



This chapter covers the following subjects:

- Frame Relay Protocols
- Frame Relay Configuration

Frame Relay

Frame Relay is the most popular WAN technology used today, so it's no surprise that Frame Relay is an important topic on the ICND exam. This chapter reviews the details of how Frame Relay accomplishes its goal of delivering frames to multiple WAN-connected sites.

Frame Relay most closely compares to the OSI data link layer (Layer 2). If you remember that the word “frame” describes the data link layer protocol data unit (PDU), it will be easy to remember that Frame Relay relates to OSI Layer 2. Like other data-link protocols, Frame Relay can be used to deliver packets (Layer 3 PDUs) between routers. Frame Relay protocol headers and trailers are simply used to let a packet traverse the Frame Relay network, just like Ethernet headers and trailers are used to help a packet traverse an Ethernet segment.

This chapter describes the Frame Relay protocol details, along with the associated configuration.

“Do I Know This Already?” Quiz

The purpose of the “Do I Know This Already?” quiz is to help you decide if you need to read the entire chapter. If you intend to read the entire chapter, you do not necessarily need to answer these questions now.

The ten-question quiz, derived from the major sections in the “Foundation Topics” section, helps you determine how to spend your limited study time.

Table 11-1 outlines the major topics discussed in this chapter and the “Do I Know This Already?” quiz questions that correspond to those topics.

Table 11-1 “Do I Know This Already?” Foundation Topics Section-to-Question Mapping

Foundations Topics Section	Questions Covered in This Section
Frame Relay Concepts	1–6
Frame Relay Configuration	7–10

CAUTION The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you don’t know the answer to a question or you’re only partially sure of the answer, you should mark this question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you guess correctly skews your self-assessment results and might give you a false sense of security.

1. Which of the following defines a protocol used between the Frame Relay DTE and the Frame Relay switch in the service provider’s network?
 - a. VC
 - b. CIR
 - c. LMI
 - d. Q.921
 - e. DLCI
 - f. FRF.5
 - g. Encapsulation
 - h. None of the above

2. Which of the following defines a protocol or feature that matters to what the provider might do inside its network but that is transparent to the DTE/router using the Frame Relay service?
 - a. VC
 - b. CIR
 - c. LMI
 - d. DLCI
 - e. Q.921
 - f. FRF.5

6. BarneyCo has five sites, with routers connected to the same Frame Relay network. VCs have been defined between each pair of routers. Barney, the president of the company, will fire anyone who configures Frame Relay without using point-to-point subinterfaces. What is the fewest subnets that BarneyCo could use on the Frame Relay network?
- a. 1
 - b. 4
 - c. 8
 - d. 10
 - e. 12
 - f. 15
7. BettyCo has five sites, with routers connected to the same Frame Relay network. VCs have been defined between each pair of routers. Betty, the president of the company, will fire anyone who configures anything that could just as easily be left as a default. Which of the following configuration commands, configured for the Frame Relay network, would get the engineer fired?
- a. `ip address`
 - b. `encapsulation`
 - c. `lmi-type`
 - d. `frame-relay map`
 - e. `inverse-arp`
8. WilmaCo has some routers connected to a Frame Relay network. R1 is a router at a remote site, with a single VC back to WilmaCo's headquarters. The R1 configuration currently looks like this:

```
interface serial 0/0
  ip address 10.1.1.1 255.255.255.0
  encapsulation frame-relay
```

Wilma, the president, has heard that point-to-point subinterfaces are cool, and she wants you to change the configuration to use a point-to-point subinterface. Which of the following commands do you need to use to migrate the configuration?

- a. `no ip address`
- b. `interface-dlci`
- c. `no encapsulation`
- d. `encapsulation frame-relay`
- e. `frame-relay interface-dlci`

9. WilmaCo has another network, with a main site router that has ten VCs connecting to the ten remote sites. Wilma now thinks that multipoint subinterfaces are even cooler than point-to-point. The current main site router’s configuration looks like this:

```
interface serial 0/0
  ip address 172.16.1.1 255.255.255.0
  encapsulation frame-relay
```

Wilma wants you to change the configuration to use a multipoint subinterface. Which of the following do you need to use to migrate the configuration? (Note: DLCIs 101 through 110 are used for the ten VCs.)

- a. `interface-dlci 101 110`
 - b. `interface dlci 101-110`
 - c. Ten different `interface-dlci` commands
 - d. `frame-relay interface-dlci 101 110`
 - e. `frame-relay interface dlci 101-110`
 - f. Ten different `frame-relay interface-dlci` commands
10. Which of the following commands lists the information learned by Inverse ARP?
- a. `show ip arp`
 - b. `show arp`
 - c. `show inverse arp`
 - d. `show frame-relay inverse-arp`
 - e. `show map`
 - f. `show frame-relay map`

The answers to the “Do I Know This Already?” quiz appear in Appendix A. The suggested choices for your next step are as follows:

- **8 or less overall score**—Read the entire chapter. This includes the “Foundation Topics,” “Foundation Summary,” and “Q&A” sections.
- **9 or 10 overall score**—If you want more review on these topics, skip to the “Foundation Summary” section and then go to the “Q&A” section. Otherwise, move to the next chapter.

Foundation Topics

Frame Relay Protocols

If you're using both books, in Chapter 4, "Fundamentals of WANs," of CCNA INTRO Exam Certification Guide, you read about the basics of Frame Relay. This chapter begins with a brief review of those concepts, and then it dives into the details of Frame Relay protocols and concepts. This chapter ends with coverage of Frame Relay configuration.

Frame Relay networks provide more features and benefits than simple point-to-point WAN links, but to do that, Frame Relay protocols are more detailed. For example, Frame Relay networks are multiaccess networks, which means that more than two devices can attach to the network, similar to LANs. Unlike LANs, you cannot send a data link layer broadcast over Frame Relay. Therefore, Frame Relay networks are called *nonbroadcast multiaccess (NBMA)* networks. Also, because Frame Relay is multiaccess, it requires the use of an address that identifies to which remote router each frame is addressed.

Figure 11-1 outlines the basic physical topology and related terminology in a Frame Relay network.

Figure 11-1 *Frame Relay Components*

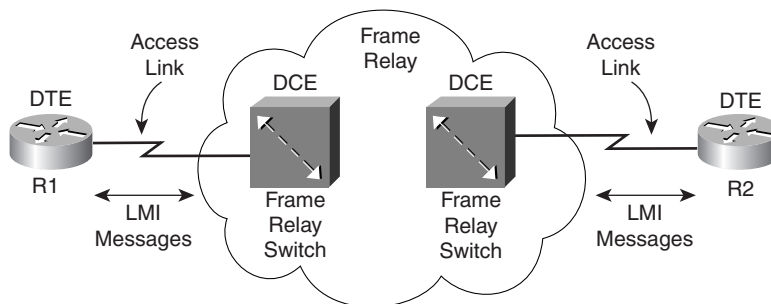
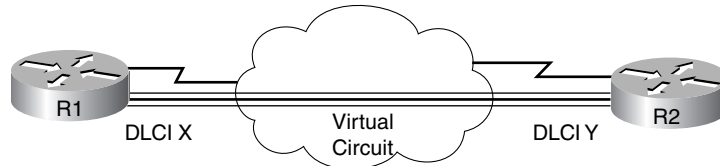


Figure 11-1 shows the most basic components of a Frame Relay network. A leased line is installed between the router and a nearby Frame Relay switch; this link is called the *access link*. To ensure that the link is working, the device outside the Frame Relay network, called the *data terminal equipment (DTE)*, exchanges regular messages with the Frame Relay switch. These keepalive messages, along with other messages, are defined by the Frame Relay *Local Management Interface (LMI)* protocol. The routers are considered DTE, and the Frame Relay switches are *data communications equipment (DCE)*.

Whereas Figure 11-1 shows the physical connectivity at each connection to the Frame Relay network, Figure 11-2 shows the logical, or virtual, end-to-end connectivity associated with a virtual circuit (VC).

Figure 11-2 *Frame Relay PVC Concepts*



The logical communications path between each pair of DTEs is a VC. The trio of parallel lines in the figure represents a single VC; this book uses a different line style to make sure you notice the line easily. Typically, the service provider preconfigures all the required details of a VC; predefined VCs are called permanent virtual circuits (PVCs).

Routers use the data-link connection identifier (DLCI) as the Frame Relay address, which identifies the VC over which the frame should travel. So, in Figure 11-2, when R1 needs to forward a packet to R2, it encapsulates the Layer 3 packet into a Frame Relay header and trailer and then sends the frame. The Frame Relay header includes the correct DLCI so that the provider's Frame Relay switches correctly forward the frame to R2.

Table 11-2 lists the components shown in Figure 11-1 and some associated terms. After the tables, the most important features of Frame Relay are described in further detail.

Table 11-2 *Frame Relay Terms and Concepts*

Term	Description
Virtual circuit (VC)	A logical concept that represents the path that frames travel between DTEs. VCs are particularly useful when comparing Frame Relay to leased physical circuits.
Permanent virtual circuit (PVC)	A predefined VC. A PVC can be equated to a leased line in concept.
Switched virtual circuit (SVC)	A VC that is set up dynamically when needed. An SVC can be equated to a dial connection in concept.
Data terminal equipment (DTE)	DTEs are connected to a Frame Relay service from a telecommunications company and typically reside at sites used by the company buying the Frame Relay service.

continues

Table 11-2 *Frame Relay Terms and Concepts (Continued)*

Term	Description
Data communications equipment (DCE)	Frame Relay switches are DCE devices. DCEs are also known as data circuit-terminating equipment. DCEs are typically in the service provider's network.
Access link	The leased line between the DTE and DCE.
Access rate (AR)	The speed at which the access link is clocked. This choice affects the connection's price.
Data-link connection identifier (DLCI)	A Frame Relay address used in Frame Relay headers to identify the VC.
Nonbroadcast multiaccess (NBMA)	A network in which broadcasts are not supported, but more than two devices can be connected.
Local Management Interface (LMI)	The protocol used between a DCE and DTE to manage the connection. Signaling messages for SVCs, PVC status messages, and keepalives are all LMI messages.

Frame Relay Standards

The definitions for Frame Relay are contained in documents from the International Telecommunications Union (ITU) and the American National Standards Institute (ANSI). The Frame Relay Forum (www.frforum.com), a vendor consortium, also defines several Frame Relay specifications, many of which predate the original ITU and ANSI specifications, with the ITU and ANSI picking up many of the forum's standards. Table 11-3 lists the most important of these specifications.

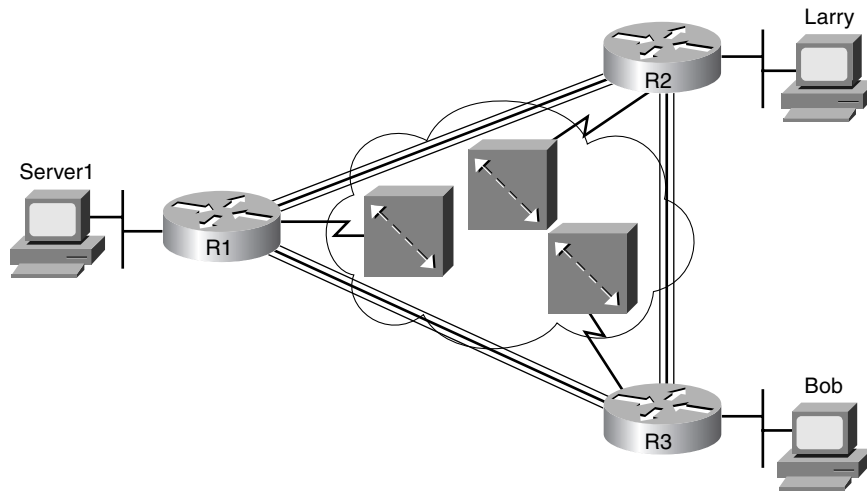
Table 11-3 *Frame Relay Protocol Specifications*

What the Specification Defines	ITU Document	ANSI Document
Data-link specifications, including LAPF header/trailer	Q.922 Annex A (Q.922-A)	T1.618
PVC management, LMI	Q.933 Annex A (Q.933-A)	T1.617 Annex D (T1.617-D)
SVC signaling	Q.933	T1.617
Multiprotocol encapsulation (originated in RFC 1490/2427)	Q.933 Annex E (Q.933-E)	T1.617 Annex F (T1.617-F)

Virtual Circuits

Frame Relay provides significant advantages over simply using point-to-point leased lines. The primary advantage has to do with virtual circuits. Consider Figure 11-3, which is a typical Frame Relay network with three sites.

Figure 11-3 Typical Frame Relay Network with Three Sites



A virtual circuit defines a logical path between two Frame Relay DTEs. The term *virtual circuit* describes the concept well. It acts like a point-to-point circuit, providing the ability to send data between two endpoints over a WAN. There is no physical circuit directly between the two endpoints, so it's virtual. For example, R1 terminates two VCs—one whose other endpoint is R2, and one whose other endpoint is R3. R1 can send traffic directly to either of the other two routers by sending it over the appropriate VC. R1 has only one physical access link to the Frame Relay network.

VCS share the access link and the Frame Relay network. For example, both VCs terminating at R1 use the same access link. In fact, many customers share the same Frame Relay network. Originally, people with leased-line networks were reluctant to migrate to Frame Relay, because they would be competing with other customers for the provider's capacity inside the cloud. To address these fears, Frame Relay is designed with the concept of a committed information rate (CIR). Each VC has a CIR, which is a guarantee by the provider that a particular VC gets at least that much bandwidth. So you can migrate from a leased line to Frame Relay, getting a CIR of at least as much bandwidth as you previously had with your leased line.

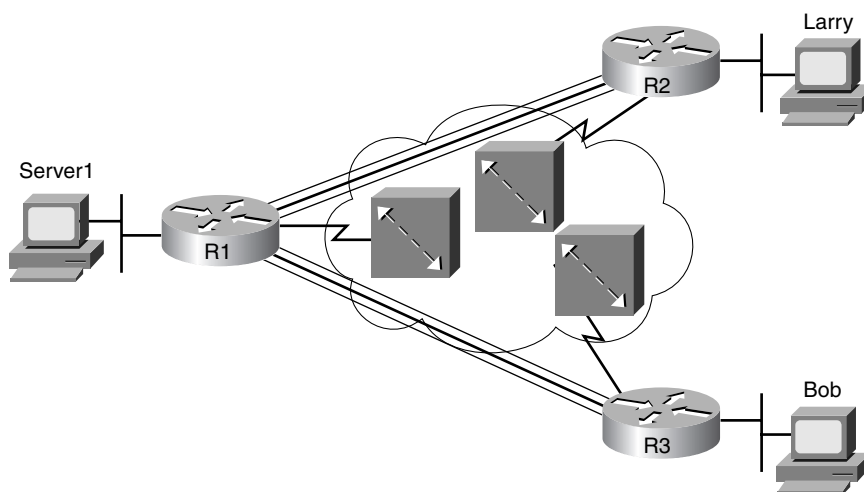
Interestingly, even with a three-site network, it's probably less expensive to use Frame Relay than to use point-to-point links. Imagine an organization with 100 sites that needs any-to-any connectivity. How many leased lines are required? 4950! And besides that, the organization would need 99 serial interfaces per router if it used point-to-point leased lines. With Frame Relay, an organization could have 100 access links to local Frame Relay switches, one per router, and have 4950 VCs running over them. That requires a lot fewer actual physical links, and you would need only one serial interface on each router!

Service providers can build their Frame Relay networks more cost-effectively than for leased lines. As you would expect, that makes it less expensive to the Frame Relay customer as well. For connecting many WAN sites, Frame Relay is simply more cost-effective than leased lines.

Two types of VCs are allowed—permanent (PVC) and switched (SVC). PVCs are predefined by the provider; SVCs are created dynamically. PVCs are by far the more popular of the two.

When the Frame Relay network is engineered, the design might not include a PVC between each pair of sites. Figure 11-3 includes PVCs between each pair of sites, which is called a full-mesh Frame Relay network. When not all pairs have a direct PVC, it is called a partial-mesh network. Figure 11-4 shows the same network as Figure 11-3, but this time with a partial mesh and only two PVCs. This is typical when R1 is at the main site and R2 and R3 are at remote offices that rarely need to communicate directly.

Figure 11-4 *Typical Partial-Mesh Frame Relay Network*



The partial mesh has some advantages and disadvantages when compared to a full mesh. The primary advantage is that partial mesh is cheaper, because the provider charges per VC. The

downside is that traffic from R2's site to R3's site must go to R1 first and then be forwarded. If that's a small amount of traffic, it's a small price to pay. If it's a lot of traffic, a full mesh is probably worth the extra money.

One conceptual hurdle with PVCs is that there is typically a single access link across which multiple PVCs flow. For example, consider Figure 11-4 from R1's perspective. Server1 sends a packet to Larry. It comes across the Ethernet. R1 gets it and matches Larry's routing table, which tells him to send the packet out Serial0, which is R1's access link. He encapsulates the packet in a Frame Relay header and trailer and then sends it. Which PVC does it use? The Frame Relay switch should forward it to R2, but why does it?

Frame Relay uses an address to differentiate one PVC from another. This address is called a data-link connection identifier (DLCI). The name is descriptive: The address is for an OSI Layer 2 (data link) protocol, and it identifies a VC, which is sometimes called a *virtual connection*. So, in this example, R1 uses the DLCI that identifies the PVC to R2, so the provider forwards the frame correctly over the PVC to R2. To send frames to R3, R1 uses the DLCI that identifies the VC for R3. DLCIs and how they work are covered in more detail later in this chapter.

LMI and Encapsulation Types

When you're first learning about Frame Relay, it's often easy to confuse the LMI and the encapsulation used with Frame Relay. The LMI is a definition of the messages used between the DTE (for example, a router) and the DCE (for example, the Frame Relay switch owned by the service provider). The encapsulation defines the headers used by a DTE to communicate some information to the DTE on 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.

The most important LMI message relating to topics on the exam is the LMI status inquiry message. Status messages perform two key functions:

- Perform a keepalive function between the DTE and DCE. If the access link has a problem, the absence of keepalive messages implies that the link is down.
- Signal whether a PVC is active or inactive. Even though each PVC is predefined, its status can change. An access link might be up, but one or more VCs could be down. The router needs to know which VCs are up and which are down. It learns that information from the switch using LMI status messages.

Three LMI protocol options are available in Cisco IOS software: Cisco, ITU, and ANSI. Each LMI option is slightly different and therefore is incompatible with the other two. As long as both the DTE and DCE on each end of an access link use the same LMI standard, LMI works fine.

The differences between LMI types are subtle. For example, the Cisco LMI calls for the use of DLCI 1023, whereas ANSI T1.617-D and ITU Q.933-A specify DLCI 0. Some of the messages have different fields in their headers. The DTE simply needs to know which of the three LMIs to use so that it can use the same one as the local switch.

Configuring the LMI type is easy. Today's most popular option is to use the default LMI setting, which uses the LMI autosense feature, in which the router simply figures out which LMI type the switch is using. So you can simply let the router autosense the LMI and never bother coding the LMI type. If you choose to configure the LMI type, it disables the autosense feature.

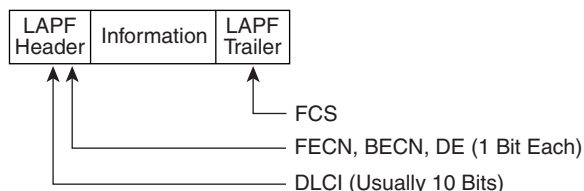
Table 11-4 outlines the three LMI types, their origin, and the keyword used in the Cisco IOS software `frame-relay lmi-type` interface subcommand.

Table 11-4 *Frame Relay LMI Types*

Name	Document	IOS LMI-Type Parameter
Cisco	Proprietary	<code>cisco</code>
ANSI	T1.617 Annex D	<code>ansi</code>
ITU	Q.933 Annex A	<code>q933a</code>

A Frame Relay-connected router encapsulates each Layer 3 packet inside a Frame Relay header and trailer before it is sent out an access link. The header and trailer are defined by the Link Access Procedure Frame Bearer Services (LAPF) specification, ITU Q.922-A. The sparse LAPF framing provides error detection with an FCS in the trailer, as well as the DLCI, DE, FECN, and BECN fields in the header (which are discussed later). Figure 11-5 diagrams the frame.

Figure 11-5 *LAPF Header*



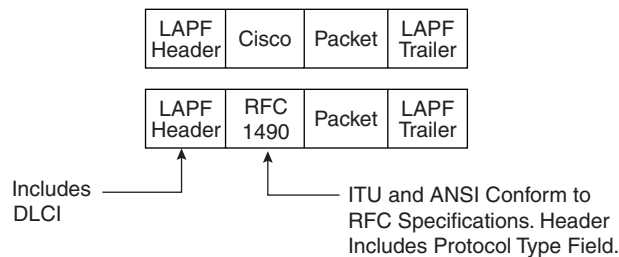
However, the LAPF header and trailer do not provide all the fields typically needed by routers. In particular, Figure 11-5 does not show a Protocol Type field. As discussed in Chapters 3, “Virtual LANs and Trunking,” and 4, “IP Addressing and Subnetting,” a field in the data-link header must define the type of packet that follows the data-link header. If Frame Relay is using only the LAPF header, DTEs (including routers) cannot support multiprotocol traffic, because there is no way to identify the type of protocol in the Information field.

Two solutions were created to compensate for the lack of a Protocol Type field in the standard Frame Relay header:

- Cisco and three other companies created an additional header, which comes between the LAPF header and the Layer 3 packet shown in Figure 11-5. It includes a 2-byte Protocol Type field, with values matching the same field used for HDLC by Cisco.
- RFC 1490 (which was later superseded by RFC 2427—you should know both numbers), “Multiprotocol Interconnect over Frame Relay,” defined the second solution. RFC 1490 was written to ensure multivendor interoperability between Frame Relay DTEs. This RFC defines a similar header, also placed between the LAPF header and Layer 3 packet, and includes a Protocol Type field as well as many other options. ITU and ANSI later incorporated RFC 1490 headers into their Q.933 Annex E and T1.617 Annex F specifications, respectively.

Figure 11-6 outlines these two alternatives.

Figure 11-6 Cisco and RFC 1490/2427 Encapsulation



DTEs use and react to the fields specified by these two types of encapsulation, but Frame Relay switches ignore these fields. *Because the frames flow from DTE to DTE, both DTEs must agree to the encapsulation used. The switches don't care.* However, each VC can use a different encapsulation. In the configuration, the encapsulation created by Cisco is called **cisco**, and the other one is called **ietf**.

DLCI Addressing Details

So far, you know some basic information about Frame Relay. First, the routers (DTEs) connect to the Frame Relay switches (DCEs) over an access link, which is a leased line between the router and the switch. The logical path between a pair of DTEs is called a virtual circuit (VC). Most networks use permanent virtual circuits (PVCs) instead of switched virtual circuits (SVCs), and the data-link connection identifier (DLCI) addresses or identifies each individual PVC. The LMI protocol manages the access link, and the LMI type must match between the router and the local switch. Finally, the routers on either end of each VC must agree on the style of encapsulation used. Both encapsulation types include a Protocol Type field, which identifies the header that follows the Frame Relay header.

DLCIs can be both simple and confusing. It was just stated that the DLCI identifies a VC, so when multiple VCs use the same access link, the Frame Relay switches know how to forward the frames to the correct remote sites. You could know just that, look at the configuration examples later in this chapter, and probably learn to create new configurations. However, a closer look at DLCIs shows how they really work. This is important for actually understanding the configurations you create. If you want to get a deeper understanding, read on. If you prefer to get the basics right now and fill in more details later, you might want to jump ahead to the “Frame Relay Configuration” section.

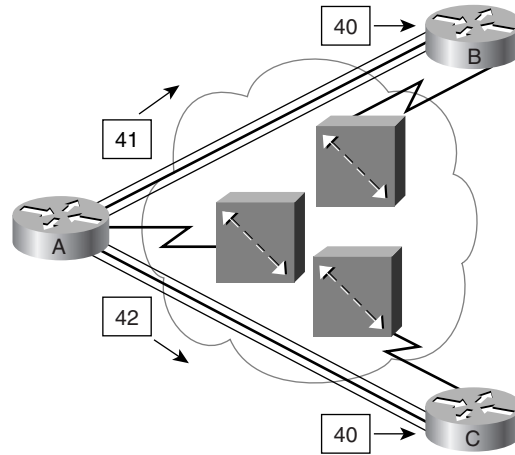
Frame Relay addressing and switching define how to deliver frames across a Frame Relay network. Because a router uses a single access link that has many VCs connecting it to many routers, there must be something to identify each of the remote routers—in other words, an address. The DLCI is the Frame Relay address.

DLCIs work slightly differently from the other data-link addresses covered on the CCNA exams. This difference is mainly because of the use of the DLCI and the fact that *the header has a single DLCI field, not both Source and Destination DLCI fields.*

A few characteristics of DLCIs are important to understand before getting into their use. Frame Relay DLCIs are locally significant; this means that the addresses need to be unique only on the local access link. A popular analogy that explains local addressing is that there can be only a single street address of 2000 Pennsylvania Avenue, Washington, DC, but there can be a 2000 Pennsylvania Avenue in every town in the United States. Likewise, DLCIs must be unique on each access link, but the same DLCI numbers can be used on every access link in your network. For example, in Figure 11-7, notice that DLCI 40 is used on two access links to describe two different PVCs. No conflict exists, because DLCI 40 is used on two different access links.

Local addressing, which is the common term for the fact that DLCIs are locally significant, is a fact. It is how Frame Relay works. Simply put, a single access link cannot use the same DLCI to represent multiple VCs on the same access link. Otherwise, the Frame Relay switch would not know how to forward frames correctly. For instance, in Figure 11-7, R1 must use different DLCI values for the PVCs on its local access link (41 and 42 in this instance).

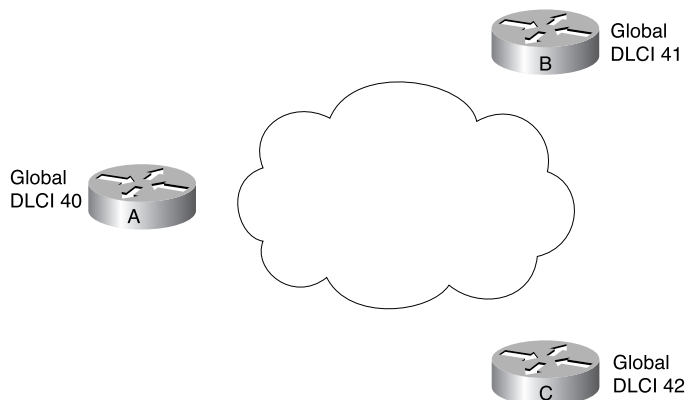
Figure 11-7 *Frame Relay Addressing with Router A Sending to Routers B and C*



Most people get confused about DLCIs the first time they think about the local significance of DLCIs and the existence of only a single DLCI field in the Frame Relay header. Global addressing solves this problem by making DLCI addressing look like LAN addressing in concept. Global addressing is simply a way of choosing DLCI numbers when planning a Frame Relay network so that working with DLCIs is much easier. Because local addressing is a fact, global addressing does not change these rules. Global addressing just makes DLCI assignment more obvious—as soon as you get used to it.

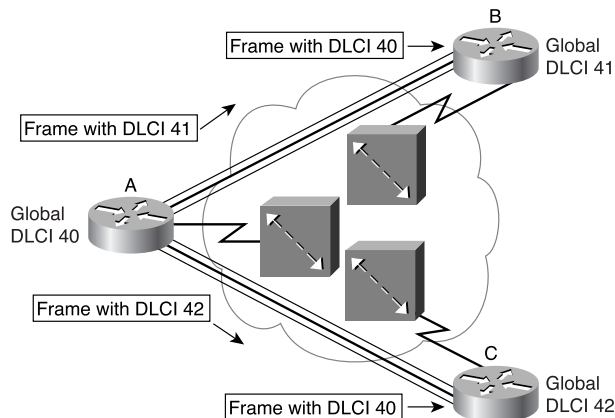
Here’s how global addressing works: The service provider hands out a planning spreadsheet and a diagram. Figure 11-8 is an example of such a diagram, with the global DLCIs shown.

Figure 11-8 *Frame Relay Global DLCIs*



Global addressing is planned as shown in Figure 11-8, with the DLCIs placed in Frame Relay frames as shown in Figure 11-9. For example, Router A uses DLCI 41 when sending a frame to Router B, because Router B's global DLCI is 41. Likewise, Router A uses DLCI 42 when sending frames over the VC to Router C. The nice thing is that global addressing is much more logical to most people, because it works like a LAN, with a single MAC address for each device. On a LAN, if the MAC addresses are MAC-A, MAC-B, and MAC-C for the three routers, Router A uses destination address MAC-B when sending frames to Router B and MAC-C as the destination to reach Router C. Likewise, with global DLCIs 40, 41, and 42 used for Routers A, B, and C, respectively, the same concept applies. Because DLCIs address VCs, the logic is something like this when Router A sends a frame to Router B: "Hey, local switch! When you get this frame, send it over the VC that we agreed to number with DLCI 41." Figure 11-9 outlines this example.

Figure 11-9 *Frame Relay Global Addressing from the Sender's Perspective*



Router A sends frames with DLCI 41, and they reach the local switch. The local switch sees the DLCI field and forwards the frame through the Frame Relay network until it reaches the switch connected to Router B. Then Router B's local switch forwards the frame out the access link to Router B. The same process happens between Router A and Router C when Router A uses DLCI 42. The beauty of global addressing is that you think of each router as having an address, like LAN addressing. If you want to send a frame to someone, you put his or her DLCI in the header, and the network delivers the frame to the correct DTE.

The final key to global addressing is that the Frame Relay switches actually change the DLCI value before delivering the frame. Did you notice that Figure 11-9 shows a different DLCI value as the frames are received by Routers B and C? For example, Router A sends a frame to Router B, and Router A puts DLCI 41 in the frame. The last switch changes the field to DLCI 40 before forwarding it to Router B. The result is that when Routers B and C receive their frames, the DLCI value is actually the sender's DLCI. Why? Well, when Router B

receives the frame, because the DLCI is 40, it knows that the frame came in on the PVC between itself and Router A. In general, the following are true:

- The sender treats the DLCI field as a destination address, using the destination's global DLCI in the header.
- The receiver thinks of the DLCI field as the source address, because it contains the global DLCI of the frame's sender.

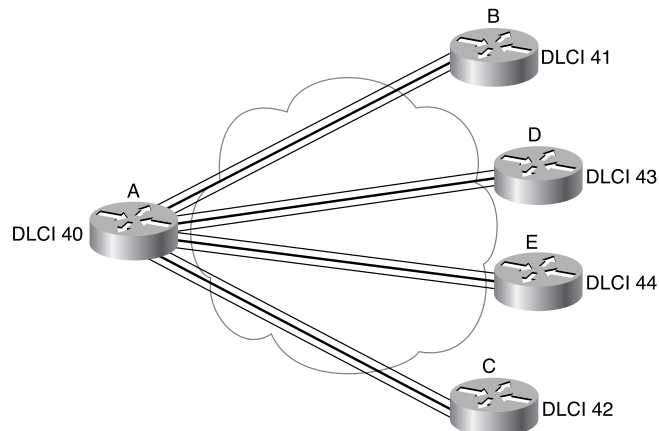
Figure 11-9 describes what happens in a typical Frame Relay network. Service providers supply a planning spreadsheet and diagrams with global DLCIs listed. Table 11-5 gives you an organized view of what DLCIs are used in Figure 11-9.

Table 11-5 *DLCI Swapping in the Frame Relay Cloud of Figure 11-9*

Frame Sent by Router	With DLCI Field	Is Delivered to Router	With DLCI Field
A	41	B	40
A	42	C	40
B	40	A	41
C	40	A	42

Global addressing makes DLCI addressing more intuitive to most people. It also makes router configuration more straightforward and lets you add new sites more conveniently. For instance, examine Figure 11-10, which adds Routers D and E to Figure 11-9's network. The service provider simply states that global DLCIs 43 and 44 are used for these two routers. If these two routers also have only one PVC to Router A, all the DLCI planning is complete. You know that Router D and Router E use DLCI 40 to reach Router A and that Router A uses DLCI 43 to reach Router D and DLCI 44 to reach Router E.

Figure 11-10 *Adding Frame Relay Sites: Global Addressing*



The remaining examples in this chapter use global addressing in any planning diagrams unless otherwise stated. One practical way to determine whether the diagram lists the local DLCIs or the global DLCI convention is this: If two VCs terminate at the same DTE, and a single DLCI is shown, it probably represents the global DLCI convention. If one DLCI is shown per VC, local DLCI addressing is depicted.

Network Layer Concerns with Frame Relay

Most of the important Frame Relay concepts have been covered. First, the routers (DTEs) connect to the Frame Relay switches (DCEs) over an access link, which is a leased line between the router and the switch. The LMI protocol is used to manage the access link, and the LMI type must match between the router and the local switch. The routers agree to the style of encapsulation used. The single DLCI field in the Frame Relay header identifies the VC used to deliver the frame. The DLCI is used like a destination address when the frame is being sent and like a source address as the frame is received. The switches actually swap the DLCI in transit.

Frame Relay networks have both similarities and differences as compared to LAN and point-to-point WAN links. These differences introduce some additional considerations for passing Layer 3 packets across a Frame Relay network. You need to concern yourself with a couple of key issues relating to Layer 3 flows over Frame Relay:

- Choices for Layer 3 addresses on Frame Relay interfaces
- Broadcast handling

The following sections cover these issues in depth.

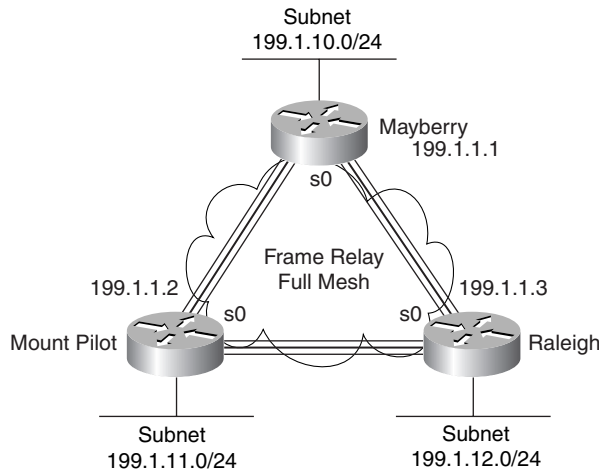
Layer 3 Addressing with Frame Relay

Cisco's Frame Relay implementation defines three different options for assigning subnets and IP addresses on Frame Relay interfaces:

- One subnet containing all Frame Relay DTEs
- One subnet per VC
- A hybrid of the first two options

Frame Relay Layer 3 Addressing: One Subnet Containing All Frame Relay DTEs

Figure 11-11 shows the first alternative, which is to use a single subnet for the Frame Relay network. The figure shows a fully meshed Frame Relay network because the single-subnet option is typically used when a full mesh of VCs exists. In a full mesh, each router has a VC to every other router, meaning that each router can send frames directly to every other router—which more closely equates to how a LAN works. So, a single subnet can be used for all the routers' Frame Relay interfaces, as configured on the routers' serial interfaces. Table 11-6 summarizes the addresses used in Figure 11-11.

Figure 11-11 *Full Mesh with IP Addresses***Table 11-6** *IP Addresses with No Subinterfaces*

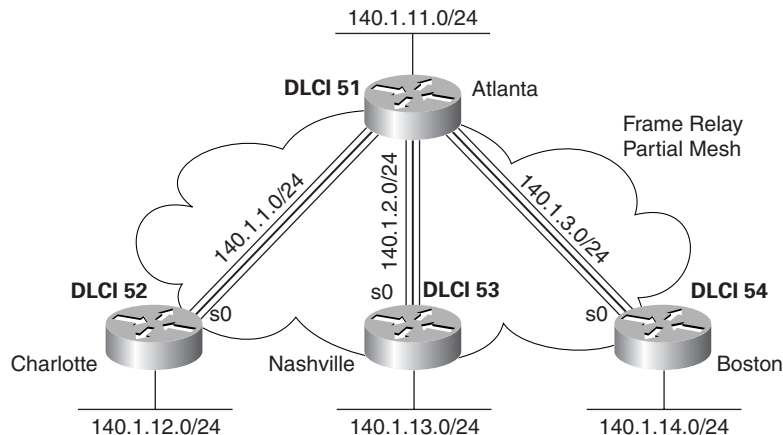
Router	IP Address of Frame Relay Interface
Mayberry	199.1.1.1
Mount Pilot	199.1.1.2
Raleigh	199.1.1.3

The single-subnet alternative is straightforward, and it conserves your IP address space. It also looks like what you are used to with LANs, which makes it easier to conceptualize. Unfortunately, most companies build partial-mesh Frame Relay networks, and the single-subnet option has some deficiencies when the network is a partial mesh.

Frame Relay Layer 3 Addressing: One Subnet Per VC

The second IP addressing alternative, the single-subnet-per-VC alternative, works better with a partially meshed Frame Relay network, as shown in Figure 11-12. Boston cannot forward frames directly to Charlotte, because no VC is defined between the two. This is a more typical Frame Relay network, because most organizations with many sites tend to group applications on servers at a few centralized locations, and most of the traffic is between each remote site and those servers.

Figure 11-12 Partial Mesh with IP Addresses



The single-subnet-per-VC alternative matches the logic behind a set of point-to-point links. Using multiple subnets instead of one larger subnet wastes some IP addresses, but it overcomes some issues with distance vector routing protocols.

Table 11-7 shows the IP addresses for the partially meshed Frame Relay network shown in Figure 11-12.

Table 11-7 IP Addresses with Point-to-Point Subinterfaces

Router	Subnet	IP Address
Atlanta	140.1.1.0	140.1.1.1
Charlotte	140.1.1.0	140.1.1.2
Atlanta	140.1.2.0	140.1.2.1
Nashville	140.1.2.0	140.1.2.3
Atlanta	140.1.3.0	140.1.3.1
Boston	140.1.3.0	140.1.3.4

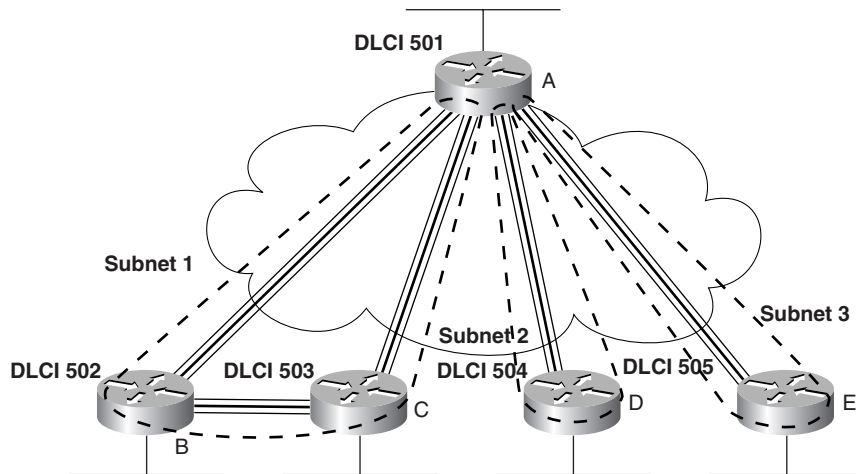
Cisco IOS software has a configuration feature called *subinterfaces* that creates a logical subdivision of a physical interface. Subinterfaces allow the Atlanta router to have three IP addresses associated with its Serial0 physical interface by configuring three separate subinterfaces. A router can treat each subinterface, and the VC associated with it, as if it were a point-to-point serial link. Each of the three subinterfaces of Serial0 on Atlanta would be assigned a different IP address from Table 11-7. (Sample configurations appear in the next section.)

NOTE The example uses IP address prefixes of /24 to keep the math simple. In production networks, point-to-point subinterfaces typically use a prefix of /30 (mask 255.255.255.252), because that allows for only two valid IP addresses—the exact number needed on a point-to-point subinterface. Of course, using different masks in the same network means your routing protocol must also support VLSM.

Frame Relay Layer 3 Addressing: Hybrid Approach

The third alternative of Layer 3 addressing is a hybrid of the first two alternatives. Consider Figure 11-13, which shows a trio of routers with VCs between each of them, as well as two other VCs to remote sites.

Figure 11-13 *Hybrid of Full and Partial Mesh*



Two options exist for Layer 3 addressing in this case. The first is to treat each VC as a separate Layer 3 group. In this case, five subnets are needed for the Frame Relay network. However, Routers A, B, and C create a smaller full mesh between each other. This allows Routers A, B, and C to use one subnet. The other two VCs—one between Routers A and D and one between Routers A and E—are treated as two separate Layer 3 groups. The result is a total of three subnets.

To accomplish either style of Layer 3 addressing in this third and final case, subinterfaces are used. Point-to-point subinterfaces are used when a single VC is considered to be all that is in the group—for instance, between Routers A and D and between Routers A and E. Multipoint subinterfaces are used when more than two routers are considered to be in the same group—for instance, with Routers A, B, and C.

Multipoint subinterfaces logically terminate more than one VC. In fact, the name “multipoint” implies the function, because more than one remote site can be reached via a VC associated with a multipoint subinterface.

Table 11-8 summarizes the addresses and subinterfaces that are used in Figure 11-13.

Table 11-8 *IP Addresses with Point-to-Point and Multipoint Subinterfaces*

Router	Subnet	IP Address	Subinterface Type
A	140.1.1.0/24	140.1.1.1	Multipoint
B	140.1.1.0/24	140.1.1.2	Multipoint
C	140.1.1.0/24	140.1.1.3	Multipoint
A	140.1.2.0/24	140.1.2.1	Point-to-point
D	140.1.2.0/24	140.1.2.4	Point-to-point
A	140.1.3.0/24	140.1.3.1	Point-to-point
E	140.1.3.0/24	140.1.3.5	Point-to-point

What will you see in a real network? Most of the time, point-to-point subinterfaces are used, with a single subnet per PVC.

The later section “Frame Relay Configuration” provides full configurations for all three cases illustrated in Figures 11-11, 11-12, and 11-13.

Broadcast Handling

After contending with Layer 3 addressing over Frame Relay, the next consideration is how to deal with Layer 3 broadcasts. Frame Relay can send copies of a broadcast over all VCs, but there is no equivalent to LAN broadcasts. In other words, no capability exists for a Frame Relay DTE to send a single frame into the Frame Relay network and have that frame replicated and delivered across multiple VCs to multiple destinations. However, routers need to send broadcasts for several features to work. In particular, routing protocol updates are often broadcasts.

The solution to the Frame Relay broadcast dilemma has two parts. First, Cisco IOS software sends copies of the broadcasts across each VC, assuming that you have configured the router to forward these necessary broadcasts. If there are only a few VCs, this is not a big problem. However, if hundreds of VCs terminate in one router, for each broadcast, hundreds of copies could be sent.

As the second part of the solution, the router tries to minimize the impact of the first part of the solution. The router places these broadcasts in a different output queue than the one for user traffic so that the user does not experience a large spike in delay each time a broadcast is replicated and sent over every VC. Cisco IOS software can also be configured to limit the amount of bandwidth that is used for these replicated broadcasts.

Although such scalability issues are more likely to appear on the CCNP Routing exam, a short example shows the significance of broadcast overhead. If a router knows 1000 routes, uses RIP, and has 50 VCs, 1.072 MB of RIP updates is sent every 30 seconds. That averages out to 285 kbps. (The math is as follows: 536-byte RIP packets, with 25 routes in each packet, for 40 packets per update, with copies sent over 50 VCs. $536 * 40 * 50 = 1.072$ MB per update interval. $1.072 * 8 / 30$ seconds = 285 kbps.) That's a lot of overhead!

Knowing how to tell the router to forward these broadcasts to each VC is covered in the section "Frame Relay Configuration." The issues that relate to dealing with the volume of these updates are more likely a topic for the CCNP and CCIE exams.

Frame Relay Service Interworking

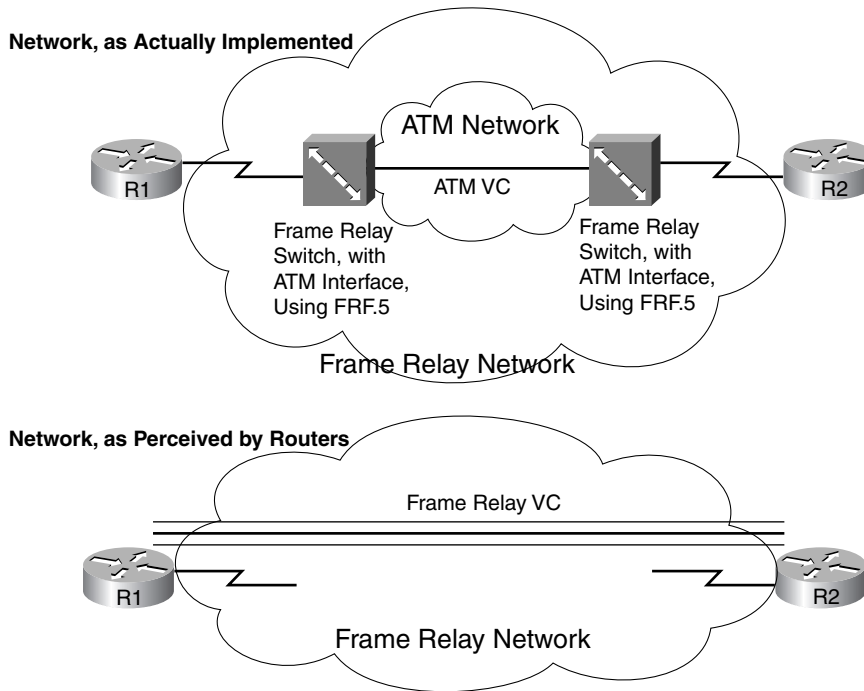
Inside a service provider's Frame Relay network, the provider can use any kind of gear it wants to create the Frame Relay network. One large vendor many years ago built its Frame Relay network with a bunch of Cisco routers! But today, most well-established Frame Relay networks use Asynchronous Transfer Mode (ATM) in the core of their Frame Relay networks.

ATM works similarly to Frame Relay, but it has many features that make it much more robust than Frame Relay. For instance, ATM uses virtual connections (VCs), which provide the same function as Frame Relay VCs. However, ATM differs in that it segments all frames into 53-byte-long cells before transmitting them across the ATM network, and it reassembles them at the other side of the network. ATM also has significantly better quality of service (QoS) features, which lets the service provider better control the use of its core networks.

Service providers can use ATM to build the core of their Frame Relay networks. This concept is depicted in Figure 11-14.

The Frame Relay Forum and other standards bodies use the term *service interworking* to refer to the use of ATM between the two Frame Relay switches. In Figure 11-14, you see that the two routers use Frame Relay to communicate with the provider. You normally think of the internals of the Frame Relay network as being hidden from the customer. Often, inside the service provider network, ATM is used between the Frame Relay switches at the edges of the network.

Figure 11-14 FRF.5 Service Interworking

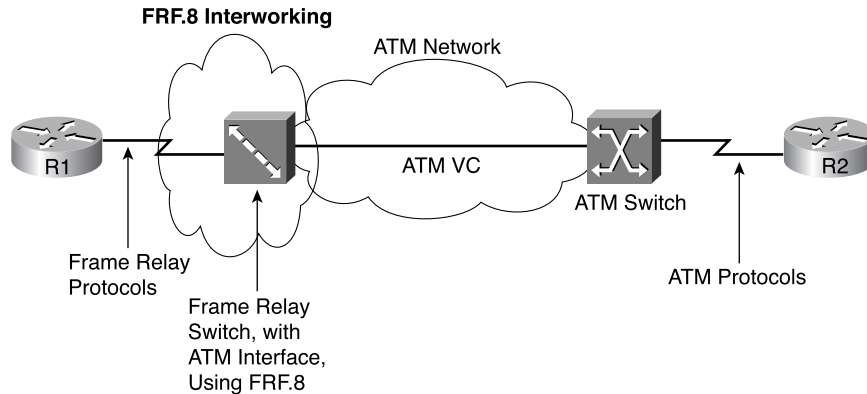


The Frame Relay Forum defined a specification about how Frame Relay switches interoperate using ATM in a Frame Relay Forum document called *FRF.5*. *FRF.5* defines how a Frame Relay switch can convert from a Frame Relay VC to an ATM VC and back to a Frame Relay VC. The end result is totally transparent to the two routers.

A second variation of service interworking was defined by the Frame Relay Forum in document *FRF.8*. *FRF.8* service interworking defines how two routers communicate when one router is connected to a Frame Relay network and the other is connected to an ATM network. ATM services can be used for WAN connectivity, much like Frame Relay. So, with *FRF.8*, a service provider can sell VCs to its customers, with some endpoints connected via Frame Relay and some with ATM. *FRF.8* defines how to convert between the ATM and Frame Relay sides of the networks, as shown in Figure 11-15.

For the exam, you should remember the basic idea behind service interworking, and the two types—*FRF.5* and *FRF.8*.

Figure 11-15 FRF.8 Service Interworking



Frame Relay Configuration

Basic configuration of Frame Relay in a Cisco router is relatively straightforward, partly because Cisco IOS software uses good default values. Of course, you still need to know the optional parameters that are described in this section and the methods of changing the default values. The configuration examples start with a configuration with all the default settings and then begin adding optional features.

There is no substitute for hands-on experience! However, in lieu of hands-on experience, this section lists commands, provides examples, and points out tricky features. Tables 11-9 and 11-10 summarize the more popular commands used for Frame Relay configuration and verification. Several configuration examples follow. If you are interested in other references as well, the Cisco IOS software documentation is an excellent reference for additional IP commands.

Table 11-9 Frame Relay Configuration Commands

Command	Description
<code>encapsulation frame-relay [ietf cisco]</code>	Interface configuration mode command that defines the Frame Relay encapsulation that is used rather than HDLC, PPP, and so on.
<code>frame-relay lmi-type {ansi q933a cisco}</code>	Interface configuration mode command that defines the type of LMI messages sent to the switch.
<code>bandwidth num</code>	Interface configuration mode command that sets the router's perceived interface speed. Bandwidth is used by some routing protocols to influence the metric and is used in link utilization calculations seen with the show interfaces command.

continues

Table 11-9 *Frame Relay Configuration Commands (Continued)*

Command	Description
<code>frame-relay map {protocol protocol-address dlc} payload-compression frf9 stac caim [element-number] [broadcast] [ietf cisco]</code>	Interface configuration mode command that statically defines a mapping between a network layer address and a DLCI.
<code>keepalive sec</code>	Interface configuration mode command that defines whether and how often LMI status inquiry messages are sent and expected.
<code>interface serial number.sub [point-to-point multipoint]</code>	Global configuration mode command that creates a subinterface or references a previously created subinterface.
<code>frame-relay interface-dlci dlc [ietf cisco] [voice-cir cir] [ppp virtual-template-name]</code>	Subinterface configuration mode command that links or correlates a DLCI to the subinterface.

Table 11-10 *Frame Relay-Related EXEC Commands*

Command	Description
<code>show interfaces [type number]</code>	Shows the physical interface status.
<code>show frame-relay pvc [interface interface][dlci]</code>	Lists information about the PVC status.
<code>show frame-relay lmi [type number]</code>	Lists LMI status information.

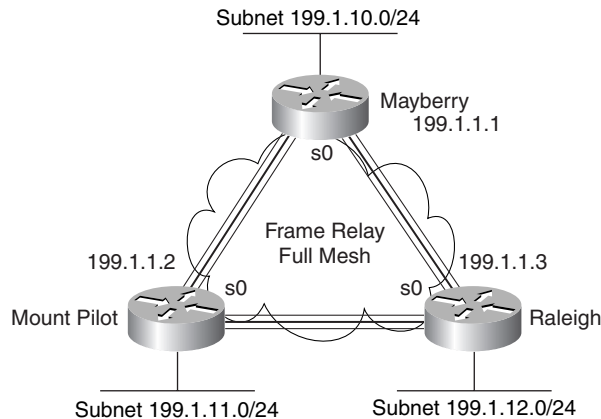
A network engineer might plan the Frame Relay configuration based on several factors. When the service is ordered, the service provider specifies the LMI type that will be used. The engineer chooses the endpoints of the VCs, including whether to use a full mesh or partial mesh. Based on the location of the VCs, the engineer then decides which IP addressing option to use: single subnet, single subnet per VC, or a combination of the two. Finally, the encapsulation type is chosen—totally without input from the provider, because only the routers on the ends of each VC need to agree on the encapsulation type. Because Frame Relay switches do not care about the encapsulation type, nor do they care about IP addressing, the only details that have to be discussed with the carrier are the VCs and the LMI type, along with the CIR and burst sizes.

Three examples of Layer 3 addressing were given earlier in this chapter, with the networks diagrammed in Figures 11-11, 11-12, and 11-13. The configurations matching those networks and addresses are shown next.

A Fully-Meshed Network with One IP Subnet

In this first example, the configuration uses all default values, and it does not use subinterfaces. The Frame Relay configuration is listed under the physical interfaces. Examples 11-1, 11-2, and 11-3 show the configuration for the network shown in Figure 11-16.

Figure 11-16 Full Mesh with IP Addresses



Example 11-1 Mayberry Configuration

```
interface serial0
encapsulation frame-relay
ip address 199.1.1.1 255.255.255.0
!
interface ethernet 0
ip address 199.1.10.1 255.255.255.0
!
router igrp 1
network 199.1.1.0
network 199.1.10.0
```

Example 11-2 Mount Pilot Configuration

```
interface serial0
encapsulation frame-relay
ip address 199.1.1.2 255.255.255.0
!
interface ethernet 0
ip address 199.1.11.2 255.255.255.0
!
router igrp 1
network 199.1.1.0
network 199.1.11.0
```

Example 11-3 *Raleigh Configuration*

```

interface serial0
encapsulation frame-relay
ip address 199.1.1.3 255.255.255.0
!
interface ethernet 0
ip address 199.1.12.3 255.255.255.0
!
router igrp 1
network 199.1.1.0
network 199.1.12.0

```

The configuration is simple in comparison with the protocol concepts. The **encapsulation frame-relay** command tells the routers to use Frame Relay data-link protocols instead of the default, which is HDLC. Note that the IP addresses on the three routers' serial interfaces are all in the same subnet. When you configure Frame Relay on the physical interfaces, all three routers must be in the same subnet.

Yes, Frame Relay configuration can be that easy, because IOS uses some very good choices for default settings:

- The LMI type is automatically sensed.
- The encapsulation is Cisco instead of IETF.
- PVC DLCIs are learned via LMI status messages.
- Inverse ARP is enabled (by default) and is triggered when the status message declaring that the VCs are up is received. (Inverse ARP is covered in the next section.)

In some cases, the default values are inappropriate. For example, you must use IETF encapsulation if one router is not a Cisco router. For the purpose of showing an alternative configuration, suppose that the following requirements were added:

- The Raleigh router requires IETF encapsulation on both VCs.
- Mayberry's LMI type should be ANSI, and LMI autosense should not be used.

Examples 11-4 and 11-5 show the changes that would be made to Mayberry and Raleigh.

Example 11-4 *Mayberry Configuration with New Requirements*

```

interface serial0
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay interface-dlci 53 ietf
ip address 199.1.1.1 255.255.255.0
! rest of configuration unchanged from Example 11-1.

```

Example 11-5 *Raleigh Configuration with New Requirements*

```

interface serial0
  encapsulation frame-relay ietf
  ip address 199.1.1.3 255.255.255.0
!
! rest of configuration unchanged from Example 11-3.

```

These configurations differ from the previous ones in two ways: Raleigh changed its encapsulation for both its PVCs with the **ietf** keyword on the **encapsulation** command. This keyword applies to all VCs on the interface. However, Mayberry cannot change its encapsulation in the same way, because only one of the two VCs terminating in Mayberry needs to use IETF encapsulation—the other needs to use Cisco encapsulation. So Mayberry is forced to code the **frame-relay interface-dlci** command, referencing the DLCI for the VC to Raleigh, with the **ietf** keyword. With that command, you can change the encapsulation setting per-VC, as opposed to the configuration on Raleigh, which changes the encapsulation for all VCs.

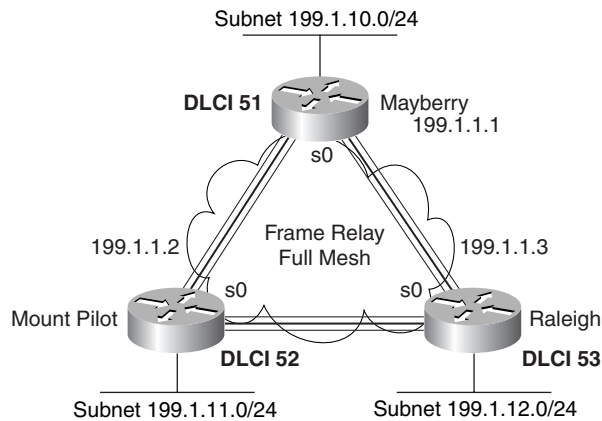
The LMI configuration in Mayberry would be fine without any changes, because autosense would recognize ANSI. However, by coding the **frame-relay lmi-type ansi** command, Mayberry must use ANSI, because this command not only sets the LMI type, it also disables autonegotiation of the LMI type.

Mount Pilot needs to configure a **frame-relay interface-dlci** command with the **ietf** keyword for its VC to Raleigh, just like Mayberry. This change is not shown in the examples.

Frame Relay Address Mapping

Figure 11-16 does not even bother listing the DLCIs used for the VCs. The configurations work as stated, and frankly, if you never knew the DLCIs, this network would work! However, for the exam, and for real networking jobs, you need to understand an important concept related to Frame Relay—namely, Frame Relay address mapping. Figure 11-17 shows the same network, this time with Global DLCI values shown.

Frame Relay “mapping” creates a correlation between a Layer 3 address and its corresponding Layer 2 address. For example, the IP Address Resolution Protocol (ARP) cache used on LANs is an example of Layer 3-to-Layer 2 address mapping. With IP ARP, you know the IP address of another device on the same LAN, but not the MAC address; when the ARP completes, you know another device’s LAN (Layer 2) address. Similarly, routers that use Frame Relay need a mapping between a router’s Layer 3 address and the DLCI used to reach that other router.

Figure 11-17 *Full Mesh with IP Addresses*

This section discusses the basics of why mapping is needed for LAN connections and Frame Relay, with a focus on Frame Relay. Here's a more general definition of mapping:

The information that correlates to the next-hop router's Layer 3 address, and the Layer 2 address used to reach it, is called mapping. Mapping is needed on multiaccess networks.

Thinking about routing helps make the need for mapping more apparent. Imagine that a host on the Mayberry Ethernet sends an IP packet to a host on the Mount Pilot Ethernet. The packet arrives at the Mayberry router over the LAN, and Mayberry discards the Ethernet header and trailer. Mayberry looks at the routing table, which lists a route to 199.1.11.0, outgoing interface Serial0, and next-hop router 199.1.1.2, which is Mount Pilot's Frame Relay IP address.

The next decision that the router must make to complete the process points out the need for mapping: What DLCI should Mayberry put in the Frame Relay header? We configured no DLCIs. However, it would work as configured! To see the answer, consider Example 11-6, which shows some important commands that can be used to see how Mayberry makes the right choice for the DLCI.

Example 11-6 *show Commands on Mayberry, Showing the Need for Mapping*

```
Mayberry#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
```

Example 11-6 *show Commands on Mayberry, Showing the Need for Mapping (Continued)*

```

P - periodic downloaded static route

Gateway of last resort is not set

I 199.1.11.0/24 [100/8576] via 199.1.1.2, 00:00:26, Serial0
C 199.1.10.0/24 is directly connected, Ethernet0
I 199.1.12.0/24 [100/8539] via 199.1.1.3, 00:01:04, Serial0
C 199.1.1.0/24 is directly connected, Serial0
C 192.68.1.0/24 is directly connected, Ethernet0
C 192.168.1.0/24 is directly connected, Ethernet0

Mayberry#show frame-relay pvc
PVC Statistics for interface Serial0 (Frame Relay DTE)

          Active      Inactive      Deleted      Static
Local          2           0            0            0
Switched       0           0            0            0
Unused         0           0            0            0

DLCI = 52, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 46          output pkts 22          in bytes 2946
out bytes 1794         dropped pkts 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 21     out bcast bytes 1730
pvc create time 00:23:07, last time pvc status changed 00:21:38

DLCI = 53, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0

input pkts 39          output pkts 18          in bytes 2564
out bytes 1584         dropped pkts 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 18     out bcast bytes 1584
pvc create time 00:23:08, last time pvc status changed 00:21:20

Mayberry#show frame-relay map
Serial0 (up): ip 199.1.1.2 dlci 52(0x34,0xC40), dynamic,
              broadcast,, status defined, active
Serial0 (up): ip 199.1.1.3 dlci 53(0x35,0xC50), dynamic,
              broadcast,, status defined, active

```

All the information needed for Mayberry to pick DLCI 52 is in the command output. The route to 199.1.11.0 points out Serial0 to 199.1.1.2 as the next-hop address. The **show frame-relay pvc** command lists two DLCIs, 52 and 53, and both are active. How does Mayberry

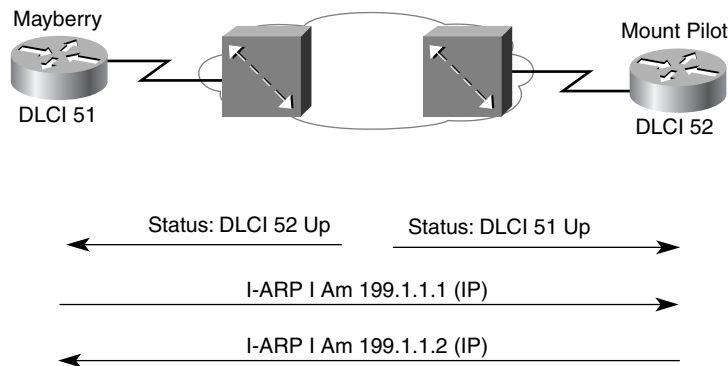
know the DLCIs? Well, the LMI status messages tell Mayberry about the VCs, the associated DLCIs, and the status (active).

Which DLCI should Mayberry use to forward the packet? The `show frame-relay map` command output holds the answer. Notice the highlighted phrase `ip 199.1.1.2 dci 52` in the output. Somehow, Mayberry has mapped 199.1.1.2, which is the next-hop address in the route, to the correct DLCI, which is 52. So, Mayberry knows to use DLCI 52 to reach next-hop IP address 199.1.1.2.

Mayberry can use two methods to build the mapping shown in Example 11-6. One uses a statically configured mapping, and the other uses a dynamic process called *Inverse ARP*.

Inverse ARP dynamically creates a mapping between the Layer 3 address (for example, the IP address) and the Layer 2 address (the DLCI). The end result of Inverse ARP is the same as IP ARP on a LAN: The router builds a mapping between a neighboring Layer 3 address and the corresponding Layer 2 address. However, the process used by Inverse ARP differs for ARP on a LAN. After the VC is up, each router announces its network layer address by sending an Inverse ARP message over that VC. This works as shown in Figure 11-18.

Figure 11-18 *Inverse ARP Process*



As shown in Figure 11-18, Inverse ARP announces its Layer 3 addresses as soon as the LMI signals that the PVCs are up. Inverse ARP starts by learning the DLCI data link layer address (via LMI messages), and then it announces its own Layer 3 addresses that use that VC.

Inverse ARP is enabled by default in Cisco IOS software Release 11.2 and later.

In Example 11-6, Mayberry shows two different entries in the **show frame-relay map** command output. Mayberry uses Inverse ARP to learn that DLCI 52 is mapped to next-hop IP address 199.1.1.2 and that DLCI 53 is mapped to next-hop IP address 199.1.1.3. Interestingly, Mayberry learns this information by receiving an Inverse ARP from Mount Pilot and Raleigh, respectively—another case of learning by listening, a great lesson for real life!

Table 11-11 summarizes what occurs with Inverse ARP in the network shown in Figure 11-17.

Table 11-11 *Inverse ARP Messages for Figure 11-17*

Sending Router	DLCI When the Frame Is Sent	Receiving Router	DLCI When the Frame Is Received	Information in the Inverse ARP Message
Mayberry	52	Mount Pilot	51	I am 199.1.1.1.
Mayberry	53	Raleigh	51	I am 199.1.1.1.
Mount Pilot	51	Mayberry	52	I am 199.1.1.2.
Mount Pilot	53	Raleigh	52	I am 199.1.1.2.
Raleigh	51	Mayberry	53	I am 199.1.1.3.
Raleigh	52	Mount Pilot	53	I am 199.1.1.3.

To understand Inverse ARP, focus on the last two columns of Table 11-11. Each router receives some Inverse ARP “announcements.” The Inverse ARP message contains the sender’s Layer 3 address, and the Frame Relay header, of course, has a DLCI in it. These two values are placed in the Inverse ARP cache on the receiving router. For example, in the third row, Mayberry receives an Inverse ARP. The DLCI is 52, and the IP address is 199.1.1.2. This is added to the Frame Relay map table in Mayberry, which is shown in the highlighted part of the **show frame-relay map** command in Example 11-6.

You can statically configure the same mapping information instead of using Inverse ARP. In a production network, you probably would just go ahead and use Inverse ARP. For the exam, you need to know how to configure the static map commands. Example 11-7 lists the static Frame Relay map for the three routers shown in Figure 11-11, along with the configuration used to disable Inverse ARP.

Example 11-7 frame-relay map *Commands*

```

Mayberry
interface serial 0
no frame-relay inverse-arp
frame-relay map ip 199.1.1.2 52 broadcast
frame-relay map ip 199.1.1.3 53 broadcast
-----
Mount Pilot
interface serial 0
no frame-relay inverse-arp
frame-relay map ip 199.1.1.1 51 broadcast
frame-relay map ip 199.1.1.3 53 broadcast
-----
Raleigh
interface serial 0
no frame-relay inverse-arp
frame-relay map ip 199.1.1.1 51 broadcast
frame-relay map ip 199.1.1.2 52 broadcast

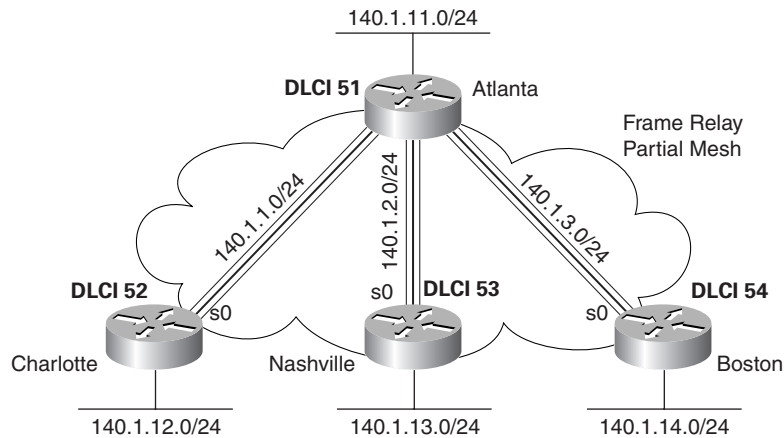
```

The `frame-relay map` command entry for Mayberry, referencing 199.1.1.2, is used for packets in Mayberry going to Mount Pilot. When Mayberry creates a Frame Relay header, expecting it to be delivered to Mount Pilot, Mayberry must use DLCI 52. Mayberry's `map` statement correlates Mount Pilot's IP address, 199.1.1.2, to the DLCI used to reach Mount Pilot—namely, DLCI 52. Likewise, a packet sent back from Mount Pilot to Mayberry causes Mount Pilot to use its `map` statement to refer to Mayberry's IP address of 199.1.1.1. Mapping is needed for each next-hop Layer 3 address for each Layer 3 protocol being routed. Even with a network this small, the configuration process can be laborious.

A Partially-Meshed Network with One IP Subnet Per VC

The second sample network, based on the environment shown in Figure 11-19, uses point-to-point subinterfaces. Examples 11-8 through 11-11 show the configuration for this network. The command prompts are included in the first example because they change when you configure subinterfaces.

Figure 11-19 Partial Mesh with IP Addresses



Example 11-8 Atlanta Configuration

```
Atlanta(config)#interface serial0
Atlanta(config-if)#encapsulation frame-relay

Atlanta(config-if)#interface serial 0.1 point-to-point
Atlanta(config-subif)#ip address 140.1.1.1 255.255.255.0
Atlanta(config-subif)#frame-relay interface-dlci 52

Atlanta(config-fr-dlci)#interface serial 0.2 point-to-point
Atlanta(config-subif)#ip address 140.1.2.1 255.255.255.0
Atlanta(config-subif)#frame-relay interface-dlci 53

Atlanta(config-fr-dlci)#interface serial 0.3 point-to-point
Atlanta(config-subif)#ip address 140.1.3.1 255.255.255.0
Atlanta(config-subif)#frame-relay interface-dlci 54

Atlanta(config-fr-dlci)#interface ethernet 0
Atlanta(config-if)#ip address 140.1.11.1 255.255.255.0
```

Example 11-9 Charlotte Configuration

```
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 140.1.1.2 255.255.255.0
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 140.1.12.2 255.255.255.0
```

Example 11-10 *Nashville Configuration*

```

interface serial0
encapsulation frame-relay
!
interface serial 0.2 point-to-point
ip address 140.1.2.3 255.255.255.0
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 140.1.13.3 255.255.255.0

```

Example 11-11 *Boston Configuration*

```

interface serial0
encapsulation frame-relay
!
interface serial 0.3 point-to-point
ip address 140.1.3.4 255.255.255.0
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 140.1.14.4 255.255.255.0

```

Again, defaults abound in this configuration, but some defaults are different than when you're configuring on the main interface, as in the preceding example. The LMI type is autosensed, and Cisco encapsulation is used, which is just like the fully-meshed example. However, Inverse ARP is disabled on each point-to-point subinterface by default. As you will see, Inverse ARP is not needed with point-to-point subinterfaces.

Two new commands create the configuration required with point-to-point subinterfaces. First, the **interface serial 0.1 point-to-point** command creates logical subinterface number 1 under physical interface Serial0. The **frame-relay interface-dlci** subinterface subcommand then tells the router which single DLCI is associated with that subinterface.

An example of how the **interface-dlci** command works can help. Consider router Atlanta in Figure 11-19. Atlanta receives LMI messages on Serial0 stating that three PVCs, with DLCIs 52, 53, and 54, are up. Which PVC goes with which subinterface? Cisco IOS software needs to associate the correct PVC with the correct subinterface. This is accomplished with the **frame-relay interface-dlci** command.

The subinterface numbers do not have to match on the router on the other end of the PVC, nor does the DLCI number. In this example, I just numbered them to be easier to remember! In real life, it is useful to encode some information about your network numbering scheme into the subinterface number. One client I worked with encoded part of the carrier's circuit ID in the subinterface number so that the operations staff could find the correct information to tell the Telco when troubleshooting the link. Many sites use the DLCI as the subinterface number. Of course, useful troubleshooting information, such as the DLCI, the name of the router on the other end of the VC, and so on, could be configured as text with the **description** command as well. In any case, there are no requirements for matching subinterface numbers. Here, all I did was match the subinterface number to the third octet of the IP address.

Example 11-12 shows the output from the most popular Cisco IOS software Frame Relay EXEC commands for monitoring Frame Relay, as issued on router Atlanta.

Example 11-12 *Output from EXEC Commands on Atlanta*

```
Atlanta#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

          Active      Inactive      Deleted      Static
Local            3           0           0           0
Switched         0           0           0           0
Unused           0           0           0           0
DLCI = 52, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 843          output pkts 876          in bytes 122723
  out bytes 134431        dropped pkts 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 876     out bcast bytes 134431
  pvc create time 05:20:10, last time pvc status changed 05:19:31
  --More--
DLCI = 53, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

  input pkts 0           output pkts 875          in bytes 0
  out bytes 142417        dropped pkts 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 875     out bcast bytes 142417
  pvc create time 05:19:51, last time pvc status changed 04:55:41
  --More--
DLCI = 54, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.3

  input pkts 10          output pkts 877          in bytes 1274
  out bytes 142069        dropped pkts 0           in FECN pkts 0
  in BECN pkts 0         out FECN pkts 0         out BECN pkts 0
```

continues

Example 11-12 *Output from EXEC Commands on Atlanta (Continued)*

```

in DE pkts 0          out DE pkts 0
out bcst pkts 877     out bcst bytes 142069
pvc create time 05:19:52, last time pvc status changed 05:17:42

Atlanta#show frame-relay map
Serial0.3 (up): point-to-point dlci, dlci 54(0x36,0xC60), broadcast
status defined, active
Serial0.2 (up): point-to-point dlci, dlci 53(0x35,0xC50), broadcast
status defined, active
Serial0.1 (up): point-to-point dlci, dlci 52(0x34,0xC40), broadcast
status defined, active

Atlanta#debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data

Serial0(out): StEnq, myseq 163, yourseen 161, DTE up
datagramstart = 0x45AED8, datagramsize = 13
FR encap = 0xFCF10309
00 75 01 01 01 03 02 A3 A1

Serial0(in): Status, myseq 163
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 162, myseq 163

```

The **show frame-relay pvc** command lists useful management information. For instance, the packet counters for each VC, plus the counters for FECN and BECN, can be particularly useful. Likewise, comparing the packets/bytes sent on one router versus the counters of what is received on the router on the other end of the VC is also quite useful, because it reflects the number of packets/bytes lost inside the Frame Relay cloud. Also, the PVC status is a great place to start when troubleshooting. In addition, an SNMP manager can better gather all this information with this command.

The **show frame-relay map** command lists mapping information. With the earlier example of a fully-meshed network, in which the configuration did not use any subinterfaces, a Layer 3 address was listed with each DLCI. In this example, a DLCI is listed in each entry, but no mention of corresponding Layer 3 addresses is made. The whole point of mapping is to correlate a Layer 3 address to a Layer 2 address, but there is no Layer 3 address in the **show frame-relay map** command output! The reason is that the information is stored somewhere else. Subinterfaces require the use of the **frame-relay interface-dlci** configuration command. Because these subinterfaces are point-to-point, when a route points out a single subinterface,

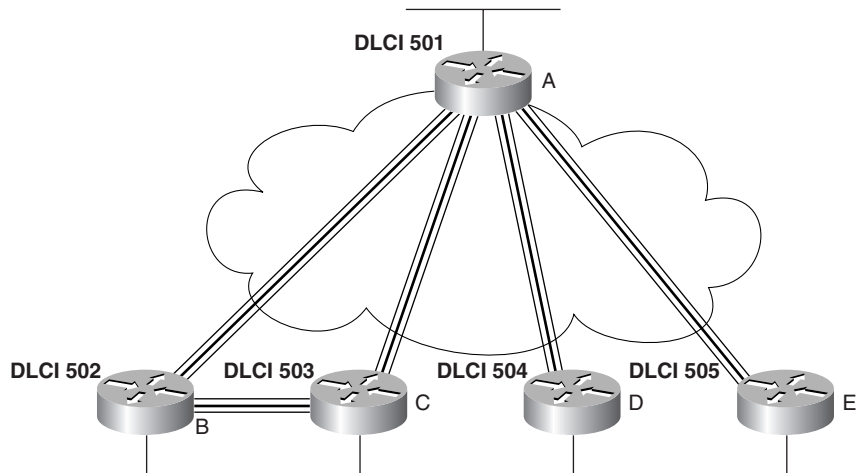
the DLCI to use to send frames is implied by the configuration. Mapping via Inverse ARP or static **frame-relay map** statements is needed only when more than two VCs terminate on the interface or subinterface, because those are the only instances in which confusion about which DLCI to use might occur.

The **debug frame-relay lmi** output lists information for the sending and receiving LMI inquiries. The switch sends the status message, and the DTE (router) sends the status inquiry. The default setting with Cisco IOS software is to send, and to expect to receive, these status messages. The Cisco IOS software **no keepalive** command is used to disable the use of LMI status messages. Unlike other interfaces, Cisco keepalive messages do not flow from router to router over Frame Relay. Instead, they are simply used to detect whether the router has connectivity to its local Frame Relay switch.

A Partially-Meshed Network with Some Fully-Meshed Parts

You can also choose to use multipoint subinterfaces for a Frame Relay configuration. This last sample network, based on the network shown in Figure 11-20, uses both multipoint and point-to-point subinterfaces. Examples 11-13 through 11-17 show the configuration for this network. Table 11-12 summarizes the addresses and subinterfaces used.

Figure 11-20 *Hybrid of Full and Partial Mesh*



Example 11-13 *Router A Configuration*

```

hostname RouterA
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 140.1.1.1 255.255.255.0
frame-relay interface-dlci 502
frame-relay interface-dlci 503
!
interface serial 0.2 point-to-point
ip address 140.1.2.1 255.255.255.0
frame-relay interface-dlci 504
!
interface serial 0.3 point-to-point
ip address 140.1.3.1 255.255.255.0
frame-relay interface-dlci 505
!
interface ethernet 0
ip address 140.1.11.1 255.255.255.0

```

Example 11-14 *Router B Configuration*

```

hostname RouterB
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 140.1.1.2 255.255.255.0
frame-relay interface-dlci 501
frame-relay interface-dlci 503
!
interface ethernet 0
ip address 140.1.12.2 255.255.255.0

```

Example 11-15 *Router C Configuration*

```

hostname RouterC
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 140.1.1.3 255.255.255.0
frame-relay interface-dlci 501
frame-relay interface-dlci 502

```

Example 11-15 Router C Configuration (Continued)

```

!
interface ethernet 0
ip address 140.1.13.3 255.255.255.0

```

Example 11-16 Router D Configuration

```

hostname RouterD
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 140.1.2.4 255.255.255.0
frame-relay interface-dlci 501
!
interface ethernet 0
ip address 140.1.14.4 255.255.255.0

```

Example 11-17 Router E Configuration

```

hostname RouterE
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 140.1.3.5 255.255.255.0
frame-relay interface-dlci 501
!
interface ethernet 0
ip address 140.1.15.5 255.255.255.0

```

Table 11-12 IP Addresses with Point-to-Point and Multipoint Subinterfaces

Router	Subnet	IP Address	Subinterface Type
A	140.1.1.0/24	140.1.1.1	Multipoint
B	140.1.1.0/24	140.1.1.2	Multipoint
C	140.1.1.0/24	140.1.1.3	Multipoint
A	140.1.2.0/24	140.1.2.1	Point-to-point
D	140.1.2.0/24	140.1.2.4	Point-to-point
A	140.1.3.0/24	140.1.3.1	Point-to-point
E	140.1.3.0/24	140.1.3.5	Point-to-point

Multipoint subinterfaces work best when you have a full mesh between a set of routers. On Routers A, B, and C, a multipoint subinterface is used for the configuration referencing the other two routers, because you can think of these three routers as forming a fully meshed subset of the network.

The term multipoint simply means that there is more than one VC, so you can send and receive to and from more than one VC on the subinterface. Like point-to-point subinterfaces, multipoint subinterfaces use the **frame-relay interface-dlci** command. Notice that there are two commands for each multipoint subinterface in this case, because each of the two PVCs associated with this subinterface must be identified as being used with that subinterface.

Router A is the only router using both multipoint and point-to-point subinterfaces. On Router A's multipoint Serial0.1 interface, DLCIs for Router B and Router C are listed. On Router A's other two subinterfaces, which are point-to-point, only a single DLCI needs to be listed. In fact, only one **frame-relay interface-dlci** command is allowed on a point-to-point subinterface, because only one VC is allowed. Otherwise, the configurations between the two types are similar.

No mapping statements are required for the configurations shown in Examples 11-13 through 11-17 because Inverse ARP is enabled on the multipoint subinterfaces by default. No mapping is ever needed for the point-to-point subinterface, because the only DLCI associated with the interface is statically configured with the **frame-relay interface-dlci** command.

Example 11-18 lists another **show frame-relay map** command, showing the mapping information learned by Inverse ARP for the multipoint subinterface. Notice that the output now includes the Layer 3 addresses! The reason is that the routes might point out a multipoint interface, but because more than one DLCI is associated with the interface, the router needs mapping information to identify the correct DLCI.

Example 11-18 *Frame Relay Maps and Inverse ARP on Router C*

```
RouterC#show frame-relay map
Serial0.1 (up): ip 140.1.1.1 dlci 501(0x1F5,0x7C50), dynamic,
                broadcast,, status defined, active
Serial0.1 (up): ip 140.1.1.2 dlci 502(0x1F6,0x7C60), dynamic,
                broadcast,, status defined, active

RouterC#debug frame-relay events
Frame Relay events debugging is on

RouterC#configure terminal
Enter configuration commands, one per line. End with Ctrl-Z.
```

Example 11-18 *Frame Relay Maps and Inverse ARP on Router C (Continued)*

```

RouterC(config)#interface serial 0.1
RouterC(config-subif)#no shutdown
RouterC(config-subif)#^Z
RouterC#

Serial0.1: FR ARP input
Serial0.1: FR ARP input
Serial0.1: FR ARP input
datagramstart = 0xE42E58, datagramsize = 30
FR encap = 0x7C510300
80 00 00 00 08 06 00 0F 08 00 02 04 00 09 00 00
8C 01 01 01 7C 51 8C 01 01 03

datagramstart = 0xE420E8, datagramsize = 30
FR encap = 0x7C610300
80 00 00 00 08 06 00 0F 08 00 02 04 00 09 00 00
8C 01 01 02 7C 61 8C 01 01 03

```

The messages about Inverse ARP in the **debug frame-relay events** output are not so obvious. One easy exercise is to search for the hex version of the IP addresses in the output. These addresses are highlighted in Example 11-18. For example, the first 3 bytes of 140.1.1.0 are 8C 01 01 in hexadecimal. This field starts on the left side of the output, so it is easy to recognize.

Foundation Summary

The “Foundation Summary” section lists the most important facts from the chapter. Although this section does not list everything that will be on the exam, a well-prepared CCNA candidate should at a minimum know all the details in each Foundation Summary before taking the exam.

Figure 11-21 outlines the basic physical topology and related terminology in a Frame Relay network.

Figure 11-21 *Frame Relay Components*

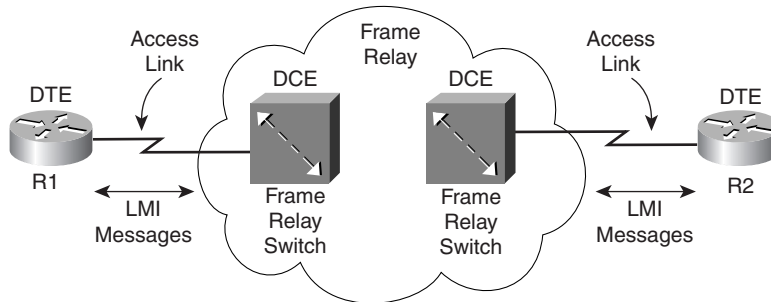


Figure 11-22 depicts a typical partially-meshed Frame Relay network.

Figure 11-22 *Typical Partial-Mesh Frame Relay Network*

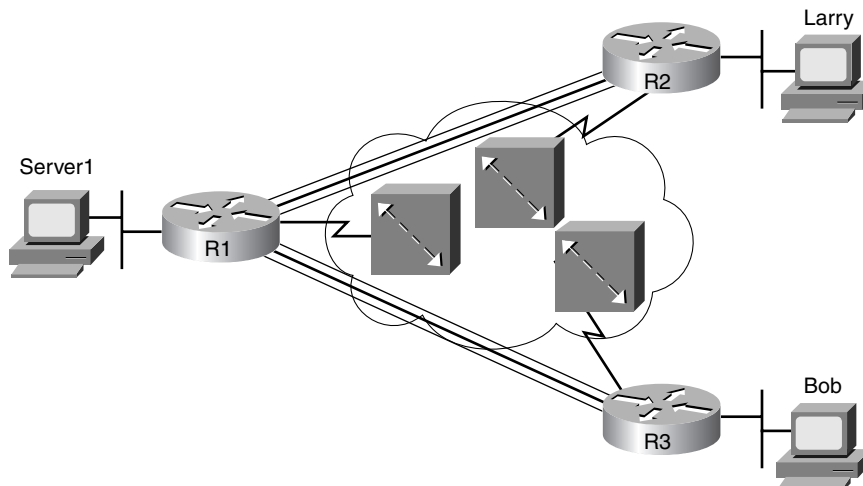


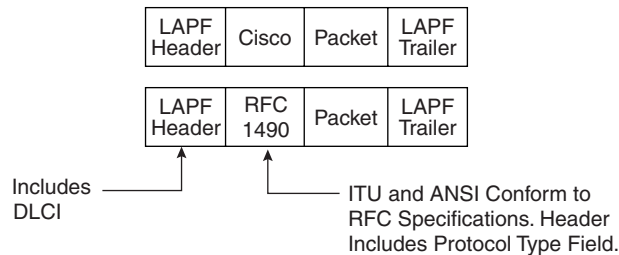
Table 11-13 outlines the three LMI types, their origin, and the keyword used in the Cisco `frame-relay lmi-type` interface subcommand.

Table 11-13 *Frame Relay LMI Types*

Name	Document	IOS LMI-Type Parameter
Cisco	Proprietary	<code>cisco</code>
ANSI	T1.617 Annex D	<code>ansi</code>
ITU	Q.933 Annex A	<code>q933a</code>

Figure 11-23 outlines the two Frame Relay encapsulation options on Cisco routers.

Figure 11-23 *Cisco and RFC 1490/2427 Encapsulation*



Cisco's Frame Relay implementation defines three different options for assigning subnets and IP addresses on Frame Relay interfaces:

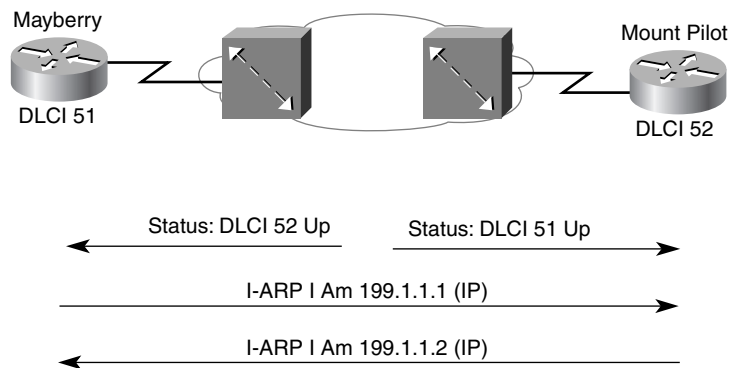
- One subnet containing all Frame Relay DTEs
- One subnet per VC
- A hybrid of the first two options

Cisco IOS software uses some very good choices for default Frame Relay settings:

- The LMI type is automatically sensed.
- The encapsulation is Cisco instead of IETF.
- PVC DLCIs are learned via LMI status messages.
- Inverse ARP is enabled (by default) and is triggered when the status message declaring that the VCs are up is received.

Figure 11-24 outlines how Inverse ARP works.

Figure 11-24 *Inverse ARP Process*



Q&A

As mentioned in the Introduction, you have two choices for review questions. The following questions give you a bigger challenge than the exam because they are open-ended. By reviewing with this more-difficult question format, you can exercise your memory better and prove your conceptual and factual knowledge of the topics covered in this chapter. The answers to these questions are found in Appendix A.

For more practice with exam-like question formats, including multiple-choice questions and those using a router simulator, use the exam engine on the CD.

1. What two WAN data-link protocols define a method of announcing the interface's Layer 3 addresses to other devices attached to the WAN?
2. Explain the purpose of Inverse ARP, as well as how it uses Frame Relay broadcasts.
3. Would a Frame Relay switch connected to a router behave differently if the IETF option were deleted from the **encapsulation frame-relay ietf** command on that attached router? Would a router on the other end of the VC behave any differently if the same change were made?
4. What does NBMA stand for? Does it apply to X.25 networks or Frame Relay networks?
5. Which layer or layers of OSI are most closely related to the functions of Frame Relay? Why?
6. When Inverse ARP is used by default, what additional configuration is needed to get IGRP routing updates to flow over each VC, assuming IGRP has already been configured correctly?
7. Define the attributes of a partial-mesh and full-mesh Frame Relay network.
8. What key pieces of information are required in the **frame-relay map** statement?
9. Create a configuration for Router1 that has Frame Relay VCs to Router2 and Router3 (DLCIs 202 and 203, respectively) on Router1's Serial1 interface. Use any IP addresses you like. Assume that the network is not fully meshed.
10. What **show** command tells you when a PVC became active? How does the router know what time the PVC became active?
11. What **show** command lists Frame Relay information about mapping? In what instances does the information displayed include the Layer 3 addresses of other routers?

12. True or false: The **no keepalive** command on a Frame Relay serial interface causes no further Cisco-proprietary keepalive messages to be sent to the Frame Relay switch.
13. What **debug** option shows Inverse ARP messages?
14. True or false: The Frame Relay **map** configuration command allows more than one Layer 3 protocol address mapping on the same configuration command.
15. What is the name of the field that identifies, or addresses, a Frame Relay virtual circuit?
16. Describe the difference between FRF.5 and FRF.8 service interworking.

