

Frame Relay

Objectives

After completing this chapter, you should be able to answer the following questions:

- What are the fundamental concepts of Frame Relay technology in terms of enterprise WAN services, including operation, implementation requirements, Frame Relay maps, and Local Management Interface (LMI) operation?
- How do you configure a basic Frame Relay permanent virtual circuit (PVC), including configuring and troubleshooting Frame Relay on a router serial interface and configuring a static Frame Relay map?
- What are the advanced concepts of Frame Relay technology in terms of enterprise WAN services, including subinterfaces, bandwidth, and flow control?
- How do you configure advanced Frame Relay PVC, including solving reachability issues, configuring subinterfaces, and verifying and troubleshooting a Frame Relay configuration?

Key Terms

This chapter uses the following key terms. You can find the definitions in the glossary at the end of the book.

dedicated lines page 128

Data Link Connection Identifier (DLCI) page 132

Frame Relay access device (FRAD) page 134

mesh page 138

Link Access Procedure for Frame Relay (LAPF)
page 140

congestion page 141

Forward Explicit Congestion Notification (FECN)
page 141

Backward Explicit Congestion Notification (BECN)
page 141

star topology page 142

hub page 142

full-mesh topology page 143

partial-mesh topology page 145

Inverse Address Resolution Protocol (ARP)
page 145

Local Management Interface (LMI) page 148

keepalive interval page 150

nonbroadcast multiaccess (NBMA) page 157

Committed Information Rate (CIR) page 163

Excess Burst Size (BE) page 165

Discard Eligible (DE) page 165

Introduction

Frame Relay is a high-performance WAN protocol that operates at the physical and data link layers of the OSI reference model.

Eric Scaee, an engineer at Sprint International, invented Frame Relay as a simpler version of the X.25 protocol to use across Integrated Services Digital Network (ISDN) interfaces. Today, it is used over a variety of other network interfaces as well. When Sprint first implemented Frame Relay in its public network, it used StrataCom switches. Cisco's acquisition of StrataCom in 1996 marked its entry into the carrier market.

Network providers commonly implement Frame Relay for voice and data as an encapsulation technique, used between LANs over a WAN. Each end user gets a private line (or leased line) to a Frame Relay node. The Frame Relay network handles the transmission over multiple paths, transparent to all end users.

Frame Relay has become one of the most extensively used WAN protocols, primarily because it is inexpensive compared to *dedicated lines*. In addition, configuring user equipment in a Frame Relay network is very simple. Frame Relay connections are created by configuring Customer Premises Equipment (CPE) routers or other devices to communicate with a service provider Frame Relay switch. The service provider configures the Frame Relay switch, which helps keep end-user configuration tasks to a minimum.

This chapter describes Frame Relay and explains how to configure Frame Relay on a Cisco router.

Basic Frame Relay Concepts

Frame Relay has become the most widely used WAN technology in the world. Large enterprises, governments, ISPs, and small businesses use Frame Relay, primarily because of its price and flexibility.

Introducing Frame Relay

As organizations grow and depend more and more on reliable data transport, traditional leased-line solutions are prohibitively expensive. The pace of technological change, along with companies expanding geographically into other areas, demand and require more flexibility.

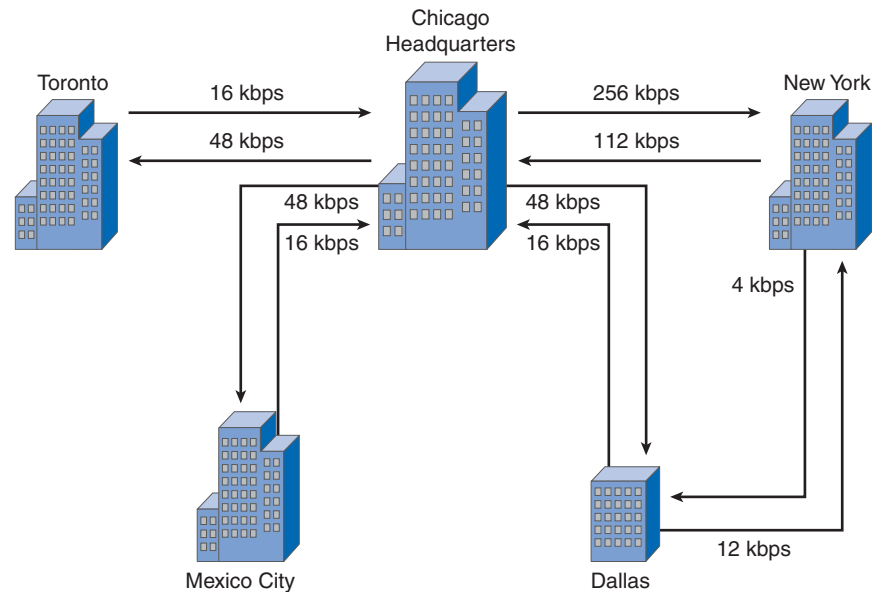
Frame Relay: An Efficient and Flexible WAN Technology

Frame Relay reduces network costs by using less equipment, less complexity, and an easier implementation. Moreover, Frame Relay provides greater bandwidth, reliability, and resiliency than private or leased lines. With increasing globalization and the growth of one-

to-many branch office topologies, Frame Relay offers simpler network architecture and lower cost of ownership.

Using an example of a large enterprise network helps illustrate the benefits of using a Frame Relay WAN. In Figure 3-1, Span Engineering has five campuses across North America. As with most organizations, Span's bandwidth requirements do not match a "one size fits all" solution.

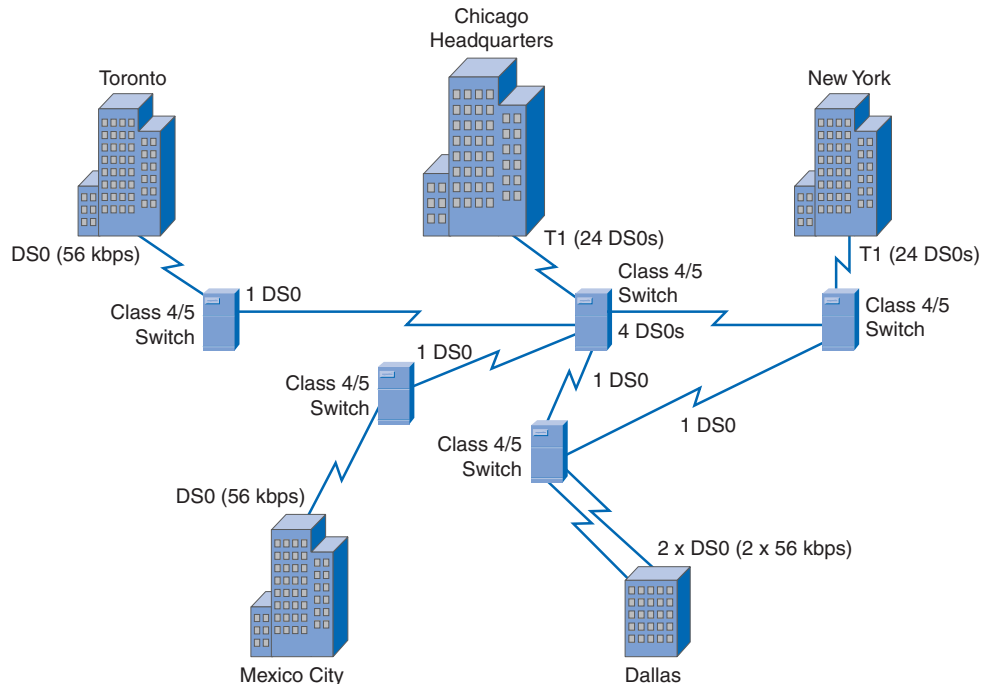
Figure 3-1 Corporate Bandwidth Requirements



The first thing to consider is the bandwidth requirements of each site. Working out from the headquarters, the Chicago-to-New York connection requires a maximum speed of 256 kbps. Three other sites need a maximum speed of 48 kbps connecting to the headquarters. The connection between the New York and Dallas branch offices requires only 12 kbps.

Before Frame Relay became available, Span leased dedicated lines.

Using leased lines, each Span site is connected through a switch at the local telephone company's central office (CO) through the local loop, and then across the entire network, as shown in Figure 3-2. The Chicago and New York sites each use a dedicated T1 line (equivalent to 24 DS0 channels) to connect to the switch, and other sites use ISDN connections (56 kbps). Because the Dallas site connects with both New York and Chicago, it has two locally leased lines. The service providers have given Span one DS0 between the respective COs, except for the larger pipe connecting Chicago to New York, which has four DS0s. DS0s are priced differently from region to region and usually are offered at a fixed price. These lines are truly dedicated in that the network provider reserves that line for Span's own use. No sharing occurs. Span pays for the end-to-end circuit regardless of how much bandwidth it uses.

Figure 3-2 Dedicated Line WAN Requirements

A dedicated line provides little practical opportunity for a one-to-many connection without getting more lines from the network provider. In the Span Engineering example, almost all communication must flow through the corporate headquarters, simply to reduce the cost of additional lines.

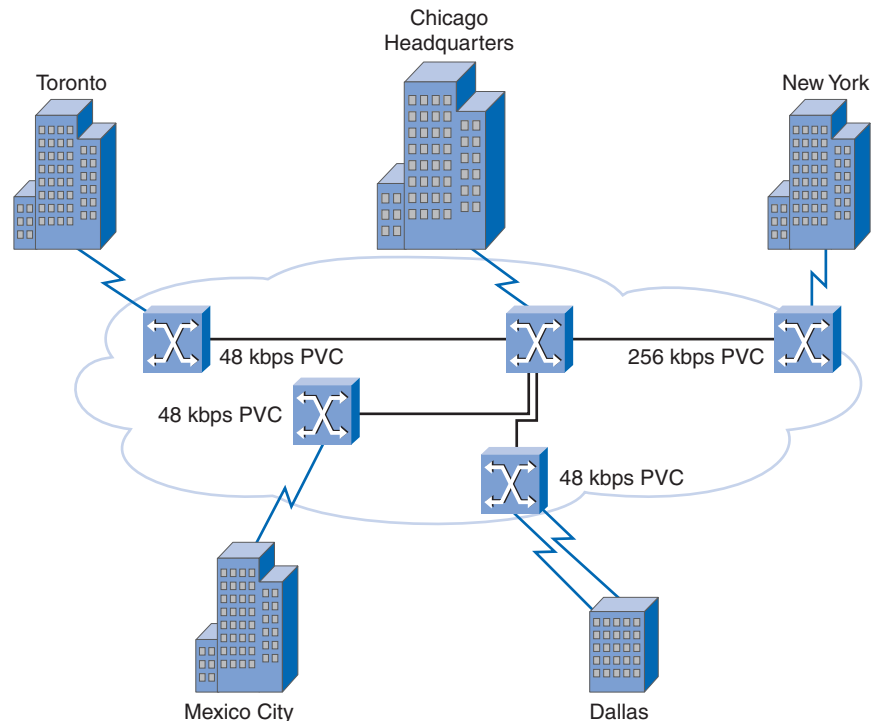
If you examine what each site requires in terms of bandwidth, you notice a lack of efficiency:

- Of the 24 DS0 channels available in the T1 connection, the Chicago site uses only seven. Some carriers offer fractional T1 connections in increments of 64 kbps, but this requires a specialized multiplexer at the customer end to channelize the signals. In this case, Span has opted for the full T1 service.
- Similarly, the New York site uses only five of its available 24 DS0s.
- Because Dallas needs to connect to Chicago and New York, two lines connect through the CO to each site.

The leased-line design also limits flexibility. Unless circuits are already installed, connecting new sites typically requires new circuit installations and takes considerable time to implement. From a network reliability point of view, imagine the additional costs in money and complexity of adding spare and redundant circuits.

In Figure 3-3, Span's Frame Relay network uses permanent virtual circuits (PVC). A PVC is the logical path along an originating Frame Relay link, through the network, and along a terminating Frame Relay link to its ultimate destination. Compare this to the physical path used by a dedicated connection. In a network with Frame Relay access, a PVC uniquely defines the path between two endpoints. The concept of virtual circuits is discussed in more detail later in this section.

Figure 3-3 Frame Relay WAN Requirements



Span's Frame Relay solution provides both cost effectiveness and flexibility.

Cost Effectiveness of Frame Relay

Frame Relay is a more cost-effective option for two reasons. First, with dedicated lines, customers pay for an end-to-end connection. That includes the local loop and the network link within the WAN cloud. With Frame Relay, customers pay for only the local loop and the bandwidth they purchase from the network provider. Distance between nodes is not important. In a dedicated-line model, customers use dedicated lines provided in increments of 64 kbps. However, Frame Relay customers can define their virtual-circuit needs in far greater granularity, often in increments as small as 4 kbps.

The second reason for Frame Relay's cost effectiveness is that it shares bandwidth across a larger base of customers. Typically, a service provider can service 40 or more 56-kbps customers over one T1 circuit. Using dedicated lines would require more DSU/CSUs (one for each line) and more complicated routing and switching. Service providers save because there is less equipment to purchase and maintain.

Table 3-1 shows a representative cost comparison (in U.S. dollars) for comparable ISDN and Frame Relay connections. Although initial costs for Frame Relay are higher than for ISDN, the monthly cost is considerably lower. Frame Relay is easier to manage and configure than ISDN. In addition, customers can increase their bandwidth as their needs grow in the future. Frame Relay customers pay for only the bandwidth they need. Frame Relay incurs no hourly charges. ISDN calls are metered and can result in unexpectedly high monthly charges from the telephone company if a full-time connection is maintained.

Table 3-1 Frame Relay Costs

	64-kbps ISDN	56-kbps Frame Relay
Local loop monthly charge	\$185	\$85
ISP setup	\$380	\$750
Equipment	\$700 (ISDN router)	\$1600 (Cisco router)
ISP monthly charge	\$195	\$195
One-time charges	\$1080	\$2660
Monthly charges	\$380	\$280

Flexibility of Frame Relay

A virtual circuit provides considerable flexibility in network design. Looking at the figure, you can see that Span's offices all connect to the Frame Relay cloud over their respective local loops. What happens in the cloud is really of no concern at this time. All that matters is that when any Span office wants to communicate with any other Span office, all it needs to do is connect to a virtual circuit leading to the other office. In Frame Relay, the end of each connection has a number to identify it called a *Data Link Connection Identifier (DLCI)*. Any station can connect with any other simply by stating the address of that station and DLCI number of the line it needs to use. In a later section, you will learn that when Frame Relay is configured, all the data from all the configured DLICs flows through the same port of the router. Try to picture the same flexibility using dedicated lines. Not only is it complicated, but it also requires considerably more equipment.

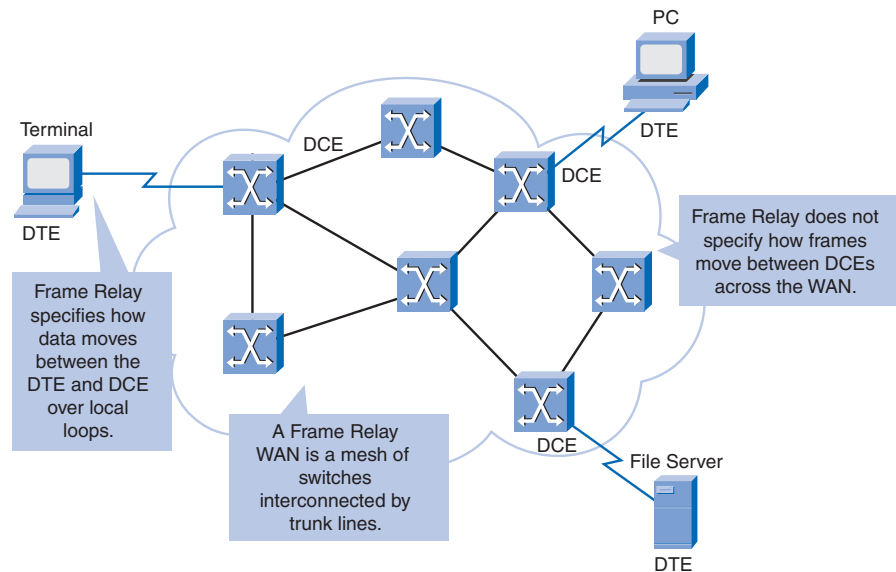
The next few topics will expand your understanding of Frame Relay by defining the key concepts introduced in the example.

Frame Relay WAN

In the late 1970s and into the early 1990s, the WAN technology joining the end sites typically used the X.25 protocol. Now considered a legacy protocol, X.25 was a popular packet-switching technology because it provided a reliable connection over unreliable cabling infrastructures. It did so by including additional error control and flow control. However, these additional features added overhead to the protocol. Its major application was for processing credit card authorizations and automatic teller machines. This course mentions X.25 only for historical purposes.

When you build a WAN, regardless of the transport you choose, the WAN always has a minimum of three basic components, or groups of components, connecting any two sites, as shown in Figure 3-4. Each site needs its own equipment (DTE) to access the telephone company's CO serving the area (DCE). The third component sits in the middle, joining the two sites. In the figure, this is the portion supplied by the Frame Relay backbone, the area within the cloud.

Figure 3-4 Frame Relay WAN



Frame Relay differs significantly from X.25 in its functionality and format. In particular, Frame Relay is a more streamlined protocol, facilitating higher performance and greater efficiency. For example, Frame Relay does not provide error correction. Modern WAN facilities offer more reliable connection services and a higher degree of reliability than older facilities. The Frame Relay node simply drops packets without notification when it detects errors. Any necessary error correction, such as retransmission of data, is left to the end-points. This makes propagation from customer end to customer end through the network very fast.

Frame Relay operates between an end-user device, such as a LAN bridge or router, and a network. The network itself can use any transmission method that is compatible with the speed and efficiency that Frame Relay applications require. Some networks use Frame Relay itself, but others use digital circuit switching or ATM cell relay systems.

Frame Relay Operation

The connection between a DTE device and a DCE device consists of both a physical layer component and a data link layer component:

- The physical component defines the mechanical, electrical, functional, and procedural specifications for the connection between the devices. One of the most commonly used physical layer interface specifications is the RS-232 specification.
- The data link layer component defines the protocol that establishes the connection between the DTE device, such as a router, and the DCE device, such as a switch.

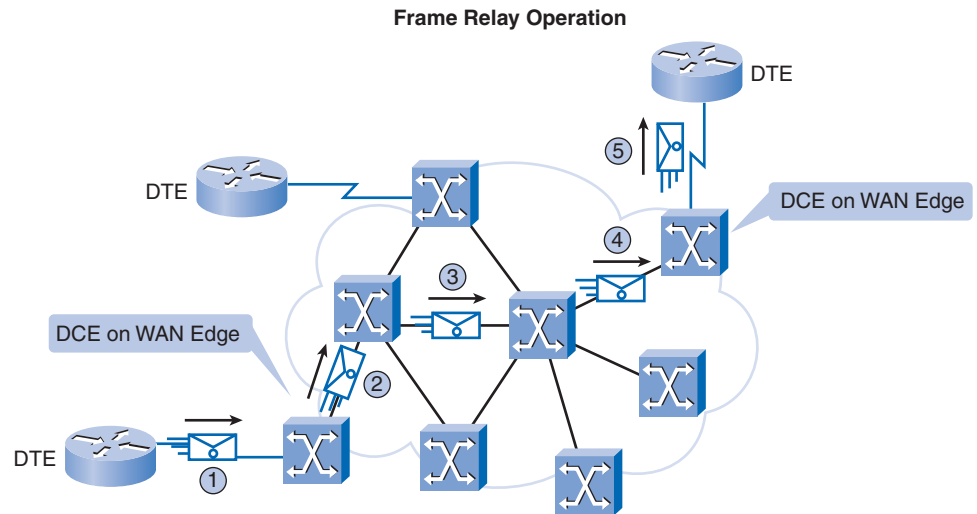
Figure 3-5 illustrates Frame Relay operation. When carriers use Frame Relay to interconnect LANs, a router on each LAN is the DTE. A serial connection, such as a T1/E1 leased line, connects the router to the carrier's Frame Relay switch at the nearest point of presence (POP) for the carrier. The Frame Relay switch is a DCE device. Network switches move frames from one DTE across the network and deliver frames to other DTEs by way of DCEs. Computing equipment that is not on a LAN may also send data across a Frame Relay network. The computing equipment uses a *Frame Relay access device (FRAD)* as the DTE. The FRAD is sometimes called a Frame Relay assembler/dissembler and is a dedicated appliance or a router configured to support Frame Relay. It is located on the customer's premises and connects to a switch port on the service provider's network. In turn, the service provider interconnects the Frame Relay switches.

Virtual Circuits

Virtual circuits provide a bidirectional communication path from one DTE device to another and are uniquely identified by a DLCI. The circuits are virtual because there is no direct electrical connection from end to end. The connection is logical, and data moves from end to end without a direct electrical circuit. With virtual circuits (VC), Frame Relay shares the bandwidth among multiple users, and any single site can communicate with any other single site without using multiple dedicated physical lines.

Frame Relay virtual circuits fall into two categories: switched virtual circuits (SVC) and permanent virtual circuits (PVC):

- SVCs are temporary connections used in situations requiring only sporadic transfer between DTE devices across the Frame Relay network. A communication session across an SVC consists of four operational states: call setup, data transfer, idle, and call termination.

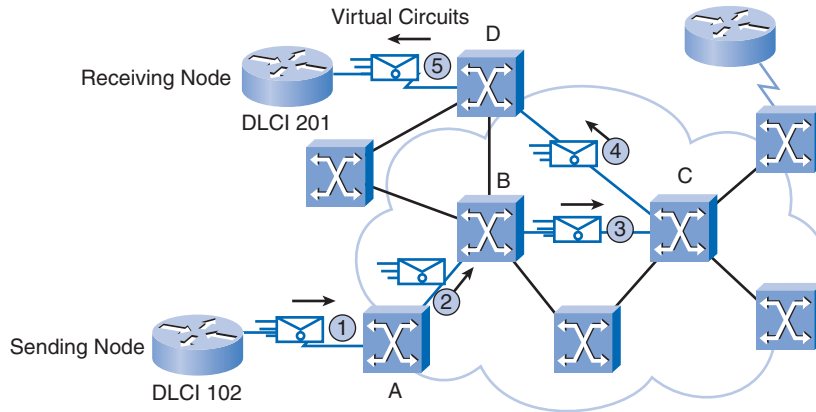
Figure 3-5 Frame Relay Operation

- 1 The DTE sends frames to the DCE switch on the WAN edge.
- 2-4 The frames move from switch to switch across the WAN to the destination DCE switch on the WAN edge.
- 5 The destination DCE delivers the frames to the destination DTE.

- PVCs are permanently established connections that are used for frequent and consistent data transfers between DTE devices across the Frame Relay network. Communication across a PVC does not require the call setup and termination states that are used with SVCs. PVCs always operate in one of two operational states: data transfer or idle mode. Note that some publications refer to PVCs as private VCs.

Figure 3-6 shows a VC between the sending and receiving nodes. The VC follows the path A, B, C, D. Frame Relay creates a VC by storing input-port-to-output-port mapping in the memory of each switch and thus links one switch to another until a continuous path from one end of the circuit to the other is identified. A VC can pass through any number of intermediate devices (switches) located in the Frame Relay network.

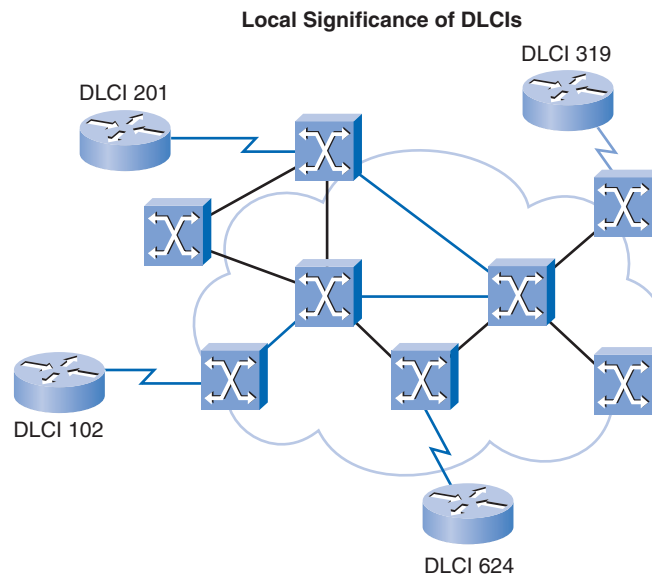
Figure 3-6 Virtual Circuits



At this point, you might be wondering how the various nodes and switches are identified.

VCs provide a bidirectional communication path from one device to another. VCs are identified by DLCIs, as shown in Figure 3-7. DLCI values typically are assigned by the Frame Relay service provider (for example, the telephone company). Frame Relay DLCIs have local significance, which means that the values themselves are not unique in the Frame Relay WAN. A DLCI identifies a VC to the equipment at an endpoint. A DLCI has no significance beyond the single link. Two devices connected by a VC may use a different DLCI value to refer to the same connection.

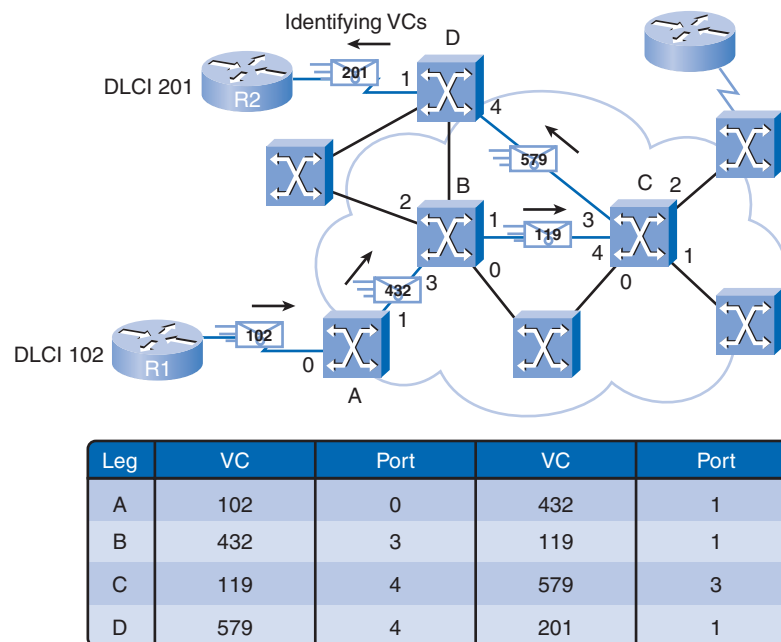
Figure 3-7 Local Significance of DLCIs



Locally significant DLCIs have become the primary method of addressing, because the same address can be reused in several different locations to refer to completely different connections. Local addressing prevents a customer from running out of DLCIs as the network grows.

Figure 3-8 is the same network as shown in Figure 3-7, but this time, as the frame moves across the network, Frame Relay labels each VC with a DLCI. The DLCI is stored in the address field of every frame transmitted to tell the network how the frame should be switched. The Frame Relay service provider assigns DLCI numbers. Usually, DLCIs 0 to 15 and 1008 to 1023 are reserved for special purposes. Therefore, service providers typically assign DLCIs in the range of 16 to 1007.

Figure 3-8 Identifying VCs



In Figure 3-8, the frame uses DLCI 102. It leaves the router (R1) using Port 0 and VC 102. At switch A, the frame exits Port 1 using VC 432. This process of VC-port mapping continues through the WAN until the frame reaches its destination at DLCI 201. The DLCI is stored in the Address field of every Frame Relay data link frame transmitted.

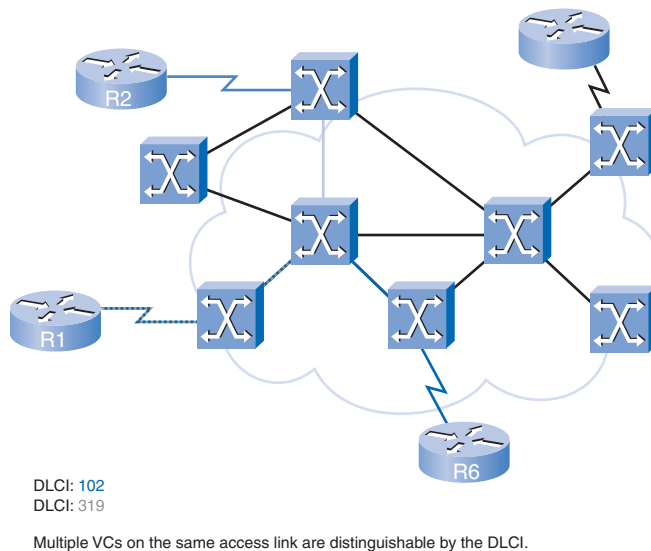
Multiple VCs

Frame Relay is statistically multiplexed, meaning that it transmits only one frame at a time, but many logical connections can coexist on a single physical line. The Frame Relay access device (FRAD) or router connected to the Frame Relay network may have multiple VCs

connecting it to various endpoints. Multiple VCs on a single physical line are distinguished because each VC has its own DLCI. Remember that the DLCI has only local significance and may be different at each end of a VC.

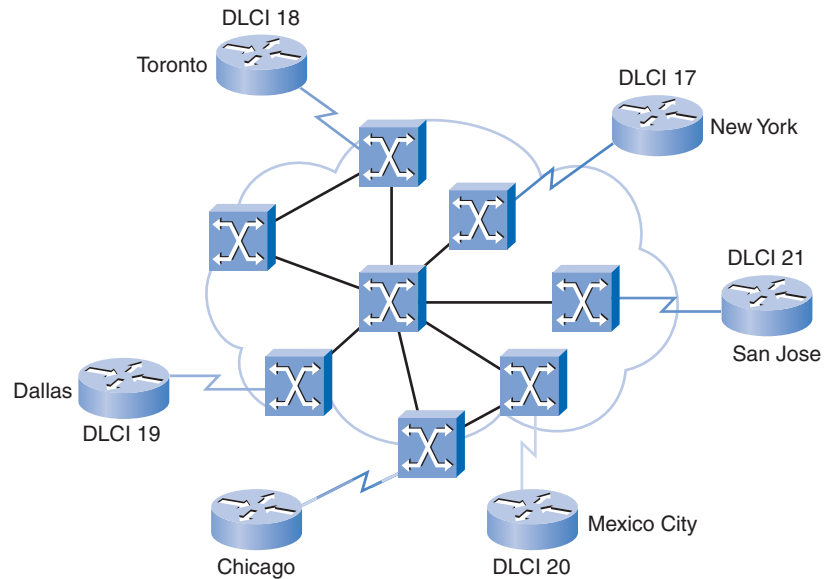
Figure 3-9 shows two VCs on a single access line, each with its own DLCI, attaching to a router (R1).

Figure 3-9 Multiple VCs on a Single Access Line



This capability often reduces the equipment and network complexity required to connect multiple devices, making it a very cost-effective replacement for a *mesh* of access lines. With this configuration, each endpoint needs only a single access line and interface. More savings arise because the capacity of the access line is based on the average bandwidth requirements of the VCs, rather than on the maximum bandwidth requirement.

For example, in Figure 3-10, Span Engineering has five locations, along with its headquarters in Chicago. Table 3-2 lists each router's local DLCI used to reach the Chicago network. Chicago is connected to the network using five VCs, and each VC is given a DLCI. The DLCIs on the Chicago router do not need to be the same values as the DLCIs on the remote routers.

Figure 3-10 Span Engineering's DLCIs from Chicago**Table 3-2** Span Engineering's DLCIs

Local Router	Local DLCI	Destination Network
New York	17	Chicago
Toronto	18	Chicago
Dallas	19	Chicago
Mexico City	20	Chicago
San Jose	21	Chicago

Cost Benefits of Multiple VCs

Recall the earlier example of how Span Engineering evolved from a dedicated-line network to a Frame Relay network. Specifically, look back at Table 3-1, which compares the cost of a single Frame Relay connection to a similar-sized ISDN connection. Note that with Frame Relay, customers pay for the bandwidth they use. In effect, they pay for a Frame Relay port. When they increase the number of ports, as described earlier, they pay for more bandwidth. But will they pay for more equipment? The short answer is “no,” because the ports are virtual. The physical infrastructure doesn’t change. Compare this to purchasing more bandwidth using dedicated lines.

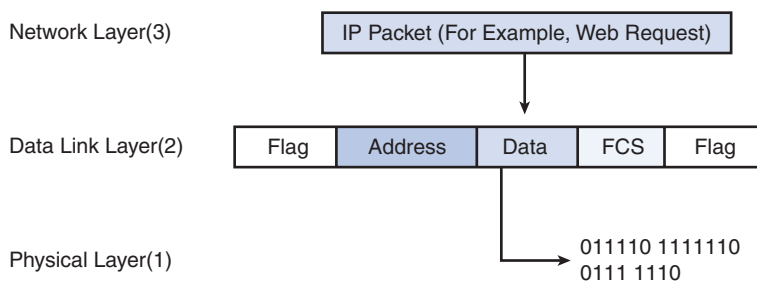
Frame Relay Encapsulation

Frame Relay takes data packets from a network layer protocol, such as IP or IPX. It encapsulates them as the data portion of a Frame Relay frame and then passes the frame to the physical layer for delivery on the wire. To understand how this process works, it is helpful to understand how it relates to the lower levels of the OSI model.

The Frame Relay Encapsulation Process

Figure 3-11 shows how Frame Relay encapsulates data for transport and moves it down to the physical layer for delivery.

Figure 3-11 Frame Relay Encapsulation and the OSI Model



First, Frame Relay accepts a packet from a network layer protocol such as IP. It then wraps it with an Address field that contains the DLCI and a checksum. Flag fields are added to indicate the beginning and end of the frame and are always the same. The flags are represented as either the hexadecimal number 7E or the binary number 0111110. After the packet is encapsulated, Frame Relay passes the frame to the physical layer for transport.

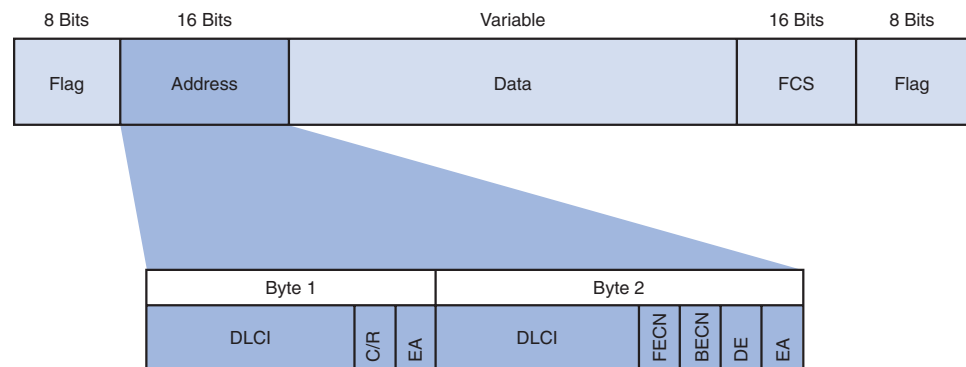
The CPE router encapsulates each Layer 3 packet inside a Frame Relay header and trailer before sending it across the VC. The header and trailer are defined by the *Link Access Procedure for Frame Relay (LAPF)* Bearer Services specification, ITU Q.922-A.

Specifically, the Frame Relay header (Address field), shown in Figure 3-12, contains the following:

- **DLCI:** The 10-bit DLCI is the essence of the Frame Relay header. This value represents the virtual connection between the DTE device and the switch. Each virtual connection that is multiplexed onto the physical channel is represented by a unique DLCI. The DLCI values have local significance only, which means that they are unique only to the physical channel on which they reside. Therefore, devices at opposite ends of a connection can use different DLCI values to refer to the same virtual connection.
- **C/R:** The bit that follows the most significant DLCI byte in the Address field. The C/R bit is not currently defined.

- **Extended Address (EA):** If the value of the EA field is 1, the current byte is determined to be the last DLCI octet. Although current Frame Relay implementations all use a two-octet DLCI, this capability does allow longer DLCIs in the future. The eighth bit of each byte of the Address field indicates the EA.
- **Congestion control:** Contains 3 bits that control the Frame Relay congestion-notification mechanisms. The *Forward Explicit Congestion Notification (FECN)*, *Backward Explicit Congestion Notification (BECN)*, and Discard Eligibility (DE) bits are the last 3 bits in the Address field. Congestion control is discussed in a later section.

Figure 3-12 Standard Frame Relay Frame



The physical layer typically is EIA/TIA-232, 449, or 530; V.35; or X.21. The Frame Relay frame is a subset of the HDLC frame type. Therefore, it is delimited with Flag fields. The 1-byte flag uses the bit pattern 01111110. The FCS determines whether any errors in the Layer 2 Address field occurred during transmission. The FCS is calculated before transmission by the sending node, and the result is inserted in the FCS field. At the distant end, a second FCS value is calculated and compared to the FCS in the frame. If the results are the same, the frame is processed. If there is a difference, the frame is discarded. Frame Relay does not notify the source when a frame is discarded. Error control is left to the upper layers of the OSI model.

Frame Relay Topologies

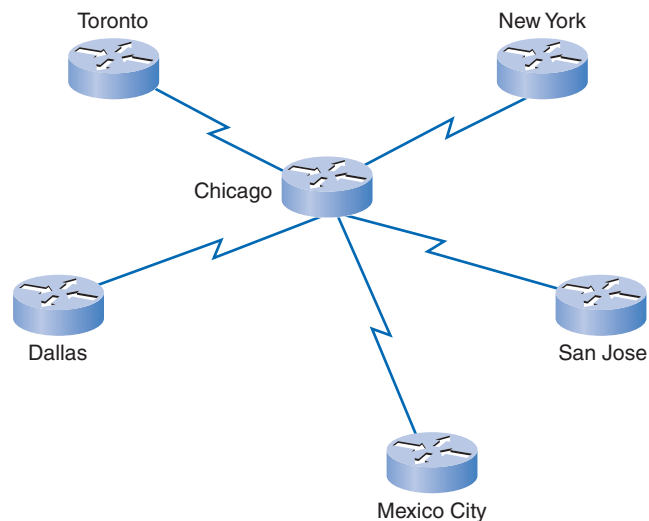
When more than two sites are to be connected, you must consider the topology of the connections between them. A topology is the map or visual layout of the Frame Relay network. You need to consider the topology from several perspectives to understand the network and the equipment used to build the network. Complete topologies for design, implementation, operation, and maintenance include overview maps, logical connection maps, functional maps, and address maps showing the detailed equipment and channel links.

Cost-effective Frame Relay networks link dozens and even hundreds of sites. Considering that a corporate network might span any number of service providers and include networks from acquired businesses differing in basic design, documenting topologies can be a very complicated process. However, every network or network segment can be viewed as being one of three topology types: star, full mesh, or partial mesh.

Star Topology (Hub and Spoke)

The simplest WAN topology is a *star topology*, as shown in Figure 3-13. In this topology, Span Engineering has a central site in Chicago that acts as a hub and hosts the primary services. Notice that Span has grown and recently opened an office in San Jose. Using Frame Relay made this expansion relatively easy.

Figure 3-13 Star (Hub-and-Spoke) Topology



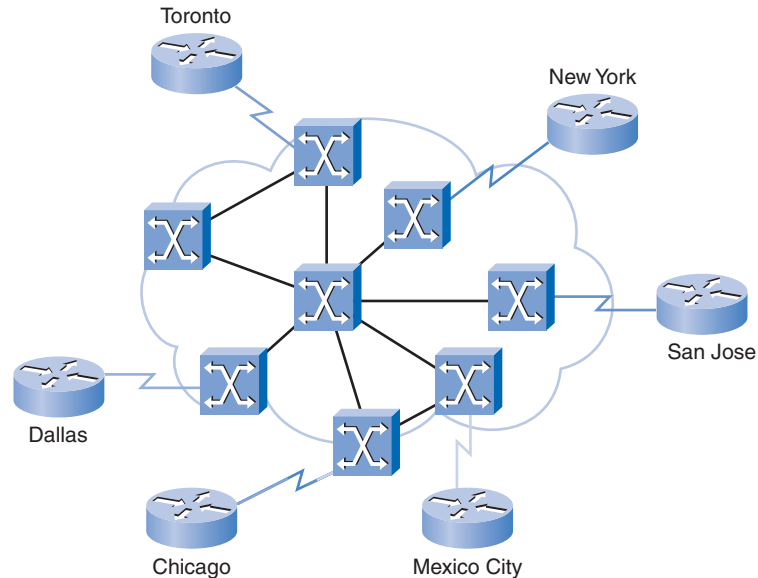
Star Topology-Hub with Five Physical Links (Spokes)

Connections to each of the five remote sites act as spokes. In a star topology, the location of the hub usually is chosen by the lowest leased-line cost. When implementing a star topology with Frame Relay, each remote site has an access link to the Frame Relay cloud with a single VC.

Figure 3-14 shows the star topology in the context of a Frame Relay cloud. The *hub* at Chicago has an access link with multiple VCs, one for each remote site. The lines going out from the cloud represent the connections from the Frame Relay service provider and terminate at the customer premises. These typically are lines ranging in speed from 56,000 bps to E1 (2.048 Mbps) and faster. One or more DLCI numbers are assigned to each line endpoint.

Because Frame Relay costs are not distance-related, the hub does not need to be in the geographic center of the network.

Figure 3-14 Frame Relay Star Topology



Frame Relay Star-Hub with One Physical Link Carrying Five VCs

Full-Mesh Topology

Figure 3-15 shows a *full-mesh topology* using dedicated lines. A full-mesh topology suits a situation in which the services to be accessed are geographically dispersed and highly reliable access to them is required. A full-mesh topology connects every site to every other site. Using leased-line interconnections, additional serial interfaces and lines add costs. In this example, ten dedicated lines are required to interconnect each site in a full-mesh topology.

Using Frame Relay, a network designer can build multiple connections simply by configuring additional VCs on each existing link. This software upgrade grows the star topology to a full-mesh topology without the expense of additional hardware or dedicated lines. Because VCs use statistical multiplexing, multiple VCs on an access link generally make better use of Frame Relay than single VCs. Figure 3-16 shows how Span Engineering has used four VCs on each link to scale its network without adding new hardware. Service providers will charge for the additional bandwidth, but this solution usually is more cost-effective than using dedicated lines.

Figure 3-15 Full-Mesh Topology

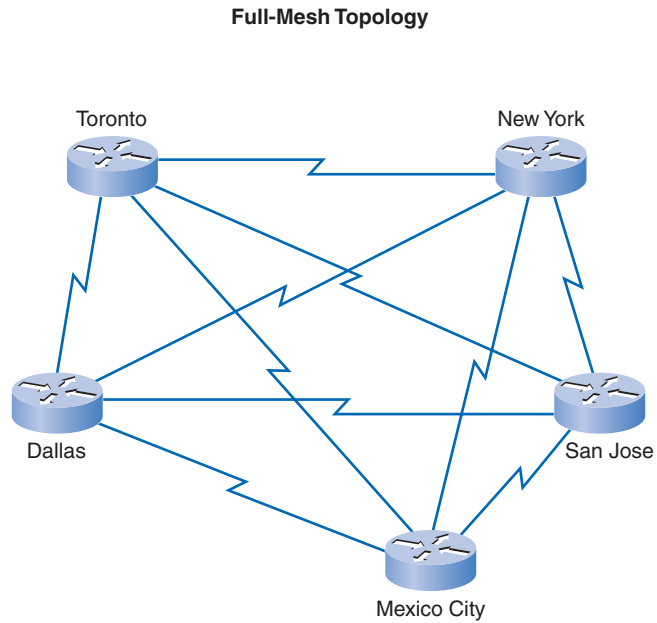
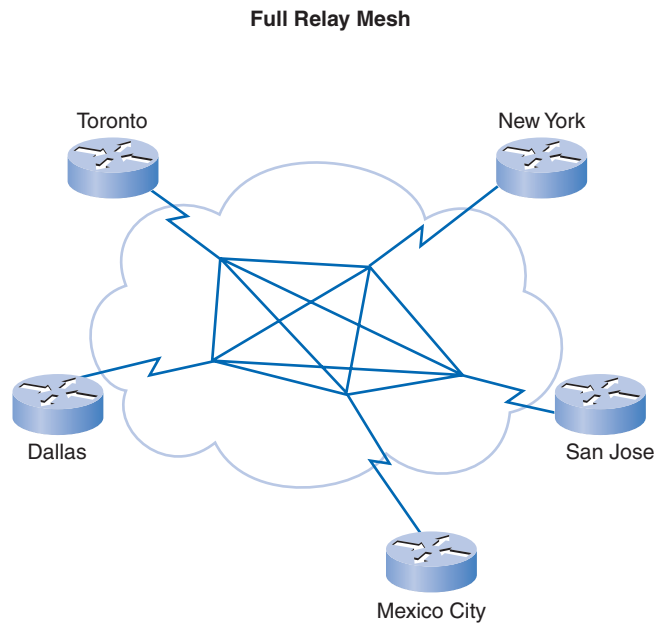


Figure 3-16 Frame Relay Mesh Topology



Partial-Mesh Topology

For large networks, a full-mesh topology is seldom affordable because the number of links required increases dramatically. The issue is not with the cost of the hardware, but because there is a theoretical limit of fewer than 1000 VCs per link. In practice, the limit is less than that.

For this reason, larger networks generally are configured in a *partial-mesh topology*. Partial mesh has more interconnections than are required for a star arrangement, but not as many as for a full mesh. The actual pattern depends on the data flow requirements.

Frame Relay Address Mapping

Before a Cisco router can transmit data over Frame Relay, it needs to know which local DLCI maps to the Layer 3 address of the remote destination. Cisco routers support all network layer protocols over Frame Relay, such as IP, IPX, and AppleTalk. This address-to-DLCI mapping can be accomplished by either static or dynamic mapping.

Inverse ARP

Inverse Address Resolution Protocol (ARP) obtains Layer 3 addresses of other stations from Layer 2 addresses, such as the DLCI in Frame Relay networks. It is used primarily in Frame Relay and ATM networks, where Layer 2 addresses of VCs are sometimes obtained from Layer 2 signaling, and the corresponding Layer 3 addresses must be available before these VCs can be used. Whereas ARP translates Layer 3 addresses to Layer 2 addresses, Inverse ARP does the opposite.

Dynamic Mapping

Dynamic address mapping relies on Inverse ARP to resolve a next-hop network protocol address to a local DLCI value. The Frame Relay router sends out Inverse ARP requests on its PVC to discover the protocol address of the remote device connected to the Frame Relay network. The router uses the responses to populate an address-to-DLCI mapping table on the Frame Relay router or access server. The router builds and maintains this mapping table, which contains dynamic entries from resolved Inverse ARP requests and static entries from manual configuration.

Example 3-1 shows the output of the **show frame-relay map** command. You can see that the interface is up and that the destination IP address is 10.1.1.2. The DLCI identifies the logical connection being used to reach this interface. This value is displayed in three ways: its decimal value (102), its hexadecimal value (0x66), and its value as it would appear in the two-octet Address field as it goes on the wire (0x1860). This is a static entry, not a dynamic entry. The link is using Cisco encapsulation as opposed to IETF encapsulation.

Example 3-1 show frame-relay map Command for Router R1

```
R1# show frame-relay map
Serial0/0/1 (up): ip 10.1.1.2 dlci 102 (0x66,0x1860), static
                  broadcast,
                  CISCO, status defined, active
R1#
```

On Cisco routers, Inverse ARP is enabled by default for all protocols enabled on the physical interface. Inverse ARP packets are not sent out for protocols that are not enabled on the interface.

The user can choose to override dynamic Inverse ARP mapping by supplying a manual static mapping for the next-hop protocol address to a local DLCI. A static map works similarly to dynamic Inverse ARP by associating a specified next-hop protocol address to a local Frame Relay DLCI. You cannot use Inverse ARP and a map statement for the same DLCI and protocol.

An example of using static address mapping is a situation in which the router at the other side of the Frame Relay network does not support dynamic Inverse ARP for a specific network protocol. To provide accessibility, a static mapping is required to complete the remote network layer address to local DLCI resolution.

Another example is on a hub-and-spoke Frame Relay network. You would use static address mapping on the spoke routers to provide spoke-to-spoke reachability. Because the spoke routers do not have direct connectivity with each other, dynamic Inverse ARP would not work between them. Dynamic Inverse ARP relies on the presence of a direct point-to-point connection between two ends. In this case, dynamic Inverse ARP works only between hub and spoke, and the spokes require static mapping to provide reachability to each other.

Configuring Static Mapping

Establishing static mapping depends on your network needs.

To map between a next-hop protocol address and a local DLCI address, use the following command:

```
Router(config)# frame-relay map protocol protocol-address dlci [broadcast] [ietf]
                [cisco]
```

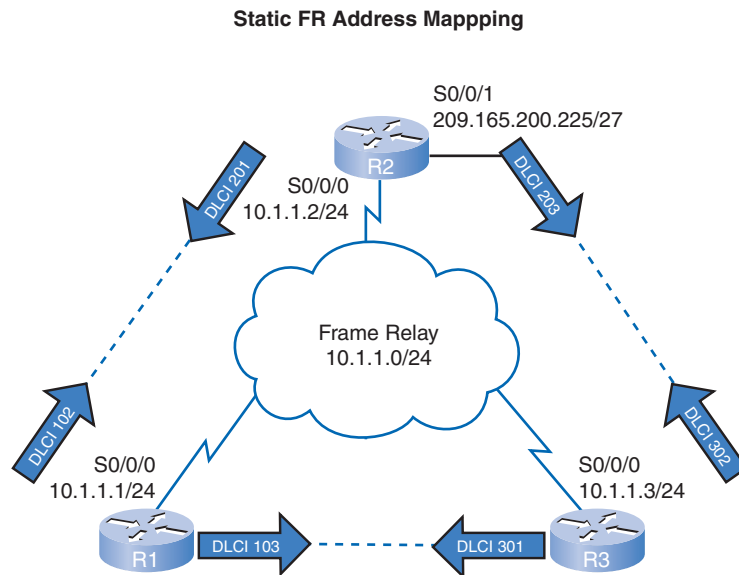
Use the keyword **ietf** when connecting to a non-Cisco router.

You can greatly simplify the configuration for the Open Shortest Path First (OSPF) protocol by adding the optional **broadcast** keyword when performing this task.

Figure 3-17 shows the topology used for the next example. Example 3-2 provides the configuration commands for statically mapping a DLCI on a Cisco router. In this example,

static address mapping is performed on R1's interface serial 0/0/0, and the Frame Relay encapsulation used on DLCI 102 is CISCO. As shown in the configuration steps, static mapping of the address using the **frame-relay map** command allows users to select the type of Frame Relay encapsulation used on a per-VC basis. Static mapping configuration is discussed in more detail in the next major section.

Figure 3-17 Static Frame Relay Mapping



Example 3-2 Static Mapping Configuration for Router R1

```
R1(config)# interface serial0/0/0
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# encapsulation frame-relay
R1(config-if)# no frame-relay inverse-arp
R1(config-if)# frame-relay map ip 10.1.1.2 102 broadcast cisco
R1(config-if)# no shut
R1(config-if)#
*Oct 16 03:37:03.391: %LINK-3-UPDOWN: Interface Serial0/0/0, Changed state to up
*Oct 16 03:37:14.443: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
  changed state to up
```

Local Management Interface (LMI)

A review of networking history will help you understand the role played by the *Local Management Interface (LMI)*. The Frame Relay design provides packet-switched data transfer with minimum end-to-end delays. The original design omits anything that might contribute to delay.

When vendors implemented Frame Relay as a separate technology rather than as one component of ISDN, they decided that there was a need for DTEs to dynamically acquire information about the network's status. However, the original design did not include this feature. A consortium of Cisco, Digital Equipment Corporation (DEC), Northern Telecom, and StrataCom extended the Frame Relay protocol to provide additional capabilities for complex internetworking environments. These extensions are collectively referred to as the LMI.

Basically, the LMI is a keepalive mechanism that provides status information about Frame Relay connections between the router (DTE) and the Frame Relay switch (DCE). Every 10 seconds or so, the end device polls the network, requesting either a dumb sequenced response or channel status information. If the network does not respond with the requested information, the user device may consider the connection down. When the network responds with a FULL STATUS response, it includes status information about DLCIs that are allocated to that line. The end device can use this information to determine whether the logical connections can pass data.

Using the topology shown in Figure 3-17, Example 3-3 shows the output of the **show frame-relay lmi** command. The output shows the LMI type used by the Frame Relay interface and the counters for the LMI status exchange sequence, including errors such as LMI timeouts.

Example 3-3 show frame-relay lmi Command for Router R1

```
R1# show frame-relay lmi

LMI Statistics for interface Serial1 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0          Invalid Msg Type 0
  Invalid Status Message 0         Invalid Lock Shift 0
  Invalid Information ID 0          Invalid Report IE Len 0
  Invalid Report Request 0         Invalid Keep IE Len 0
  Num Status Enq. Sent 9           Num Status msgs Rcvd 0
  Num Update Status Rcvd 0         Num Status Timeouts 9
```

It is easy to confuse the LMI and encapsulation. The LMI is a definition of the messages used between the DTE (R1) and the DCE (the Frame Relay switch owned by the service provider). Encapsulation defines the headers used by a DTE to communicate information to the DTE at the other end of a VC. The switch and its connected router care about using the

same LMI. The switch does not care about the encapsulation. The endpoint routers (DTEs) do care about the encapsulation.

LMI Extensions

In addition to the Frame Relay protocol functions for transferring data, the Frame Relay specification includes optional LMI extensions that are extremely useful in an internetworking environment. The LMI enhancements offer a number of features for managing complex internetworks, including the following:

- **VC status messages** provide information about PVC integrity by communicating and synchronizing between devices, periodically reporting the existence of new PVCs and the deletion of already existing PVCs. VC status messages prevent data from being sent into black holes (PVCs that no longer exist).
- **Multicasting** allows a sender to transmit a single frame that is delivered to multiple recipients. Multicasting supports the efficient delivery of routing protocol messages and address resolution procedures that typically are sent to many destinations simultaneously.
- **Global addressing** gives connection identifiers global rather than local significance, allowing them to be used to identify a specific interface to the Frame Relay network. Global addressing makes the Frame Relay network resemble a LAN in terms of addressing, and ARPs perform exactly as they do over a LAN.
- **Simple flow control** provides an XON/XOFF flow-control mechanism that applies to the entire Frame Relay interface. It is intended for devices whose higher layers cannot use the congestion notification bits and need some level of flow control.

Table 3-3 lists the LMI identifiers. The 10-bit DLCI field supports 1024 VC identifiers: 0 through 1023. The LMI extensions reserve some of these identifiers, thereby reducing the number of permitted VCs. LMI messages are exchanged between the DTE and DCE using these reserved DLCIs.

Table 3-3 LMI Identifiers

VC Identifiers	VC Types
0	LMI (ANSI, ITU)
1 to 15	Reserved for future use
1008 to 1022	Reserved for future use (ANSI, ITU)
1019 and 1020	Multicasting (Cisco)
1023	LMI (Cisco)

There are several LMI types, each of which is incompatible with the others. The LMI type configured on the router must match the type used by the service provider. Cisco routers support three types of LMIs:

- **Cisco:** Original LMI extension
- **Ansi:** Corresponds to the ANSI standard T1.617 Annex D
- **q933a:** Corresponds to the ITU standard Q933 Annex A

Starting with Cisco IOS software Release 11.2, the default LMI autosense feature detects the LMI type supported by the directly connected Frame Relay switch. Based on the LMI status messages it receives from the Frame Relay switch, the router automatically configures its interface with the supported LMI type acknowledged by the Frame Relay switch.

If it is necessary to set the LMI type, use the following interface configuration command:

```
frame-relay lmi-type [cisco | ansi | q933a]
```

Configuring the LMI type disables the autosense feature.

When manually setting up the LMI type, you must configure the *keepalive interval* on the Frame Relay interface to prevent status exchanges between the router and switch from timing out. The LMI status exchange messages determine the status of the PVC connection. For example, a large mismatch in the keepalive interval on the router and the switch can cause the switch to declare the router dead.

By default, the keepalive time interval is 10 seconds on Cisco serial interfaces. You can change the keepalive interval with the **keepalive** interface configuration command.

You'll set the LMI type and configure the keepalive in a later activity.

LMI Frame Format

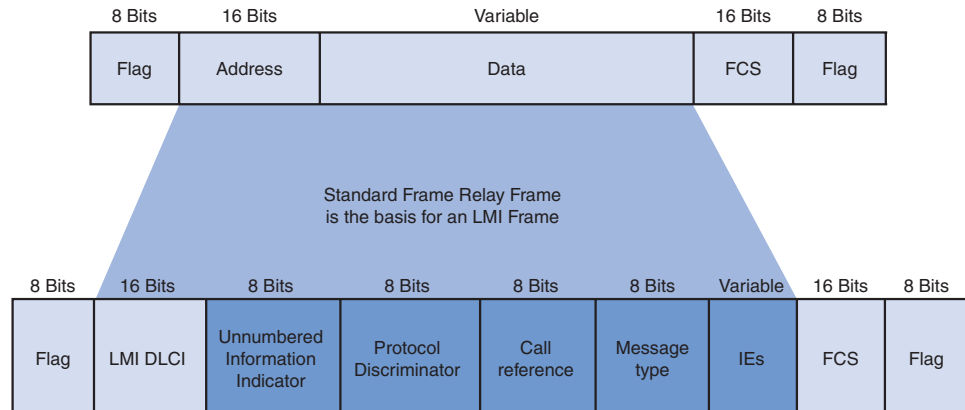
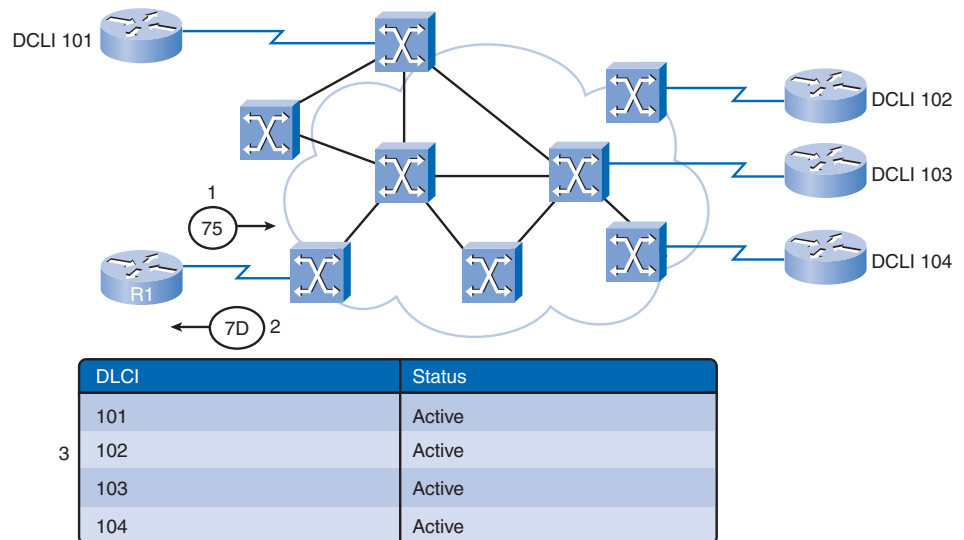
Figure 3-18 shows an LMI frame format. LMI messages are carried in a variant of LAPF frames. The Address field carries one of the reserved DLCIs. Following the DLCI field are the Unnumbered Information Indicator, Protocol Discriminator, and Call Reference fields, which do not change. The fourth field indicates the LMI message type.

Status messages help verify the integrity of logical and physical links. This information is critical in a routing environment because routing protocols make decisions based on link integrity.

Using LMI and Inverse ARP to Map Addresses

LMI status messages combined with Inverse ARP messages allow a router to associate network layer and data link layer addresses.

In Figure 3-19, when R1 connects to the Frame Relay network, it sends an LMI status inquiry message to the network. The network replies with an LMI status message containing the details of every VC configured on the access link.

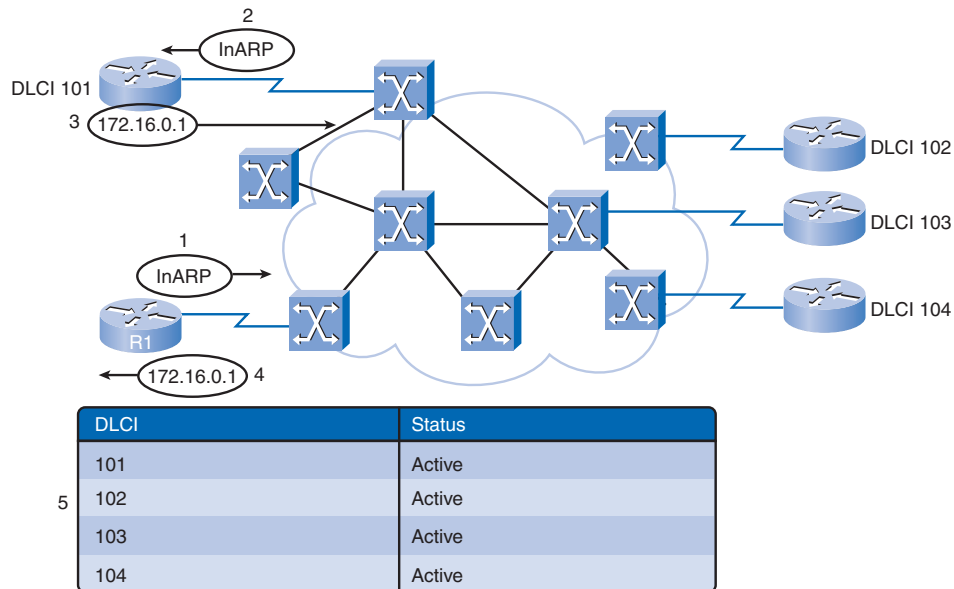
Figure 3-18 LMI Frame Format**Figure 3-19** Stages of Inverse ARP and LMI Operation

- 1 DTE sends Status Enquiry Message (75) to DCE.
- 2 DCE responds with Status Message (7D) – includes configured DLCIs.
- 3 DTE learns what VCs it has.

Periodically, the router repeats the status inquiry, but subsequent responses include only status changes. After a set number of these abbreviated responses, the network sends a full status message.

If the router needs to map the VCs to network layer addresses, it sends an Inverse ARP message on each VC, as shown in Figure 3-20. The Inverse ARP message includes the router's network layer address, so the remote DTE, or router, can also perform the mapping. The Inverse ARP reply allows the router to make the necessary mapping entries in its address-to-DLCI mapping table. If several network layer protocols are supported on the link, Inverse ARP messages are sent for each one.

Figure 3-20 Using LMI and Inverse ARP to Map Addresses



1,2 DTE sends inverse ARP on a VC, mapping VC to network address.

3,4 Remote DTE replies with Layer 3 address.

5 DTE maps Layer 2–Layer 3.

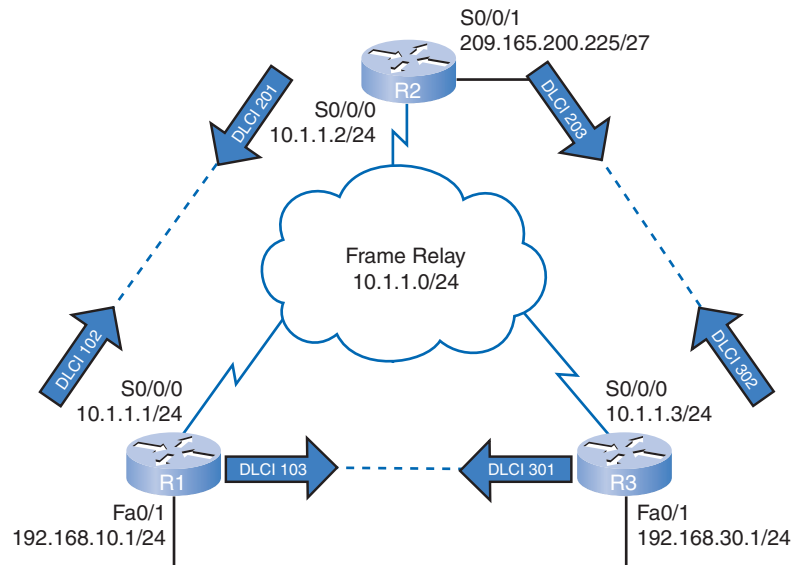
This process is repeated for each VC and each Layer 3 protocol.

Configuring Frame Relay

Frame Relay is configured on a Cisco router from the Cisco IOS command-line interface (CLI).

This section outlines the required steps to enable Frame Relay on your network, as well as some of the optional steps you can follow to enhance or customize your configuration.

Figure 3-21 shows the basic setup model used for this discussion. Later in this section, additional hardware will be added to the diagram to help explain more-complex configuration tasks.

Figure 3-21 Frame Relay Topology for Configuration

In this section, you will configure the Cisco routers as Frame Relay access devices, or DTE, connected directly to a dedicated Frame Relay switch, or DCE. The following are the required and optional tasks for configuring Frame Relay:

Required tasks:

- Enable Frame Relay encapsulation on an interface.
- Configure dynamic or static address mapping.

Optional tasks:

- Configure the LMI.
- Configure Frame Relay SVCs.
- Configure Frame Relay traffic shaping.
- Customize Frame Relay for your network.
- Monitor and maintain Frame Relay connections.

Enabling Frame Relay Encapsulation

Using the topology shown in Figure 3-21, Examples 3-4 and 3-5 show how Frame Relay has been configured on the serial interfaces for routers R1 and R2. This involves assigning an IP address, setting the encapsulation type, and allocating bandwidth.

Example 3-4 Router R1 Configuration Script

```
interface s0/0/0
ip address 10.1.1.1 255.255.255.0
encapsulation frame-relay
bandwidth 64
```

Example 3-5 Router R2 Configuration Script

```
interface s0/0/0
ip address 10.1.1.2 255.255.255.0
encapsulation frame-relay
bandwidth 64
```

The steps for enabling Frame Relay encapsulation are as follows:

- Step 1. Set the IP address on the interface.** On a Cisco router, Frame Relay is most commonly supported on synchronous serial interfaces. Use the **ip address** command to set the interface's IP address. You can see in Figure 3-21 that R1 has been assigned IP address 10.1.1.1/24, and R2 has been assigned 10.1.1.2/24.
- Step 2. Configure the encapsulation.** The **encapsulation frame-relay** interface configuration command enables Frame Relay encapsulation and allows Frame Relay processing on the supported interface. You can choose from two encapsulation options, as described next.
- Step 3. Set the bandwidth (optional).** If needed, use the **bandwidth** command to set the bandwidth of the serial interface. You specify the bandwidth in kbps. This command notifies the routing protocol that bandwidth is statically configured on the link. The EIGRP and OSPF routing protocols use the bandwidth value to calculate and determine the link's metric.
- Step 4. Set the LMI type (optional).** This step is optional because Cisco routers autosense the LMI type. Recall that Cisco supports three LMI types—Cisco, ANSI Annex D, and Q933-A Annex A—and that the default LMI type for Cisco routers is cisco.

Recall that the default encapsulation type on a serial interface on a Cisco router is the Cisco-proprietary version of HDLC. To change the encapsulation from HDLC to Frame Relay, use the following command:

```
encapsulation frame-relay [cisco | ietf]
```

The **no** form of this command removes the Frame Relay encapsulation on the interface and returns the interface to the default HDLC encapsulation.

The default Frame Relay encapsulation enabled on supported interfaces is the Cisco encapsulation. Use this option if you're connecting to another Cisco router. Many non-Cisco devices also support this encapsulation type. It uses a 4-byte header, with 2 bytes to identify the DLCI and 2 bytes to identify the packet type.

The IETF encapsulation type complies with RFC 1490 and RFC 2427. Use this option if you're connecting to a non-Cisco router.

The output of the **show interfaces serial** command shown in Examples 3-6 and 3-7 verifies the configuration for routers R1 and R2.

Example 3-6 show interface serial Command for Router R1

```
R1# show interface serial0/0/0

Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.1.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  LMI enq sent 18, LMI stat recvd 19, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:08, output 00:00:08, output hang never
  Last clearing of "show interface" counters 00:04:06
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1558 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    24 packets input, 815 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    27 packets output, 807 bytes, 0 underruns
    0 output errors, 0 collisions, 1 interface resets
    0 output buffer failures, 0 output buffers swapped out
    2 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

Example 3-7 show interface serial Command for Router R2

```
R2# show interface serial0/0/0

Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.1.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  LMI enq sent 17, LMI stat recvd 18, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/64, broadcasts sent/dropped 0/0, interface broadcasts 0
  Last input 00:00:07, output 00:00:07, output hang never
  Last clearing of "show interface" counters 00:03:40
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1558 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    23 packets input, 786 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    29 packets output, 1197 bytes, 0 underruns
    0 output errors, 0 collisions, 1 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up
```

Configuring Static Frame Relay Maps

Cisco routers support all network layer protocols over Frame Relay, such as IP, IPX, and AppleTalk. The address-to-DLCI mapping can be accomplished by either dynamic or static address mapping.

Dynamic mapping is performed by the Inverse ARP feature. Because Inverse ARP is enabled by default, no additional command is required to configure dynamic mapping on an interface.

Static mapping is manually configured on a router. Establishing static mapping depends on your network needs. To map between a next-hop protocol address and a DLCI destination address, use the following command:

```
frame-relay map protocol protocol-address dlcid [broadcast]
```

Table 3-4 shows how to use the **frame-relay map** command parameters when configuring static address maps.

Table 3-4 Command Parameters

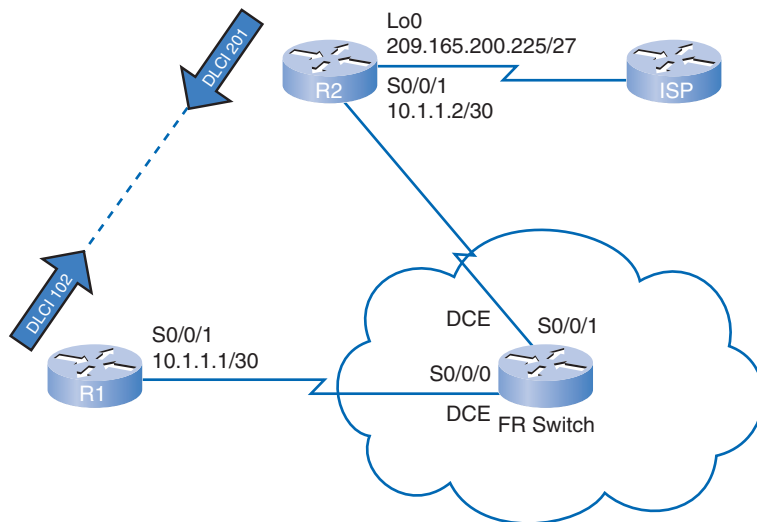
Command Parameter	Description
<i>protocol</i>	Defines the supported protocol, bridging, or logical link control. Possible values are appletalk , decnet , dls , ip , ipx , llc2 , rsrb , vines , and xns .
<i>protocol-address</i>	Defines the network layer address of the destination router interface.
<i>dlci</i>	Defines the local DLCI used to connect to the remote protocol address.
broadcast	(Optional) Allows broadcasts and multicasts over the VC. This permits the use of dynamic routing protocols over the VC.

Frame Relay, ATM, and X.25 are *nonbroadcast multiaccess (NBMA)* networks. NBMA networks only allow data transfer from one computer to another over a VC or across a switching device. NBMA networks do not support multicast or broadcast traffic, so a single packet cannot reach all destinations. To achieve a broadcast, you are required to replicate the packets manually to all destinations.

Some routing protocols may require additional configuration options. For example, RIP, EIGRP, and OSPF require additional configurations to be supported on NBMA networks.

Because NBMA does not support broadcast traffic, using the **broadcast** keyword for the **frame-relay map** command is a simplified way of forwarding routing updates. The **broadcast** keyword allows broadcasts and multicasts over the PVC and, in effect, turns the broadcast into a unicast so that the other node gets the routing updates.

Using the topology shown in Figure 3-22, R1 uses the **frame-relay map** command to map the VC to R2, as shown in Example 3-8.

Figure 3-22 Configuring the Static Frame Relay Map

Note: The Frame Relay switch is shown as a router in this graphic. Routers can be configured as a Frame Relay switches.

Example 3-8 Static Map Configuration for Router R1

```
R1#(config)# interface s0/0/1
R1#(config-if)# ip address 10.1.1.1 255.255.255.252
R1#(config-if)# encapsulation frame-relay
R1#(config-if)# bandwidth 64
R1#(config-if)# frame-relay map ip 10.1.1.2 102 broadcast
```

To verify the Frame Relay mapping, use the **show frame-relay map** command, as shown in Example 3-9 for router R1 and in Example 3-10 for router R2.

Example 3-9 Verifying a Static Frame Relay Map on R1

```
R1# show frame-relay map

Serial0/0/1 (up): ip 10.1.1.2 dlci 102(0x66,0x1860), static,
broadcast,
CISCO, status defined, active
```


Example 3-10 Verifying a Static Frame Relay Map on R2

```
R2# show frame-relay map
Serial0/0/1 (up): ip 10.1.1.1 dlci 201(0xC9,0x3090), static,
broadcast,
CISCO, status defined, active
```

Packet Tracer
Activity**Configuring Basic Frame Relay with Static Maps (3.2.2)**

In this activity, you configure two static Frame Relay maps on each router to reach two other routers. Although the LMI type is autosensed on the routers, you will statically assign the type by manually configuring the LMI. Detailed instructions are provided within the activity. Use File e4-322.pka on the CD-ROM that accompanies this book to perform this activity using Packet Tracer.

Advanced Frame Relay Concepts

Frame Relay involves several advanced concepts, including understanding the effects of routing protocols that use split horizon, the use of subinterfaces, and the complexities of cost structure used by Frame Relay providers.

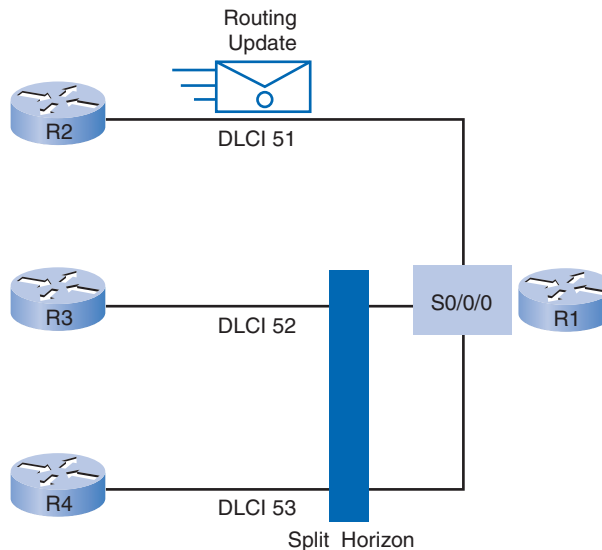
Solving Reachability Issues

By default, a Frame Relay network provides NBMA connectivity between remote sites. NBMA clouds usually use a hub-and-spoke topology. Unfortunately, a basic routing operation based on the split-horizon principle can cause reachability issues on a Frame Relay NBMA network.

Split Horizon

Recall that split horizon is a technique used to prevent a routing loop in networks using distance vector routing protocols. Split-horizon updates reduce routing loops by preventing a routing update received on one interface from being forwarded out the same interface.

Figure 3-23 shows R2, a spoke router, sending a broadcast routing update to R1, the hub router.

Figure 3-23 Split-Horizon Rule

Routers that support multiple connections over a single physical interface have many PVCs terminating on a single interface. R1 must replicate broadcast packets, such as routing update broadcasts, on each PVC to the remote routers. The replicated broadcast packets can consume bandwidth and cause significant latency to user traffic. The amount of broadcast traffic and the number of VCs terminating at each router should be evaluated during the design of a Frame Relay network. Overhead traffic, such as routing updates, can affect the delivery of critical user data, especially when the delivery path contains low-bandwidth (56-kbps) links.

In Figure 3-23, router R1 has multiple PVCs on a single physical interface. The split-horizon rule prevents R1 from forwarding the routing updates, initially coming from router R2, through the same physical interface to other remote spoke routers (R3 and R4).

Disabling split horizon may seem like a simple solution, because it allows routing updates to be forwarded out the same physical interface from which they came. However, only IP allows you to disable split horizon; IPX and AppleTalk do not. Also, disabling split horizon increases the chance of routing loops in any network. Split horizon could be disabled for physical interfaces with a single PVC.

The next obvious solution to solve the split-horizon problem is to use a fully meshed topology. However, this is expensive because more PVCs are required. The preferred solution is to use subinterfaces, as explained in the next section.

Frame Relay Subinterfaces

Frame Relay can partition a physical interface into multiple virtual interfaces called subinterfaces. A subinterface is simply a logical interface that is directly associated with a physical interface. Therefore, a Frame Relay subinterface can be configured for each of the PVCs coming into a physical serial interface.

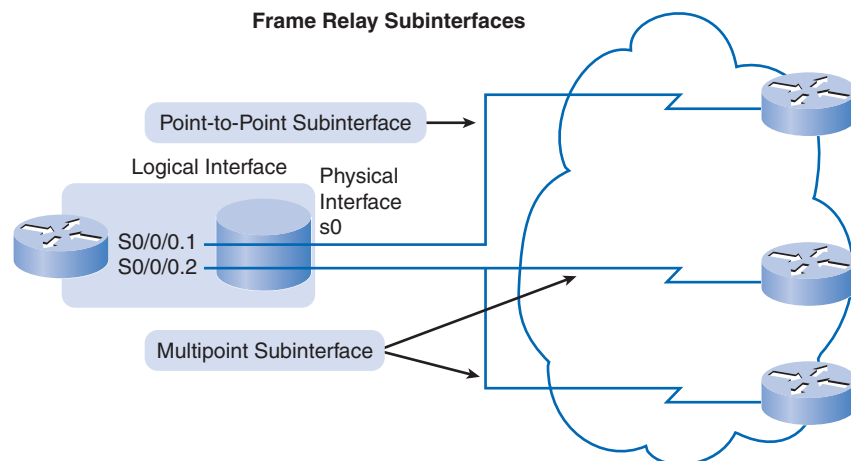
To enable the forwarding of broadcast routing updates in a Frame Relay network, you can configure the router with logically assigned subinterfaces. A partially meshed network can be divided into a number of smaller, fully meshed, point-to-point networks. Each point-to-point subnetwork can be assigned a unique network address. This allows packets received on a physical interface to be sent out the same physical interface, because the packets are forwarded on VCs in different subinterfaces.

Frame Relay subinterfaces can be configured in either point-to-point or multipoint mode:

- **Point-to-point:** A single point-to-point subinterface establishes one PVC connection to another physical interface or subinterface on a remote router. In this case, each pair of the point-to-point routers is on its own subnet, and each point-to-point subinterface has a single DLCI. In a point-to-point environment, each subinterface acts like a separate, conventional point-to-point interface. Typically, each point-to-point VC has a separate subnet. Therefore, routing update traffic is not subject to the split-horizon rule.
- **Multipoint:** A single multipoint subinterface establishes multiple PVC connections to multiple physical interfaces or subinterfaces on remote routers. All the participating interfaces are in the same subnet. The subinterface acts like an NBMA Frame Relay interface, so routing update traffic is subject to the split-horizon rule. Typically, all multipoint VCs belong to the same subnet.

Figure 3-24 shows two types of subinterfaces that Cisco routers support: point-to-point and multipoint.

Figure 3-24 Frame Relay Subinterfaces



In split-horizon routing environments, routing updates received on one subinterface can be sent out another subinterface. In a subinterface configuration, each VC can be configured as a point-to-point connection. This allows each subinterface to act similarly to a leased line. Using a Frame Relay point-to-point subinterface, each pair of the point-to-point routers is on its own subnet.

The **encapsulation frame-relay** command is assigned to the physical interface. All other configuration items, such as the network layer address and DLCIs, are assigned to the subinterface.

You can use multipoint configurations to conserve addresses. This can be especially helpful if Variable-Length Subnet Masking (VLSM) is not being used. However, multipoint configurations may not work properly given the broadcast traffic and split-horizon considerations. The point-to-point subinterface option was created to avoid these issues.

The following list summarizes point-to-point subinterfaces and multipoint subinterfaces:

Point-to-point subinterfaces (in hub-and-spoke topologies):

- They act as leased lines.
- Each point-to-point subinterface requires its own subnet.

Multipoint subinterfaces (in partial-mesh and full-mesh topologies):

- They act as NBMA, so they do not resolve the split-horizon issue.
- They can save address space, because it uses a single subnet.

You will learn more about how to configure subinterfaces in the later section “Configuring Frame Relay Subinterfaces.”

Paying for Frame Relay

Service providers build Frame Relay networks using very large and powerful switches, but as a customer, your devices see only the switch interface of the service provider. Customers usually are not exposed to the network’s inner workings, which may be built on very high-speed technologies, such as T1, T3, SONET, or ATM.

Key Terminology

From a customer’s point of view, then, Frame Relay is an interface and one or more PVCs. Customers simply buy Frame Relay services from a service provider. However, before considering how to pay for Frame Relay services, you need to know some key terms and concepts:

- **Access rate or port speed:** From a customer’s point of view, the service provider provides a serial connection or access link to the Frame Relay network over a leased line. The line’s speed is the access speed or port speed. The access rate is the rate at which your access circuits join the Frame Relay network. These typically are 56 kbps, T1

(1.536 Mbps), or Fractional T1 (a multiple of 56 kbps or 64 kbps). Port speeds are clocked on the Frame Relay switch. It is not possible to send data at higher than port speed.

- **Committed Information Rate (CIR):** Customers negotiate CIRs with service providers for each PVC. The CIR is the amount of data that the network receives from the access circuit. The service provider guarantees that the customer can send data at the CIR. All frames received at or below the CIR are accepted.

A great advantage of Frame Relay is that any unused network capacity is made available or shared with all customers, usually at no extra charge. This allows customers to “burst” over their CIR as a bonus. Bursting is explained in a moment.

In this example, aside from any CPE costs, the customer pays for three Frame Relay cost components:

- **Access or port speed:** The cost of the access line from the DTE to the DCE (customer to service provider). This line is charged based on the port speed that has been negotiated and installed.
- **PVC:** This cost component is based on the PVCs. After a PVC is established, the additional cost to increase CIR typically is small and can be done in small (4-kbps) increments.
- **CIR:** Customers normally choose a CIR lower than the port speed or access rate. This allows them to take advantage of bursts.

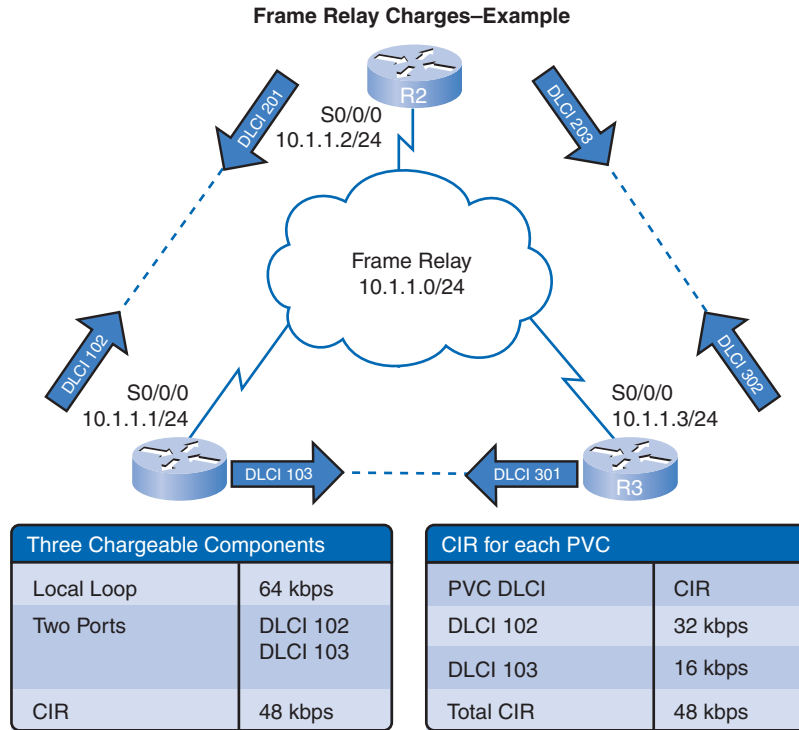
In Figure 3-25, the customer is paying for the following:

- An access line with a rate of 64 kbps connecting the customer’s DCE to the DCE of the service provider through serial port S0/0/0.
- Two virtual ports, one at 32 kbps and the other at 16 kbps.
- A CIR of 48 kbps across the entire Frame Relay network. This usually is a flat charge and is not connected to the distance.

Oversubscription

Service providers sometimes sell more capacity than they have on the assumption that not everyone will demand their entitled capacity all the time. This oversubscription is analogous to airlines selling more seats than they have in the expectation that some of the booked customers will not show up. Because of oversubscription, there will be instances when the sum of CIRs from multiple PVCs to a given location is higher than the port or access channel rate. This can cause traffic issues, such as congestion and dropped traffic.

Figure 3-25 Frame Relay Charges Example



Bursting

As stated earlier, Frame Relay takes advantage of any unused network capacity, making it available to other customer and usually at no extra charge.

Using the topology shown in Figure 3-25, Table 3-5 shows an access rate on serial port S0/0/0 of router R1 of 64 kbps. This is higher than the combined CIRs of the two PVCs. Under normal circumstances, the two PVCs should not transmit more than 32 kbps and 16 kbps, respectively. If the amount of data the two PVCs are sending does not exceed the combined CIR, the data should get through the network.

Table 3-5 Access Rates

PVC DLCI	CIR (Normal)	CBIR (Example)	BE
DLCI 102	32 kbps	48 kbps	16 kbps
DLCI 103	16 kbps	0 kbps	48 kbps
	All frames are forwarded	Frames are forwarded but marked DE	Frames most likely will be dropped

Because the physical circuits of the Frame Relay network are shared between subscribers, there are often times when excess bandwidth is available. Frame Relay can allow customers to dynamically access this extra bandwidth and “burst” over their CIR for free.

Bursting allows devices that temporarily need additional bandwidth to borrow it at no extra cost from other devices that aren’t using it. For example, if PVC 102 is transferring a large file, it could use any of the 16 kbps not being used by PVC 103. A device can burst up to the access rate and still expect the data to get through. The duration of a burst transmission should be short—less than 3 or 4 seconds.

Various terms are used to describe burst rates, including Committed Burst Information Rate (CBIR) and *Excess Burst Size (BE)*.

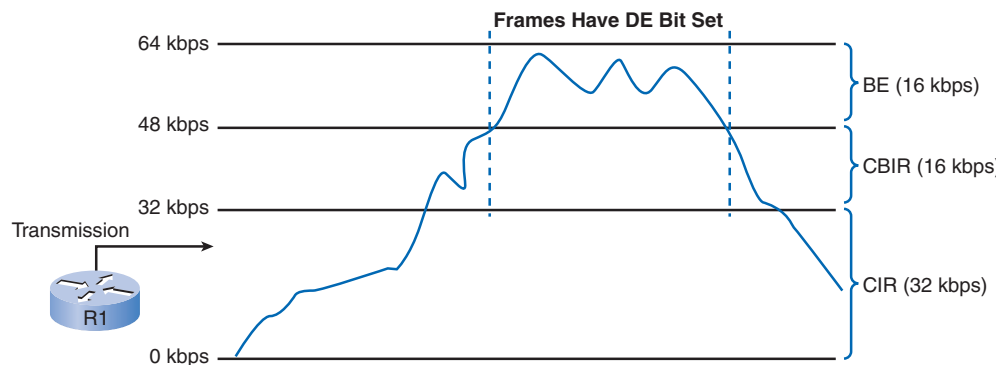
The CBIR is a negotiated rate above the CIR that the customer can use to transmit for short bursts. It allows traffic to burst to higher speeds, as available network bandwidth permits. However, it cannot exceed the port speed of the link. A device can burst up to the CBIR and still expect the data to get through. If long bursts persist, a higher CIR should be purchased.

For example, DLCI 102 has a CIR of 32 kbps, with an additional CBIR of 16 kbps, for a total of up to 48 kbps. Frames submitted at the CBIR level are marked as *Discard Eligible (DE)* in the frame header, indicating that they may be dropped if congestion occurs or the network doesn’t have enough capacity. Frames within the negotiated CIR are ineligible for discard (DE = 0). Frames above the CIR have the DE bit set to 1, marking it as eligible to be discarded if the network becomes congested.

BE is the term used to describe the bandwidth available above the CBIR up to the access rate of the link. Unlike the CBIR, it is not negotiated. Frames may be transmitted at this level but most likely will be dropped.

Figure 3-26 illustrates the relationship between the various bursting terms.

Figure 3-26 Frame Relay Bursting

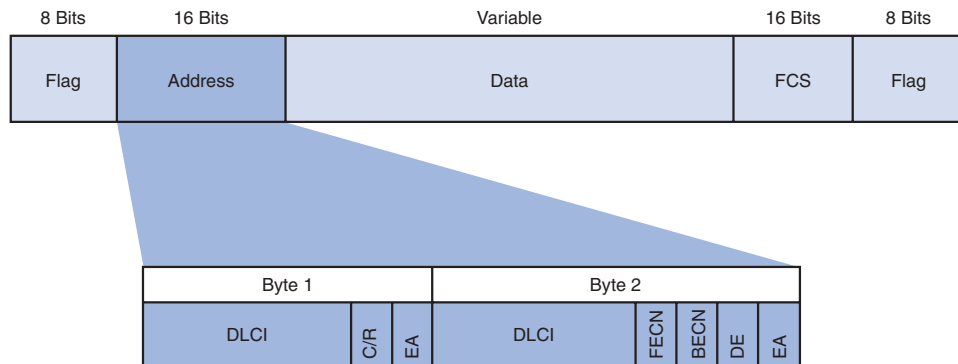


Frame Relay Flow Control

Frame Relay reduces network overhead by implementing simple congestion-notification mechanisms rather than explicit, per-VC flow control. These congestion-notification mechanisms are Forward Explicit Congestion Notification (FECN) and Backward Explicit Congestion Notification (BECN).

To help you understand the mechanisms, Figure 3-27 reviews the structure of the Frame Relay frame. FECN and BECN are each controlled by a single bit contained in the frame header. They let the router know that congestion is occurring and that the router may implement traffic shaping or throttling until the condition is reversed. BECN is a direct notification. FECN is an indirect one.

Figure 3-27 Standard Frame Relay Frame



The frame header also contains a Discard Eligibility (DE) bit, which identifies “less important” traffic that can be dropped during periods of congestion. DTE devices can set the value of the DE bit to 1 to indicate that the frame has lower importance than other frames. When the network becomes congested, DCE devices discard the frames that have the DE bit set to 1 before discarding those that do not. This reduces the likelihood of this data’s being dropped during periods of congestion.

Note

The concept of “importance” and the Discard Eligibility (DE) bit can be decided by various factors, including quality of service (QoS) mechanisms and or whether this is burst traffic.

In periods of congestion, the provider’s Frame Relay switch applies the following logic rules to each incoming frame based on whether the CIR is exceeded:

- If the incoming frame does not exceed the CBIR, the frame is passed.
- If an incoming frame exceeds the CBIR, it is marked DE.
- If an incoming frame exceeds the CBIR plus the BE, it is discarded.

Frames arriving at a switch are queued or buffered before forwarding. As in any queuing system, it is possible that an excessive buildup of frames will occur at a switch. This causes delays. Delays lead to unnecessary retransmissions that occur when higher-level protocols receive no acknowledgment within a set time. In severe cases, this can cause a serious drop in network throughput. To avoid this problem, Frame Relay incorporates a flow-control feature.

Figures 3-28 through 3-30 show a switch with a filling queue. To reduce the flow of frames to the queue, the switch notifies DTEs of the problem using the Explicit Congestion Notification bits in the frame's Address field:

- The BECN bit, indicated by the B's in Figure 3-30, is set on every frame that the switch receives on the congested link.
- The FECN bit, indicated by an F in Figure 3-29, is set on every frame that the switch places on the congested link.

DTEs receiving frames with the ECN bits set are expected to try to reduce the flow of frames until the congestion clears.

To summarize, Figure 3-28 shows that while switch A is putting a large frame on interface 1, other frames for this interface are queued. In Figure 3-29, the Frame Relay switch setting the FECN bit warns the downstream devices about the queue. In Figure 3-30, the Frame Relay switch setting the BECN bit warns the upstream devices about the queue—even though they may not have contributed to the congestion.

Figure 3-28 Frame Relay Bandwidth Control: Queuing

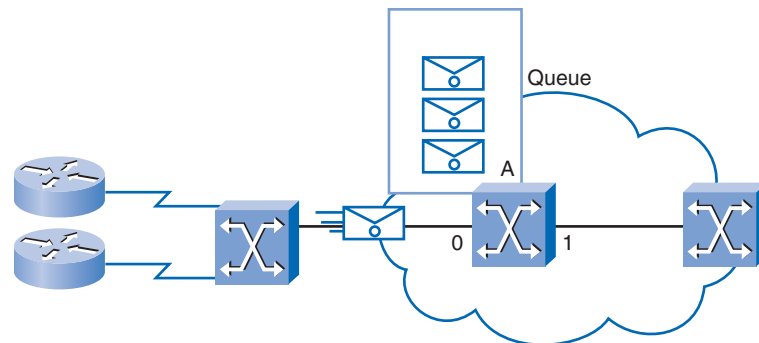
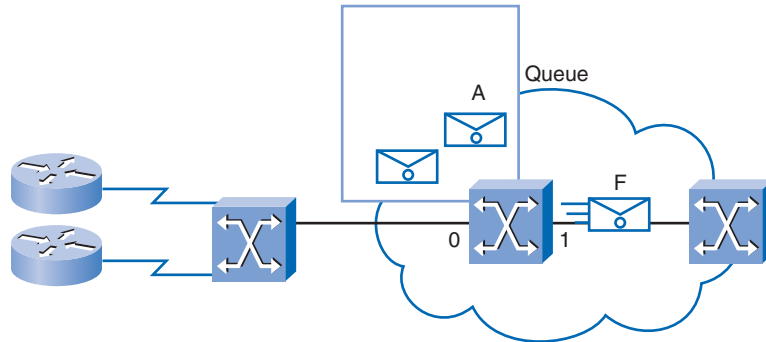
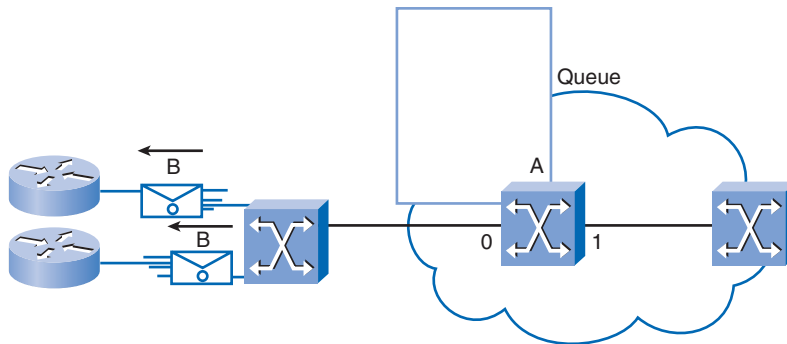


Figure 3-29 Frame Relay Bandwidth Control: FECN**Figure 3-30** Frame Relay Bandwidth Control: BECN

If the congestion occurs on an internal trunk, DTEs may be notified even though they are not the cause of the congestion.

Configuring Advanced Frame Relay

Advanced Frame Relay configuration includes configuring subinterfaces and DLCI information. Configuring Frame Relay on subinterfaces is a little different from configuring it on physical interfaces.

Configuring Frame Relay Subinterfaces

Recall that using Frame Relay subinterfaces ensures that a single physical interface is treated as multiple virtual interfaces to overcome split-horizon rules. Packets received on one virtual interface can be forwarded to another virtual interface, even if they are configured on the same physical interface.

Subinterfaces address the limitations of Frame Relay networks by providing a way to subdivide a partially meshed Frame Relay network into a number of smaller, fully meshed (or point-to-point) subnetworks. Each subnetwork is assigned its own network number and appears to the protocols as if it can be reached through a separate interface. Point-to-point subinterfaces can be unnumbered for use with IP, reducing the addressing burden that might otherwise result.

To create a subinterface, use the **interface serial** command. You can do this in either global configuration mode (config) or interface configuration mode (config-if). Specify the port number, followed by a period (.) and the subinterface number. To make troubleshooting easier, use the DLCI as the subinterface number. You must also specify whether the interface is point-to-point or point-to-multipoint using either the **multipoint** or **point-to-point** keyword, because there is no default. It is important to note that the **encapsulation frame-relay** command must be configured on the physical interface before the subinterfaces can be created. The command syntax for configuring subinterfaces is as follows:

```
router(config-if)# interface serial number.subinterface-number [multipoint |
point-to-point]
```

Table 3-6 defines the parameters for this command.

Table 3-6 Point-to-Point **interface serial** Command Parameters

interface serial Command Parameters	Description
<i>number.subinterface-number</i>	A subinterface number in the range of 1 to 4294967293. The interface number that precedes the period (.) must match the physical interface number to which this subinterface belongs.
multipoint	Select this if all routers exist in the same subnet.
point-to-point	Select this for each pair of point-to-point routers to have its own subnet. Point-to-point links normally use a subnet mask of 255.255.255.252.

To create a point-to-point subinterface for PVC 103 to R3, you would enter the following:

```
R1(config-if)# interface serial 0/0/0.103 point-to-point
```

If the subinterface is configured as point-to-point, the local DLCI for the subinterface must also be configured to distinguish it from the physical interface. The DLCI is also required for multipoint subinterfaces for which Inverse ARP is enabled. It is not required for multipoint subinterfaces configured with static route maps.

The Frame Relay service provider assigns the DLCI numbers. These numbers range from 16 to 992 and usually have only local significance. The range varies depending on the LMI used.

The **frame-relay interface-dlci** command configures the local DLCI on the subinterface. The command syntax for the **frame-relay interface-dlci** command is as follows:

```
router(config-subif)# frame-relay interface-dlci dlci-number
```

where the *dlci-number* parameter defines the local DLCI number being linked to the subinterface. This is the only way to link an LMI-derived DLCI to a subinterface, because LMI does not know about subinterfaces. Use the **frame-relay interface-dlci** command on subinterfaces only.

Here's an example:

```
R1(config-subif)# frame-relay interface-dlci 103
```

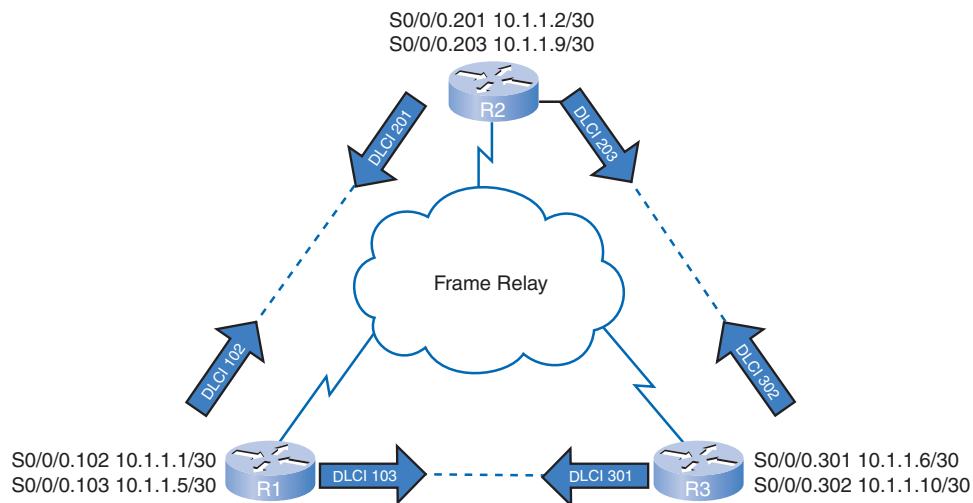
Note

Unfortunately, altering an existing Frame Relay subinterface configuration may fail to provide the expected result. In these situations, it may be necessary to save the configuration and reload the router.

Consider the following example of configuring Frame Relay subinterfaces.

In Figure 3-31, R1 has two point-to-point subinterfaces. The s0/0/0.102 subinterface connects to R2, and the s0/0/0.103 subinterface connects to R3. Each subinterface is on a different subnet.

Figure 3-31 Configuring Point-to-Point Subinterfaces



To configure subinterfaces on a physical interface, follow these steps:

- Step 1.** Remove any network layer address assigned to the physical interface. If the physical interface has an address, frames are not received by the local subinterfaces.

- Step 2.** Configure Frame Relay encapsulation on the physical interface using the **encapsulation frame-relay** command.
- Step 3.** For each of the defined PVCs, create a logical subinterface. Specify the port number, followed by a period (.) and the subinterface number. To make troubleshooting easier, it is suggested that the subinterface number match the DLCI number.
- Step 4.** Configure an IP address for the subinterface, and set the bandwidth.
- At this point, you will configure the DLCI. Recall that the Frame Relay service provider assigns the DLCI numbers.
- Step 5.** Configure the local DLCI on the subinterface using the **frame-relay interface-dlci** command.

Figure 3-32 illustrates these configuration steps for R1.

Figure 3-32 Point-to-Point Subinterface Configuration Steps

```

Step 1 → interface s/0/0/0
Step 2 → no ip address
Step 2 → encapsulation frame-relay
        no shutdown
        exit
        !
Step 3 → interface s/0/0/0.102 point-to-point
Step 4 → ip address 10.1.1.1 255.255.255.252
        bandwidth 64
Step 5 → frame-relay interface-dlci 102
        exit
        !
Step 3 → interface s/0/0/0.103 point-to-point
Step 4 → ip address 10.1.1.5 255.255.255.252
        bandwidth 64
Step 5 → frame-relay interface-dlci 103

```

Verifying Frame Relay Operation

Frame Relay generally is a very reliable service. Nonetheless, there are times when the network performs at less than expected levels and troubleshooting is necessary. For example, users may report slow and intermittent connections across the circuit. Circuits may go down. Regardless of the reason, network outages are very expensive in terms of lost productivity. A recommended best practice is to verify your configuration before problems appear.

In this section, you will step through a verification procedure to ensure that everything is working correctly before you launch your configuration on a live network.

Verifying Frame Relay Interface Configuration

After configuring a Frame Relay PVC and when troubleshooting an issue, you should verify that Frame Relay is operating correctly on that interface using the **show interfaces** command.

Note

Recall that with Frame Relay, the router normally is considered a DTE device. However, a Cisco router can be configured as a Frame Relay switch. In such cases, the router becomes a DCE device when it is configured as a Frame Relay switch.

The **show interfaces** command displays how the encapsulation is set up, along with useful Layer 1 and Layer 2 status information, including

- LMI type
- LMI DLCI
- Frame Relay DTE/DCE type

The first step is always to confirm that the interfaces are properly configured. Examples 3-11 and 3-12 show sample output for the **show interfaces** command for R1 and R2. Among other things, you can see details about the encapsulation, the DLCI on the Frame Relay-configured serial interface, and the DLCI used for the LMI. You should confirm that these values are the expected values. If not, you may need to make changes.

Example 3-11 show interface serial Commands for Router R1

```
R1# show interface serial 0/0/0

Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  CRC checking enabled
  LMI enq sent 59, LMI stat recvd 59, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/64, broadcasts sent/dropped 11/0, interface broadcasts 0
  Last input 00:00:05, output 00:00:05, output hang never
  Last clearing of "show interface" counters 00:09:55
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations 0/1/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
```

```

    Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    67 packets input, 2367 bytes, 0 no buffer
      Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    75 packets output, 4906 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up DSR=up DTR=up RTS=up CTS=up

```

R1# **show interface serial 0/0/0.102**

```

Serial0/0/0.102 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.1.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY
  CRC checking enabled
  Last clearing of "show interface" counters never

```

Example 3-12 show interface serial Commands for Router R2

R2# **show interface serial0/0/0**

```

Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
  Keepalive set (10 sec)
  LMI enq sent 37, LMI stat recvd 37, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent 0, LMI upd sent 0
  LMI DLCI 1023 LMI type is CISCO frame relay DTE
  FR SVC disabled, LAPF state down
  Broadcast queue 0/64, broadcasts sent/dropped 5/0, interface broadcasts 0
  Last input 00:00:06, output 00:00:06, output hang never
  Last clearing of "show interface" counters 00:06:16
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)

```

```

Conversations 0/1/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
Available Bandwidth 1158 kilobits/sec
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 46 packets input, 3498 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 42 packets output, 2121 bytes, 0 underruns
 0 output errors, 0 collisions, 2 interface resets
 0 output buffer failures, 0 output buffers swapped out
 0 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up

R2# show interface serial0/0/0.201

Serial0/0/0.201 is up, line protocol is up
Hardware is GT96K Serial
Internet address is 10.1.1.2/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation FRAME-RELAY
Last clearing of "show interface" counters never

```

Verifying LMI Statistics to Confirm Communication Between Routers and Provider

The next step is to look at some LMI statistics using the **show frame-relay lmi** command, as shown in Examples 3-13 and 3-14. In the output, look for any nonzero “Invalid” items. This helps isolate the problem to a Frame Relay communication issue between the carrier’s switch and your router.

Example 3-13 show frame-relay lmi Command for Router R1

```

R1# show frame-relay lmi

LMI Statistics for interface Serial0/0/0 (Frame Relay DTE) LMI TYPE = CISCO
Invalid Unnumbered info 0          Invalid Prot Disc 0
Invalid dummy Call Ref 0          Invalid Msg Type 0
Invalid Status Message 0          Invalid Lock Shift 0
Invalid Information ID 0           Invalid Report IE Len 0
Invalid Report Request 0           Invalid Keep IE Len 0
Num Status Enq. Sent 76           Num Status msgs Rcvd 76
Num Update Status Rcvd 0           Num Status Timeouts 0
Last Full Status Req 00:00:48      Last Full Status Rcvd 00:00:48

```


Example 3-14 show frame-relay lmi Command for Router R2

```

R2# show frame-relay lmi

LMI Statistics for interface Serial0/0/0 (Frame Relay DTE) LMI TYPE = CISCO
  Invalid Unnumbered info 0          Invalid Prot Disc 0
  Invalid dummy Call Ref 0          Invalid Msg Type 0
  Invalid Status Message 0          Invalid Lock Shift 0
  Invalid Information ID 0          Invalid Report IE Len 0
  Invalid Report Request 0          Invalid Keep IE Len 0
  Num Status Enq. Sent 78          Num Status msgs Rcvd 78
  Num Update Status Rcvd 0          Num Status Timeouts 0
  Last Full Status Req 00:00:02     Last Full Status Rcvd 00:00:02

```

Examples 3-13 and 3-14 show the number of status messages exchanged between the local router and the local Frame Relay switch.

Now look at the statistics for the interface.

Displaying PVC and Traffic Statistics

Use the **show frame-relay pvc [interface *interface*] [dlci]** command to view PVC and traffic statistics, as shown in Examples 3-15 and 3-16. This command is also useful for viewing the number of BECN and FECN packets received by the router. The PVC status can be active, inactive, or deleted.

Example 3-15 show frame-relay pvc Command for Router R1

```

R1# show frame-relay pvc 102

PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)
DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0.102
  input pkts 12          output pkts 20          in bytes 2816
  out bytes 5455         dropped pkts 0          in pkts dropped 0
  out pkts dropped 0     out bytes dropped 0
  in FECN pkts 0        in BECN pkts 0         out FECN pkts 0
  out BECN pkts 0       in DE pkts 0           out DE pkts 0
  out bcast pkts 15     out bcast bytes 4935
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:13:27, last time pvc status changed 00:07:47

```

Example 3-16 show frame-relay pvc Command for Router R2

```

R2# show frame-relay pvc 201

PVC Statistics for interface Serial0/0/0 (Frame Relay DTE)
DLCI = 201, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0.201
  input pkts 11          output pkts 8          in bytes 3619
  out bytes 2624        dropped pkts 0         in pkts dropped 0
  out pkts dropped 0    out bytes dropped 0
  in FECN pkts 0       in BECN pkts 0        out FECN pkts 0
  out BECN pkts 0     in DE pkts 0          out DE pkts 0
  out bcast pkts 8     out bcast bytes 2624
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:08:23, last time pvc status changed 00:08:23

```

The **show frame-relay pvc** command displays the status of all the PVCs configured on the router. You can also specify a particular PVC.

When an Inverse ARP request is made, the router updates its map table with three possible LMI connection states:

- **ACTIVE** state indicates a successful end-to-end (DTE-to-DTE) circuit.
- **INACTIVE** state indicates a successful connection to the switch (DTE-to-DCE) without a DTE detected on the other end of the PVC. This can occur because of residual or incorrect configuration on the switch.
- **DELETED** state indicates that the DTE is configured for a DLCI the switch does not recognize as valid for that interface.

The possible values of the Status field are as follows:

- **0x0:** The switch has this DLCI programmed, but for some reason it is unusable. The reason could possibly be that the other end of the PVC is down.
- **0x2:** The Frame Relay switch has the DLCI, and everything is operational.
- **0x4:** The Frame Relay switch does not have this DLCI programmed for the router, but it was programmed at some point in the past. This could also be caused by the DLCIs being reversed on the router, or by the PVC being deleted by the service provider in the Frame Relay cloud.

As soon as you have gathered all the statistics, use the **clear counters** command to reset the statistics counters. Wait 5 or 10 minutes after clearing the counters before issuing the **show** commands again. Note any additional errors. If you need to contact the carrier, these statistics help resolve the issues.

Verifying Remote IP Address-to-Local DLCI Translation

A final task is to confirm whether the **frame-relay inverse-arp** command resolved a remote IP address to a local DLCI. Use the **show frame-relay map** command to display the current map entries and information about the connections, as shown in Example 3-17.

Example 3-17 **show frame-relay map** Command for Router R1

```
R1# show frame-relay map

Serial0/0/0.102 (up): ip 10.1.1.2 dlci 100(0x64,0x1840), dynamic broadcast,
                    CISCO, status defined, active
```

The output shows the following information:

- 10.1.1.2 is the IP address of the remote router, dynamically learned via the Inverse ARP process.
- 100 is the decimal value of the local DLCI number.
- 0x64 is the hex conversion of the DLCI number; $0x64 = 100$ decimal.
- 0x1840 is the value as it would appear on the wire because of the way the DLCI bits are spread out in the Address field of the Frame Relay frame.
- Broadcast/multicast is enabled on the PVC.
- PVC status is active.

To clear dynamically created Frame Relay maps that are created using Inverse ARP, use the **clear frame-relay inarp** command, as shown in Examples 3-18 and 3-19. It may take several moments for the Inverse ARP process. So you may not see any output until this process is complete, before you issue the **show frame-relay map** command.

Example 3-18 **clear frame-relay inarp** and **show frame-relay map** Commands for Router R1

```
R1# clear frame-relay inarp
R1# show frame-relay map

Serial0/0/0.102 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
                    status defined, active
```

Example 3-19 clear frame-relay inarp and show frame-relay map Commands for Router R2

```
R2# clear frame-relay inarp
R2# show frame-relay map

Serial0/0/0.201 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
                    status defined, active
```

Troubleshooting Frame Relay Configuration

If the verification procedure indicates that your Frame Relay configuration is not working properly, you need to troubleshoot the configuration.

Use the **debug frame-relay lmi** command to determine whether the router and the Frame Relay switch are sending and receiving LMI packets properly.

Look at Examples 3-20 and 3-21. Examine the output of an LMI exchange.

Example 3-20 debug frame-relay lmi Command for Router R1

```
R1# debug frame-relay lmi

Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
R1#
*Sep 12 00:09:35.425: Serial0/0/0(out): StEnq, myseq 110, yourseen 109, DTE up
*Sep 12 00:09:35.425: datagramstart = 0x3F4055D4, datagramsize = 13
*Sep 12 00:09:35.425: FR encap = 0xFCF10309
*Sep 12 00:09:35.425: 00 75 01 01 01 03 02 6E 6D
*Sep 12 00:09:35.425:
*Sep 12 00:09:35.425: Serial0/0/0(in): Status, myseq 110, pak size 13
*Sep 12 00:09:35.425: RT IE 1, length 1, type 1
*Sep 12 00:09:35.425: KA IE 3, length 2, yourseq 110, myseq 110
R1#
*Sep 12 00:09:45.425: Serial0/0/0(out): StEnq, myseq 111, yourseen 110, DTE up
*Sep 12 00:09:45.425: datagramstart = 0x3F4050D4, datagramsize = 13
*Sep 12 00:09:45.425: FR encap = 0xFCF10309
*Sep 12 00:09:45.425: 00 75 01 01 01 03 02 6F 6E
*Sep 12 00:09:45.425:
*Sep 12 00:09:45.425: Serial0/0/0(in): Status, myseq 111, pak size 13
*Sep 12 00:09:45.425: RT IE 1, length 1, type 1
*Sep 12 00:09:45.425: KA IE 3, length 2, yourseq 111, myseq 111
R1# undebg all

All possible debugging has been turned off
R1#
```

Example 3-21 debug frame-relay lmi Command for Router R2

```

R2# debug frame-relay lmi

Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
R2#
*Sep 12 00:07:12.773: Serial0/0/0(out): StEnq, myseq 82, yourseen 81, DTE up
*Sep 12 00:07:12.773: datagramstart = 0x3F401B14, datagramsize = 13
*Sep 12 00:07:12.773: FR encap = 0xFCF10309
*Sep 12 00:07:12.773: 00 75 01 01 01 03 02 52 51
*Sep 12 00:07:12.773:
*Sep 12 00:07:12.773: Serial0/0/0(in): Status, myseq 82, pak size 13
*Sep 12 00:07:12.773: RT IE 1, length 1, type 1
*Sep 12 00:07:12.773: KA IE 3, length 2, yourseq 82, myseq 82
R2#
*Sep 12 00:07:22.773: Serial0/0/0(out): StEnq, myseq 83, yourseen 82, DTE up
*Sep 12 00:07:22.773: datagramstart = 0x3F6AEFD4, datagramsize = 13
*Sep 12 00:07:22.773: FR encap = 0xFCF10309
*Sep 12 00:07:22.773: 00 75 01 01 01 03 02 53 52
*Sep 12 00:07:22.773:
*Sep 12 00:07:22.773: Serial0/0/0(in): Status, myseq 83, pak size 13
*Sep 12 00:07:22.773: RT IE 1, length 1, type 1
*Sep 12 00:07:22.773: KA IE 3, length 2, yourseq 83, myseq 83
*Sep 12 00:07:22.773: PVC IE 0x7 , length 0x3 , dlc1 100, status 0x2
R2# undebg all

All possible debugging has been turned off
R2#

```

Some of the information in the **debug frame-relay lmi** command is explained as follows:

- “out” is an LMI status message sent by the router.
- “in” is a message received from the Frame Relay switch.
- A full LMI status message is a “type 0” (not shown in either example).
- An LMI exchange is a “type 1.”
- “dlci 100, status 0x2” means that the status of DLCI 100 is active.

Summary

Frame Relay provides greater bandwidth, reliability, and resiliency than private or leased lines. Frame Relay has reduced network costs by using less equipment, by being less complex, and by being easier to implement. For these reasons, Frame Relay has become the most widely used WAN technology in the world.

A Frame Relay connection between a DTE device at the LAN edge and a DCE device at the carrier edge has a link layer component and a physical layer component. Frame Relay takes data packets and encapsulates them in a Frame Relay frame and then passes the frame to the physical layer for delivery on the wire. The connection across the carrier network is a VC identified by a DLCI. Multiple VCs can be multiplexed using a FRAD. Frame Relay networks usually use a partial-mesh topology optimized to the data flow requirements of the carrier's customer base.

Frame Relay uses Inverse ARP to map DLCIs to the IP addresses of remote locations. Dynamic address mapping relies on Inverse ARP to resolve a next-hop network protocol address to a local DLCI value. The Frame Relay router sends out Inverse ARP requests on its PVC to discover the protocol address of the remote device connected to the Frame Relay network. DTE Frame Relay routers use the LMI to provide status information about their connection with the DCE Frame Relay switch. LMI extensions provide additional internet-working information.

The first two tasks in configuring Frame Relay on a Cisco router are to enable Frame Relay encapsulation on the interface and then to configure either static or dynamic mapping. After this, a number of optional tasks can be completed as required, including configuring the LMI and VCs, traffic shaping, and customizing Frame Relay on your network. Monitoring and maintaining Frame Relay connections is the final task.

Frame Relay configuration must consider the split-horizon problem that arises when multiple VCs converge on a single physical interface. Frame Relay can partition a physical interface into multiple virtual interfaces called subinterfaces. Subinterface configuration was also explained and practiced.

The configuration of Frame Relay is affected by the way service providers charge for connections using units of access rates and committed information rates (CIR). An advantage of these charging schemes is that unused network capacity is available to or shared with all customers, usually at no extra charge. This allows users to burst traffic for short periods.

Configuring flow control in a Frame Relay network is also affected by service provider charging schemes. You can configure queuing and shape traffic according to the CIR. DTEs can be prompted to control congestion in the network by adding BECN and FECN bits to frame addresses. DTEs can also be configured to set a discard eligible (DE) bit indicating that the frame may be discarded in preference to other frames if congestion occurs. Frames that are sent in excess of the CIR are marked as DE, which means that they can be dropped if congestion occurs within the Frame Relay network.

Finally, after configuring Frame Relay, you learned how to verify and troubleshoot the connections.

Labs

The activities and labs available in the companion *Accessing the WAN, CCNA Exploration Labs and Study Guide* (ISBN 1-58713-201-x) provide hands-on practice with the following topics introduced in this chapter:



Lab 3-1: Basic Frame Relay (3.5.1)

In this lab, you learn how to configure Frame Relay encapsulation on serial links using the network shown in the topology diagram. You also learn how to configure a router as a Frame Relay switch. Both Cisco standards and Open standards apply to Frame Relay. You will learn about both. Pay special attention in the lab section, in which you intentionally break the Frame Relay configurations. This will help you in the Troubleshooting lab associated with this chapter.



Lab 3-2: Challenge Frame Relay Configuration (3.5.2)

In this lab, you configure Frame Relay using the network shown in the topology diagram. If you need assistance, refer to the Basic Frame Relay lab. However, try to do as much on your own as possible.



Lab 3-3: Troubleshooting Frame Relay (3.5.3)

In this lab, you practice troubleshooting a misconfigured Frame Relay environment. Load or have your instructor load the configurations into your routers. Locate and repair all errors in the configurations, and establish end-to-end connectivity. Your final configuration should match the topology diagram and addressing table.



Many of the Hands-on Labs include Packet Tracer Companion Activities where you can use Packet Tracer to complete a simulation of the lab. Look for this icon in the *Accessing the WAN, CCNA Exploration Labs and Study Guide* for Hands-on Labs that have a Packet Tracer Companion.

Check Your Understanding

Complete all the review questions listed here to test your understanding of the topics and concepts in this chapter. Answers are listed in Appendix, “Check Your Understanding and Challenge Questions Answer Key.”

1. What is used to identify the path to the next Frame Relay switch in a Frame Relay network?
 - A. CIR
 - B. DLCI
 - C. FECN
 - D. BECN

2. Why are Frame Relay paths referred to as virtual?
 - A. There are no dedicated circuits to and from the Frame Relay carrier.
 - B. Frame Relay PVCs are created and discarded on demand.
 - C. The connections between PVC endpoints act like dialup circuits.
 - D. There are no dedicated circuits inside the Frame Relay carrier cloud.

3. Which statement accurately describes the split-horizon problem with regard to a multipoint topology?
 - A. Split horizon must be disabled for all non-IP protocols.
 - B. Split horizon creates IP routing loops in multipoint domains.
 - C. Split horizon does not apply to broadcasts, so it does not protect protocols that use broadcast updates.
 - D. Split horizon prevents any interface from accepting a valid update and forwarding to all the other interfaces.

4. Why is Frame Relay more cost-effective than leased lines? (Choose two.)
 - A. Time division multiplexing
 - B. It uses less equipment.
 - C. Optimized packet routing
 - D. It shares bandwidth across a large customer base.
 - E. Dynamic IP addressing

5. Match each status from the **show frame-relay pvc** command with its meaning:

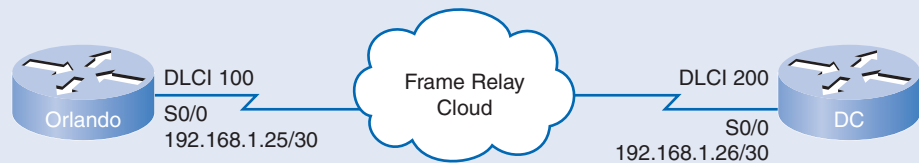
Active

Inactive

Deleted

- A. The DLCI is programmed in the Frame Relay switch, but the other end of the PVC may be down.
 - B. The DLCI does not exist on the router.
 - C. The DLCI is programmed in the Frame Relay switch and is usable.
 - D. The DLCI does not exist on the Frame Relay switch for this router.
 - E. The DLCI is programmed on the other end of the PVC for this router.
6. What reliability advantage does Frame Relay offer over leased lines?
- A. Frame Relay access circuits are higher-grade circuits than leased lines.
 - B. The pathways for virtual circuits inside the carrier are meshed.
 - C. From end to end, a single virtual circuit uses a fixed error-checked path.
 - D. Frame Relay uses more-sophisticated error-detection methods.
7. Refer to Figure 3-33. What is placed in the Address field of a frame that will travel from the Orlando office to the DC office?

Figure 3-33 Network Topology for Question 7



- A. MAC address of the Orlando router
- B. MAC address of the DC router
- C. 192.168.1.25
- D. 192.168.1.26
- E. DLCI 100
- F. DLCI 200

8. Which situation favors a multipoint topology over point-to-point?
 - A. When VLSM cannot be used to conserve addresses
 - B. When using routing protocols other than IP
 - C. When using a frame mesh topology to save access circuits
 - D. When using a routing protocol that requires broadcast updates
9. What is an advantage of configuring subinterfaces in a Frame Relay environment?
 - A. It makes the DLCIs globally significant.
 - B. It eliminates the need for using Inverse ARP.
 - C. It solves split-horizon issues.
 - D. It improves flow control and bandwidth usage.
10. Which protocol can provide error correction for data that is transmitted over a Frame Relay link?
 - A. FECN
 - B. FTP
 - C. LMI
 - D. TCP
 - E. UDP

11. Match each command with its description:

show interface

show frame-relay lmi

show frame-relay pvc

show frame-relay map

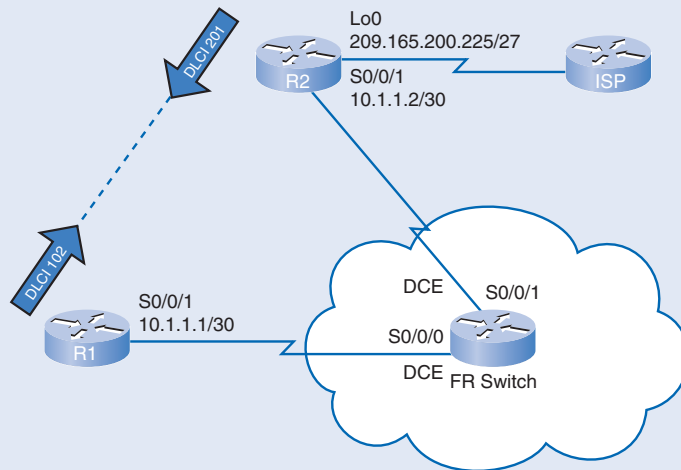
debug frame-relay lmi

- A. Shows the status of the virtual circuit and FECN/BECN statistics
- B. Verifies that the router and Frame Relay switch are sending and receiving LMI packets properly
- C. Verifies encapsulation, LMI type, LMI DLCI, and LMI status
- D. Verifies LMI statistics
- E. Verifies the destination IP address mapping to the DLCI

-
- 12.** At which rate does a service provider guarantee to transfer data into the Frame Relay network?
- A. Baud rate
 - B. Timing rate
 - C. Data transfer rate
 - D. Committed information rate
- 13.** How are DLCI numbers assigned?
- A. They are assigned by a DLCI server.
 - B. They are assigned arbitrarily by the user.
 - C. They are assigned by the service provider.
 - D. They are assigned based on the host IP address.
- 14.** A router can reach multiple networks through a Frame Relay interface. How does the router know which DLCI to assign to the IP address of the destination network?
- A. It consults the Frame Relay map.
 - B. It consults the routing table to find the DLCI.
 - C. It uses Frame Relay switching tables to map DLCIs to IP addresses.
 - D. It uses RARP to find the IP address of the corresponding DLCI.
- 15.** Match each term with its definition:
- CIR
 - DE
 - FECN
 - BECN
- A. A bit that marks the frame to be dropped when congestion is present
 - B. A bit set on every frame that a switch places on a congested link
 - C. The rate at which the service provider agrees to accept bits on the VC
 - D. A bit set on every frame that a switch receives on a congested link
- 16.** Compare and contrast the following terms: DLCI, LMI, and Inverse ARP.
- 17.** Refer to Figure 3-34.

Figure 3-34 Network Topology for Question 17

```
interface s0/0/1
ip address 10.1.1.1 255.255.255.252
encapsulation frame-relay
bandwidth 64
```



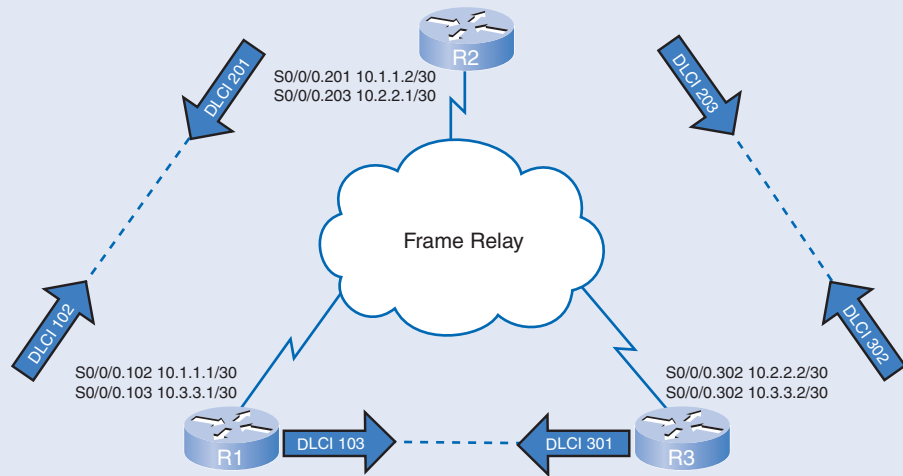
The following configuration is for router R1:

```
interface s0/0/1
ip address 10.1.1.1 255.255.255.252
encapsulation frame-relay
bandwidth 64
```

What command on R1 is required to statically configure a Frame Relay connection to R2? Traffic between sites must also support OSPF.

18. Compare and contrast the following terms: access rate, CIR, CBIR, and BE.
19. Refer to Figure 3-35 and the following configuration. R1 cannot establish connectivity with the routers over the Frame Relay cloud. What are the problems with this configuration?

Figure 3-35 Network Topology for Question 19



```

hostname R1
interface s0/0/1
 encapsulation frame-relay
 !
interface s0/0/1.201 point-to-point
 ip address 10.1.1.1 255.255.255.0
 frame-relay interface-dlci 201
 !
interface s0/0/1.301 point-to-point
 ip address 10.3.3.1 255.255.255.0
 frame-relay interface-dlci 301

```

```

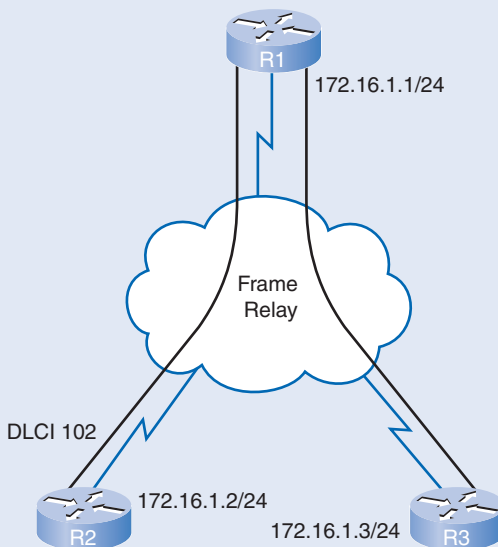
hostname R1
interface s0/0/1
 encapsulation frame-relay
 !
interface s0/0/1.201 point-to-point
 ip address 10.1.1.1 255.255.255.0
 frame-relay interface-dlci 201
 !
interface s0/0/1.301 point-to-point
 ip address 10.3.3.1 255.255.255.0
 frame-relay interface-dlci 301
 !

```

Challenge Questions and Activities

1. Refer to Figure 3-36. Routers R2 and R3 both have Frame Relay connections to the R1 router. R1 is acting as the hub router in this hub-and-spoke topology. R2 is using Inverse ARP to map its DLCI with R1's network address.

Figure 3-36 Network Topology for Challenge Question 1



To reach R3, the network administrator configured the following static map on R2:

```
frame-relay map ip 172.16.1.3 102
```

This seems to work, giving R2 reachability to both R1 and R3 (assuming that R3 has also been appropriately configured). However, when the router is reloaded (the running-config was saved to the startup config), R2 can no longer reach R1. The **show frame-relay map** command on R2 shows only the static map to R3; it no longer shows the dynamic Inverse ARP map to R1. What happened? What would be the solution?

To Learn More

Here are some suggested books for further reading on Frame Relay:

- *Cisco Frame Relay Solutions Guide* by Jonathan Chin (Cisco Press, 2004)
- *Frame Relay: Technology and Practice* by Jeff T. Buckwalter (Addison-Wesley Professional, 1999)