Exam Topics in This Chapter

29 Recognize key Frame Relay terms and features.

30 List commands to configure Frame Relay LMIs, maps, and subinterfaces.

31 List commands to monitor Frame Relay operation in the router.

CHAPTER 10

Frame Relay Concepts and Configuration

Frame Relay networks deliver variable-sized data frames between devices connected to the network. Engineers deploy Frame Relay more than any other WAN protocol today, so it's no surprise that Frame Relay is an important topic on the CCNA exam. This chapter reviews the details of how Frame Relay accomplishes its goal of delivering frames to multiple WAN-connected sites.

Frame Relay is considered a data link layer protocol (Layer 2). If you remember that the word *frame* describes the data link layer protocol data unit, it will be easy to remember that Frame Relay relates to OSI Layer 2. Because Frame Relay is a Layer 2 protocol, it can be used to deliver packets (Layer 3 protocol data units) 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. (Refer to Chapter 3, "OSI Reference Model and Layered Communication," for a review of OSI layers.)

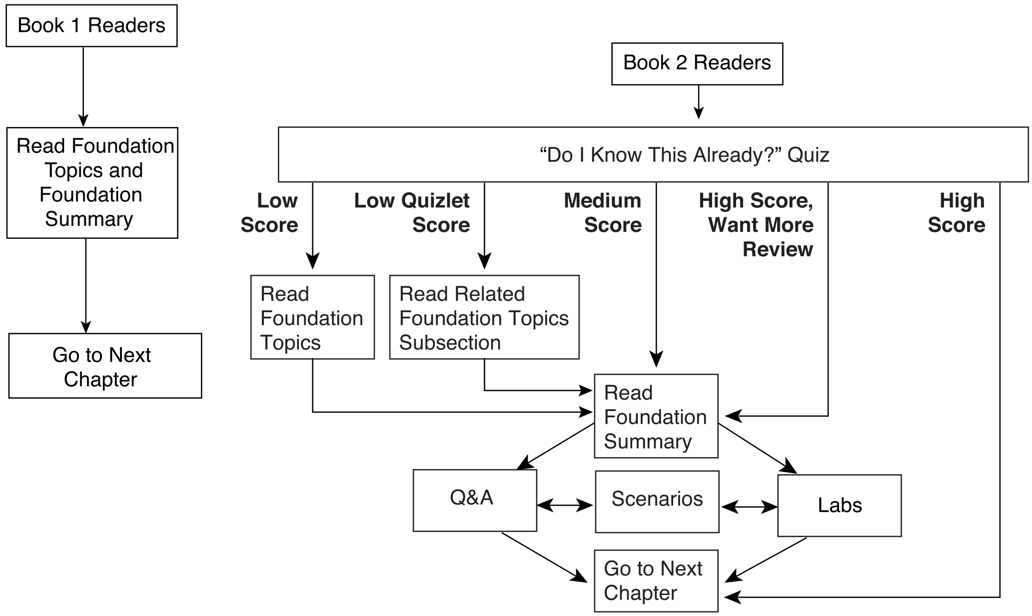
This chapter summarizes the Frame Relay protocol details that are expected to be on the exam.

How to Best Use This Chapter

By following these steps, you can make better use of your study time:

- Keep your notes and the answers for all your work with this book in one place for easy reference.
- Take the "Do I Know This Already?" quiz, and write down your answers. Studies show that retention is significantly increased through writing down facts and concepts, even if you never look at the information again.
- Use Figure 10-1 to guide you to the next step.

Figure 10-1 *How to Use This Chapter*



"Do I Know This Already?" Quiz

The purpose of the "Do I Know This Already?" quiz is to help you decide what parts of this chapter to use. If you already intend to read the entire chapter, you do not necessarily need to answer these questions now.

This eight-question quiz helps you determine how to spend your limited study time. The quiz is sectioned into two smaller four-question "quizlets" that correspond to the two major headings in this chapter. Figure 10-1 outlines suggestions on how to spend your time in this chapter based on your quiz score. Use Table 10-1 to record your scores.

Table 10-1 *Scoresheet for Quiz and Quizlets*

Quizlet Number	Foundation Topics Section Covering These Questions	Questions	Score
1	Frame Relay Protocols	1 to 4	
2	Frame Relay Configuration	5 to 8	
All questions		1 to 8	

1 Explain the purpose of Inverse ARP. Explain how Inverse ARP uses Frame Relay broadcasts.

2 What does NBMA stand for? Does it apply to X.25 networks or Frame Relay networks?

3 What is the name of the field that identifies, or addresses, a Frame Relay virtual circuit?

4 Which layer or layers of OSI are most closely related to the functions of Frame Relay? Why?

5 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?

6 What **show** command tells you when a PVC became active? How does the router know what time the PVC became active?

7 What **debug** option shows Inverse ARP messages?

____

____

____

8 What **show** command lists Frame Relay information about mapping? In what instances does the information displayed include the Layer 3 addresses of other routers?

____

____

____

The answers to the "Do I Know This Already?" quiz are found in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Q&A Sections." The suggested choices for your next step are as follows:

- **4 or less overall score**—Read the entire chapter. This includes the "Foundation Topics" and "Foundation Summary" sections, the "Q&A" section, and the scenarios at the end of the chapter.
- **2 or less on any quizlet**—Review the subsections of the "Foundation Topics" section, based on Table 10-1. Then move to the "Foundation Summary" section, the "Q&A" section, and the scenarios at the end of the chapter.
- **5 or 6 overall score**—Begin with the "Foundation Summary" section, and then go to the "Q&A" section and the scenarios at the end of the chapter.
- **7 or 8 overall score**—If you want more review of these topics, skip to the "Foundation Summary" section and then go to the "Q&A" section and the scenarios at the end of the chapter. Otherwise, move to the next chapter.

Foundation Topics

Frame Relay Protocols

29 Recognize key Frame Relay terms and features.

Frame Relay networks provide more features and benefits than simple point-to-point WAN links, but in order 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. Because Frame Relay is multiaccess, Frame Relay addressing is important. First, consider Figure 10-2, which shows some connectivity concepts for Frame Relay.

Figure 10-2 *Frame Relay Components*

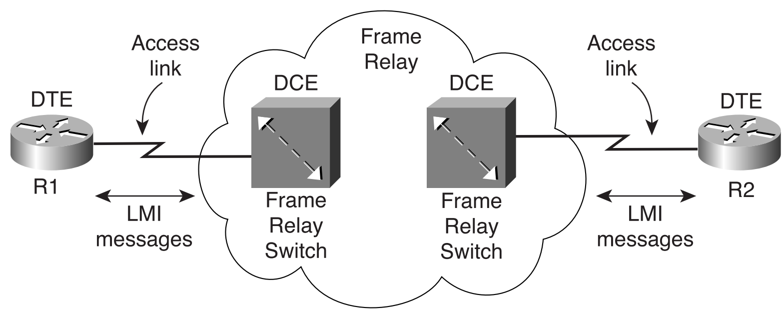
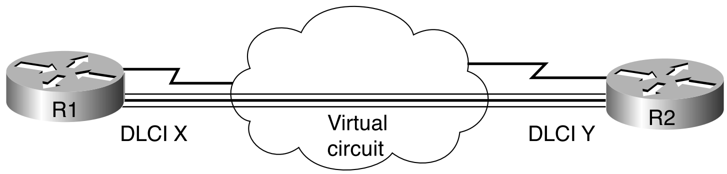


Figure 10-2 shows the most basic components of a Frame Relay network. A leased line is installed between the router and a nearby Frame Relay switch; this link is called the *access link*. To ensure that the link is working, the *Local Management Interface* (LMI) protocol is used between the router and the switch to notify each other of status and problems on the link. The routers are considered *data terminal equipment* (DTE), and the switches are *data communications equipment* (DCE).

Whereas, Figure 10-2 shows the physical and logical connectivity at each connection to the Frame Relay network, Figure 10-3 shows the end-to-end connectivity associated with a virtual circuit.

Figure 10-3 *Frame Relay PVC Concepts*



The logical path between each pair of DTEs is called a *virtual circuit* (VC). It is represented by the trio of parallel lines in the figure. Typically, the service provider preconfigures all the required details of a VC; these VCs are called permanent virtual circuits (PVCs). 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. R1 uses a Frame Relay address called a *data-link connection identifier* (DLCI) in the Frame Relay header. This allows the switches to deliver the frame to R2, ignoring the details of the Layer 3 packet, only caring to look at the Frame Relay header and trailer. Just like on a point-to-point serial link, when the service provider forwards the frame over a physical circuit between R1 and R2, with Frame Relay the provider forwards the frame over a logical virtual circuit from R1 to R2.

Table 10-2 lists the components shown in Figure 10-2 and some associated terms. Table 10-3 lists some of the associated standards for Frame Relay. After the tables, the most important features of Frame Relay are described in further detail.

Table 10-2 *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.
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 DTE and DCE.
Access rate (AR)	The speed at which the access link is clocked. This choice affects the price of the connection.

Table 10-2 *Frame Relay Terms and Concepts (Continued)*

Term	Description
Committed information rate (CIR)	The rate at which the DTE can send data for an individual VC, for which the provider commits to deliver that amount of data. The provider sends any data in excess of this rate for this VC if its network has capacity at the time. This choice typically affects the price of each VC.
Burst rate	The rate and length of time which, for a particular VC, the DTE can send faster than the CIR, and the provider agrees to forward the data. Often it is expressed as a *burst size*. A Frame Relay DTE can send burst size bits, wait a moment, send burst size bits, wait, and so on, with the average being the CIR. This choice typically affects the price of each VC.
Data-link connection identifier (DLCI)	A Frame Relay address used in Frame Relay headers to identify the VC.
Forward explicit congestion notification (FECN)	The bit in the Frame Relay header that signals to anyone receiving the frame (switches and DTEs) that congestion is occurring in the same direction as the frame. Switches and DTEs can react by slowing the rate at which data is sent in that direction.
Backward explicit congestion notification (BECN)	The bit in the Frame Relay header that signals to anyone receiving the frame (switches and DTEs) that congestion is occurring in the opposite (backward) direction as the frame. Switches and DTEs can react by slowing the rate at which data is sent in that direction.
Discard eligibility (DE)	The bit in the Frame Relay header that, if frames must be discarded, signals a switch to choose this frame to discard instead of another frame without the DE bit set.
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.
Link Access Procedure Frame Mode Bearer Services (LAPF)	Defines the basic Frame Relay header and trailer. The header includes DLCI, FECN, BECN, and DE bits.

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, a vendor consortium, also defines several Frame Relay specifications, many of which have been added to the standards bodies' documents. Table 10-3 lists the most important of these specifications.

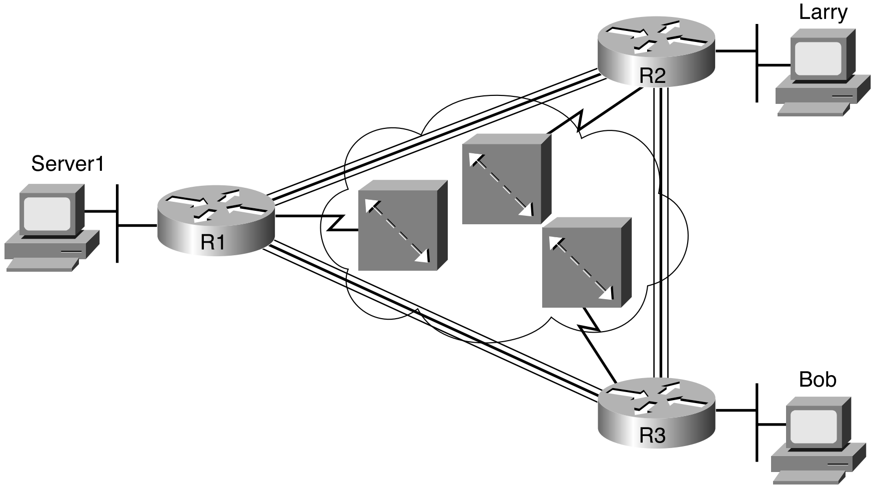
Table 10-3 *Frame Relay Protocol Specifications*

What the Specification Defines	ITU Document	ANSI Document
Data-link specifications, including LAPF header/trailer	Q.922 Annex A	T1.618
PVC management, LMI	Q.933 Annex A	T1.617 Annex D
SVC signaling	Q.933	T1.617
Multiprotocol encapsulation (originated in RFC 1490/2427)	Q.933 Annex E	T1.617 Annex 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 10-4, which is a typical Frame Relay network with three sites.

Figure 10-4 *Typical Frame Relay Network with Three Sites*



A virtual circuit is a concept that describes a logical path between two Frame Relay DTEs. The term *virtual circuit* describes the concept well. It acts like a point-to-point circuit, but physically it is not, 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.

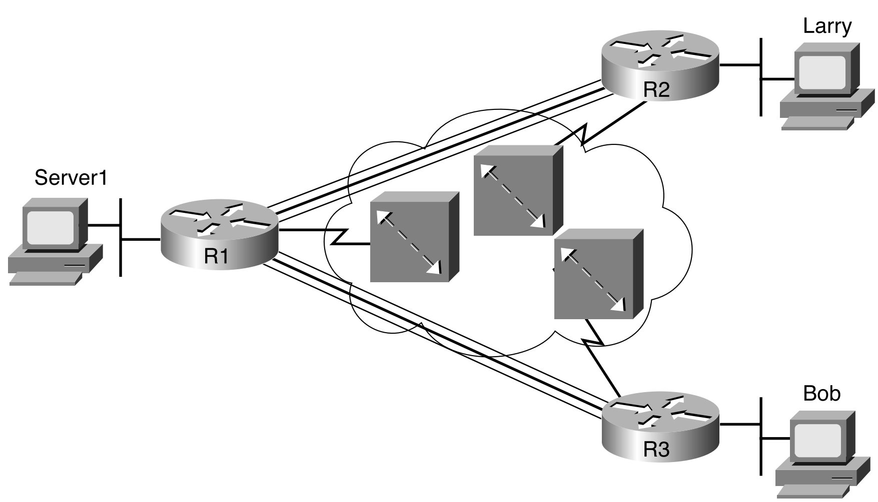
Originally, people with leased-line networks were reluctant to migrate to Frame Relay, because they would be competing with other customers for the provider's capacity inside the cloud. To address these fears, Frame Relay is designed with the concept of a committed information rate (CIR). Each VC has a CIR, which is a guarantee by the provider that a particular VC gets at least that much bandwidth.

It's interesting that, even in this three-site network, it's probably less expensive to use Frame Relay than to use point-to-point links. Now imagine an organization with 100 sites that needs any-to-any connectivity. How many leased lines are required? 4950! And besides that, you would need 99 serial interfaces per router. Or you could have 100 access links to local Frame Relay switches, one per router, and have 4950 VCs running over them. Also, you would need only one serial interface on each router! The Frame Relay topology is easier for the service provider to implement, costs the provider less, and makes better use of the core of the provider's network. 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; whereas, SVCs are created dynamically. PVCs are by far the more popular of the two. PVCs are covered on the CCNA exam, but SVCs are not.

When the Frame Relay network is engineered, the design might not include a PVC between each pair of sites. Figure 10-4 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 10-5 shows the same network, but this time with 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 10-5 *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 and because there are fewer VCs. 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, but multiple PVCs flow across the one link. For example, consider Figure 10-5 from R1's perspective. Server1 sends a packet to Larry. It comes across the Ethernet, R1 gets it and matches his 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? Well, 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.

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, but Cisco expects CCNAs to master the differences. 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 in order 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. First, they 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. The second important function of the status message is to signal whether a PVC is active or inactive. Even though each PVC is predefined, its status can change. So an access link might be up, but it could be down. The router could then remove all routes using that PVC but leave other routes that use other working PVCs alone.

Three LMI protocol options are available in Cisco IOS Software: Cisco, ITU, and ANSI. Each LMI option is slightly different and, therefore, incompatible with the other two. For example, the Cisco and ANSI Q.933-A LMIs call for the use of DLCI 1023 for LMI messages; whereas, T1.617-D specifies 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, and the local switch must agree.

Configuring the LMI type is easy. Today's most popular option is to use the default LMI setting, which uses the LMI autosense feature. Because the default LMI setting is supported by Cisco IOS Software Release 11.2 and later, you do not need to code the LMI type. The LMI type can be configured, but this disables the autosense feature.

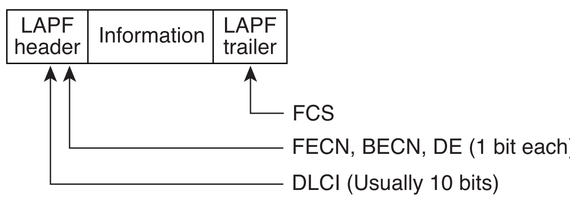
Table 10-4 outlines the three LMI types, their origin, and the keyword used in the Cisco **frame-relay lmi-type** interface subcommand.

Table 10-4 *Frame Relay LMI Types*

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

A Frame Relay header and trailer are used to encapsulate a packet before it is sent out an access link. Frame Relay uses the standard Link Access Procedure Frame Bearer Services (LAPF) header, defined by 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 10-6 diagrams the frame.

Figure 10-6 *LAPF Header*



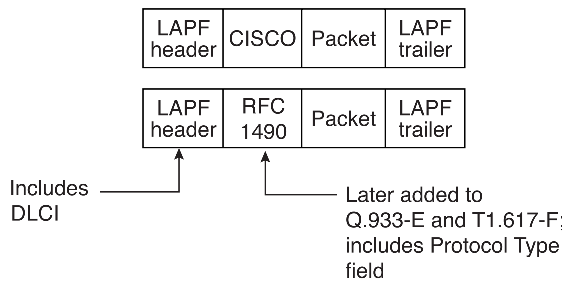
The LAPF header and trailer do not provide all the fields typically needed by routers. In particular, Figure 10-6 does not show the presence of a Protocol Type field. As discussed in Chapter 3, a field in the header must define the type of header, typically a Layer 3 packet, that follows the data-link 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. (See Chapter 3 for more information on the concept of a Protocol Type field in data-link headers.)

Two solutions were created to compensate for the lack of a Protocol Type field. Cisco and three other companies created an additional header, which comes first in the Information field shown in Figure 10-6. It includes a two-byte Protocol Type field, with values matching the same field used for HDLC by Cisco. The second solution was defined in RFC 1490, “Multiprotocol Interconnect over Frame Relay,” which was written to ensure multivendor interoperability between Frame Relay DTEs. This solution includes use of a Protocol Type field and adds many other options, including support for bridged frames. ITU and ANSI later incorporated RFC 1490 headers into specs Q.933 Annex E and T1.617 Annex F, respectively. The encapsulation option defined by Cisco and others, and the option as originally defined in RFC 1490, are the two encapsulation options in the Cisco IOS Software today. They are called cisco and ietf.

NOTE RFC 1490 has been superceded by RFC 2427. You will want to remember both numbers, particularly the older 1490, because it is referred to often in documentation from Cisco and other vendors.

DTEs use and react to the fields specified by these two types of encapsulation; Frame Relay switches ignore these fields. Figure 10-7 provides a conceptual diagram of the two forms of encapsulation. *Because the frames flow from DTE to DTE, both DTEs must agree to the encapsulation used*. The switches do not care. However, each VC can use a different encapsulation.

Figure 10-7 *Cisco and RFC 1490/2427 Encapsulation*



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). Permanent virtual circuits (PVCs) are typically used, and the data-link connection identifier (DLCI) is used to address or identify each individual PVC. The LMI protocol is used to manage the access link, and the LMI type must match between the router and the local switch. Finally, the routers agree to the style of encapsulation used. Both encapsulation types include a protocol type field, which identifies the next header that follows the Frame Relay header.

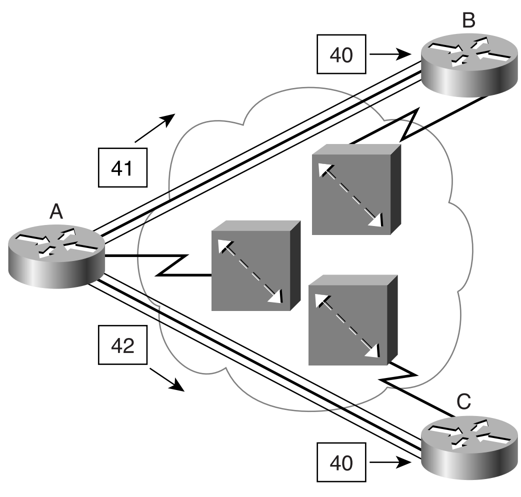
DLCIs can be both simple and confusing. The earlier explanation simply states that the DLCI is used to identify a VC, so when multiple VCs use the same access link, the Frame Relay switches know how to forward the frames correctly. You could know just that, look at the configuration examples later in this chapter, and probably learn to create new configurations. You would probably even get all the exam questions right with that depth of knowledge. 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.

Start Extra Credit

Frame Relay addressing and switching define how to deliver frames across a Frame Relay network. Because a router uses a single access link but can send to many other routers, there must be something to identify the other device—in other words, an address. The DLCI is the Frame Relay address. However, DLCIs are used to address VCs. The logic and use of DLCIs are different from the addresses seen for other protocols covered in this book. This difference is mainly due to the use of the DLCI and the fact that *there is a single DLCI field in the header—there are not both source and destination DLCI fields.*

A few characteristics of DLCIs are important to understand before getting into their actual 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. In Figure 10-8, notice that DLCI 40 is used on two access links to describe two different PVCs. Because DLCI 40 is used on two different access links, there is no conflict.

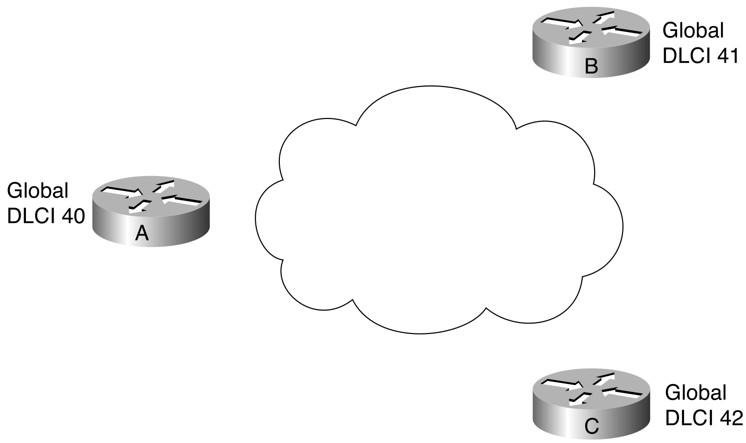
Figure 10-8 *Frame Relay Addressing with A Sending to B and C*



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 one DLCI to represent more than one VC on the same link. Otherwise, the Frame Relay switch would not know how to forward frames correctly.

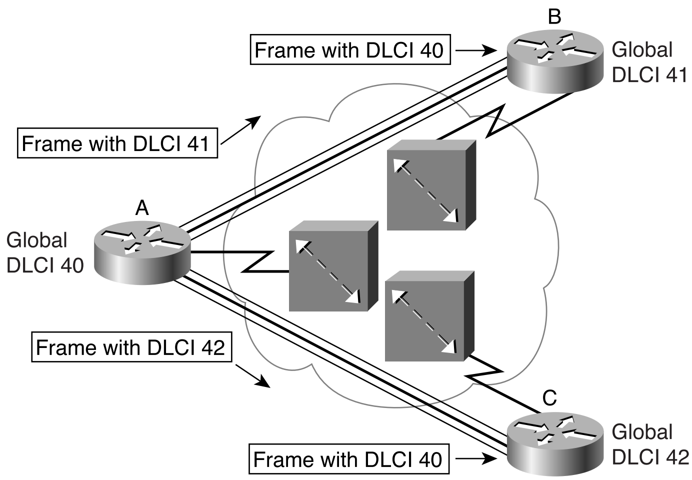
Most people get confused about DLCIs the first time they think about the local significance of DLCIs and the existence of only a single DLCI field in the Frame Relay header. Global addressing solves this problem by making DLCI addressing look like LAN addressing in concept. Global addressing is simply a way of choosing DLCI numbers when planning a Frame Relay network. Because local addressing is a fact, global addressing does not change the rules. Global addressing just makes DLCI assignment more obvious—once you get used to it. Here's how it works: The service provider hands out a planning spreadsheet and a diagram. Figure 10-9 is an example of such a diagram, with the "global" DLCIs shown.

Figure 10-9 *Frame Relay Global DLCIs*



Global addressing is planned as shown in Figure 10-9, with the resulting use of DLCIs as shown in Figure 10-8. For example, Router A uses DLCI 41 when sending a frame to Router B, because B's "global" DLCI is 41. Likewise, A uses 42 for the PVC 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 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 10-10 outlines this example.

Figure 10-10 *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 I want to send a frame to someone, I 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 10-10 shows a different DLCI value as the frames are received by Routers B and C? For example, Router A is sending 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 B and C receive their frames, the DLCI value is actually the DLCI of the sender. Why? Well, when 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 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.

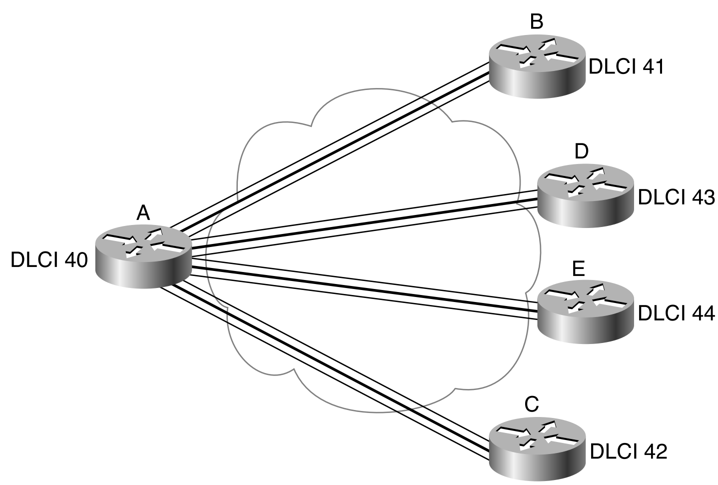
Figure 10-10 describes what really happens in a typical Frame Relay network. Service providers supply a planning spreadsheet and diagrams with global DLCIs listed. Table 10-5 gives a tabular view of what DLCIs are used.

Table 10-5 *DLCI Swapping in the Frame Relay Cloud*

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

One benefit of global addressing is that new sites can be added more conveniently. Examine Figure 10-11, which adds Routers D and E. The service provider simply states that global DLCI 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 10-11 *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.

NOTE If you're taking the ICND course or reading the ICND book, you might notice that it does not cover the concepts behind global addressing. Can you build router configurations if you just understand local DLCI addressing? Yes. But if you expect to work with Frame Relay, understanding both local and global addressing concepts is very important, because some people will want to use one convention, and some will want to use the other.

End Extra Credit

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 is both similar to and a little different from LAN and point-to-point WAN links. These differences introduce some additional considerations for passing Layer 3 packets across a Frame Relay network. As a CCNA, you need to concern yourself with three key issues relating to Layer 3 flows over Frame Relay:

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

The following sections cover these three 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

Figure 10-12 shows the first alternative, which is to use a single subnet. The illustration 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, which makes the

Frame Relay network behave like a LAN, at least in how IP addressing works. (These concepts also apply to IPX networks. In this case, a single IPX network is needed.) Figure10-12 also shows IPX and IP addresses. The IPX and IP addresses are configured as subcommands on the serial interface. (Configuration details are shown in a later section.) Table 10-6 summarizes the addresses used in Figure 10-12.

Figure 10-12 *Full Mesh with IP and IPX Addresses*

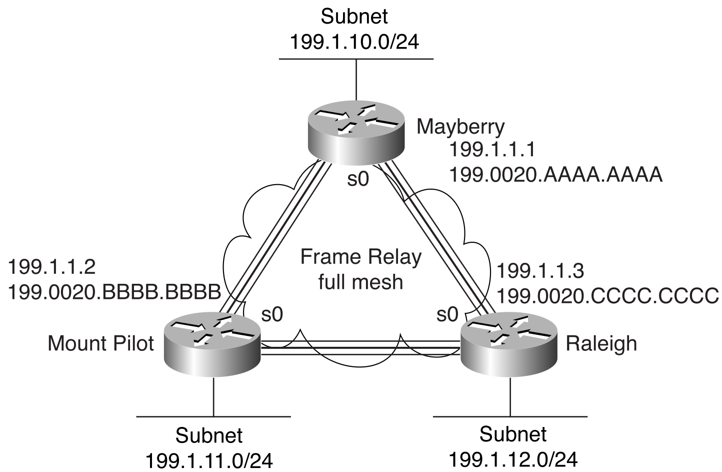


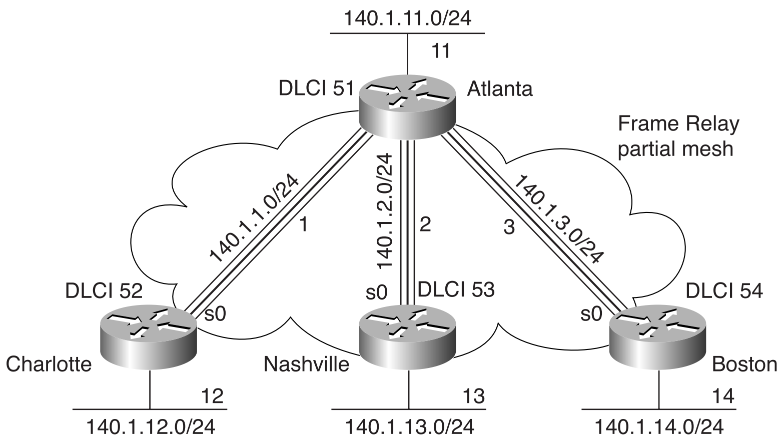
Table 10-6 *IP and IPX Addresses with No Subinterfaces*

Router	IP Address of Frame Relay Interface	IPX Network of Frame Relay Interface	IPX Address
Mayberry	199.1.1.1	199	199.0200.aaaa.aaaa
Mount Pilot	199.1.1.2	199	199.0200.bbbb.bbbb
Raleigh	199.1.1.3	199	199.0200.cccc.cccc

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. The problems are that most Frame Relay networks are not full mesh, and the single-subnet option has some deficiencies when the network is a partial mesh.

The second IP addressing alternative, the single-subnet-per-VC alternative, is most useful with a partially meshed Frame Relay network (see Figure 10-13). 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 onto servers at a few locations, and most of the traffic is between a remote site and those servers.

Figure 10-13 *Partial Mesh with IP and IPX Addresses*



The single-subnet-per-VC alternative matches the logic behind a set of point-to-point links. Because there is a separate subnet for each point-to-point link, using a single subnet per VC has some advantages in this case. Table 10-7 shows the IP and IPX addresses for the partially meshed Frame Relay network illustrated in Figure 10-13. The addresses would be configured as subcommands on serial subinterfaces.

NOTE The notation /24 signifies a subnet mask with 24 binary 1s—in other words, 255.255.255.0.

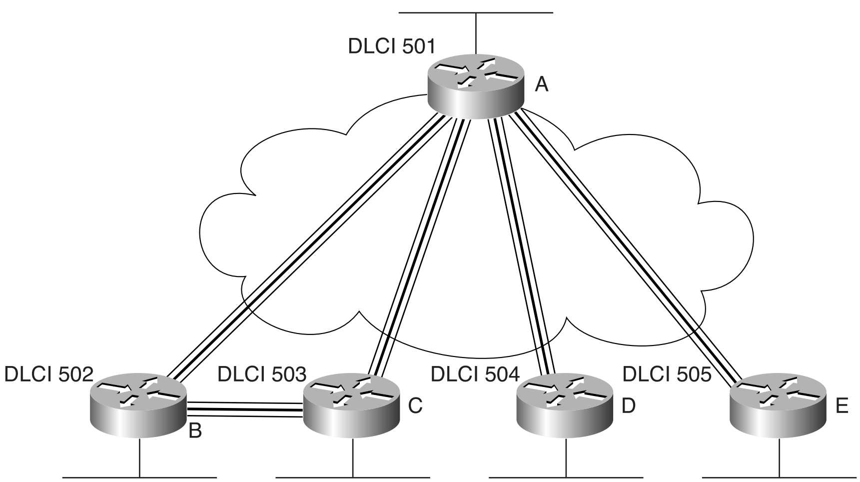
Table 10-7 *IP and IPX Addresses with Point-to-Point Subinterfaces*

Router	Subnet	IP Address	IPX Network	IPX Address
Atlanta	140.1.1.0/24	140.1.1.1	1	1.0200.aaaa.aaaa
Charlotte	140.1.1.0/24	140.1.1.2	1	1.0200.bbbb.bbbb
Atlanta	140.1.2.0/24	140.1.2.1	2	2.0200.aaaa.aaaa
Nashville	140.1.2.0/24	140.1.2.3	2	2.0200.cccc.cccc
Atlanta	140.1.3.0/24	140.1.3.1	3	3.0200.aaaa.aaaa
Boston	140.1.3.0/24	140.1.3.4	3	3.0200.dddd.dddd

Frame Relay VCs essentially create a subdivision of the traffic on a serial interface per VC. 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 and three IPX addresses associated with its Serial0 interface by configuring three separate subinterfaces associated with the single physical interface. Subinterfaces can treat each VC as though it were a point-to-point serial link. Each of the three subinterfaces of Serial0 on Atlanta would be assigned a different IP address and IPX address from the list in Table 10-7. (Sample configurations appear in the next section.)

The third alternative of Layer 3 addressing is a hybrid of the first two alternatives. Consider Figure 10-14, which shows a trio of routers with VCs between each of them, as well as two other VCs to remote sites.

Figure 10-14 *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; five subnets and five IPX networks are needed for the Frame Relay network. However, Routers A, B, and C create a full mesh between each other. This allows Routers A, B, and C to use one subnet and IPX network. The other two VCs—one between A and D and one between A and E—are treated as two separate Layer 3 groups. The result is a total of three subnets and three IPX network numbers.

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. Multipoint subinterfaces are used among Routers A, B, and C in Figure 10-14. (A multipoint subinterface is a subinterface that is used when multiple VCs terminate at a router.) Table 10-8 summarizes the addresses and subinterfaces that are used.

Table 10-8 *IP and IPX Addresses with Point-to-Point and Multipoint Subinterfaces*

Router	Subnet	IP Address	IPX Network	IPX Address	Subinterface Type
A	140.1.1.0/24	140.1.1.1	1	1.0200.aaaa.aaaa	Multipoint
B	140.1.1.0/24	140.1.1.2	1	1.0200.bbbb.bbbb	Multipoint
C	140.1.1.0/24	140.1.1.3	1	1.0200.cccc.cccc	Multipoint
A	140.1.2.0/24	140.1.2.1	2	2.0200.aaaa.aaaa	Point-to-point
D	140.1.2.0/24	140.1.2.4	2	2.0200.dddd.dddd	Point-to-point
A	140.1.3.0/24	140.1.3.1	3	3.0200.aaaa.aaaa	Point-to-point
E	140.1.3.0/24	140.1.3.5	3	3.0200.eeee.eeee	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 and IPX network per PVC. All three alternatives are fair game on the exam, however!

The later section “Frame Relay Configuration” provides full configurations for all three cases illustrated in Figures 10-12, 10-13, and 10-14.

Broadcast Handling

The second consideration for Layer 3 over Frame Relay 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 in order for several features to work. In particular, routing protocol updates and SAP updates are broadcasts.

The solution to the broadcast dilemma for Frame Relay has two parts. First, Cisco IOS Software sends copies of the broadcasts across each VC that you tell it to. 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 into 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.

NOTE Although the CCNP exam, not the CCNA exam, covers such issues as dealing with overhead, a short example shows the significance of this overhead. If a router knows 1000 routes, uses RIP, and has 50 VCs, 1.072 MB of RIP updates are 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.)

Knowing how to tell the router to forward these broadcasts to each VC is important on the CCNA exam and, therefore, is covered later 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.

Start Extra Credit

Split Horizon

The third network layer consideration when you're using Frame Relay is understanding how split horizon works over Frame Relay. Split horizon is useful for preventing routing loops by preventing a router from advertising a route onto the same interface on which the route was learned. (Refer to Chapter 7, "Routing and Routing Protocols," for a full explanation.) However, split horizon can cause some problems with Frame Relay. Thankfully, several configuration options help you deal with this issue.

Understanding the problem is difficult without an example. Refer back to Figure 10-13. Atlanta uses a single serial interface—say, Serial0. With split horizon enabled on Atlanta's Serial0, Atlanta learns about 140.1.12.0/24 from Charlotte, but Atlanta does not advertise Charlotte's 140.1.12.0/24 subnet in its updates to Nashville or Boston. So no traffic could flow from Boston or Nashville to Charlotte.

Two solutions to this problem are supported in the Cisco IOS Software. First, split-horizon logic applies to subinterfaces, as if they were separate interfaces. In other words, Atlanta uses a different subinterface for each VC to the three remote sites. Split horizon is enabled on each subinterface. However, because the routing updates from Charlotte are considered to enter Atlanta via one subinterface, and because routing updates to Nashville and Boston exit two other subinterfaces, advertising 140.1.12.0/24 to Nashville and Boston is allowed. No special action is required.

The other solution is to disable split horizon. Normally that would be a bad idea, but with Frame Relay, it is generally more acceptable. Consider Figure 10-12. When all three VCs are up, no problem exists. However, if the VC from Mount Pilot to Raleigh went down, split horizon on Mayberry would be harmful. Mount Pilot advertises its route to 199.1.11.0 on its local subnet to Mayberry. Mayberry receives that route. However, because subinterfaces are not used, Mayberry does not advertise Mount Pilot's 199.1.11.0 subnet to Raleigh when split horizon is enabled.

The multipoint subinterfaces used between Routers A, B, and C in Figure 10-14 would experience the same problems for the same reasons described for Figure 10-12.

The second solution to the split horizon problem is to disable split horizon when not using subinterfaces or when using multipoint subinterfaces. Conveniently, the Cisco IOS Software defaults to disabling split horizon on Frame Relay interfaces in all cases except for point-to-point subinterfaces. Table 10-9 summarizes these settings and shows that the current default settings work around the split horizon issues just described.

Table 10-9 *Split Horizon and Frame Relay Interfaces*

Type of Configuration	Split Horizon Is
No subinterfaces	Disabled
Point-to-point subinterfaces	Enabled
Multipoint subinterfaces	Disabled

If the default value for split horizon is not what you want, the **ip split horizon** interface configuration command can be used to enable split horizon. Similarly, the **no ip split horizon** interface configuration command disables split horizon on that interface.

End Extra Credit

Frame Relay Configuration

30 List commands to configure Frame Relay LMIs, maps, and subinterfaces.

31 List commands to monitor Frame Relay operation in the router.

This chapter describes Frame Relay concepts. For example, three LMI types and two encapsulation types are available. Depending on the placement of your VCs, you might want to use one subnet for the whole Frame Relay network, one subnet per VC, or a mixture of the two. You might need to configure static mapping of IP addresses and their corresponding DLCIs. And you will definitely need to tell the router that its serial interface is using Frame Relay instead of the default of HDLC.

Basic configuration of Frame Relay in a Cisco router is relatively straightforward. The Cisco IOS Software uses good default values. Of course, Cisco expects CCNAs to know the optional parameters that are described in this section and the methods of changing the default values.

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 10-10 and 10-11 summarize the more popular commands used for Frame Relay configuration and verification. Two 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. The Cisco Press book *Interconnecting Cisco Network Devices* is also a good reference, particularly if you can't attend the instructor-led version of the ICND class.

Table 10-10 *Frame Relay Configuration Commands*

Command	Configuration Mode	Description
encapsulation frame-relay [**ietf** | **cisco**]	Interface	Defines the Frame Relay encapsulation that is used rather than HDLC, PPP, and so on.
frame-relay lmi-type {**ansi** | **q933a** | **cisco**}	Interface	Defines the type of LMI messages sent to the switch.
bandwidth *num*	Interface	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.
frame-relay map {*protocol protocol-address dlci*} **payload-compression frf9 stac caim** [*element-number*] [**broadcast**] [**ietf** | **cisco**]	Interface	Statically defines a mapping between a network layer address and a DLCI.
keepalive *sec*	Interface	Defines whether and how often LMI status inquiry messages are sent and expected.
interface serial *number.sub* [**point- to-point** | **multipoint**]	Global	Creates a subinterface or references a previously created subinterface.
frame-relay interface-dlci *dlci* [**ietf** | **cisco**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*]	Subinterface	Links or correlates a DLCI to the subinterface.

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

Command	Function
show interfaces [*type number*]	Shows the physical interface status.
show frame-relay pvc [**interface** *interface*][*dlci*]	Lists information about the PVC status.
show frame-relay lmi [*type number*]	Lists LMI status information.

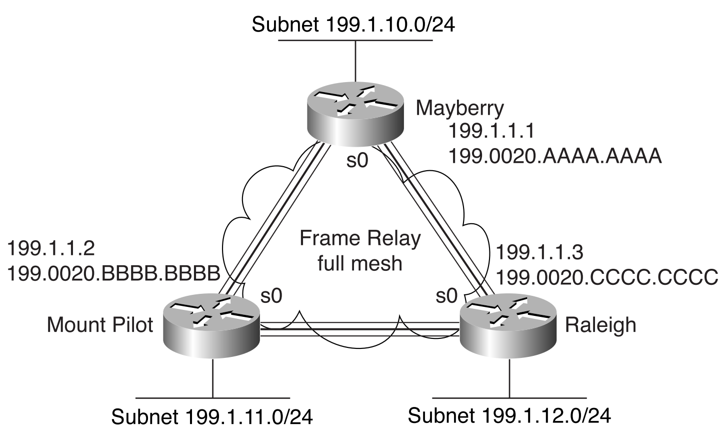
The network engineer plans the Frame Relay configuration based on several factors. When the service is ordered, the service provider specifies the LMI type that will be used. The engineer chooses the endpoints of the VCs, including whether to use a full mesh or partial mesh. Based on the location of 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. 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 rates.

Three examples of Layer 3 addressing were given earlier in this chapter, with the networks diagrammed in Figures 10-12, 10-13, and 10-14. The configurations matching those networks and addresses are shown next.

Fully-Meshed Network with One IP Subnet/IPX Network

The network engineer designed a fully-meshed network for the first example. This first sample network, based on the environment depicted in Figure 10-12, does not use subinterfaces, but rather includes all Frame Relay configuration under the physical interface. Multipoint subinterfaces could have been used instead. Examples 10-1, 10-2, and 10-3 show the configuration for the network shown in Figure 10-15.

Figure 10-15 *Full Mesh with IP and IPX Addresses*



Example 10-1 *Mayberry Configuration*

```
ipx routing 0200.aaaa.aaaa
!
interface serial0
encapsulation frame-relay
ip address  199.1.1.1  255.255.255.0
ipx network  199
!
interface ethernet 0
ip address  199.1.10.1  255.255.255.0
ipx network  1
!
router igrp 1
network  199.1.1.0
network 199.1.10.0
```

Example 10-2 *Mount Pilot Configuration*

```
ipx routing 0200.bbbb.bbbb
!
interface serial0
encapsulation frame-relay
ip address  199.1.1.2  255.255.255.0
ipx network  199
!
interface ethernet 0
ip address  199.1.11.2   255.255.255.0
ipx network  2
!
router igrp 1
network  199.1.1.0
network 199.1.11.0
```

Example 10-3 *Raleigh Configuration*

```
ipx routing 0200.cccc.cccc
!
interface serial0
encapsulation frame-relay
ip address  199.1.1.3  255.255.255.0
ipx network  199
!
interface ethernet 0
ip address  199.1.12.3   255.255.255.0
ipx network  3
!
router igrp 1
network  199.1.1.0
network 199.1.12.0
```

The configuration is simple in comparison with the protocol concepts. All default settings (Cisco IOS Software Release 12.2) are used. They are as follows:

- 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 has been received.
- Because either RIP or IGRP is being used, and all the configuration is on the real interface, split horizon is disabled.

In some cases, the default values are inappropriate. For example, if one router is not a Cisco router and does not support Cisco encapsulation, IETF encapsulation is required. 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 10-4 and 10-5 show the changes that would be made to Mayberry and Raleigh.

Example 10-4 *Mayberry Configuration with New Requirements*

```
ipx routing 0200.aaaa.aaaa
!
interface serial0
encapsulation frame-relay
frame-relay lmi-type ansi
frame-relay interface-dlci 42 ietfip address  199.1.1.1  255.255.255.0
ipx network  199
! rest of configuration unchanged from Example 10-1.
```

Example 10-5 *Raleigh Configuration with New Requirements*

```
ipx routing 0200.cccc.cccc
!
interface serial0
encapsulation frame-relay ietf
ip address  199.1.1.3  255.255.255.0
ipx network  199
!
! rest of configuration unchanged from Example 10-3.
```

The encapsulation was changed 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 could not change its encapsulation in the same way, because only one of the two VCs to Mayberry was directed to use IETF encapsulation. So Mayberry was forced to code the **frame-relay interface-dlci** command, coding the DLCI for the VC to Raleigh, with the **ietf** keyword being used to change the encapsulation just for this VC.

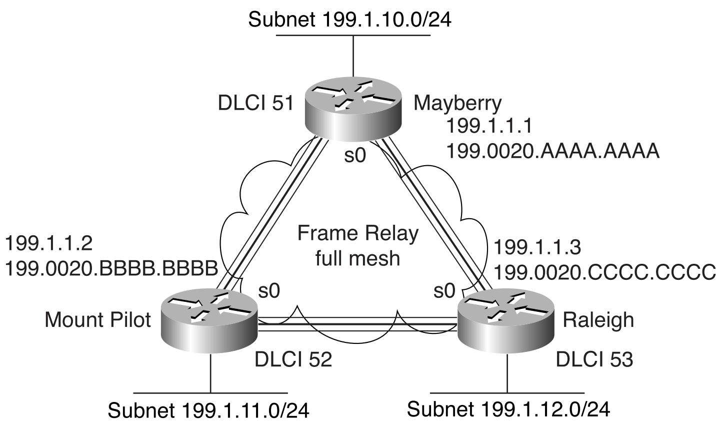
The LMI configuration in Mayberry would have been fine without any changes, because autosense would have recognized ANSI. However, by coding the **frame-relay lmi-type ansi**, Mayberry is forced to use ANSI, because this command disables autonegotiation of the LMI type.

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

Frame Relay Address Mapping

The DLCIs are missing from Figure 10-16 and the original configurations (Examples 10-1, 10-2, and 10-3). The configurations work as stated, and frankly, if you never knew the DLCIs, this network would work! However, knowing why you can make it work with no knowledge of the DLCIs means that you need to understand an important concept related to Frame Relay—namely, Frame Relay address mapping.

Figure 10-16 *Full Mesh with IP and IPX Addresses*



Mapping, as used here, means a correlation between a Layer 3 address and its corresponding Layer 2 address. For example, the IP ARP cache is an example of mapping. With IP ARP, you know the IP address of another device on the same LAN. When the ARP completes, you know another device's LAN (Layer 2) address. (For a review of IP ARP, see Chapter 6, "TCP/IP and IP Routing.") Likewise, we need a mapping between a router's Layer 3 address and the DLCI we use to reach that other router.

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

> 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.

The need for mapping is more apparent when you think about the routing process. A host in Mayberry sends an IP packet to a host in Mount Pilot. The packet arrives at the Mayberry router, which 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. 199.1.11.0 is Mount Pilot's subnet on its Ethernet interface.

The issue is simply this: What DLCI does Mayberry put into the Frame Relay header? We configured no DLCIs. However, the LMI uses status messages to tell Mayberry about the DLCIs. If the network works, obviously Mayberry knows the right DLCI value to use. To see the answer, consider Example 10-6, which shows some important commands that can be used to see how Mayberry makes the right choice for the DLCI.

Example 10-6 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
       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
```

continues

Example 10-6 show *Commands on Mayberry, Showing the Need for Mapping (Continued)*

```
  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
Serial0 (up): ipx 1.0200.bbbb.bbbb dlci 52(0x34,0xC40), dynamic,
              broadcast,, status defined, active
Serial0 (up): ipx 199.0000.3089.b170 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. Which one should be used? The **show frame-relay map** command output holds the answer. Notice the phrase **ip 199.1.1.2 dlci 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.

Mayberry can use two methods to build the mapping shown in the example. One uses a statically configured mapping, and the other uses a dynamic process called *Inverse ARP*. Before these two options are described, you need more background information. Table 10-12 lists the IP and IPX addresses of the three routers shown in Figure 10-16.

Table 10-12 *Layer 3 Addresses and DLCIs Used with Figure 10-16*

Router	Global DLCI	IP Address	IPX Address
Mayberry	51	199.1.1.1	199.0200.aaaa.aaaa
Mount Pilot	52	199.1.1.2	199.0200.bbbb.bbbb
Raleigh	53	199.1.1.3	199.0200.cccc.cccc

Example 10-7 lists the static Frame Relay map for the three routers shown in Figure 10-12. The DLCIs in Table 10-12 are the same as those used in Figure 10-16.

Example 10-7 frame-relay map *Commands*

```
Mayberry
interface serial 0
frame-relay map ip 199.1.1.2 52 broadcast
frame-relay map ipx 199.0200.bbbb.bbbb 52 broadcast
frame-relay map ip 199.1.1.3 53 broadcast
frame-relay map ipx 199.0200.cccc.cccc 53 broadcast
```

Example 10-7 frame-relay map *Commands (Continued)*

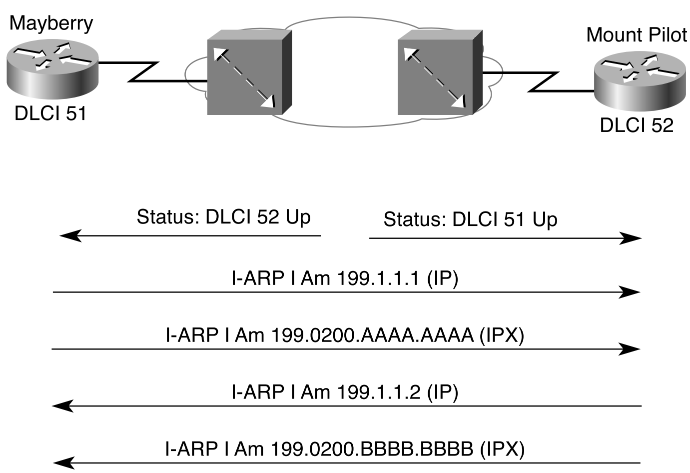
```
Mount Pilot
interface serial 0
frame-relay map ip 199.1.1.1 51 broadcast
frame-relay map ipx 199.0200.aaaa.aaaa 51 broadcast
frame-relay map ip 199.1.1.3 53 broadcast
frame-relay map ipx 199.0200.cccc.cccc 53 broadcast
```

```
Raleigh
interface serial 0
frame-relay map ip 199.1.1.1 51 broadcast
frame-relay map ipx 199.0200.aaaa.aaaa 51 broadcast
frame-relay map ip 199.1.1.2 52 broadcast
frame-relay map ipx 199.0200.bbbb.bbbb 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.

The alternative mapping solution is a dynamic protocol called Inverse ARP. Inverse ARP still creates a mapping between the Layer 3 address (for example, the IP address) and Layer 2 address (the DLCI). The process it uses is different from ARP on a LAN. After the VC is up, each DTE announces its network layer address to the DTE on the other end of the VC. It works as shown in Figure 10-17.

Figure 10-17 *Inverse ARP Process*



As shown in Figure 10-17, Inverse ARP announces its Layer 3 addresses as soon as the LMI signals that the PVCs are up. IP ARP reacts to an incoming packet and begins knowing the IP address but not the data link layer LAN address. Inverse ARP starts by learning the DLCI data link layer address and announces its own Layer 3 addresses right away. Inverse ARP is enabled by default in Cisco IOS Software Release 11.2 and later. Table 10-13 summarizes what occurs in the network shown in Figure 10-16.

Table 10-13 *Inverse ARP Messages for Figure 10-16*

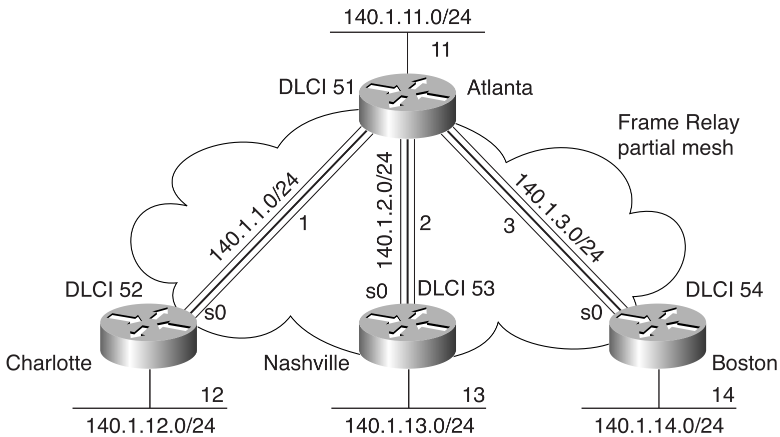
Sending Router	DLCI in Header of Inverse ARP Frame When Sent	Receiving Router	DLCI in Header of Inverse ARP Frame When Received	Information in Inverse ARP
Mayberry	52	Mount Pilot	51	I am 199.1.1.1.
Mayberry	52	Mount Pilot	51	I am 199.0200.aaaa.aaaa.
Mayberry	53	Raleigh	51	I am 199.1.1.1.
Mayberry	53	Raleigh	51	I am 199.0200.aaaa.aaaa.
Mount Pilot	51	Mayberry	52	I am 199.1.1.2.
Mount Pilot	51	Mayberry	52	I am 199.0200.bbbb.bbbb.
Mount Pilot	53	Raleigh	52	I am 199.1.1.2.
Mount Pilot	53	Raleigh	52	I am 199.0200.bbbb.bbbb.
Raleigh	51	Mayberry	53	I am 199.1.1.3.
Raleigh	51	Mayberry	53	I am 199.0200.cccc.cccc.
Raleigh	52	Mount Pilot	53	I am 199.1.1.3.
Raleigh	52	Mount Pilot	53	I am 199.0200.cccc.cccc.

To understand Inverse ARP, focus on the last two columns of Table 10-13. Each router receives some Inverse ARP "announcements." The Inverse ARP contains the Layer 3 address of the sender, and the Frame Relay header, of course, has a DLCI in it. These two values are placed into the Inverse ARP cache on the receiving router. For example, in the fifth 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 seen with the **show frame-relay map** command in Example 10-6.

Partially-Meshed Network with One IP Subnet/IPX Network Per VC

The second sample network, based on the environment shown in Figure 10-18, uses point-to-point subinterfaces. Examples 10-8 through 10-11 show the configuration for this network. The command prompts are included in the first example because they change when you're configuring subinterfaces.

Figure 10-18 *Partial Mesh with IP and IPX Addresses*



Example 10-8 *Atlanta Configuration*

```
Atlanta(config)#ipx routing 0200.aaaa.aaaa
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)#ipx network 1
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)#ipx network 2
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)#ipx network 3
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
Atlanta(config-if)#ipx network 11
```

Example 10-9 *Charlotte Configuration*

```
ipx routing 0200.bbbb.bbbb
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 140.1.1.2  255.255.255.0
ipx network 1
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 140.1.12.2 255.255.255.0
ipx network 12
```

Example 10-10 *Nashville Configuration*

```
ipx routing 0200.cccc.cccc
!
interface serial0
encapsulation frame-relay
!
interface serial 0.2 point-to-point
ip address 140.1.2.3 255.255.255.0
ipx network 2
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 140.1.13.3 255.255.255.0
ipx network 13
```

Example 10-11 *Boston Configuration*

```
ipx routing 0200.dddd.dddd
!
interface serial0
encapsulation frame-relay
!
interface serial 0.3 point-to-point
ip address 140.1.3.4 255.255.255.0
ipx network 3
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 140.1.14.4  255.255.255.0
ipx network 14
```

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 and split horizon is enabled, because these are the defaults when you're using a point-point subinterface. As you will see, Inverse ARP is not needed, and because there is only one VC per subinterface, the split horizon problem described earlier is not an issue.

Point-to-point subinterfaces are used in this configuration because the network is not fully meshed. If only the main interface were used, or if multipoint subinterfaces were used, the routing problems described in the preceding section would prevent remote sites from communicating with each other.

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 is needed when you're using subinterfaces. Consider router Atlanta in Figure 10-18. 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. 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 call during a failed access link. Many sites use the DLCI as the subinterface number. 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 10-12 shows the output from the most popular Cisco IOS Software Frame Relay EXEC commands for monitoring Frame Relay, as issued on router Atlanta.

Example 10-12 *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
```

continues

Example 10-12 *Output from EXEC Commands on Atlanta (Continued)*

```
  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
  in DE pkts 0             out DE pkts 0
  out bcast pkts 877        out bcast 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, all this information can be better gathered by an SNMP manager 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 was 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 is implied by the configuration. Mapping is needed only when more than two devices are attached to the link, and with a point-to-point subinterface, logically speaking, there are only two DTEs.

The **debug frame-relay lmi** output lists information for the sending and receiving LMI inquiries. The status message is sent by the switch; whereas, the status inquiry is sent by the DTE (router). The default setting with Cisco IOS Software is to send, and 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.

Partially-Meshed Network with Some Fully-Meshed Parts

Frame Relay networks built by CCNAs usually include both point-to-point and multipoint subinterfaces. This last sample network (based on the environment shown in Figure 10-19) uses both types of subinterfaces. Examples 10-13 through 10-17 show the configuration for this network. Table 10-14 summarizes the addresses and subinterfaces used.

Figure 10-19 *Hybrid of Full and Partial Mesh*

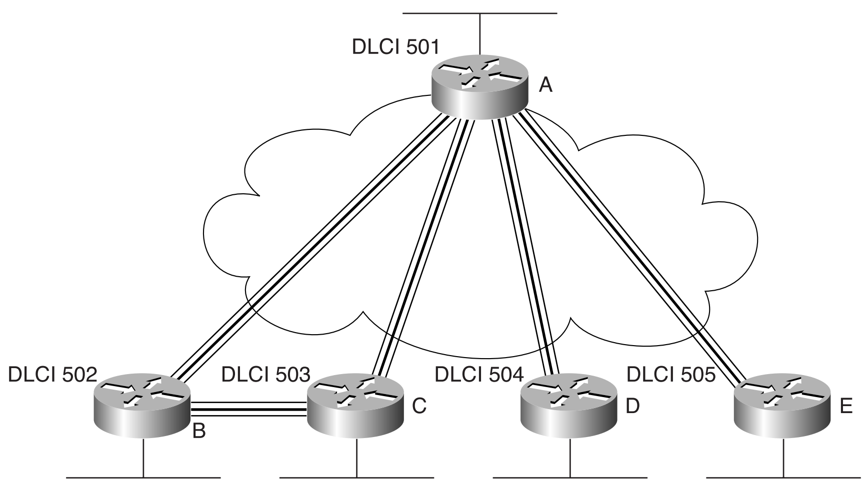


Table 10-14 *IP and IPX Addresses with Point-to-Point and Multipoint Subinterfaces*

Router	Subnet	IP Address	IPX Network	IPX Address	Subinterface Type
A	140.1.1.0/24	140.1.1.1	1	1.0200.aaaa.aaaa	Multipoint
B	140.1.1.0/24	140.1.1.2	1	1.0200.bbbb.bbbb	Multipoint
C	140.1.1.0/24	140.1.1.3	1	1.0200.cccc.cccc	Multipoint
A	140.1.2.0/24	140.1.2.1	2	2.0200.aaaa.aaaa	Point-to-point
D	140.1.2.0/24	140.1.2.4	2	2.0200.dddd.dddd	Point-to-point
A	140.1.3.0/24	140.1.3.1	3	3.0200.aaaa.aaaa	Point-to-point
E	140.1.3.0/24	140.1.3.5	3	3.0200.eeee.eeee	Point-to-point

Example 10-13 *Router A Configuration*

```
hostname RouterA
!
ipx routing 0200.aaaa.aaaa
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 140.1.1.1  255.255.255.0
ipx network 1
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
ipx network 2
frame-relay interface-dlci 504
!
interface serial 0.3 point-to-point
ip address 140.1.3.1 255.255.255.0
ipx network 3
frame-relay interface-dlci 505
!
interface ethernet 0
ip address 140.1.11.1 255.255.255.0
ipx network 11
```

Example 10-14 *Router B Configuration*

```
hostname RouterB
!
ipx routing 0200.bbbb.bbbb
!
```

Example 10-14 *Router B Configuration (Continued)*

```
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 140.1.1.2  255.255.255.0
ipx network 1
frame-relay interface-dlci 501
frame-relay interface-dlci 503
!
interface ethernet 0
ip address 140.1.12.2 255.255.255.0
ipx network 12
```

Example 10-15 *Router C Configuration*

```
hostname RouterC
!
ipx routing 0200.cccc.cccc
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 140.1.1.3  255.255.255.0
ipx network 1
frame-relay interface-dlci 501
frame-relay interface-dlci 502
!
interface ethernet 0
ip address 140.1.13.3 255.255.255.0
ipx network 13
```

Example 10-16 *Router D Configuration*

```
hostname RouterD
!
ipx routing 0200.dddd.dddd
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 140.1.2.4  255.255.255.0
ipx network 2
frame-relay interface-dlci 501
!
interface ethernet 0
ip address 140.1.14.4 255.255.255.0
ipx network 14
```

Example 10-17 *Router E Configuration*

```
hostname RouterE
!
ipx routing 0200.eeee.eeee
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 140.1.3.5 255.255.255.0
ipx network 3
frame-relay interface-dlci 501
!
interface ethernet 0
ip address 140.1.15.5 255.255.255.0
ipx network 15
```

On Routers A, B, and C, a multipoint subinterface is used. These three routers each have a PVC to the other two, making the use of multipoint reasonable. The term *multipoint* simply means that there is more than one DTE. Like point-to-point subinterfaces, multipoint subinterfaces use the **frame-relay interface-dlci** command. Notice that there are two for each multipoint subinterface in this case. The reason is that each PVC associated with this subinterface must be identified.

No mapping statements are required for the configurations shown in Examples 10-13 through 10-17 because Inverse ARP is enabled on the multipoint subinterfaces by default. The **show frame-relay map** command lists the mapping information learned by Inverse ARP. 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.

Router A is the only router using both multipoint and point-to-point subinterfaces. On Router A's Serial0.1 interface, multipoint is in use, with DLCIs for Router B and Router C 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.

Example 10-18 shows the contents of the Frame Relay map table, which are a result of Inverse ARP. This example also shows a copy of the **debug frame-relay events**, showing the contents of the Inverse ARP messages. The **debug** in Example 10-18 provides some insight into Inverse ARP operation.

Example 10-18 *Frame Relay Maps and Inverse ARP on Router C*

```
RouterC#show frame-relay map
Serial0.10 (up): ip 140.1.1.1 dlci 501(0x1F5,0x7C50), dynamic,
              broadcast,, status defined, active
Serial0.10 (up): ip 140.1.1.2 dlci 502(0x1F6,0x7C60), dynamic,
              broadcast,, status defined, active
Serial0.10 (up): ipx 1.0200.aaaa.aaaa dlci 501(0x1F5,0x7C50), dynamic,
              broadcast,, status defined, active
Serial0.10 (up): ipx 1.0200.bbbb.bbbb 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.
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 = 0xE427A0, datagramsize = 46
FR encap = 0x7C510300
80 00 00 00 08 06 00 0F 81 37 02 0A 00 09 00 00
00 00 00 01 02 00 AA AA AA AA 7C 51 00 00 00 01
02 00 CC CC CC CC 1B 99 D0 CC

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

Serial0.1: FR ARP input
datagramstart = 0xE47188, datagramsize = 46
FR encap = 0x7C610300
80 00 00 00 08 06 00 0F 81 37 02 0A 00 09 00 00
00 00 00 01 02 00 BB BB BB BB 7C 61 00 00 00 01
02 00 CC CC CC CC 1B 99 D0 CC
```

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 and IPX addresses in the output. These addresses are highlighted in Example 10-18. For example, the first three 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. The IPX address should be even easier to recognize, because it is already in hexadecimal format in the configuration.

NOTE Enabling **debug** options increases the router's CPU utilization. Depending on how much processing is required and how many messages are generated, it is possible to significantly degrade performance and possibly crash the router. This is a result of the memory and processing used to look for the requested information and to process the messages. You might want to first type the command **no debug all** and then type your **debug** command. If your **debug** creates too much output, you can easily go back to the **no debug all** command by pressing Ctrl-P twice.

Foundation Summary

The Foundation Summary is a collection of tables and figures that provide a convenient review of many key concepts in this chapter. If you are already comfortable with the topics in this chapter, this summary will help you recall a few details. If you just read this chapter, this review will help solidify some key facts. If you are doing your final preparation before the exam, these tables and figures are a convenient way to review the day before the exam.

Figure 10-20 shows some connectivity used for Frame Relay.

Figure 10-20 *Frame Relay Components*

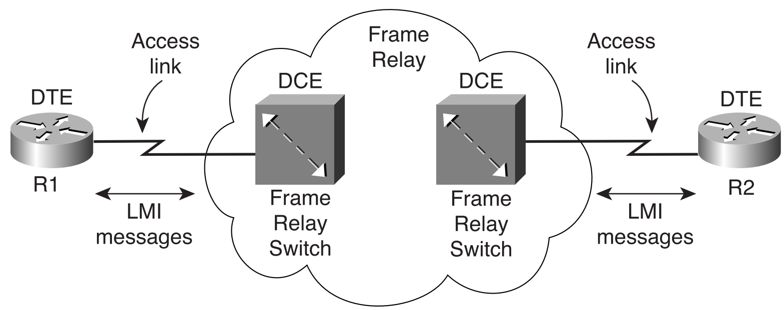
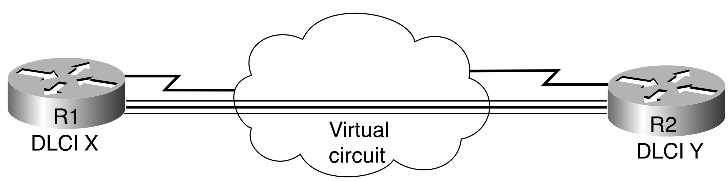


Figure 10-21 shows the physical and logical connectivity at each connection to the Frame Relay network.

Figure 10-21 *Frame Relay Concepts*



For reference, Table 10-15 lists the components shown in Figure 10-20 and some associated terms.

Table 10-15 *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.
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 DTE and DCE.
Access rate (AR)	The speed at which the access link is clocked. This choice affects the price of the connection.
Committed information rate (CIR)	The rate at which the DTE can send data for an individual VC, for which the provider commits to deliver that amount of data. The provider sends any data in excess of this rate for this VC if its network has capacity at the time. This choice typically affects the price of each VC.
Burst rate	The rate and length of time which, for a particular VC, the DTE can send faster than the CIR, and the provider agrees to forward the data. Often it is expressed as a *burst size*. A Frame Relay DTE can send burst size bits, wait a moment, send burst size bits, wait, and so on, with the average being the CIR. This choice typically affects the price of each VC.
Data-link connection identifier (DLCI)	A Frame Relay address used in Frame Relay headers to identify the VC.
Forward explicit congestion notification (FECN)	The bit in the Frame Relay header that signals to anyone receiving the frame (switches and DTEs) that congestion is occurring in the same direction as the frame. Switches and DTEs can react by slowing the rate at which data is sent in that direction.
Backward explicit congestion notification (BECN)	The bit in the Frame Relay header that signals to anyone receiving the frame (switches and DTEs) that congestion is occurring in the opposite (backward) direction as the frame. Switches and DTEs can react by slowing the rate at which data is sent in that direction.

Table 10-15 *Frame Relay Terms and Concepts (Continued)*

Term	Description
Discard eligibility (DE)	The bit in the Frame Relay header that, if frames must be discarded, signals a switch to choose this frame to discard instead of another frame without the DE bit set.
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.
Link Access Procedure Frame Mode Bearer Services (LAPF)	Defines the basic Frame Relay header and trailer. The header includes DLCI, FECN, BECN, and DE bits.

Table 10-16 lists the most important of the Frame Relay specifications.

Table 10-16 *Frame Relay Protocol Specifications*

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

Table 10-17 outlines the three LMI types, their origin, and the keyword used in the Cisco **frame-relay lmi-type** interface subcommand.

Table 10-17 *Frame Relay LMI Types*

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

Table 10-18 summarizes the default split horizon settings used for each type of Frame Relay interface.

Table 10-18 *Split Horizon and Frame Relay Interfaces*

Type of Configuration	Split Horizon Is
No subinterfaces	Disabled
Point-to-point subinterfaces	Enabled
Multipoint subinterfaces	Disabled

Tables 10-19 and 10-20 summarize the more popular commands used for Frame Relay configuration and verification.

Table 10-19 *Frame Relay Configuration Commands*

Command	Configuration Mode	Description
encapsulation frame-relay [**ietf** | **cisco**]	Interface	Defines the Frame Relay encapsulation that is used rather than HDLC, PPP, and so on.
frame-relay lmi-type {**ansi** | **q933a** | **cisco**}	Interface	Defines the type of LMI messages sent to the switch.
bandwidth *num*	Interface	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.
frame-relay map {*protocol protocol-address dlci*} **payload-compression frf9 stac caim** [*element-number*] [**broadcast**] [**ietf** | **cisco**]	Interface	Statically defines a mapping between a network layer address and a DLCI.
keepalive *sec*	Interface	Defines whether and how often LMI status inquiry messages are sent and expected.
interface serial *number.sub* [**point-to-point** | **multipoint**]	Global	Creates a subinterface or references a previously created subinterface.
frame-relay interface-dlci *dlci* [**ietf** | **cisco**] [**voice-cir** *cir*] [**ppp** *virtual-template-name*]	Subinterface	Links or correlates a DLCI to the subinterface.

Table 10-20 *Frame Relay-Related EXEC Commands*

Command	Function
show interfaces [*type number*]	Shows the physical interface status.
show frame-relay pvc [**interface** *interface*][*dlci*]	Lists information about the PVC status.
show frame-relay lmi [*type number*]	Lists LMI status information.

Q&A

As mentioned in Chapter 1, "All About the Cisco Certified Network Associate Certification," the questions and scenarios in this book are more difficult than what you should experience on the exam. The questions do not attempt to cover more breadth or depth than the exam, but they are designed to make sure that you know the answer. Rather than allowing you to derive the answer from clues hidden in the question, the questions challenge your understanding and recall of the subject. Questions from the "Do I Know This Already?" quiz at the beginning of this chapter are repeated here to ensure that you have mastered this chapter's topics. Hopefully these questions will help limit the number of exam questions on which you narrow your choices to two options and then guess.

The answers to these questions can be found in Appendix A.

1. Name two WAN data-link protocols that 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. Explain how Inverse ARP 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 Define the terms DCE and DTE in the context of the physical layer and a point-to-point serial link.

6 Which layer or layers of OSI are most closely related to the functions of Frame Relay? Why?

7 When Inverse ARP is used by default, what additional configuration is needed to get IGRP routing updates to flow over each VC?

8 Define the attributes of a partial-mesh and full-mesh Frame Relay network.

9 What key pieces of information are required in the **frame-relay map** statement?

10 When creating a partial-mesh Frame Relay network, are you required to use subinterfaces?

11 What benefit related to routing protocols can be gained by using subinterfaces with a partial mesh?

12 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 and IPX addresses you like. Assume that the network is not fully meshed.

13 What **show** command tells you when a PVC became active? How does the router know what time the PVC became active?

14 What **show** command lists Frame Relay information about mapping? In what instances does the information displayed include the Layer 3 addresses of other routers?

15 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.

16 What **debug** option shows Inverse ARP messages?

17 True or false: The Frame Relay **map** configuration command allows more than one Layer 3 protocol address mapping on the same configuration command.

18 What is the name of the field that identifies, or addresses, a Frame Relay virtual circuit?

Scenarios

Scenario 10-1: Frame Relay Verification

Use Examples 10-19 through 10-22 when completing the exercises and answering the questions that follow.

Example 10-19 *Atlanta Command Output, Scenario 10-1*

```
Atlanta#show interface s 0
Serial0 is up, line protocol is up
  Hardware is HD64570
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation FRAME-RELAY, loopback not set, keepalive set (10 sec)
  LMI enq sent  32, LMI stat recvd 32, 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
  Broadcast queue 0/64, broadcasts sent/dropped 75/0, interface broadcasts 59
  Last input 00:00:00, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Queuing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     74 packets input, 5697 bytes, 0 no buffer
     Received 32 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     110 packets output, 9438 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

Atlanta#show interface s 0.1
Serial0.1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 168.10.202.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation FRAME-RELAY

Atlanta#show interface s 0.2
Serial0.2 is up, line protocol is up
  Hardware is HD64570
  Internet address is 168.10.203.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation FRAME-RELAY

Atlanta#show interface s 0.3
Serial0.3 is up, line protocol is up
  Hardware is HD64570
  Internet address is 168.10.204.1/24
```

Example 10-19 *Atlanta Command Output, Scenario 10-1 (Continued)*

```
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation FRAME-RELAY

Atlanta#show frame-relay map
Serial0.3 (up): point-to-point dlci, dlci 54(0x36,0xC60), broadcast, IETF
          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#show frame-relay lmi

LMI Statistics for interface Serial0 (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 43          Num Status msgs Rcvd 43
  Num Update Status Rcvd 0         Num Status Timeouts 0

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

Atlanta#configure terminal
Enter configuration commands, one per line.  End with Ctrl-Z.
Atlanta(config)#interface serial 0
Atlanta(config-if)#shutdown

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0.1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0.2, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0.3, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down
%LINK-5-CHANGED: Interface Serial0, changed state to administratively down
%FR-5-DLCICHANGE: Interface Serial0 - DLCI 54 state changed to DELETED
%FR-5-DLCICHANGE: Interface Serial0 - DLCI 53 state changed to DELETED
%FR-5-DLCICHANGE: Interface Serial0 - DLCI 52 state changed to DELETED

Atlanta(config-if)#no shutdown
Atlanta(config-if)#^Z

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0.1, changed state to up
%FR-5-DLCICHANGE: Interface Serial0 - DLCI 52 state changed to ACTIVE
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0.2, changed state to up
%FR-5-DLCICHANGE: Interface Serial0 - DLCI 53 state changed to ACTIVE
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0.3, changed state to up
%FR-5-DLCICHANGE: Interface Serial0 - DLCI 54 state changed to ACTIVE
%SYS-5-CONFIG_I: Configured from console by console
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up
%LINK-3-UPDOWN: Interface Serial0, changed state to up
```

continues

Example 10-19 *Atlanta Command Output, Scenario 10-1 (Continued)*

```
Atlanta#show frame map
Serial0.3 (up): point-to-point dlci, dlci 54(0x36,0xC60), broadcast, IETF
          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
Atlanta#

Serial0(out): StEnq, myseq 6, yourseen 5, DTE up
datagramstart = 0x45B25C, datagramsize = 13
FR encap = 0xFCF10309
00 75 01 01 01 03 02 06 05

Serial0(in): Status, myseq 6
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 6 , myseq 6
```

Example 10-20 *Charlotte Command Output, Scenario 10-1*

```
Charlotte#show interface s 0.1
Serial0.1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 168.10.202.2/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation FRAME-RELAY

Charlotte#show cdp neighbor detail
-------------------------
Device ID: Atlanta
Entry address(es):
  IP address: 168.10.202.1
  Novell address: 202.0200.aaaa.aaaa
Platform: Cisco 2500,  Capabilities: Router
Interface: Serial0.1,  Port ID (outgoing port): Serial0.1
Holdtime : 164 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-AINR-L), Version 11.2(11), RELEASE SOFTWARE (fc1)
Copyright  1986-1997 by Cisco Systems, Inc.
Compiled Mon 29-Dec-97 18:47 by ckralik

Charlotte#show frame-relay map
Serial0.1 (up): point-to-point dlci, dlci 51(0x33,0xC30), broadcast
          status defined, active
Charlotte#show frame-relay pvc
```

Example 10-20 *Charlotte Command Output, Scenario 10-1 (Continued)*

```
PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 51, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 36            output pkts 28           in bytes 4506
  out bytes 2862           dropped pkts 1           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 26         out bcast bytes 2774
  pvc create time 00:08:54, last time pvc status changed 00:01:26

Charlotte#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = CCITT
  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 54        Num Status msgs Rcvd 37
  Num Update Status Rcvd 0       Num Status Timeouts 17
```

Example 10-21 *Nashville Command Output, Scenario 10-1*

```
Nashville#show cdp neighbor detail
-------------------------
Device ID: Atlanta
Entry address(es):
  IP address: 168.10.203.1
  Novell address: 203.0200.aaaa.aaaa
Platform: Cisco 2500,  Capabilities: Router
Interface: Serial0.1,  Port ID (outgoing port): Serial0.2
Holdtime : 139 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-AINR-L), Version 11.2(11), RELEASE SOFTWARE (fc1)
Copyright  1986-1997 by Cisco Systems, Inc.
Compiled Mon 29-Dec-97 18:47 by ckralik

Nashville#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 51, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 52            output pkts 47           in bytes 6784
  out bytes 6143           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 46         out bcast bytes 6099
  pvc create time 00:13:50, last time pvc status changed 00:06:51
```

continues

Example 10-21 *Nashville Command Output, Scenario 10-1 (Continued)*

```
Nashville#show frame-relay traffic
Frame Relay statistics:
    ARP requests sent 0, ARP replies sent 0
    ARP requests recvd 0, ARP replies recvd 0

Nashville#show frame-relay lmi

LMI Statistics for interface Serial0 (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 84         Num Status msgs Rcvd 84
  Num Update Status Rcvd 0        Num Status Timeouts 0
```

Example 10-22 *Boston Command Output, Scenario 10-1*

```
Boston#show interface s 0.1
Serial0.1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 168.10.204.4/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation FRAME-RELAY

Boston#show cdp neighbor detail
-------------------------
Device ID: Atlanta
Entry address(es):
  IP address: 168.10.204.1
  Novell address: 204.0200.aaaa.aaaa
Platform: Cisco 2500,  Capabilities: Router
Interface: Serial0.1,  Port ID (outgoing port): Serial0.3
Holdtime : 125 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-AINR-L), Version 11.2(11), RELEASE SOFTWARE (fc1)
Copyright  1986-1997 by Cisco Systems, Inc.
Compiled Mon 29-Dec-97 18:47 by ckralik

Boston#show frame-relay map
Serial0.1 (up): point-to-point dlci, dlci 51(0x33,0xC30), broadcast, IETF
          status defined, active

Boston#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

DLCI = 51, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1
```

Example 10-22 *Boston Command Output, Scenario 10-1 (Continued)*

```
  input pkts 65            output pkts 54           in bytes 8475
  out bytes 6906           dropped pkts 1           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 52        out bcast bytes 6792
  pvc create time 00:15:43, last time pvc status changed 00:07:54
Num Update Status Rcvd 0   Num Status Timeouts 0
```

Assuming the details established in Examples 10-19 through 10-22 for Scenario 10-1, do the following:

1. Create a diagram for the network based on the command output shown in Examples 10-19 through 10-22.
2. Complete Table 10-21 with the Layer 3 addresses on the serial links.

Table 10-21 *Layer 3 Addresses for Scenario 10-1*

Router	Port	Subinterface	IP Address	IPX Address
Atlanta	S0			
Atlanta	S0			
Atlanta	S0			
Atlanta	S0			
Charlotte	S0			
Charlotte	S0			
Nashville	S0			
Nashville	S0			
Boston	S0			
Boston	S0			

3. Complete Table 10-22 with the LMI types and encapsulations used.

Table 10-22 *LMIs and Encapsulations Used in Scenario 10-1*

Router	Port	Subinterface	LMI Type	Encapsulation
Atlanta	s0			
Atlanta	s0			
Atlanta	s0			
Atlanta	s0			

continues

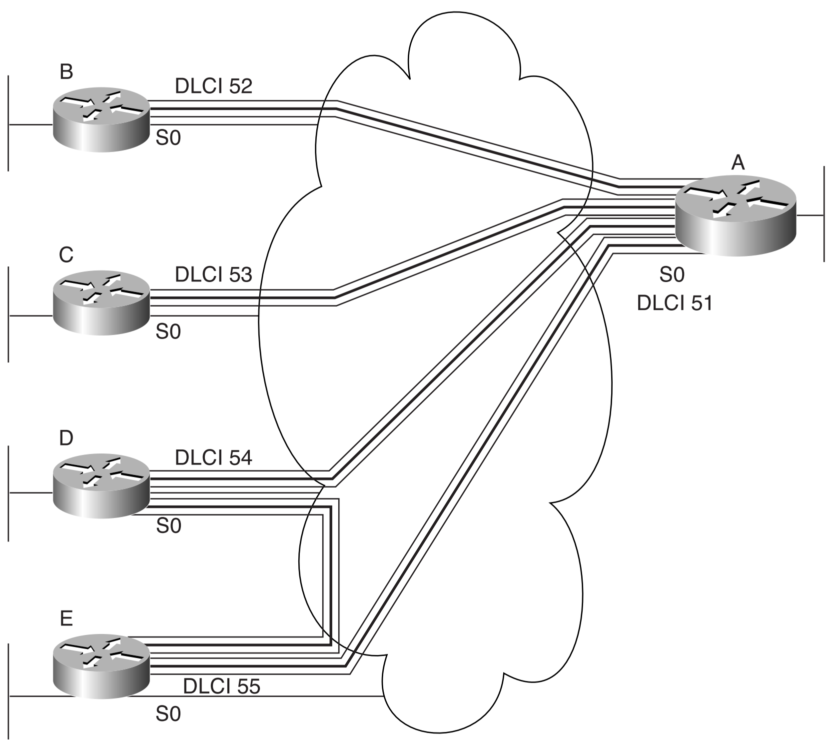
Table 10-22 *LMIs and Encapsulations Used in Scenario 10-1 (Continued)*

Router	Port	Subinterface	LMI Type	Encapsulation
Charlotte	s0			
Charlotte	s0			
Nashville	s0			
Nashville	s0			
Boston	s0			
Boston	s0			

Scenario 10-2: Frame Relay Configuration

Your job is to deploy a new network. Site A is the main site, with PVC connections to the other four sites. Sites D and E also have a PVC between them. Examine Figure 10-22 and perform the following activities.

Figure 10-22 *Scenario 10-2 Frame Relay Network*



1 Plan the IP and IPX addresses to be used. Use Table 10-23 if helpful. Use IP network 168.15.0.0.

2 Using the DLCIs shown in Figure 10-22, create configurations for Routers A, B, and E. Use multipoint subinterfaces for the VCs between Routers A, D, and E.

3 Create alternative configurations for Routers A and E using point-to-point subinterfaces instead of multipoint.

4 Describe the contents of the IP and IPX routing tables on Router A, assuming that the network you just created is working properly. Describe the routing table, assuming that you are using point-to-point subinterfaces only, as in Step 3. Use Table 10-24 if useful.

Table 10-23 *Layer 3 Address Planning Chart*

Interface	Subinterface	IP Address	IPX Address
A's Ethernet			
B's Ethernet			
C's Ethernet			
D's Ethernet			
E's Ethernet			
A's S0			
A's S0			
A's S0			
A's S0			
A's S0			
B's S0			
C's S0			
D's S0			
D's S0			
E's S0			
E's S0			

Table 10-24 *Frame Relay Configuration Commands*

Layer 3 Group	Outgoing Interface	Next-Hop IP Address, or Connected	Next-Hop IPX Address, or Connected

Scenario 10-3: Frame Relay Configuration Dissection

A four-router Frame Relay network has been configured. Consider the configurations in Examples 10-23 through 10-26, and answer the questions that follow.

Example 10-23 *Scenario 10-3, Atlanta Configuration*

```
hostname Atlanta
!
ipx routing 0200.1111.1111
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1
ip address 180.1.1.1 255.255.255.0
ipx network AAA1801
frame-relay interface-dlci 501
!
interface serial 0.2
ip address 180.1.2.1 255.255.255.0
ipx network AAA1802
frame-relay interface-dlci 502
!
interface serial 0.3
ip address 180.1.3.1 255.255.255.0
ipx network AAA1803
frame-relay interface-dlci 503
!
interface ethernet 0
ip address 180.1.10.1 255.255.255.0
ipx network AAA18010
!
router igrp 1
network 180.1.0.0
```

Example 10-24 *Scenario 10-3, Charlotte Configuration*

```
hostname Charlotte
!
ipx routing 0200.2222.2222
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1
ip address 180.1.1.2 255.255.255.0
ipx network AAA1801
frame-relay interface-dlci 500
!
interface ethernet 0
ip address 180.1.11.2 255.255.255.0
ipx network AAA18011
!
router igrp 1
network 180.1.0.0
```

Example 10-25 *Scenario 10-3, Nashville Configuration*

```
hostname Nashville
!
ipx routing 0200.3333.3333
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1
ip address 180.1.2.3 255.255.255.0
ipx network AAA1802
frame-relay interface-dlci 500
!
interface ethernet 0
ip address 180.1.12.3 255.255.255.0
ipx network AAA18012
!
router igrp 1
network 180.1.0.0
```

Example 10-26 *Scenario 10-3, Boston Configuration*

```
hostname Boston
!
ipx routing 0200.4444.4444
!
interface serial0
encapsulation frame-relay
!
```

continues

Example 10-26 *Scenario 10-3, Boston Configuration (Continued)*

```
interface serial 0.1
ip address 180.1.3.4 255.255.255.0
ipx network AAA1803
frame-relay interface-dlci 500
!
interface ethernet 0
ip address 180.1.13.4 255.255.255.0
ipx network AAA18013
!
router igrp 1
network 180.1.0.0
```

Assuming the details established in Examples 10-23 through 10-26 for Scenario 10-3, complete or answer the following:

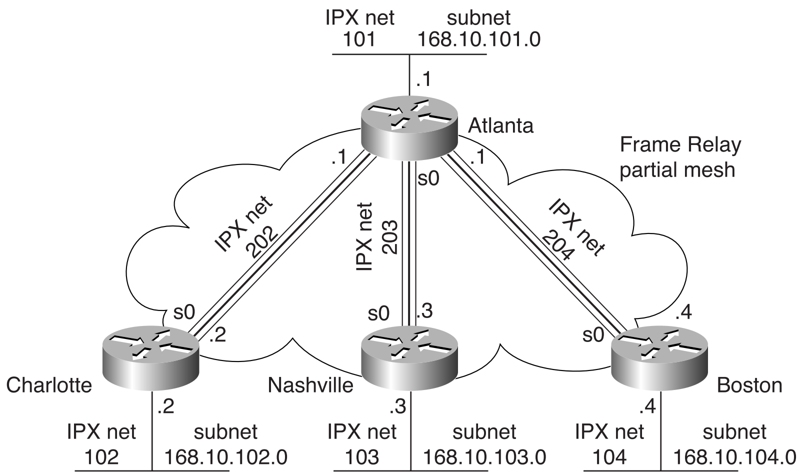
1. Draw a diagram of the network.
2. Is IGRP split horizon on or off? How can you tell?
3. What type of Frame Relay encapsulation is used?
4. Create the commands on Router 1 and Router 2 to disable Inverse ARP and instead use static mapping.

Answers to Scenarios

Answers to Scenario 10-1: Frame Relay Verification

Figure 10-23 is a diagram that matches the configuration.

Figure 10-23 *Scenario 10-1 Network Derived from* **show** *and* **debug** *Commands*



Discovering the IP addresses and subinterfaces is relatively straightforward. The **show** commands for most subinterfaces are provided. They list the IP address and mask used. The **show cdp neighbor detail** commands also mention the IP address of the connected routers.

The full IPX addresses are more challenging to deduce. The only command that lists the IPX addresses is the **show cdp neighbor detail** command, which is used in Examples 10-20, 10-21, and 10-22. The **show frame-relay map** command should seemingly provide that information, but because all the subinterfaces are point-to-point, no direct mapping is needed. The subinterface acts like a point-to-point link, so the neighboring router's IPX address is not shown in the **show frame-relay map** command output. A **debug frame-relay events** command, which shows output for Inverse ARP flows, could have identified the IPX addresses, but Inverse ARP is not enabled on point-to-point subinterfaces because it is not needed.

In short, there is no way to deduce all IPX addresses from the scenario.

Table 10-25 completes Table 10-21 by listing the Layer 3 addresses and subinterface numbers.

Table 10-25 *Completed Layer 3 Addresses for Scenario 10-1*

Router	Port	Subinterface	IP Address	IPX Address
Atlanta	S0	1	168.10.202.1	202.0200.AAAA.AAAA
Atlanta	S0	2	168.10.203.1	203.0200.AAAA.AAAA
Atlanta	S0	3	168.10.204.1	204.0200.AAAA.AAAA
Atlanta	s0	N/A	N/A	N/A
Charlotte	s0	1	168.10.202.2	202.????.????.????
Charlotte	s0	N/A	N/A	N/A
Nashville	s0	1	168.10.203.3	203.????.????.????
Nashville	s0	N/A	N/A	N/A
Boston	s0	1	168.10.204.4	204.????.????.????
Boston	s0	N/A	N/A	N/A

NOTE There is not enough information to derive the IPX addresses for Charlotte, Nashville, and Boston. The IPX network numbers are implied by the **show cdp neighbor detail** command output.

The LMI type is discovered only by examining the output of the **show frame-relay lmi** command. However, this command does not list whether the LMI type was learned via autosensing or whether it was configured.

The encapsulation type is more obscure. The **show frame-relay map** command output holds the answer. Table 10-26 completes Table 10-22 by summarizing the answers.

Table 10-26 *Completed LMIs and Encapsulations for Scenario 10-1*

Router	Port	Subinterface	LMI Type	Encapsulation
Atlanta	S0	N/A	Cisco	N/A
Atlanta	S0	1	N/A	cisco
Atlanta	S0	2	N/A	cisco
Atlanta	S0	3	N/A	ietf
Charlotte	S0	N/A	Q933A	N/A
Charlotte	s0	1	N/A	cisco
Nashville	s0	N/A	Cisco	N/A

Table 10-26 *Completed LMIs and Encapsulations for Scenario 10-1 (Continued)*

Router	Port	Subinterface	LMI Type	Encapsulation
Nashville	s0	1	N/A	cisco
Boston	s0	N/A	Cisco	N/A
Boston	s0	1	N/A	ietf

Answers to Scenario 10-2: Frame Relay Configuration

Check your IP and IPX address design against the ones chosen in Table 10-27. Of course, your choices most likely are different. However, you should have one subnet per VC when using only point-to-point subinterfaces. With the original criteria of Routers A, D, and E each using multipoint subinterfaces, these three subinterfaces should have been in the same IP subnet and IPX network. Table 10-27 lists the planned Layer 3 addresses for the configurations using multipoint among these three routers.

Table 10-27 *Layer 3 Planning Chart, Multipoint A-D-E*

Interface	Subinterface	IP Address	IPX Address
A's Ethernet	N/A	168.15.101.1	101.0200.AAAA.AAAA
B's Ethernet	N/A	168.15.102.1	102.0200.BBBB.BBBB
C's Ethernet	N/A	168.15.103.1	103.0200.CCCC.CCCC
D's Ethernet	N/A	168.15.104.1	104.0200.DDDD.DDDD
E's Ethernet	N/A	168.15.105.1	105.0200.EEEE.EEEE
A's S0	2	168.15.202.1	202.0200.AAAA.AAAA
A's S0	3	168.15.203.1	203.0200.AAAA.AAAA
A's S0	1	168.15.200.1	200.0200.AAAA.AAAA
B's S0	2	168.15.202.2	202.0200.BBBB.BBBB
C's S0	3	168.15.203.3	203.0200.CCCC.CCCC
D's S0	1	168.15.200.4	200.0200.DDDD.DDDD
E's S0	1	168.15.200.5	200.0200.EEEE.EEEE

Assuming the DLCIs shown in Figure 10-22, Examples 10-27, 10-28, and 10-29 show the configurations for Routers A, B, and E, respectively, using multipoint subinterfaces for the VCs between A, D, and E.

Example 10-27 *Router A Configuration, Scenario 10-2*

```
ipx routing 0200.aaaa.aaaa
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 168.15.200.1  255.255.255.0
ipx network 200
frame-relay interface-dlci 54
frame-relay interface-dlci 55
!
interface serial 0.2 point-to-point
ip address 168.15.202.1  255.255.255.0
ipx network 202
interface-dlci 52
!
interface serial 0.3 point-to-point
ip address 168.15.203.1  255.255.255.0
ipx network 203
interface-dlci 53
!

interface ethernet 0
ip address 168.15.101.1 255.255.255.0
ipx network 101
!
router igrp 1
network 168.15.0.0
```

Example 10-28 *Router B Configuration, Scenario 10-2*

```
ipx routing 0200.bbbb.bbbb
!
interface serial0
encapsulation frame-relay
!
interface serial 0.2 point-to-point
ip address 168.15.202.2  255.255.255.0
ipx network 202
frame-relay interface-dlci 51
!
interface ethernet 0
ip address 168.15.102.1 255.255.255.0
ipx network 102
!
router igrp 1
network 168.15.0.0
```

Example 10-29 *Router E Configuration, Scenario 10-2*

```
ipx routing 0200.eeee.eeee
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 multipoint
ip address 168.15.200.5  255.255.255.0
ipx network 200
frame-relay interface-dlci 51
frame-relay interface-dlci 54
!
interface ethernet 0
ip address 168.15.105.1 255.255.255.0
ipx network 105
!
router igrp 1
network 168.15.0.0
```

Multipoint subinterfaces work perfectly well in this topology. Using multipoint also conserves IP subnets, as you will see in the next task of this scenario. When you change strategy to use only point-to-point subinterfaces, each of the three VCs in the triangle of Routers A, D, and E requires a different subnet and IPX network number. Table 10-28 shows the choices made here. Examples 10-30 and 10-31 show alternative configurations for Routers A and E using point-to-point instead of multipoint subinterfaces.

Table 10-28 *Scenario 10-2 Layer 3 Address Planning Chart, All Point-to-Point Subinterfaces*

Interface	Subinterface	IP Address	IPX Address
A's Ethernet	N/A	168.15.101.1	101.0200.AAAA.AAAA
B's Ethernet	N/A	168.15.102.1	102.0200.BBBB.BBBB
C's Ethernet	N/A	168.15.103.1	103.0200.CCCC.CCCC
D's Ethernet	N/A	168.15.104.1	104.0200.DDDD.DDDD
E's Ethernet	N/A	168.15.105.1	105.0200.EEEE.EEEE
A's S0	2	168.15.202.1	202.0200.AAAA.AAAA
A's S0	3	168.15.203.1	203.0200.AAAA.AAAA
A's S0	4	168.15.204.1	204.0200.AAAA.AAAA
A's S0	5	168.15.205.1	205.0200.AAAA.AAAA

continues

Table 10-28 *Scenario 10-2 Layer 3 Address Planning Chart, All Point-to-Point Subinterfaces (Continued)*

Interface	Subinterface	IP Address	IPX Address
B's S0	2	168.15.202.2	202.0200.BBBB.BBBB
C's S0	3	168.15.203.3	203.0200.CCCC.CCCC
D's S0	4	168.15.204.4	204.0200.DDDD.DDDD
D's S0	1	168.15.190.4	190.0200.DDDD.DDDD
E's S0	5	168.15.205.5	205.0200.EEEE.EEEE
E's S0	1	168.15.190.5	190.0200.EEEE.EEEE

Example 10-30 *Router A Configuration, Scenario 10-2, All Point-to-Point Subinterfaces*

```
ipx routing 0200.aaaa.aaaa
!
interface serial0
encapsulation frame-relay
!
interface serial 0.2 point-to-point
ip address 168.15.202.1  255.255.255.0
ipx network 202
frame-relay interface-dlci 52
!
interface serial 0.3 point-to-point
ip address 168.15.203.1  255.255.255.0
ipx network 203
frame-relay interface-dlci 53
!
interface serial 0.4 point-to-point
ip address 168.15.204.1  255.255.255.0
ipx network 204
frame-relay interface-dlci 54
!
interface serial 0.5 point-to-point
ip address 168.15.205.1  255.255.255.0
ipx network 205
frame-relay interface-dlci 55
!
interface ethernet 0
ip address 168.15.101.1 255.255.255.0
ipx network 101
!
router igrp 1
network 168.15.0.0
```

Example 10-31 *Router E Configuration, Scenario 10-2, Subinterfaces*

```
ipx routing 0200.eeee.eeee
!
interface serial0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 168.15.190.5  255.255.255.0
ipx network 190
interface-dlci 54
!
interface serial 0.5 point-to-point
ip address 168.15.200.5  255.255.255.0
ipx network 200
interface-dlci 51
!
interface ethernet 0
ip address 168.15.105.1 255.255.255.0
ipx network 105
!
router igrp 1
network 168.15.0.0
```

The contents of the IP and IPX routing tables asked for in task 4 of this scenario are listed in shorthand in Table 10-29. The third byte of the IP address is shown in the Layer 3 Group column because the third byte (octet) fully comprises the subnet field. Not coincidentally, the IPX network number was chosen as the same number, mainly to make network operation easier.

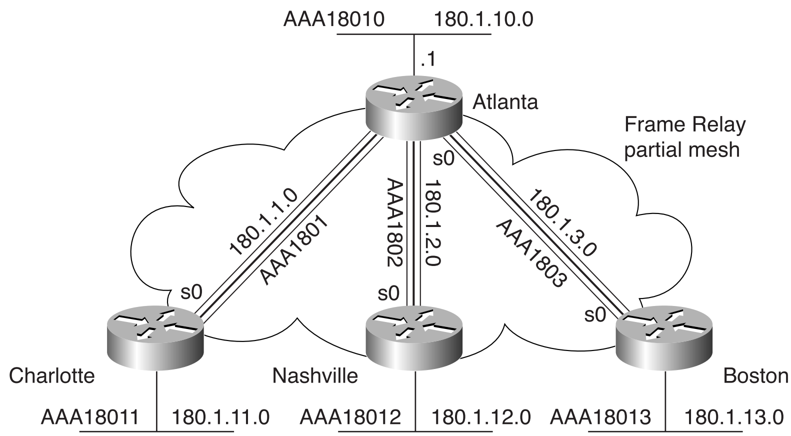
Table 10-29 *Scenario 10-2 IP and IPX Routing Table Contents, Router A*

Layer 3 Group	Outgoing Interface	Next-Hop IP Address, or Connected	Next-Hop IPX Address, or Connected
101	E0	Connected	Connected
102	S0.2	168.15.202.2	202.0200.bbbb.bbbb
103	S0.3	168.15.203.3	203.0200.cccc.cccc
104	S0.4	168.15.204.4	204.0200.dddd.dddd
105	S0.5	168.15.205.5	205.0200.eeee.eeee

Answers to Scenario 10-3: Frame Relay Configuration Dissection

Figure 10-24 supplies the network diagram described in Scenario 10-3. The subinterfaces are all point-to-point, which is a clue that each VC has a subnet and IPX network associated with it. An examination of the IP addresses or IPX network numbers should have been enough for you to deduce which routers are attached to each end of each VC.

Figure 10-24 *Diagram of Scenario 10-3 Frame Relay Network*



Split horizon is turned off on all interfaces because that is the default with point-to-point subinterfaces and because no command has been configured to turn it on.

Cisco encapsulation is used in each case. The **encapsulation frame-relay** command defaults to the use of Cisco encapsulation.

Disabling Inverse ARP is unlikely in real networks. However, this exercise was included so that you are ready for the exam. Examples 10-32 and 10-33 show the commands used to migrate to not using Inverse ARP. The maps are necessary for both IP and IPX because both need to be routed across the Frame Relay network.

Example 10-32 *Scenario 10-3, Atlanta Router: Changes for Static Mapping*

```
Atlanta(config)# interface serial 0.1
Atlanta(config-subif)#no frame-relay inverse-arp
Atlanta(config-subif)#frame-relay map ip 180.1.1.2 501 broadcast
Atlanta(config-subif)#frame-relay map ipx aaa1801.0200.2222.2222 501 broadcast
Atlanta(config-subif)# interface serial 0.2
Atlanta(config-subif)#no frame-relay inverse-arp
Atlanta(config-subif)#frame-relay map ip 180.1.2.3 502 broadcast
Atlanta(config-subif)#frame-relay map ipx aaa1802.0200.3333.3333 502 broadcast
Atlanta(config-subif)# interface serial 0.3
Atlanta(config-subif)#no frame-relay inverse-arp
Atlanta(config-subif)#frame-relay map ip 180.1.3.4 503 broadcast
Atlanta(config-subif)#frame-relay map ipx aaa1803.0200.4444.4444 503 broadcast
```

Example 10-33 *Scenario 10-3, Charlotte Router: Changes for Static Mapping*

```
Charlotte(config)# interface serial 0.1
Charlotte(config-subif)#no frame-relay inverse-arp
Charlotte(config-subif)#frame-relay map ip 180.1.1.1 500 broadcast
Charlotte(config-subif)#frame-relay map ipx aaa1801.0200.1111.1111 500 broadcast
```