CHAPTER 15

Frame Relay

Frame Relay is a Layer 2 (data link) wide-area networking (WAN) protocol that operates at both Layer 1 (physical) and Layer 2 (data link) of the OSI networking model. Although Frame Relay internetworking services were initially designed to operate over Integrated Services Digital Network (ISDN), the more common deployment today involves dedicated access to WAN resources.

NOTE ISDN and Frame Relay both use the signaling mechanisms specified in ITU-T Q.933 (Frame Relay Local Management Interface [LMI] Type Annex-A) and American National Standards Institute (ANSI) T1.617 (Frame Relay LMI Type Annex-D).

Frame Relay is considered to be a more efficient version of X.25 because it does not require the windowing and retransmission features found with X.25. This is primarily due to the fact that Frame Relay services typically are carried by more reliable access and backbone facilities.

Frame Relay networks are typically deployed as a cost-effective replacement for point-to-point private line, or leased line, services. Whereas point-to-point customers incur a monthly fee for local access and long-haul connections, Frame Relay customers incur the same monthly fee for local access, but only a fraction of the long-haul connection fee associated with point-to-point private line services. The long-haul charges are typically usage-based across the virtual circuit (VC).

NOTE The long-haul fee associated with point-to-point private (leased) line services is sometimes known as the inter-office connection fee. Service providers generally file a tariff with the FCC regarding these fees, comprising a base cost plus a per-mile charge.

NOTE X.25 was designed for use over less reliable transmission medium than what is available in the marketplace today. Due to this unreliable nature, X.25 took on the error detection and correction (windowing and retransmission) mechanisms within the protocol stack. This resulted in higher overhead on the network, yielding less available bandwidth for data throughput.

NOTE Frame Relay is a packet-switched technology, enabling end nodes to dynamically share network resources.

Frame Relay was standardized by two standards bodies—internationally by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) and domestically by ANSI.

Frame Relay Terms and Concepts

Frame Relay is a frame-switched technology, meaning that each network end user, or end node, will share backbone network resources, such as bandwidth. Connectivity between these end nodes is accomplished with the use of Frame Relay virtual circuits (VCs). Figure 15-1 illustrates the components of a Frame Relay WAN.

Figure 15-1 *Frame Relay WAN*

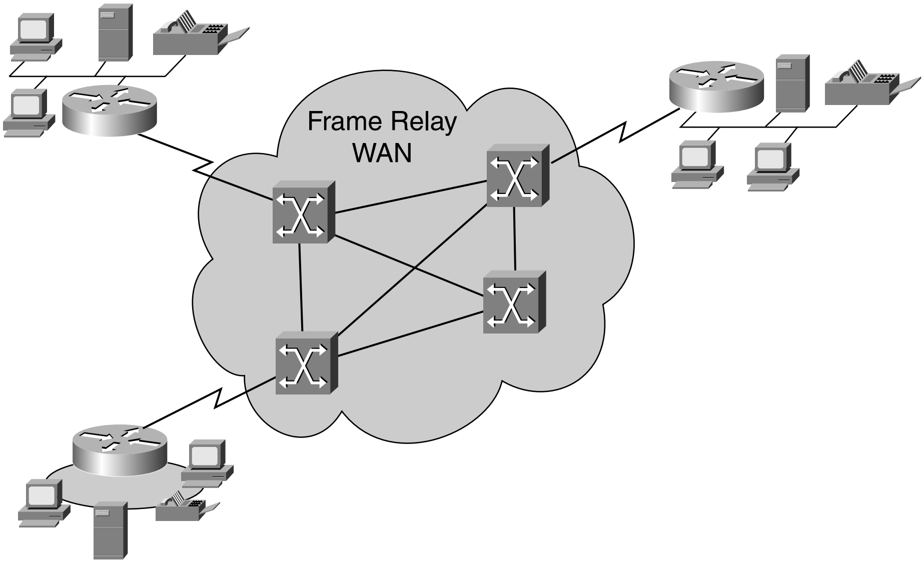


Table 15-1 defines the common and relevant Frame Relay terms.

Table 15-1 *Frame Relay Terms and Definitions*

Acronym	Definition
B_c	Committed burst. Negotiated tariff metric in Frame Relay internetworks. The maximum amount of data (measured in bits) that a Frame Relay internetwork is committed to accept and transmit at the committed information rate (CIR). B_c can be represented by the formula $B_c = \text{CIR} \times T_c$.
B_e	Excess burst. Negotiated tariff metric in Frame Relay internetworks. The number of bits that a Frame Relay internetwork will attempt to transfer after B_c is accommodated. B_e data is generally delivered with a lower probability than B_C data because B_e data is marked as discard eligible (DE) by the network.
BECN	Backward explicit congestion notification. A Frame Relay network in frames traveling in the opposite direction of frames that are encountering a congested path sets this bit. Data terminal equipment (DTE) receiving frames with the BECN bit set can request that higher-level protocols take flow control action as appropriate, such as the throttling back of data transmission.
CIR	Committed information rate. Rate at which a Frame Relay network agrees to transfer information under normal conditions, averaged over a minimum increment of time. CIR, measured in bits per second (bps), is one of the key negotiated tariff metrics.
DCE	Data communications equipment. The DCE provides a physical connection to the network, forwards traffic, and provides a clocking signal used to synchronize data transmission between DCE and DTE.

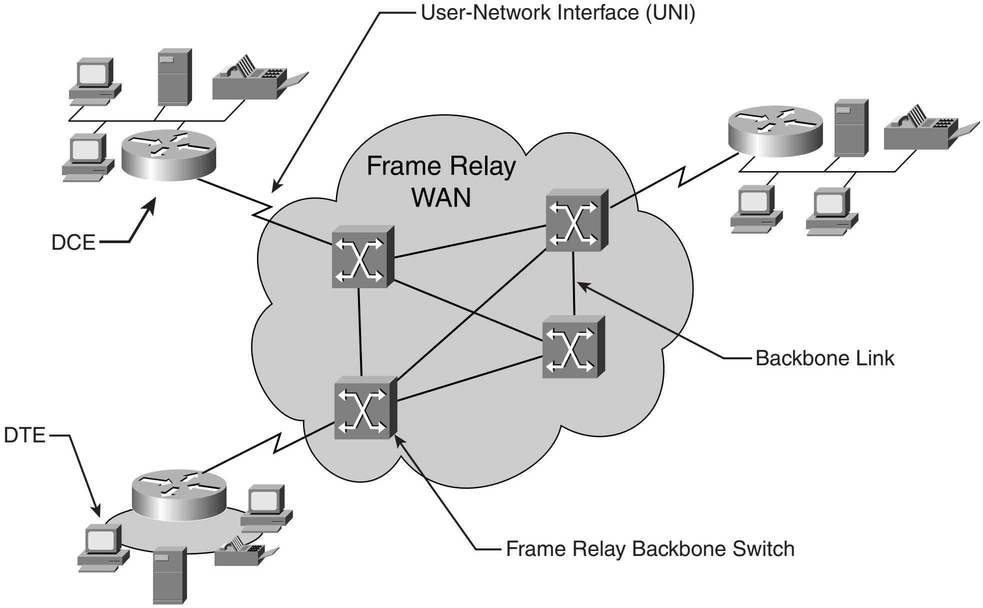
(continues)

Table 15-1 *Frame Relay Terms and Definitions (Continued)*

Acronym	Definition
DE	Discard eligible. If the Frame Relay network is congested, DE-marked frames can be dropped to ensure delivery of higher-priority traffic—in this case, CIR-marked frames.
DLCI	Data-link connection identifier. Values used to identify a specific PVC or SVC. In Frame Relay internetworks, DLCIs are locally significant. In a Frame Relay LMI extended environment, DLCIs are globally significant because they indicate end devices.
DTE	Data terminal equipment. Device at the end of a User-Network Interface (UNI) that serves as either a data source or destination.
FECN	Forward explicit congestion notification. Bit set by a Frame Relay network to inform the DTE receiving the frame that congestion was experienced in the path from origination to destination. The DTE that is receiving frames with the FECN bit set can request that higher-level protocols take flow control action as appropriate, such as throttling back data transmission.
LMI	Local Management Interface. Set of enhancements to the basic Frame Relay specification. LMI includes support for keepalive mechanisms, verifying the flow of data; multicast mechanisms, providing the network server with local and multicast DLCI information; global addressing, giving DLCIs global rather than local significance; and status mechanisms, providing ongoing status reports on the switch-known DLCIs.
NNI	Network-to-Network Interface. Standard interface between two Frame Relay switches that are both located in either a private or public network.
PVC	Permanent virtual circuit. Frame Relay virtual circuit that is permanently established (does not require call-setup algorithms).
SVC	Switched virtual circuit. Frame Relay virtual circuit that is dynamically established via call-setup algorithms. Usually found in sporadic data transfer environments.
T_c	T_c is a periodic interval. This interval is triggered anew when data is incoming to the network. When there is no data traffic when time T_c has elapsed, a new interval does not begin until new data traffic is sent to the network.
UNI	User-Network Interface. Frame Relay interface between a Frame Relay switch in a private network (such as a customer premise) and a public network (such as a service provider). Sometimes referred to as a Subscriber Network Interface (SNI).

Figure 15-2 illustrates some of the Frame Relay terminology used. The remainder of the terms will be illustrated where appropriate throughout this chapter.

Figure 15-2 *Frame Relay Terminology, Part I*



Frame Relay Components

Frame Relay WAN service comprises four primary functional components:

- Customer premise Frame Relay access device (FRAD)
- Local access loop to the service provider network
- Frame Relay switch access port

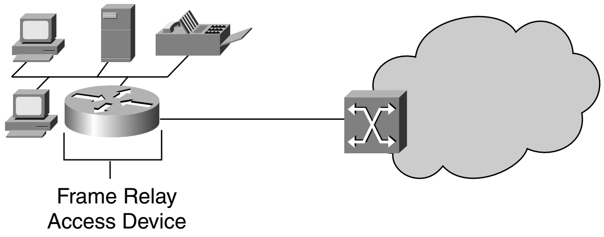
  Link Management Interface parameters are defined here
- Frame Relay VC parameters to each end site

Customer Premise FRAD

This device is either a dedicated FRAD, such as a Cisco *XXX*; or a router, such as a Cisco 26xx Series Router. Figure 15-3 illustrates a typical FRAD implementation.

NOTE A router with an integrated CSU/DSU acts as a FRAD/Router.

Figure 15-3 *Frame Relay Access Device*



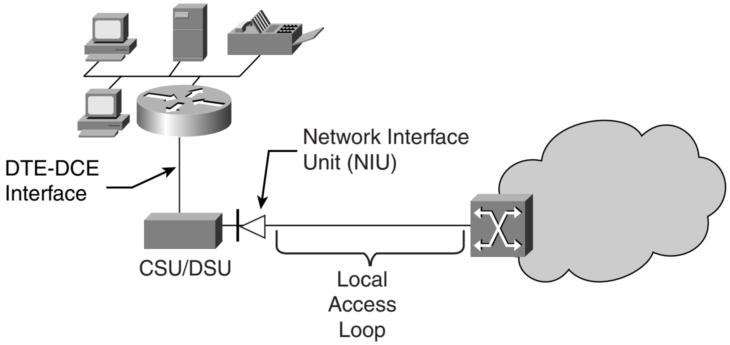
Local Access Loop to the Service Provider Network

Local access loop is the physical wiring that interconnects the customer premise FRAD and the service provider network's Frame Relay switch access port. This local loop is typically a DS0, DS1, NxDS1, DS3 service, or some fraction of DS1/DS3 service (such as Frac-T1).

In telephony, a local loop is the wired connection from a telephone company's central office (CO) in a locality to its customers' telephones at homes and businesses. This connection is usually on a pair of copper wires called twisted pair. The local loop system was originally designed for voice transmission only using analog transmission technology on a single voice channel. A modem is used to handle the conversion between analog and digital signals. With the advent of ISDN or digital subscriber line (DSL), the local loop can carry digital signals directly and at a much higher bandwidth than for voice only.

The local loop requires termination into a network interface unit (NIU) at the customer premise, and subsequent connection to the customer DCE device, usually a CSU/DSU. The DTE port of this CSU/DSU provides connectivity to the FRAD/Router. Figure 15-4 illustrates a typical local loop configuration.

Figure 15-4 *Frame Relay Local Access Loop*



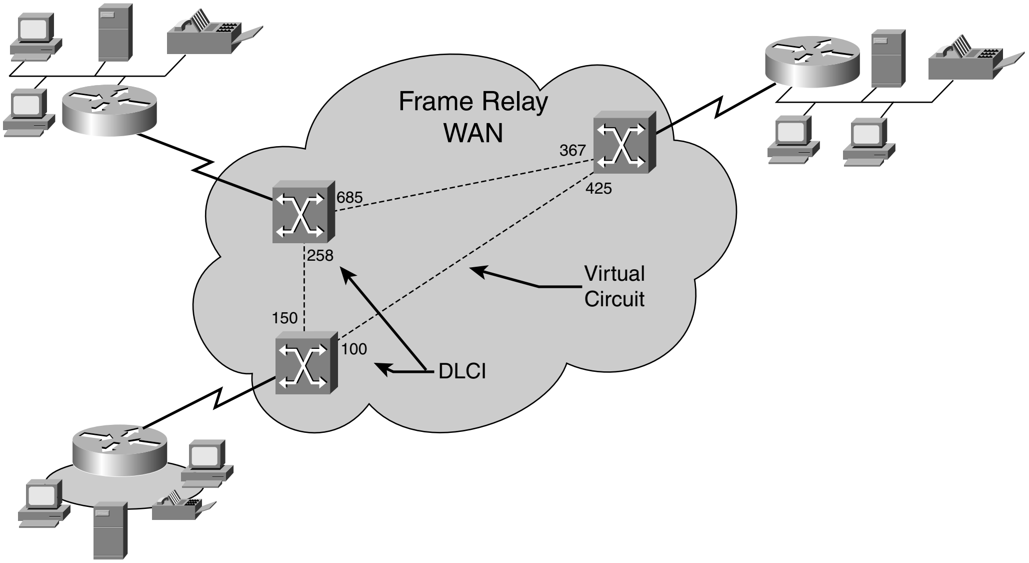
Frame Relay Virtual Circuits

Frame Relay is a connection-oriented service, operating at the data link layer (Layer 2) of the OSI model. A DLCI is used to identify this dedicated communication path between two end nodes: origination and termination. This path, or VC, is a bidirectional logical connection across the wide-area network between two end node DTE devices.

Figure 15-5 illustrates a fully meshed (all sites with connectivity to each other) Frame Relay WAN, with DLCI assignments for each location.

NOTE Sometimes the originating node of a VC will be annotated as Site A and the terminating node of a VC will be annotated as Site B or Site Z.

Figure 15-5 *Frame Relay WAN with Virtual Circuit and DLCI*



DLCIs

DLCIs are used to identify the PVC that is provisioned to transport data traffic. DLCIs are of local significance, unless an agreement has been made with the network service provider to deploy global DLCIs. Local significance means that DLCIs are of use only to the local Frame Relay network device. Frame Relay DLCIs are analogous to an organization's telephone network that is utilizing speed-dial functions. The most common Frame Relay

WAN deployment involves the use of local DLCIs because a network size limitation exists for the use of global DLCIs.

NOTE Global DLCI addresses are assigned so that each DLCI has universal significance, meaning that the DLCI number is pointed to the same destination (termination point) regardless of the origination point.

The concept behind global DLCI addressing is to simplify Frame Relay network addressing administration; however, global addressing has an inherent limitation in that no more than 992 DLCIs (1024 DLCIs less the 32 reserved DLCIs) can be used. In a Frame Relay network of more than 992 sites, global addressing will not work.

The use of global DLCIs requires that they each be preassigned. (Typically, the assignments are negotiated between the customer and the network service provider.) In addition, each DLCI can be used only once throughout the network. (If two sites had the same DLCI, the network would not know which termination site was the intended destination.) The Frame Relay switch within the network service provider's network will have tables that route the traffic between each origination and termination pair.

Suppose that an organization has deployed the speed-dialing scheme illustrated for reference in Figure 15-6 and detailed in the following list:

- The CEO speed-dials 1 to talk with the COO.
- The CEO speed-dials 2 to talk with the VP of Marketing.
- The COO speed-dials 1 to talk with the VP of Marketing.
- The COO speed-dials 5 to talk with the CEO.
- The VP of Marketing speed-dials 7 to talk with the CEO.
- The VP of Marketing speed-dials 9 to talk with the COO.

Figure 15-6 *Telephone Speed-Dial Network*

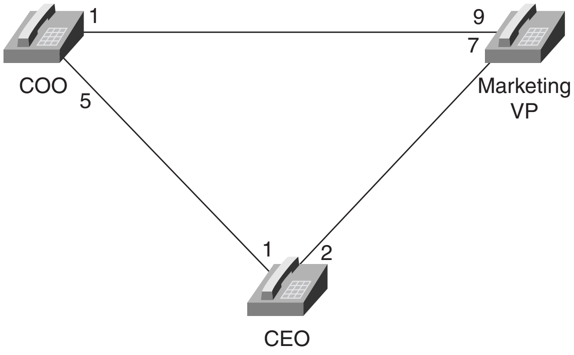


Table 15-2 provides another view of the Telephone Speed-Dial Network.

Table 15-2 *Telephone Speed-Dial Configuration Table*

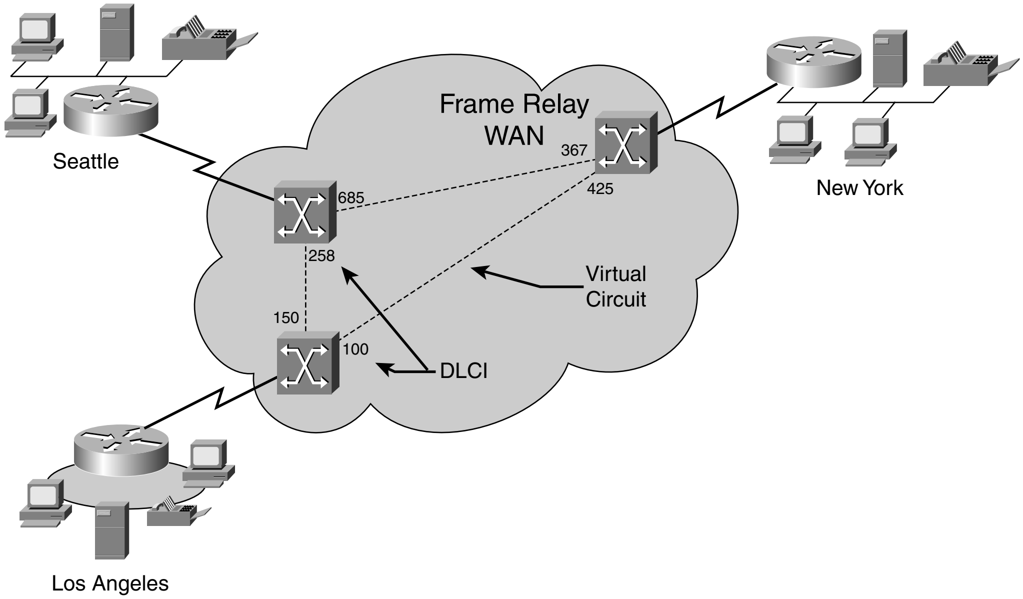
Site A	Site B	A ➠ B Speed Dial	B ➠ A Speed Dial
COO	Marketing VP	1	9
COO	CEO	5	1
CEO	Marketing VP	2	7
CEO	COO	1	5
Marketing VP	CEO	7	5
Marketing VP	COO	9	1

For the CEO to speak with the COO, the CEO will press speed-dial 1 on the telephone set. However, the COO will see speed-dial 5 on the telephone because that is the local speed-dial assignment given to the CEO, as the speed-dial 1 local assignment on the COO's telephone set is assigned to the VP of Marketing.

If the COO presses speed-dial 1, the VP of Marketing will answer the phone and speed-dial 9 will show on the Marketing VP's phone because that is the local assignment given to the COO.

This same concept applies to Frame Relay DLCIs; the DLCI assignment is locally significant. The distant-end of the VC is unaware of this number because it has its own local DLCI assignment to identify the distant-end node. This is illustrated in Figure 15-7.

Figure 15-7 *Frame Relay Network with DLCI Assignment*



In this example, for Los Angeles to send traffic to New York, the FRAD will map the network layer information, such as IP address, to the DLCI. In this case, the DLCI is 100. New York will see traffic arrive on DLCI 425 and will be able to identify within its Frame Relay mapping tables that this traffic has arrived from Los Angeles.

Table 15-3 provides a view of the Frame Relay network shown in Figure 15-7.

Table 15-3 *Frame Relay DLCI Table for Figure 15-7*

Site A	Site B	A ➠ B DLCI	B ➠ A DLCI
Los Angeles	New York	100	425
Los Angeles	Seattle	150	258
Seattle	New York	685	367
Seattle	Los Angeles	258	150
New York	Seattle	367	685
New York	Los Angeles	425	100

NOTE The **show frame-relay map** command can be used in Privileged Exec mode to view the mapping of network addressing to a DLCI.

NOTE To define the mapping between a destination protocol address and the DLCI used to connect to the destination address, use the **frame-relay map** interface configuration command. Use the **no** form of this command to delete the map entry.

```
frame-relay map protocol protocol-address dlci [broadcast] [ietf ¦ cisco]
[payload-compress {packet-by-packet ¦ frf9 stac [hardware-options]}]
no frame-relay map protocol protocol-address
```

Table 15-4 provides details of the **frame-relay map** interface configuration command.

Table 15-4 ***frame-relay map*** *Command Field Descriptions*

Field	Description
Protocol	Supported protocol, bridging, or logical link control keywords: **appletalk**, **decnet**, **dlsw, ip**, **ipx**, **llc2**, **rsrb**, **vines** and **xns**.
protocol-address	Destination protocol address.
Dlci	DLCI number used to connect to the specified protocol address on the interface.
Broadcast	(Optional) Forwards broadcasts to this address when multicast is not enabled (see the **frame-relay multicast-dlci** command for more information about multicasts). This keyword also simplifies the configuration of Open Shortest Path First (OSPF).
Ietf	(Optional) Internet Engineering Task Force (IETF) form of Frame Relay encapsulation. Used when the router or access server is connected to another vendor's equipment across a Frame Relay network.
Cisco	(Optional) Cisco encapsulation method.
payload-compress packet-by-packet	(Optional) Packet-by-packet payload compression using the Stacker method.
payload-compress frf9 stac	(Optional) Enables FRF.9 compression using the Stacker method. If the router contains a compression service adapter (CSA), compression is performed in the CSA hardware (hardware compression). If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression). If the VIP2 is not available, compression is performed in the router's main processor (software compression).

(continues)

Table 15-4 ***frame-relay map*** *Command Field Descriptions (Continued)*

Field	Description
hardware-options	**distributed** (Optional)—Specifies that compression is implemented in the software that is installed in a VIP2. If the VIP2 is not available, compression is performed in the router's main processor (software compression). This option applies only to the Cisco 7500 series. **software** (Optional)—Specifies that compression is implemented in the Cisco IOS software installed in the router's main processor. **csa** csa_number (Optional)—Specifies the CSA to use for a particular interface. This option applies only to Cisco 7200 series routers.

PVCs

PVCs are Frame Relay VC connections that are permanently established. PVCs are used for frequent communication between end nodes, such as file sharing, file transfer, and CAD/CAM imaging.

Frame Relay PVCs use DLCIs for Layer 2 addressing.

PVCs operate in one of two modes:

- Idle—The connection between end nodes is active albeit with no data transfer occurring. PVCs are not terminated or "taken-down" when in an idle state.
- Data Transfer—Data traffic is being transmitted between end nodes over the VC.

Even though PVCs are generally discussed as being full-duplex, PVCs are simplex connections, each with its own DLCI/CIR assignment.

The three duplex modes are as follows:

- Full-duplex communication involves origination and termination points transmitting and receiving at the same time; this is two-way communication full-time.
- Half-duplex communication is origination and termination points transmitting and receiving, but not at the same time. Only one flow of traffic is allowed across the connection; this is two-way communication, one-way at a time.
- Simplex communication is origination or termination points transmitting or receiving; this is one-way communication only.

SVCs

Unlike PVCs, which are permanently established connections, SVCs require a call setup process. SVCs are temporary connections that are traditionally used when communication between end nodes is infrequent or sporadic, such as in Voice over Frame Relay (VoFr) situations.

NOTE Frame Relay SVCs use E.164 or X.121 addresses for Layer 2 addressing.

Whereas PVCs are permanently established, SVCs require a call setup and termination process, defined by the following process and functions:

1. Call setup—Establishes the VC between Frame Relay end nodes. This includes negotiation of VC parameters, such as CIR.
2. Data transfer—Data traffic is transmitted between end nodes (originating and terminating) across the VC.
3. Idle—Like PVCs, when the VC is idle (no data traffic) the connection between end nodes remains active and available for communication. However, unlike PVCs, which do not terminate the connection, an SVC will terminate the connection if it is in an idle state for a configured time period.
4. Call termination—The VC between Frame Relay end nodes is terminated, or "taken down."

NOTE In bidirectional mode, both ends of a VC send and respond to keepalive requests. If one end of the VC is configured in the bidirectional mode, the other end must also be configured in the bidirectional mode.

In request mode, the router sends keepalive requests and expects replies from the other end of the VC. If one end of a VC is configured in the request mode, the other end must be configured in the reply or passive-reply mode.

In reply mode, the router does not send keepalive requests, but waits for keepalive requests from the other end of the VC and replies to them. If no keepalive request has arrived within the timer interval, the router times out and increments the error counter by 1. If one end of a VC is configured in the reply mode, the other end must be configured in the request mode.

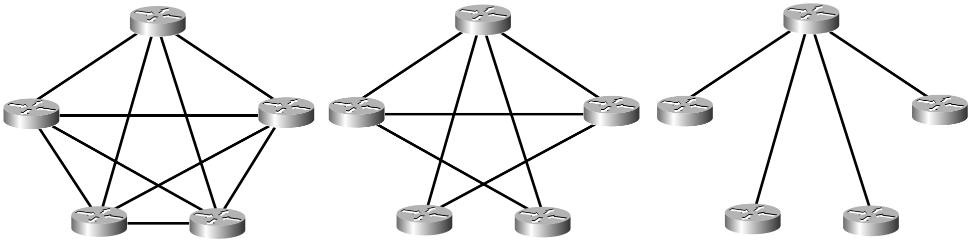
In passive-reply mode, the router does not send keepalive requests, but waits for keepalive requests from the other end of the VC and replies to them. No timer is set when in this mode, and the error counter is not incremented. If one end of a VC is configured in the passive-reply mode, the other end must be configured in the request mode.

The command to configure end-to-end keepalive (Cisco IOS 12.0.5(T) or greater) is frame-relay end-to-end keepalive mode {bidirectional | request | reply | passive-reply}.

X.121 is a hierarchical addressing scheme that was originally designed to number X.25 nodes. E.164 is a hierarchical global telecommunications numbering plan, similar to the North American Number Plan (NANP, 1-NPA-NXX-XXXX).

As one would expect, determining the number of Virtual Circuits required for a network configuration is based on the number of end nodes, and the communication requirements, such as fully meshed (all-to-all), partial meshed (some-to-all), or hub-and-spoke (all-to-one), as illustrated by Figure 15-8.

Figure 15-8 *Fully Meshed, Partially Meshed, and Hub-and-Spoke Networks*



In a fully meshed network environment, the number of VCs required can be represented by the following formula:

$$[(N \times (N-1)) / 2]$$

where *N* is the number of end nodes in the network. This formula is sometimes referred to as the "N^2 Formula" because it is derived from $((N^2-N) / 2)$.

In a partial meshed network environment, the number of VCs required is not easily represented by a mathematical formula. You must consider many variables, the least of which is based on the determination of which end nodes require communication with which other end nodes. It is a fair assumption to estimate that the number of VCs required would fall between those of a fully meshed and those of a hub-and-spoke environment:

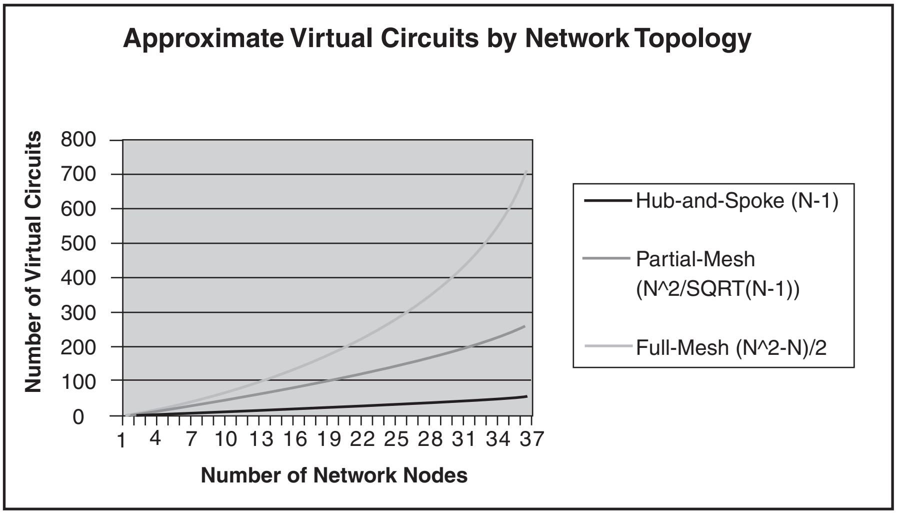
$$[((N \times (N-1)) / 2) \geq X \geq (N-1)]$$

where *N* is the number of end nodes in the network and *X* is the number of VCs required to support a partially meshed configuration. The following formula can be used as an approximation to determine the number of VCs necessary to support a partial mesh configuration: $[N^2 / \sqrt{}(N-1)]$, where *N* is the number of network nodes. This formula is useful from a network planning and cost-determination standpoint; however, because partial mesh connectivity is determined by application and user requirements at the end node, an exact number of partial-mesh VCs is almost impossible to determine.

In a hub-and-spoke network environment, the number of VCs required can be represented by the formula $[N-1]$, where *N* is the number of end nodes in the network.

Figure 15-9 illustrates the number of VCs necessary to support fully meshed [($N \times (N–1)) / 2$], partially meshed approximation [$N^2 / \sqrt{}(N–1)$], and hub-and-spoke (N–1) Frame Relay network configurations.

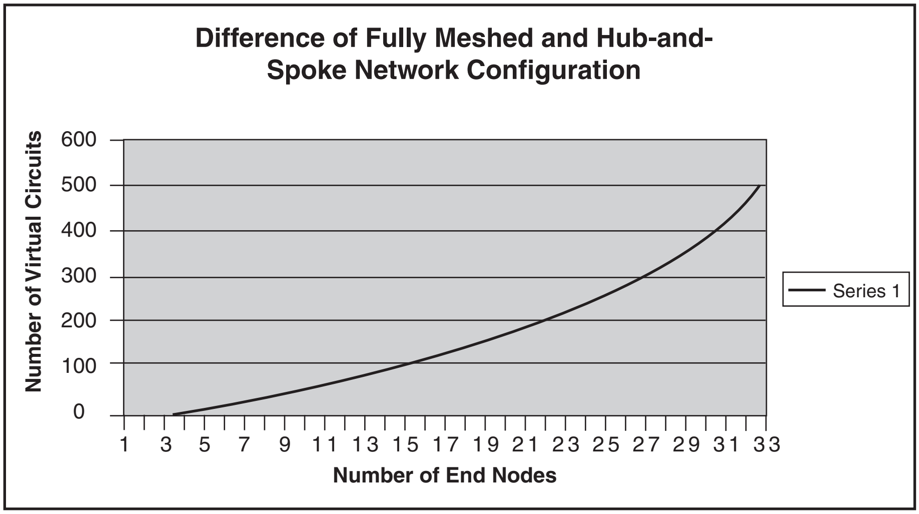
Figure 15-9 *Graph of Approximate VCs Required by Network Topology*



VCs often incur a financial obligation to the service provider. As Figure 15-9 illustrates, even relatively small networks can become quite costly very quickly based on the number of VCs alone. For this reason, hub-and-spoke network configurations are fairly common. As illustrated here, a 30-node network would require approximately 450 VCs in a fully meshed configuration, compared to the 29 VCs necessary to support a hub-and-spoke configuration.

To illustrate the potential cost savings of deploying a hub-and-spoke network over a fully meshed network environment, see Figure 15-10. As is reflected here, a difference of nearly 500 VCs exists between a fully meshed and a hub-and-spoke configuration. If it is not mandated that a fully meshed network be used, it is certainly more cost effective to design and implement a hub-and-spoke or partial-mesh configuration.

Figure 15-10 *Difference Between Fully Meshed (N^2) and Hub-and-Spoke (N-1) Network Configuration*



FECN and BECN

Congestion is inherent in any packet-switched network. Frame Relay networks are no exception. Frame Relay network implementations use a simple congestion-notification method rather than explicit flow control (such as the Transmission Control Protocol, or TCP) for each PVC or SVC; effectively reducing network overhead.

Two types of congestion-notification mechanisms are supported by Frame Relay:

- FECN
- BECN

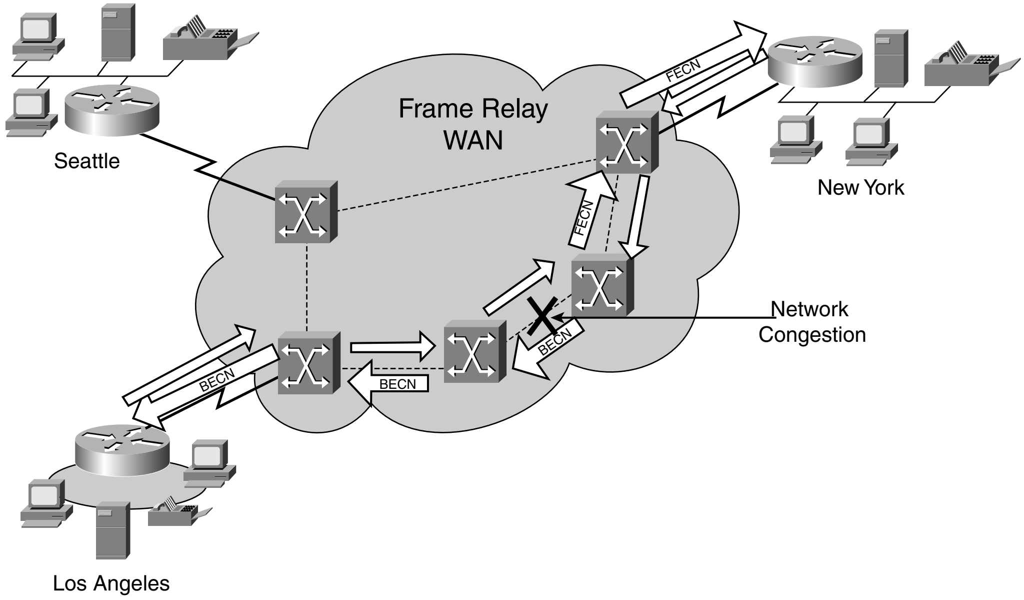
FECN and BECN are each controlled by a single bit in the Frame Relay frame header.

NOTE A Frame Relay frame is defined as a variable-length unit of data, in frame-relay format, that is transmitted through a Frame Relay network as pure data. Frames are found at Layer 2 of the OSI model, whereas packets are found at Layer 3.

The FECN bit is set by a Frame Relay network device, usually a switch, to inform the Frame Relay networking device that is receiving the frame that congestion was experienced in the path from origination to destination. The Frame Relay networking device that is receiving frames with the FECN bit will act as directed by the upper-layer protocols in operation. Depending on which upper-layer protocols are implemented, they will initiate flow-control operations. This flow-control action is typically the throttling back of data transmission, although some implementations can be designed to ignore the FECN bit and take no action.

Much like the FECN bit, a Frame Relay network device sets the BECN bit, usually a switch, to inform the Frame Relay networking device that is receiving the frame that congestion was experienced in the path traveling in the opposite direction of frames encountering a congested path. The upper-layer protocols (such as TCP) will initiate flow-control operations, dependent on which protocols are implemented. This flow-control action, illustrated in Figure 15-11, is typically the throttling back of data transmission, although some implementations can be designed to ignore the BECN bit and take no action.

Figure 15-11 *Frame Relay with FECN and BECN*



NOTE The Cisco IOS can be configured for Frame Relay Traffic Shaping, which will act upon FECN and BECN indications. Enabling Frame Relay traffic shaping on an interface enables both traffic shaping and per-VC queuing on the interface's PVCs and SVCs. Traffic shaping enables the router to control the circuit's output rate and react to congestion notification information if it is also configured. To enable Frame-Relay Traffic shaping within the Cisco IOS on a per-VC basis, use the **frame-relay traffic-shaping** command.

Cisco also implements a traffic control mechanism called ForeSight. ForeSight is the network traffic control software used in some Cisco switches. The Cisco Frame Relay switch can extend ForeSight messages over a UNI, passing the backward congestion notification for VCs.

ForeSight allows Cisco Frame Relay routers to process and react to ForeSight messages and adjust VC level traffic shaping in a timely manner. ForeSight must be configured explicitly on both the Cisco router and the Cisco switch. ForeSight is enabled on the Cisco router when Frame Relay traffic shaping is configured. The router's response to ForeSight is not applied to any VC until the **frame-relay adaptive-shaping foresight** command is added to the VC's map-class. When ForeSight is enabled on the switch, the switch will periodically send out a ForeSight message based on the time value configured. The time interval can range from 40 to 5,000 milliseconds (ms).

For router ForeSight to work, the following conditions must exist on the Cisco router:

- Frame Relay traffic shaping must be enabled on the interface.
- The traffic shaping for a circuit must be adapted to ForeSight.

In addition, the UNI connecting to the router consolidated link layer management (CLLM) must be enabled, with the proper time interval specified.

Frame Relay Router ForeSight is enabled automatically when the **frame-relay traffic-shaping** command is used. However, the **map-class frame-relay** command and the **frame-relay adaptive-shaping foresight** command must both be issued before the router will respond to ForeSight and apply the traffic shaping effect on a specific interface, subinterface, or VC.

When a Cisco router receives a ForeSight message indicating that certain DLCIs are experiencing congestion, the Cisco router reacts by activating its traffic shaping function to slow down the output rate. The router reacts as it would if it were to detect the congestion by receiving a frame with the BECN bit set.

Frame Relay Virtual Circuit (VC) Parameters

Frame Relay VCs, both permanent (PVC) and switched (SVC), have three configurable parameters that must be agreed upon between each end node (origination and termination) and the Frame Relay network service provider.

These parameters are as follows:

- CIR
- DE
- VC identifiers
  - — DLCIs for PVCs
  - — X.121/E.164 addresses for SVCs

Frame Relay CIR

The CIR is the amount of bandwidth that will be delivered as "best-effort" across the Frame Relay backbone network. Network providers typically have provisions in their tariffs guaranteeing delivery of CIR traffic at some percentage. For example, a tariff might state a guarantee such as guaranteeing delivery of 99.9% CIR marked traffic.

CIR is measured in bytes over a periodic interval of time, expressed as T_C. B_C is the committed burst rate across the VC for that period of time. B_c can be represented by the formula $B_c = CIR \times T_c$.

B_c is the negotiated maximum amount of bits that a Frame Relay internetwork is committed to accept and transmit at the CIR. Excess of CIR is measured as B_e.

B_E is the number of bits that a Frame Relay internetwork will attempt to transfer after B_c is accommodated and is marked as DE.

T_C is the periodic interval of time over which B_C and B_E are measured. The T_C interval counter starts when data begins to enter into the Frame Relay network, and ends when data is no longer entering the network. When a new data stream enters the network, the T_C counter starts over.

Frame Relay PVCs are simplex connections. Each VC is configured with its own CIR, meaning an A ➠ B PVC could be configured for 64 kbps CIR, and a B ➠ A PVC could be configured with a 32 kbps CIR. It is up to the network designer/engineer to determine the proper amount of CIR required, generally based on user and application traffic.

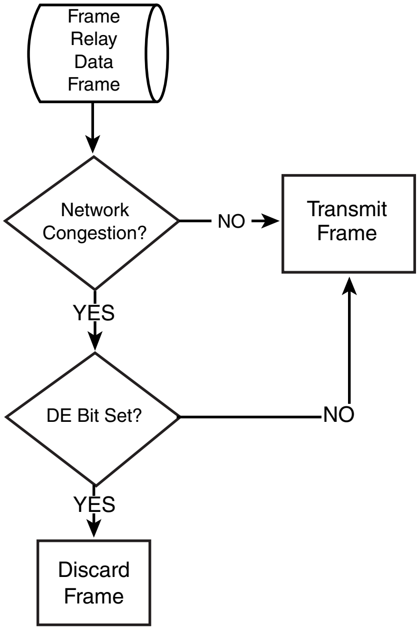
Frame Relay Discard Eligibility (DE)

Frame Relay uses a bit in the frame header to indicate whether that frame can be discarded if congestion is encountered during transmission. The DE bit is part of the Frame Relay frame header's address field.

The DE bit can be set by the transmitting Frame Relay networking device to prioritize the frame because it has a lower priority than other outgoing frames. If the network becomes congested, frames with the DE bit marked will be discarded prior to frames that do not have the DE bit marked to relieve this congestion.

Figure 15-12 illustrates the process that each Frame Relay switch runs upon receipt of a frame for transmission.

Figure 15-12 *Frame Relay Data Frame Transmission Flowchart*



DE requires that the Frame Relay network interpret, and in some cases act on, the DE bit. Some networks take no action when the DE bit is set, and other networks will use the DE bit to determine which frames to discard. Often the DE bit is used to determine which frames should be dropped first or which frames have lower time sensitivity.

DE lists can be created within the Cisco routing device to identify the characteristics of frames eligible for discarding. DE groups can be specified to identify the DLCI that is affected.

When a DE frame is discarded, it is up to the upper-layer protocols, such as TCP, to determine the loss and effect corrective actions as determined by the protocol's data correction and retransmission algorithms.

NOTE Within the Cisco IOS, the command used to define a DE list specifying the frames that can be dropped when the Frame Relay switch is congested is as follows (this command is entered in global configuration mode):

```
frame-relay de-list list-number {protocol protocol ¦
interface type number} characteristic
```

For example, the following command specifies that IP packets larger than 512 bytes (including the 4 byte Frame Relay encapsulation) will have the DE bit set:

```
frame-relay de-list 1 protocol ip gt 512
```

DE lists can be created based on the protocol or the interface, and on characteristics such as fragmentation of the frame, a specific TCP or User Datagram Protocol (UDP) port, an access list number, or a packet size (Layer 3 maximum transmission unit (MTU).

To define a DE group that is specifying the DE list and is DLCI-affected, the following command is used in interface configuration mode:

```
frame-relay de-group group-number dlci
```

For example, the following command specifies that group number 3 will be used for DLCI 170:

```
frame-relay de-group 3 170
```

PVC DLCIs

Although DLCI values can be 10, 16, or 23 bits in length, 10-bit DLCIs have become the *de facto* standard for Frame Relay WAN implementations.

The 10-bit DLCI values, as recommended by the Frame Relay Forum, are allocated as Table 15-5 indicates.

Table 15-5 *Frame Relay Forum 10 Bit DLCI Recommendations*

DLCI Value	Function
0	FRF—In-channel signaling
1 to 15	Reserved
16 to 1007	Available for VC endpoint assignment
1008 to 1022	Reserved
1023	LMI

The 10-bit DLCI values, as recommended by both the ANSI (T1.618) and the ITU-T (Q.922), are allocated as Table 15-6 indicates.

Table 15-6 *ANSI (T1.618) and ITU-T (Q.922) 10-Bit DLCI Recommendations*

DLCI Value	Function
0	In-channel signaling and management (LMI)
1 to 15	Reserved
16 to 991	Available for VC endpoint assignment
992 to 1007	Frame Relay bearer service Layer 2 management
1008 to 1022	Reserved
1023	Reserved for in-channel layer management

NOTE The number of DLCIs configurable per port varies depending on the traffic level. All 1,000 DLCIs can be used. However, 200 to 300 is a common maximum. If the DLCIs are used for broadcast traffic, 30 to 50 is a more realistic number due to CPU overhead in generating broadcasts.

Within the Cisco IOS, the number of PVCs that is configurable per interface is limited to 255; this means that a Frame Relay serial interface is limited to 255 subinterfaces. The 255 subinterface limit is dependent on how they are configured; however, no more than 255 point-to-point subinterfaces with one DLCI each can exist.

You must consider and deal with some practical performance issues on an individual case basis. The higher the CIR on the DLCIs, the more impact that the individual interface's ability will have on supporting the traffic flow.

A T1 could certainly be expected to handle 24 56 K DLCIs with little problem. However, substantial broadcast traffic could affect the performance. If, for example, 50 56 K DLCIs are configured into a T1 interface, traffic issues, such as congestion and dropped traffic, will arise. This configuration is referred to as *oversubscription*. For example, consider the following two scenarios:

A network configuration that consists of 24 (DLCIs/PVCs) × 56 kbps (CIR per PVC) = 1.344 Mbps is well within the T1 bandwidth limitation of 1.344/1.536 Mbps (depending on physical line coding; AMI = 1.344 Mbps, B8ZS = 1.536 Mbps).

This configuration is not oversubscribing the interface because available bandwidth is sufficient to support the traffic requirement: 1.344 Mbps ≤ 1.344/1.536 Mbps.

A network configuration of 50 (DLCIs/PVCs) × 56 kbps (CIR per PVC) = 2.800 Mbps far exceeds the maximum bandwidth supported by a T1 limitation of 1.344/1.536 Mbps (depending on physical line coding; AMI = 1.344 Mbps, B8ZS = 1.536 Mbps).

This configuration is oversubscribing the interface because the bandwidth available is not sufficient to support the traffic requirement: 2.800 Mbps ≥ 1.344/1.536 Mbps.

SVC X.121/E.164 Addressing

X.121 is a hierarchical addressing scheme that was originally designed to number X.25 nodes. X.121 addresses are up to 14 digits in length and are structured as follows:

- Country Code: 3 digits

  The first digit is a zone number that identifies a part of the world. For example, Zone 2 covers Europe and Zone 3 includes North America). The zone numbers can be found in Appendix C, “List of ITU-TX.121 Data Country or Geographical Codes.” These codes can also be found in ITU-T Recommendation X.121.

  — Service Provider: 1 digit

  — Terminal Number: Up to 10 digits

- E.164 is a hierarchical global telecommunications numbering plan, similar to the North American Number Plan (NANP). E.164 addresses are up to 15 digits in length and are structured as follows:

- Country Code: 1, 2, or 3 digits

  This code is based on the international telephony numbering plan and can be found in Appendix D, “International Country Codes.” These codes can also be found in any phone book.

- National Destination Code and Subscriber Number: Up to 14 digits in length (maximum length is dependent on the length of the Country Code).

  Subaddress: Up to 40 digits

Frame Relay Status Polling

The Frame Relay Customer Premises Equipment (CPE) polls the switch at set intervals to determine the status of both the network and DLCI connections. A Link Integrity Verification (LIV) packet exchange takes place about every 10 seconds, verifying that the connection is still good. The LIV also provides information to the network that the CPE is active, and this status is exported at the other end. Approximately every minute, a Full Status (FS) exchange occurs, passing information regarding which DLCIs are configured and active. Until the first FS exchange occurs, the CPE does not know which DLCIs are active, and as such, no data transfer can take place.

Frame Relay Error Handling

Frame Relay uses the Cyclic Redundancy Check (CRC) method for error detection. Frame Relay services perform error detection rather than error checking; error detection is based on the premise that the underlying network media is reliable. Frame Relay error detection uses the CRC checksum to determine if the frame is received by the Frame Relay networking device (router or switch) with, or without, error. Error correction is left to the upper-layer protocols, such as the TCP (of the TCP/IP protocol suite).

NOTE Error detection detects errors, but does not make attempts to correct the condition. Error correction detects errors and attempts to correct the condition, usually under control or direction of a higher-layer protocol. The termination node performs error detection.

Frame Relay Frame Format

Figure 15-13 illustrates the standard Frame Relay frame format.

Figure 15-13 *Frame Relay Standard Frame Format*

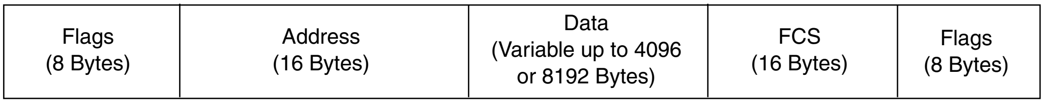


Table 15-7 presents a description of each of the Frame Relay standard frame fields.

Table 15-7 *Frame Relay Standard Frame Field Descriptions*

Field	Description
Flags	Delimits the beginning and end of the frame. The value of this field is always the same and is represented as hexadecimal 7E or as binary 0111110.
Address	Contains the following information: • DLCI—The 10-bit DLCI is the most significant part of the Frame Relay header. This value identifies and represents the VC* between the FRAD and the Frame Relay [network service provider] switch. Each VC that is multiplexed onto a physical channel will be represented by a unique DLCI. The DLCI values have local significance only, meaning they are only significant to the physical channel on which they reside. Devices on each end of a VC can use different DLCIs to identify the same VC. • Extended Address (EA)—Used to indicate whether the byte in which the EA value is 1 is the last addressing field. If the value is 1, then the current byte is determined to be the last DLCI byte. Although current Frame Relay implementations all use a two-byte DLCI, this capability does allow for the use of longer DLCIs in the future. The eighth bit of each byte of the Address field is used to indicate the EA.

Table 15-7 *Frame Relay Standard Frame Field Descriptions*

Field	Description
	MPLS labels use the extended address field of the Frame Relay frame header. • C/R—The C/R (Command/Response) bit that follows is the most significant DLCI byte in the Address field. The C/R bit is not currently defined. • Congestion Control—Consists of the 3 bits that control the Frame Relay congestion-notification mechanism. These are the FECN, BECN, and DE bits; they are the last 3 bits in the Address field. • Forward-Explicit Congestion Notification (FECN)—A single-bit field that can be set to a value of 1 by a switch to indicate to an end DTE device (router) that congestion was encountered in the direction of the frame transmission from source to destination. The primary benefit of both the FECN and BECN fields is the capability of higher-layer protocols to react intelligently to these congestion indicators. Currently, DECnet and OSI are the only higher-layer protocols that implement these capabilities. • Backward-Explicit Congestion Notification (BECN)—A single-bit field that, when set to a value of 1 by a switch, indicates that congestion was encountered in the network in the direction opposite of the frame transmission from source to destination. Explicit congestion notification is proposed as the congestion avoidance policy. It tries to keep the network operating at its desired equilibrium point so that a certain quality of service (QOS) for the network can be met. To do so, special congestion control bits have been incorporated into the address field of the Frame Relay: FECN and BECN. The basic idea is to avoid data accumulation inside the network. • Discard Eligibility (DE)—Set by the Frame Relay networking device (router) to indicate that the marked frame is of lesser importance relative to other frames being transmitted. Frames that are marked as "DE" should be discarded before other frames in a congested network. This allows for basic prioritization in Frame Relay networks.
Data	Contains encapsulated upper-layer data. Each frame in this variable-length field includes a user data or payload field that will vary in length up to 4096 bytes. This field serves to transport the higher-layer protocol data unit (PDU) through a Frame Relay network.

(continues)

Table 15-7 *Frame Relay Standard Frame Field Descriptions (Continued)*

Field	Description
Frame Check Sequence (FCS)	Ensures the integrity of transmitted data. This value is computed by the source device and is verified by the receiver to ensure integrity of the data transmission.

*Note: Physical channel = serial interface; VC = sub-interface.

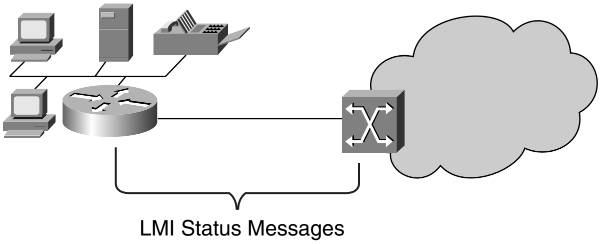
Frame Relay LMI

The Frame Relay LMI is a set of Frame Relay specification enhancements. The original LMI was developed in 1990 by the Gang of Four (Cisco, DEC, Nortel, and StrataCom). LMI includes support for the following:

- Keepalive mechanisms—Verify the flow of data
- Multicast mechanisms—Provide the network server with local and multicast DLCI information
- Global addressing—Give DLCIs global rather than local significance
- Status mechanisms—Provide ongoing status reports on the switch-known DLCIs

Figure 15-14 illustrates the endpoints for LMI status messages.

Figure 15-14 *LMI Status Message Endpoints*



The original LMI supports a number of features, or enhancements, to the original Frame Relay protocol, for managing Frame Relay internetworks. The most notable Frame Relay LMI extensions include support for the following:

- Global addressing—The LMI global addressing extension gives Frame Relay DLCI values a global, rather than local, significance. These global DLCI values become Frame Relay networking device addresses that are unique in the Frame Relay WAN.

NOTE As discussed earlier in this chapter, global addressing has an inherent limitation in that no more than 992 DLCIs (1024 DLCIs less the 32 reserved DLCIs) can be used. In a Frame Relay network of more than 992 sites, global addressing will not work. Apart from global addressing of DLCIs, the LMI status message presents an inherent limitation on the number of DLCIs that can be supported by an interface. Cisco has published a brief detailing these limitations at http://www.cisco.com/warp/public/125/lmidlci.html.

- Virtual circuit status messages—Provide communication and synchronization between Frame Relay network access devices (FRADs) and the network provider devices (switches). These messages report (in a regular interval) the status of PVCs, which prevents data from being pointed to a PVC that does not exist.
- Multicasting—Supports the assignment management of multicast groups. Multicasting preserves bandwidth by enabling routing updates and address-resolution (such as ARP, RARP) messages to be sent only to specific groups of routers.

LMI VC status messages provide communication and synchronization between Frame Relay DTE and DCE devices. These messages are used to periodically report on the status of PVCs, which prevents data from being sent into black holes (over PVCs that no longer exist).

LMI Types

Three types of LMI are found in Frame Relay network implementations:

- ANSI T1.617 (Annex D)—Maximum number of connections (PVCs) supported is limited to 976. LMI type ANSI T1.627 (Annex D) uses DLCI 0 to carry local (link) management information.
- ITU-T Q.933 (Annex A)—Like LMI type Annex D, the maximum number of connections (PVCs) supported is limited to 976. LMI type ITU-T Q.933 (Annex A) also uses DLCI 0 to carry local (link) management information.
- LMI (Original)—Maximum number of connections (PVCs) supported is limited to 992. LMI type LMI uses DLCI 1023 to carry local (link) management information.

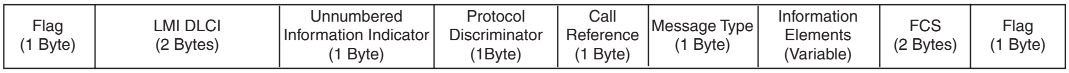
NOTE LMI Type LMI (Original) is annotated as LMI type Cisco within the Cisco IOS.

NOTE The frame MTU setting impacts LMI messages. If PVCs appear to be "bouncing," (that is, repeated up/down indications), it might be because of the MTU size of the Frame Relay frame. If the MTU size is too small, not all PVC status messages will be communicated between the service provider edge and the Frame Relay access router. If this condition is suspected, the next step is to contact the network service provider to troubleshoot.

LMI Frame Format

Figure 15-15 illustrates the LMI frame format to which Frame Relay LMI frames must conform, as deemed by the LMI specification.

Figure 15-15 *LMI Frame Format*



Flag (1 Byte)	LMI DLCI (2 Bytes)	Unnumbered Information Indicator (1 Byte)	Protocol Discriminator (1Byte)	Call Reference (1 Byte)	Message Type (1 Byte)	Information Elements (Variable)	FCS (2 Bytes)	Flag (1 Byte)

Table 15-8 presents a description of each LMI field.

Table 15-8 *LMI Frame Format Field Description*

Field	Description
Flag	Delimits the start and end of the LMI frame.
LMI DLCI	Identifies the frame as an LMI frame rather than a Frame Relay data frame. The DLCI value is dependent on the LMI specification used; LMI (original) uses DLCI 1023, LMI (Annex A) and LMI (Annex D) use DLCI 0.
Unnumbered Information Indicator	Sets the poll/final bit to zero (0).
Protocol Discriminator	Always contains a value indicating that the frame is an LMI frame.
Call Reference	Always contains zeros. This field is currently not used for any purpose.
Message Type	Labels the frame as one of the following message types: • Status-inquiry message—Allows a user device to inquire about the status of the network. • Status message—Responds to status-inquiry messages. Status messages include keepalive and PVC status messages.

Table 15-8 *LMI Frame Format Field Description*

Field	Description
Information Elements	Contains a variable number of individual information elements (IEs). IEs consist of the following fields: • IE Identifier—Uniquely identifies the IE. • IE Length—Indicates the length of the IE. • Data—Consists of one or more bytes containing encapsulated upper-layer data.
Frame Check Sequence (FCS)	Ensures the integrity of transmitted data.

LMI Extensions

The LMI global addressing extension gives Frame Relay DLCI values global rather than local significance. DLCI values become DTE addresses that are unique in the Frame Relay WAN. The global addressing extension adds functionality and manageability to Frame Relay internetworks. Individual network interfaces and the end nodes attached to them can be identified by using standard address-resolution and discovery techniques. Additionally, the entire Frame Relay WAN appears as a LAN to routers on the periphery.

The LMI multicasting extension allows multicast groups to be assigned. Multicasting saves bandwidth by allowing routing updates and address-resolution messages to be sent only to specific groups of routers. The extension also transmits reports on the status of multicast groups in update messages.

Frame Relay Applications

Traditionally, four networking suites are deployed using Frame Relay as the Layer 2 transport mechanism:

- TCP/IP Suite
- Novell IPX Suite
- IBM Systems Network Architecture (SNA) Suite
- VoFr

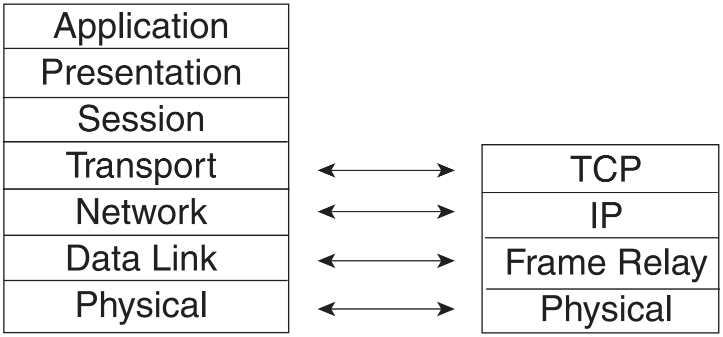
The following sections will discuss the application of these protocol suites across Frame Relay WANs and some of the special issues, challenges, and solutions that have been developed.

Frame Relay and the TCP/IP Suite

The TCP/IP Suite comprises two components: the IP operating at Layer 3 (Network) and the Transmission Control Protocol (TCP) operating at Layer 4. IP is a best-effort delivery protocol, relying on the transmission control mechanisms (that is, packet acknowledgement, sequencing) supported by TCP. IP datagrams, or *packets*, are routed from source to destination based on the address information found in the packet header. IP traffic is typically bursty in nature, making it an ideal network-layer protocol for Frame Relay WANs.

Figure 15-16 illustrates the correlation between the OSI model and an IP-over-Frame Relay implementation.

Figure 15-16 *OSI Reference Model with IP/Frame Relay*



Virtual LANs (VLANs) and IP Subnets

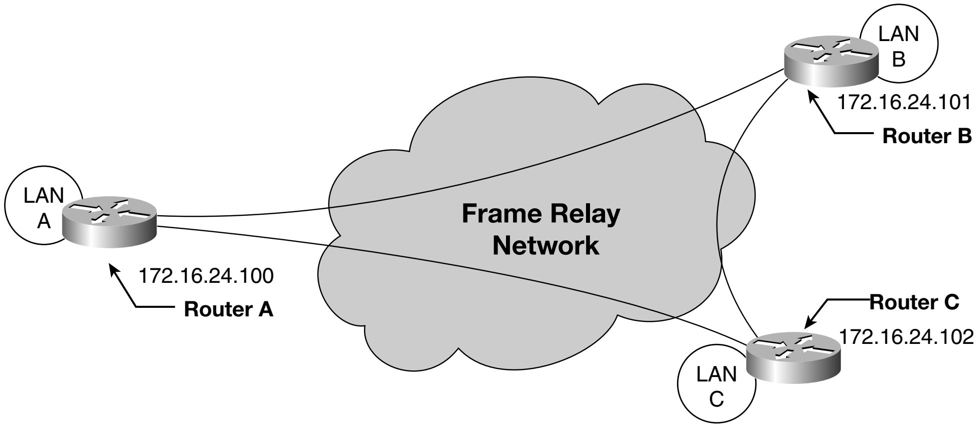
Multiple IP FRADs or routers can be interconnected via a Frame Relay WAN in a configuration behaving like a Virtual LAN (VLAN), or an IP subnet.

NOTE An IP subnet is a set of systems, or nodes/hosts, that share certain characteristics, such as the following:

- Their IP addressing starts with the same network and subnet numbers.
- Any system, or node/host, can communicate directly with any other system in the subnet. Data traffic in the same subnet will not flow through an intermediate router.

Figure 15-17 illustrates three routers that are interconnected by Frame Relay PVCs in a fully meshed configuration.

Figure 15-17 *Frame Relay with IP Subnet*



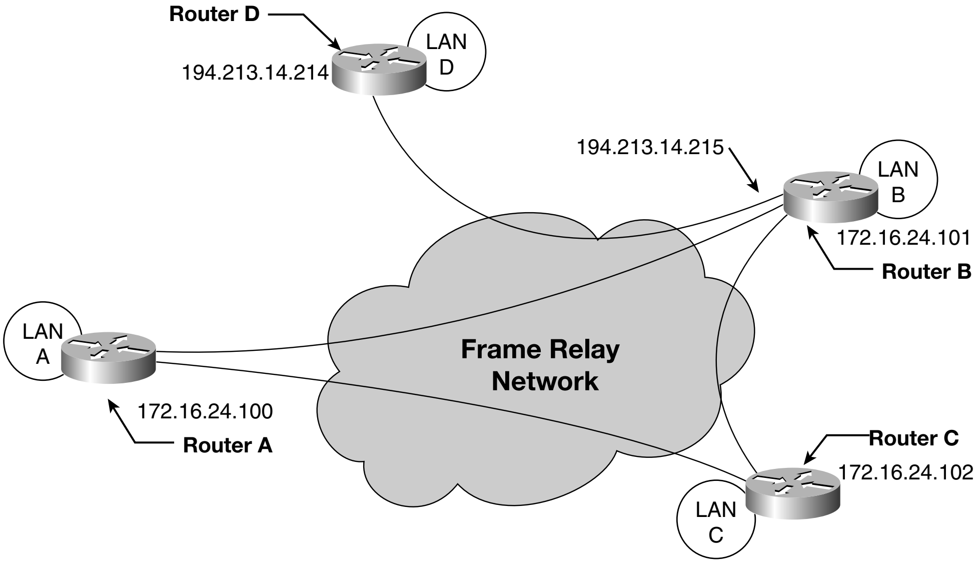
NOTE Fully meshed configurations enable every device in the system to directly communicate with every other device in the same system. Partially meshed configurations, generally known as hub-and-spoke, enable communication between devices with a central hub point providing the interconnection between end nodes.

A fully meshed Frame Relay network can be treated as a VLAN or a single IP subnet. As illustrated in Figure 15-17, three end nodes, each with an IP address assigned, are interconnected across the Frame Relay network via the Frame Relay interface on each FRAD/router. Each of these addresses shares the same network and subnet numbers, with only the host portion of the address differing.

It is worth noting that in this configuration, each Frame Relay interface is supporting multiple VCs with only a single IP address assigned to that interface, in a broadcast configuration. Another common Frame Relay configuration would have an IP address for each VC on the Frame Relay interface. This is enabled by the use of subinterfaces and will be discussed in more detail later in this chapter.

Figure 15-18 reflects the addition of a fourth router (Router D) with a different IP address from the rest of the subnet assigned to the Frame Relay interface. Router D is not part of the full mesh because it is only connected to Router B. Because Router D is part of a different subnet, new IP addresses are assigned to each endpoint of the permanent virtual circuit (PVC).

Figure 15-18 *Frame Relay with Two IP Subnets*



In summary, the following rules apply when dealing with Frame Relay implementations using virtual IP subnets:

- A single IP address can be assigned to the entire Frame Relay interface
- When one or more DLCIs are used requiring the use of subinterfaces, each subinterface is assigned an IP address.

Address Resolution Protocol (ARP)

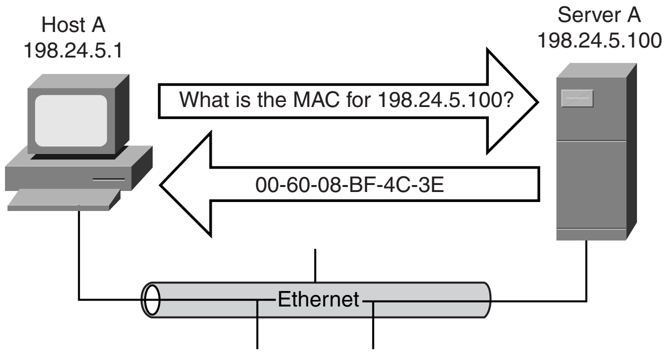
LAN systems transmit data to one another by wrapping, or encapsulating, the payload in a LAN frame whose header contains the Media Access Control (MAC) address of the destination's LAN host network interface card (NIC). A LAN system cannot communicate with a neighbor until it has discovered this neighbor's MAC address. The discovery method used is set by the procedures outlined in the Address Resolution Protocol (ARP).

To send an IP datagram to a particular IP address, the network driver must have a method to find out whether the IP address belongs to a computer on the network. If the IP address does belong to a computer on the network, the driver must know the hardware address of the computer to transmit the packet over the network. This is accomplished in Ethernet-type devices using an Internet protocol called the Address Resolution Protocol (ARP). This protocol is described in detail in RFC 826.

Figure 15-19 illustrates how ARP operates in an IP network. Host A, with an IP address of 198.24.5.1, wants to establish a connection with Server A, with an IP address of

198.24.5.100. Host A will broadcast an ARP query message across the medium asking the system, or host, with IP address 198.24.5.100 to respond. Server A replies to the ARP query providing its Layer 2 MAC address. In this example, the MAC address is 00-60-08-BF-4C-3E.

Figure 15-19 *Host A ARP Discovery of Server A MAC Address*



Host A maintains an Ethernet MAC table that records the Layer 3 Network (IP) address with its associated Layer 2 (MAC) address. In this example, Host A's ARP table would look something similar to Table 15-9.

Table 15-9 *Host A ARP Table*

IP Address	MAC Address
192.24.5.100	00-60-08-BF-4C-3E

NOTE The ARP table can be viewed from any host by entering the command **arp** at a DOS prompt:

```
C:\>arp
```

then it displays and modifies the IP-to-Physical address translation tables used by ARP:

```
ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]
```

-a Displays current ARP entries by interrogating the current protocol data. If inet_addr is specified, the IP and Physical addresses for only the specified computer are displayed. If more than one network interface uses ARP, entries for each ARP table are displayed.

-g Same as -a.

inet_addr Specifies an Internet address.

-N if_addr Displays the ARP entries for the network interface specified by if_addr.

-d Deletes the host specified by inet_addr. inet_addr can be wildcarded with * to delete all hosts.

-s Adds the host and associates the Internet address inet_addr with the physical address eth_addr. The physical address is given as six hexadecimal bytes separated by hyphens. The entry is permanent.

eth_addr Specifies a physical address.

if_addr If present, this specifies the Internet address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used.

Example:

```
> arp -s 157.55.85.212   00-aa-00-62-c6-09         .... Adds a static entry.
> arp -a                                           .... Displays the arp table.
```

Server A, upon receipt of the ARP query, will place the MAC address of Host A into its ARP table.

Inverse ARP

Inverse ARP will be discussed here as it applies to IP networking and address discovery. Inverse ARP operates in a similar fashion for Frame Relay DLCI discovery for AppleTalk, Banyan VINES, DECnet, Novell IPX, and Xerox Network Services (XNS).

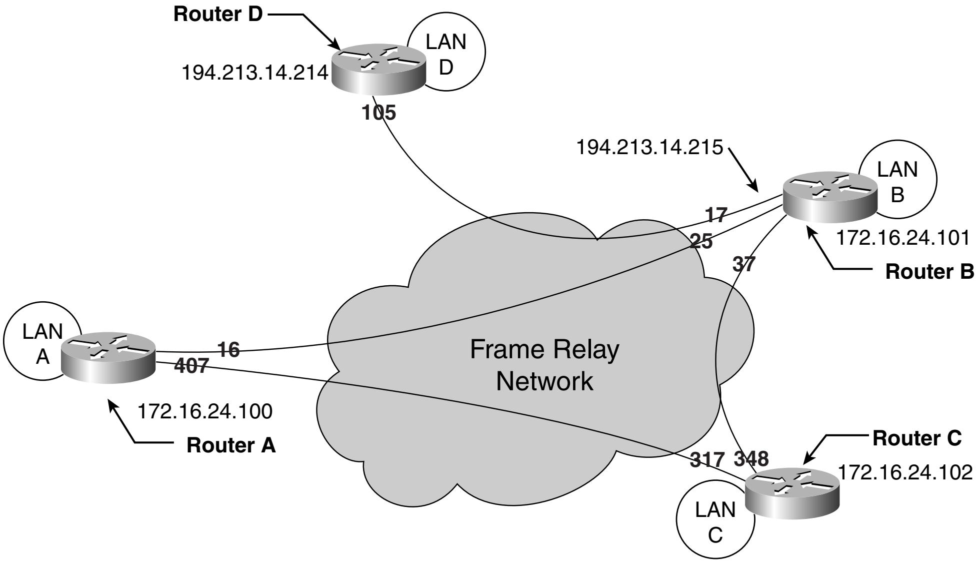
The motivation for the development of Inverse ARP is a result of the desire to make dynamic address resolution within Frame Relay both possible and efficient. PVCs and, eventually, SVCs are identified by a DLCI. These DLCIs define a single virtual connection through the WAN and are the Frame Relay equivalent to a hardware address.

Periodically, through the exchange of signaling messages, a network might announce a new VC with its corresponding DLCI. Unfortunately, protocol addressing is not included in the announcement. The station receiving such an indication will learn of the new connection, but will not be able to address the other side. Without a new configuration or mechanism for discovering the protocol address of the other side, this new VC is unusable. RFC 1293 defines Inverse ARP in more detail.

Whereas ARP enables a system to build a table mapping the LAN system's IP address to the Layer 2 MAC address, Inverse ARP is used to build a similar table mapping the connected system's IP address (by Frame Relay Virtual Circuit) to the Layer 2 DLCI on the connected system.

Figure 15-20 illustrates a four-node Frame Relay WAN, each site interconnected by a Frame Relay PVC.

Figure 15-20 *Four-Node Frame Relay WAN with DLCI Assignments*



Router B needs to determine the IP address of its neighbors prior to the forwarding of traffic across the interconnection. Router B also needs to map each neighbor's IP address to the DLCI that will be used to reach that neighbor. Router B essentially needs a table similar to Table 15-10.

Table 15-10 *Router B's Inverse ARP Table*

IP Address	DLCI
172.16.24.102	37
172.16.24.100	25
194.213.14.214	17

The IP Address column identifies the IP address of Router B's neighbor; the associated right column identifies the corresponding DLCI assignment.

Although this table could be built manually, it is far more efficient to let the Inverse ARP mechanism build it. Each router uses a simple two-step procedure to build this table. These steps include the following:

1. Each router sends a message across each connected VC asking for the IP address of the distant end.

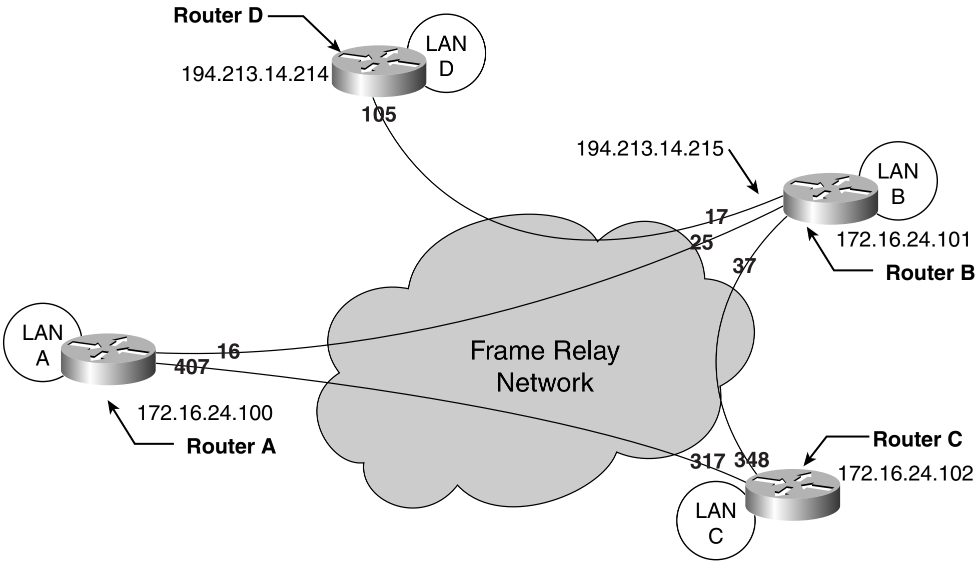
2. The sending router then records the IP address and the corresponding DLCI into its tables.

NOTE ARP is used when the Layer 3 address is known but the corresponding Layer 2 address is unknown, typically the MAC address. Inverse ARP is used when the Layer 2 address is known, typically the DLCI, but the corresponding Layer 3 address is unknown.

Inverse ARP and Routing Tables

The information learned via Inverse ARP is of significant use to IP routers. Using the network represented by Figure 15-21, suppose that Router D has message traffic to forward to Router A.

Figure 15-21 *Four-Node Frame Relay WAN with IP Address and DLCI Assignments*



Router D's routing table would look something like Table 15-11.

Table 15-11 *Router D's Routing Table*

Destination IP Address	Forwarding IP Address	Via DLCI
172.16.24.100	194.213.14.215	105
172.16.24.102	194.213.14.215	105
172.16.24.101	194.213.14.215	105

Router B’s routing table would look something like Table 15-12.

Table 15-12 *Router B’s Routing Table*

Destination IP Address	Forwarding IP Address	Via DLCI
194.213.14.214	194.213.14.214*	17
172.16.24.100	172.16.24.100*	25
172.16.24.102	172.16.24.102*	37

*This would be identified as a directly connected route in the routing table, identified by “C” when using the **show ip route** command from the IOS command prompt.

Router C’s routing table would look something like Table 15-13.

Table 15-13 *Router C’s Routing Table*

Destination IP Address	Forwarding IP Address	Via DLCI
172.16.24.100	172.16.24.100*	317
194.213.14.214	172.16.24.101	348
172.16.24.101	172.16.24.101*	348

*This would be identified as a directly connected route in the routing table, identified by “C” when using the **show ip route** command from the IOS command prompt.

Frame Relay and the Novell IPX Suite

Novell IPX implementations over Frame Relay are similar to IP network implementation. Whereas a TCP/IP implementation would require the mapping of Layer 3 IP addresses to a DLCI, Novell IPX implementations require the mapping of the Layer 3 IPX address to a DLCI. Special consideration needs to be made with IPX over Frame Relay implementations regarding the impact of Novell RIP (distance-vector algorithm) or NLSP (NetWare Link Services Protocol, link-state algorithm) and SAP (Service Advertising Protocol) message traffic to a Frame Relay internetwork.

Frame Relay IPX Bandwidth Guidelines

IPX can consume large amounts of bandwidth very quickly by virtue of its broadcast announcement-based design. Following are some considerations that demonstrate some methods to consider to manage the IPX traffic and minimize its impact on a Frame Relay WAN.

To reduce overhead in a Frame Relay network, implement the Burst Mode NetWare Loadable Module (NLM). Burst Mode opens the IPX window to avoid waiting for one acknowledgement (ACK) per IPX packet, and allows a maximum window of 128.

Another consideration is the implementation of the Large Internet Packet EXchange (LIPX) NLM if not version 4.X or higher. LIPX will allow for larger-sized packets between client and server. (Often in the case of Frame Relay WANs, the client and server will be connected via Frame Relay VC.) Native IPX without LIPX allows for a maximum payload frame size of 512 bytes; LIPX extends the packet size to 1000 to 4000 bytes. The larger packet size consumes less processing power from the Frame Relay access devices, in turn increasing throughput.

NOTE Because Ethernet and Token Ring LANs support higher frame sizes, the native IPX 512 byte frame limitation has an adverse effect on network throughput across WAN routers.

If you are working with an older version of Novell NetWare (v3.11), implement the NLSP NLM for network routing. NLSP only sends routing information when an event happens (link failure) or every two hours. The standard RIP [routing] protocol sends its entire routing table to all other routers every 30 seconds. NLSP uses less bandwidth over the WAN, ensuring more bandwidth is available for user data traffic.

NOTE SAP utilizes IPX packets to broadcast or advertise available services on a NetWare LAN. NetWare servers use these SAP packets to advertise their address, services available, and name to all clients every 60 seconds. All servers on a NetWare LAN listen for these SAP messages and store them in their own server information table. Because most Novell clients utilize local resources, the resources should be advertised on a local basis and not broadcast across the Frame Relay WAN.

Novell SAP

Novell SAP traffic can consume an adverse amount of WAN bandwidth. The router will send, without delay, a SAP update when a change is detected. It is recommended that you modify the SAP delay timer to “slow down” these delays, enabling more user traffic to get through the WAN.

The Cisco IOS command **ipx output-sap-delay 55** will send SAP packets with a 55 ms delay between packets. Without a delay, all packets in the update are sent immediately, and the Frame Relay router consumes all available buffer space. With no bandwidth or buffer space available, user traffic will be dropped, requiring retransmission.

The **ipx output-sap-delay** command causes the router to grab only one buffer a time, leaving the remaining buffer space available to queue user traffic for transmission across the WAN.

The **ipx sap-interval** and **ipx update-interval** IOS commands can be used to change the frequency of the updates between IPX-enabled devices. All IPX-enabled devices (routers) interconnected across the Frame Relay WAN must be set to the same update interval. Otherwise, updates will not be synchronized, resulting in *phantom routes*—routes that appear and disappear with each update.

In a multipoint Frame Relay "broadcast" environment, in which message traffic is propagated to all sites and subinterfaces are not employed, SAP advertisements will be propagated to all sites as well.

NOTE The absence of the **broadcast** parameter in the Frame Relay map configuration will prevent both IPX RIP and SAP advertisements from being propagated.

Whereas point-to-point Frame Relay WAN links do not employ a map statement, IPX RIP and SAP updates will be propagated at will between each site. It is recommended that you use IPX RIP and SAP filters in this configuration to minimize the Frame Relay WAN traffic.

Frame Relay and the IBM SNA Suite

IBM mainframes and SNA were the *de facto* standard of the networking community for many years, predominantly in the 1970s and 1980s. IP has since replaced SNA as the dominant internetworking protocol suite. SNA is still very much "at large," especially in large legacy networking systems, such as those found within the banking and financial industries.

These IBM SNA networking environments were ideally suited to the internetworking environment enabled by Frame Relay network implementations due to the lower-cost and cleaner architecture, compared to that of traditional point-to-point private line interconnections. Whereas point-to-point network architecture requires several lines and interfaces, Frame Relay networks enable a single line (serial interface) and multiple subinterfaces, one for each SNA communication session (Frame Relay VC).

Migration of a legacy SNA network from a point-to-point infrastructure to a more economical and manageable Frame Relay infrastructure is attractive; however, some challenges exist when SNA traffic is sent across Frame Relay connections. IBM SNA was designed to operate across reliable communication links that supported predictable response times. The challenge that arises with Frame Relay network implementations is that Frame Relay service tends to have unpredictable and variable response times, for which SNA was not designed to interoperate or able to manage within its traditional design.

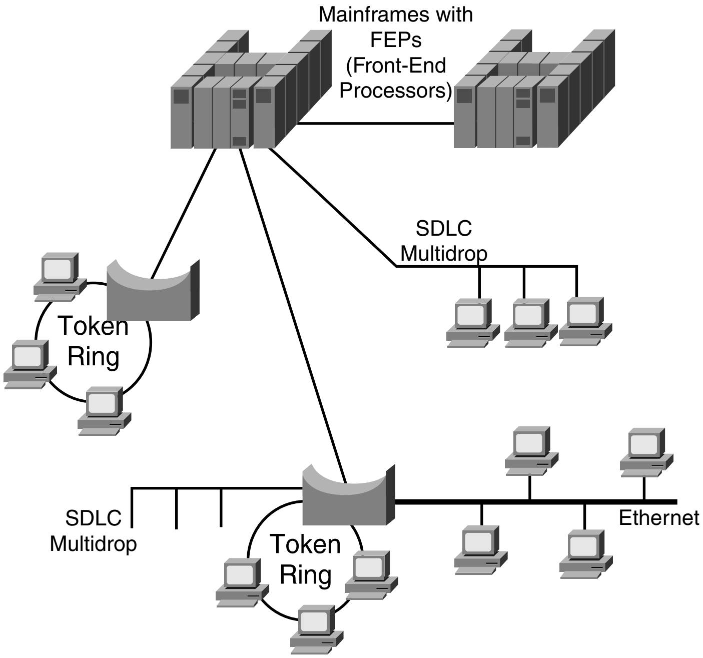
NOTE Migration from SDLC to Frame Relay networking environments will require an upgrade to the communications software packages in both the FEPs and SNA controllers.

Typically, SNA controllers, routers, and FRADs encapsulate SNA traffic as multiprotocol data, as described in the Frame Relay Forum's FRF 3.1 Implementation Agreement.

Traditional IBM SNA Network Configuration

Figure 15-22 illustrates a traditional IBM SNA network configuration.

Figure 15-22 *Traditional IBM SNA Network Configuration*



An SNA network has two primary components:

- Front-end processors (FEPs)—FEPs offload the coordination effort required to enable and support communication between IBM hosts and (potentially) thousands of remote devices.
- Remote controllers—Remote controllers are located at remote locations and are used to interconnect LANs and low-bandwidth (typically 9.6 kbps or 56 kbps) leased lines. Remote controllers concentrate several sources of remote traffic onto one high bandwidth connection to the front-end processor.

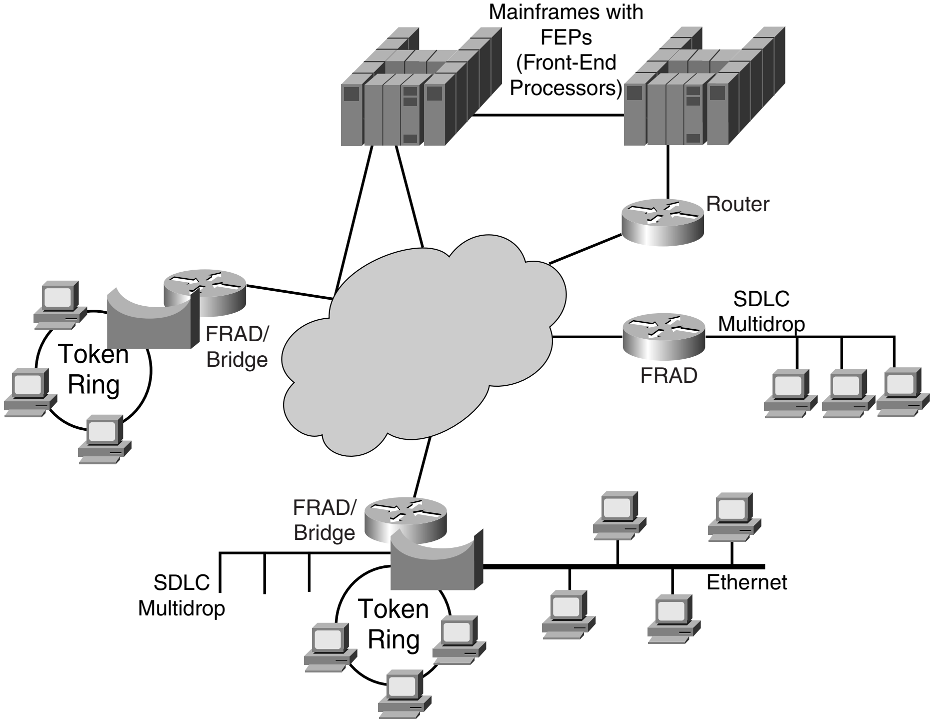
IBM's SNA environment supports the implementation of multidrop technology, making multipoint leased line configurations more cost effective versus each SNA drop possessing its own leased line. In the multidrop environment, several devices share the same leased line with the front-end processor or remote controller, polling these devices and allowing each device a turn to communicate with the mainframe.

The IBM SNA environment relies heavily upon the FEP's polling mechanisms because the FEP controls when each of its connected remote devices can send and receive data. The SNA infrastructure is based on this polling methodology.

When the FEP polls the remote device, it expects to see a response within a preconfigured timeout period. This timeout threshold is typically a fairly small period of time, generally a few seconds. If the timeout period expires, the poll is retransmitted. Frame discards and late frame arrivals (usually caused by network congestion) can disrupt SNA communication.

Figure 15-23 illustrates a Frame Relay implementation, replacing point-to-point leased lines, supporting an IBM SNA infrastructure.

Figure 15-23 *IBM SNA Implementation over Frame Relay*



SNA Data Link Protocols

Two reliable data link protocols are used for FEP/controller communication in the IBM SNA environment: Synchronous Data Link Control (SDLC) and Logical Link Control, type 2 (LLC2).

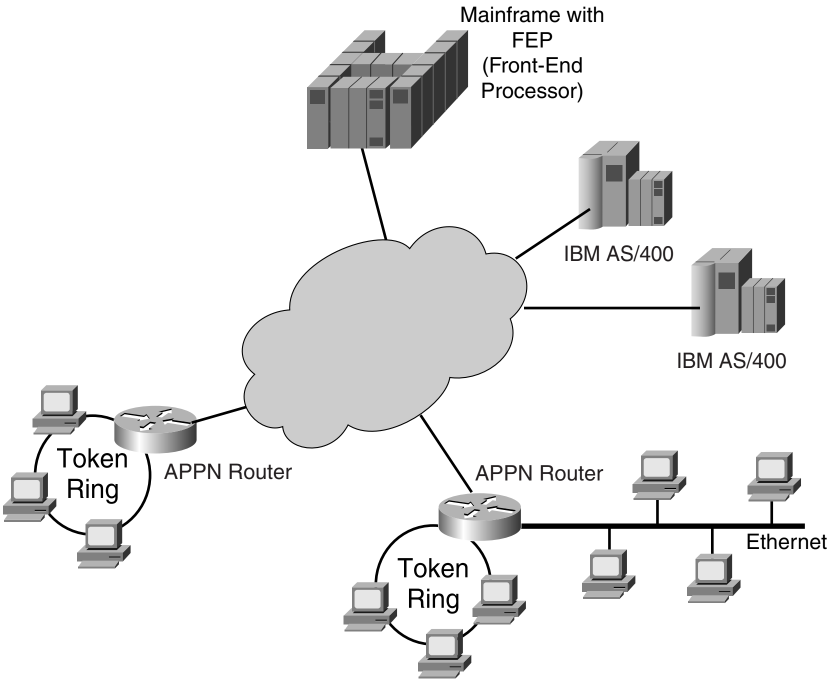
Modern SNA networks also support end-to-end sessions set up by the Advanced Peer-to-Peer Networking (APPN) protocol. Figure 15-24 illustrates an APPN infrastructure supporting communication between a mainframe, AS/400 hosts, and LAN systems.

NOTE APPN relies on LLC2 links.

IBM offers an extension to APPN that can optionally operate without LLC2: High Performance Routing (HPR). HPR can operate without an underlying [reliable] data link protocol. Retransmission and flow control is performed end-to-end by a higher layer protocol, similar to TCP within the TCP/IP protocol suite.

HPR traffic that does not operate on top of an LLC2 link can support transmission across Frame Relay links without encountering the issues associated with reliable links, such as SDLC or LLC2. An example of reliable link issues not found with HPR is an SDLC or LLC2 poll timeout.

Figure 15-24 *APPN Network*



SDLC and LLC2

The IBM SDLC protocol was designed for SNA-based networks and has a number of features that must be addressed when leased-lines are replaced by Frame Relay circuits. These features include the following:

- SDLC is a master/slave polling protocol—An FEP or controller polls remote devices to ascertain whether they have data to be sent or received. SDLC polling traffic is heavy and consumes bandwidth. In addition, the FEP or controller must receive poll responses within a strictly predictable time limit, usually measured in a few seconds.

- SDLC makes liberal use of control frames for flow control—A Frame Relay circuit that is carrying raw SDLC traffic will be congested with frequent SDLC polls and other control traffic.
- Each SDLC information frame is numbered in sequence and contains frame acknowledgements—After a preset number of frames have been sent, data transmission will not proceed unless the sender receives an acknowledgement from the terminating partner (receiver).
- SDLC is not used for LAN peer-to-peer communications—SNA LAN frames contain an LLC2 header that contains both the frame sequence and the acknowledgement numbers.
- LLC2 does not have the polling overhead attributed to SDLC—LLC2 does have the overhead associated with reliable, ordered, flow-controlled delivery of data across a communications link.

Data-Link Switching (DLSw)

Data-link switching (DLSw) is a means of transporting SNA and NetBIOS traffic across a network using many different protocols. The original RFC 1434 described DLSw, but that RFC has been superceded by RFC 1795, which describes DLSw version 1. More recently, scalability enhancements have been introduced in DLSw version 2. Cisco has introduced some enhancements in its DLSw+ implementation that are backward compatible with both version 1 and version 2.

DLSw has the following advantages over SRB:

- DLSw gets around the SRB 7-hop limit.
- DLSw allows multiple connections across a network.
- DLSw increases session response times.
- DLSw provides flow control.
- DLSw reroutes traffic around broken links.
- DLSw removes the SRB heavy broadcast traffic.

Additionally, DLSw implementations provide SDLC to LLC2 conversion, eliminating the need for many Front End Processor (FEP) ports. DLSw supports RFC 1490, enabling LLC2 over Frame Relay and DLSw prioritization.

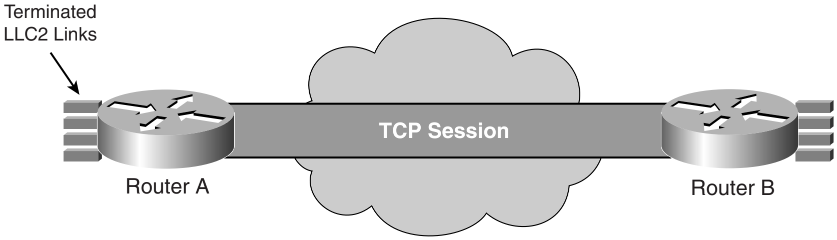
DLSw uses the Switch-to-Switch Protocol (SSP) in place of source route bridging (SRB) between routers. SSP is used to create DLSw peer connections, locate resources, forward data, and handle flow control and error recovery. TCP is used for DLSw encapsulation. A newer, standard version of DLSw is not restricted to TCP for encapsulation services.

The routers are called data-link switches. The data-link connections (DLCs) are terminated at the router, or data-link switch, so that the Routing Information Field (RIF) ends at a virtual ring within the router. Because DLCs are locally terminated, they can be locally acknowledged. This local acknowledgement means that the necessity for link layer acknowledgements or keeping alive messages to run across the WAN do not exist, minimizing session timeouts. Because the RIF ends at the peer router at each end, six hops can be added on each side of the virtual ring, thereby extending the network. With remote source-route bridging (RSRB), the RIF is carried all the way through the virtual ring, thereby limiting the number of hops. With DLSw, the virtual ring can be different in each peer because of the RIF termination.

Frame relay circuits that are carrying reliable link traffic incur a substantial amount of increased overhead. One Frame Relay circuit has the potential to carry several separate reliable links. Each link requires acknowledgement and flow control messages, which in turn require available bandwidth to carry the additional traffic.

The carrying of LLC2 links across a frame circuit can be avoided with the use of DLSw, as illustrated in Figure 15-25.

Figure 15-25 *Data Link Switching (DLSw)*



When DLSw is implemented, the LLC2 links are terminated at each router. Incoming data is transmitted across the Frame Relay WAN via a TCP session and is then forwarded across a new LLC2 link.

NOTE DLSw is not constrained to Frame Relay WANs; DLSw interoperates with any WAN technology.

The SNA traffic is preserved by the TCP sessions that support reliable data transfer. The TCP protocol, by its nature and design, adjusts well to sporadic transmission delays, efficiently manages acknowledgements, and carries out flow control without adding overhead to the communications flow.

Implementing DLSw has a disadvantage in that the TCP/IP headers add extra overhead to the transmitted data. This is generally worth the tradeoff compared to the overhead involved with the management of multiple independent LLC2 links.

SNA and DLSw Traffic Management

Following is an example of an access list enabling SNA traffic to be passed across a DLSw link:

```
access-list 200 permit 0x0d0d 0x0101
access-list 200 deny 0x0000 0xffff
dlsw remote-peer 0 tcp 1.1.1.1 lsap-output-list 200
```

If non-SNA traffic is to be blocked, it is recommended that you prevent the traffic from coming into the router and being classified. After traffic is classified, the router's resources begin to be consumed.

```
source-bridge input-lsap-list 200
```

Custom Versus Priority Queuing

To ensure that SNA traffic is managed (that is, sessions do not time out), Cisco recommends the use of either custom or priority queuing.

Priority-queuing is easier to configure than custom-queuing, but priority-queuing can potentially "break" the Frame Relay network. Priority queuing *always* checks the higher priority queues before checking the lower priority ones. Therefore, if IP is configured in a high priority queue and IPX in a normal priority queue, the possibility exists to completely choke out IPX traffic if an IP packet is always ready in the high queue (such as infinite preemption). This results in lost IPX sessions, which creates problems for network users. If known low bandwidth protocols are placed on the high queue, this possibility can be eliminated. For example, small numbers of SNA users who are running interactive 3270 traffic on a LAN, or SNA users residing off a slow SDLC line, would not be able to keep the high queues full constantly. This would also apply to other protocols that are bandwidth constrained on the in-bound side. This is the ideal situation in which to use priority queuing.

Custom-queuing removes the possibility of infinite preemption by permitting the administrator to customize how the various queues are serviced.

The following example demonstrates the process of queue servicing:

- For example, if starting with 10 possible queues, the router polls all queues all the time.
- If queue 1 is configured to contain IP traffic and queue 2 to contain IPX traffic, the router services X number of bytes on queue 1, then moves on to queue 2 and services X number of bytes there. (The router administrator can configure the value for X.)
- After servicing queue 1, if queue 2 has no packets, the router immediately moves on to the next queue, which in this case will be queue 1, allowing traffic on queue 1 to use all available bandwidth, if no other protocols require it.

NOTE Custom-queuing and priority-queuing will be discussed in more detail in Chapter 17, “Frame Relay WAN Analysis.”

When the serial line [interface] is saturated, queues can be configured to the proper byte count values (*Q*) if the average size of the packets is known. This essentially configures bandwidth allocation on a per-protocol basis. In this scenario, some “tweaking” will likely be required.

NOTE As described here, per-protocol bandwidth allocation is a powerful feature that is not easy to implement. Care should be taken to review all configurations prior to implementing this strategy.

Voice over Frame Relay (VoFr)

Voice over Frame Relay (VoFr) has been recently enjoying the general acceptance of any efficient and cost-effective technology. In the traditional plain old telephone service (POTS) network, a conventional (with no compression) voice call is encoded, as defined by the ITU pulse code modulation (PCM) standard, and utilizes 64 kbps of bandwidth. Several compression methods have been developed and deployed that reduce the bandwidth required by a voice call down to as little as 4 kbps, thereby allowing more voice calls to be carried over a single Frame Relay serial interface (or subinterface PVC). Table 15-14 demonstrates these different compression algorithms and the bandwidth utilized per algorithm.

Table 15-14 *Voice Compression Algorithms with Bandwidth Utilization*

Encoding/Compression	Bit Rate (Bandwidth Required)
G.711 PCM (A-Law/U-Law)	64 kbps (DS0)
G.726 ADPCM	16, 24, 32, 40 kbps
G.729 CS-ACELP	8 kbps
G.728 LD-CELP	16 kbps
G.723.1 CELP	6.3/5.3 kbps variable

NOTE A common concern regarding VoFR is keeping latency and jitter within acceptable limits. This can be a challenge in a network that is built around applications that can tolerate both; however, it is possible to pull a “trick or two” within the network.

One approach is to increase the CIR over each Frame Relay PVC that will carry voice traffic and not mark the voice traffic as DE.

Another approach is to implement separate PVCs for voice and data applications, segregating the traffic prior to transmission from the router.

A third approach is to work with a network service provider that offers PVC of different levels of delay/priority. Some providers offer as many as three PVC levels, or priorities:

- Top priority for delay-sensitive traffic (such as voice and SDLC)
- No (or middle) priority for traffic that can tolerate some level of delay (such as LAN traffic)
- Low priority for applications that can tolerate significant levels of delay (such as Internet access and e-mail)

Voice Coders-Decoders (Codecs)

The issue with packetized voice is the ability of the sending and receiving voice codecs (coders-decoders) to be able to clock against each other to ensure the synchronization of the data flow. Two of the more common Voice over X (VoX) implementations are Voice over Frame Relay (VoFr) and Voice over ATM (VoATM). The reason for ATM's popularity in the VoX arena is that ATM utilizes fixed cell lengths of 53 bytes, enabling the sending and receiving codecs to clock against each other in synchronicity, ensuring the seamless flow of the voice traffic.

NOTE VoIP recently has come to the forefront of Voice over X (VoX) and will be discussed in detail in Chapter 20, "Voice Technology."

Frame Relay frames operate in a similar fashion to ATM (packet-switching versus cell-switching). However, one of the significant differences with regard to voice traffic is that Frame Relay frames are of variable length, up to 4096 bytes, making it difficult for the sending and receiving codecs to clock against each other because the "start" and "stop" flags appear at seemingly random intervals.

Several VoFr, or Voice FRAD (VFRAD), vendors are on the market today, each with its own workaround to the variable frame length issue. Each workaround is similar in that the sending codec limits the payload size of the transmitted frame, usually to 2000 bytes, or 2 kilobytes (KB). By utilizing this fixed length, the sending and receiving codecs can now clock against each other because the receiving codec now knows when the "stop" flag will appear, enabling a more seamless voice conversation.

VoFr Quality

A direct correlation exists between the quality of voice and the compression algorithm used. This quality is measured with something called the mean opinion score (MOS). Table 15-15 compares these different compression algorithms and their respective MOS.

Table 15-15 *Voice Compression Mean Opinion Scores*

Encoding Compression	Mean Opinion Score	Native Bit Rate kbps	Voice Quality	BW	DTMF	Dual	Comp	CPU Music on Hold
G.711 PCM	4.1	64	A	D	A	A	A	A
G.726 ADPCM	3.85	32	B	C	B	B	B	B
G.728 LD-CELP	3.61	16	C	B	B	C	C	C
G.729 CS-ACELP	3.92	8	A	A	B	B	C	C
G.729a CS-ACELP	3.7	8	B	A	C	C	B	D
G.723.1	3.65	5.3	C	A	C	D	C	D

It is considered efficient to send compressed voice over data circuits, initially over point-to-point leased lines and more recently over Frame Relay. Because of this, it is natural for enterprise users to consider supporting voice service across an existing, or planned, Frame Relay WAN.

Generally, VoFr implementations utilize CIR/DE to prioritize voice over data traffic across the VC. To do this effectively, the proper amount of CIR bandwidth needs to be determined prior to implementation. The formula used to determine this is as follows:

$$FRL_{CIR} = ((VOX_{MODULE} \times MODULE_{BANDWIDTH}) + VOX_{BUFFER})$$

$$VOX_{BUFFER} = ((VOX_{MODULE} \times MODULE_{BANDWIDTH}) \times 20\%)$$

where VOX_{MODULE} is the number of VoFr modules; $MODULE_{BANDWIDTH}$ is the amount of bandwidth per voice module, and VOX_{BUFFER} is the amount of additional buffer space on the VC for the voice traffic.

For example, assume a FRAD with a 4-port voice module, utilizing G.728 compression (16 kbps). The CIR required to support the VoFr service is determined by the previous formulae:

$$FRL_{CIR} = ((4 \times 16) + 12.8) = 76.8 \text{ kbps}$$

$$VOX_{BUFFER} = ((4 \times 16 \text{ kbps}) \times 20\%) = 12.8 \text{ kbps}$$

The minimum amount of CIR required to support this configuration is 76.8 kbps. Typically, a network provider provisions CIR in multiples of 16 kbps or 64 kbps; therefore, the minimum CIR to support the voice traffic here would be 80 kbps. This is for voice traffic only; it is reasonable to expect to add incremental CIR to support data traffic requirements as well across the VC.

VoFR, as with all VoX, is subjected to and unforgiving of quality issues, most notably delay and jitter. These and other concerns will be discussed in greater detail in Chapter 20.

VFRADs

Voice over Frame Relay service is enabled by the use of Frame Relay communications devices, such as routers or FRADs, configured with voice modules. These devices are sometimes referred to as Voice FRADs, or VFRADs.

Although VoFR implementations are efficient and cost effective for intra-enterprise or intra-corporate communication, there are considerations, such as quality, to be made regarding the use of packetized voice when dealing with non-Frame Relay WAN, or off-net, users.

Figure 15-26 illustrates how a typical Voice over Frame Relay implementation might look, supporting simultaneously both on-net (VoFr) and off-net (POTS) implementations.

Figure 15-26 *On-Net (VoFr) and Off-Net (POTS) Voice Implementation*

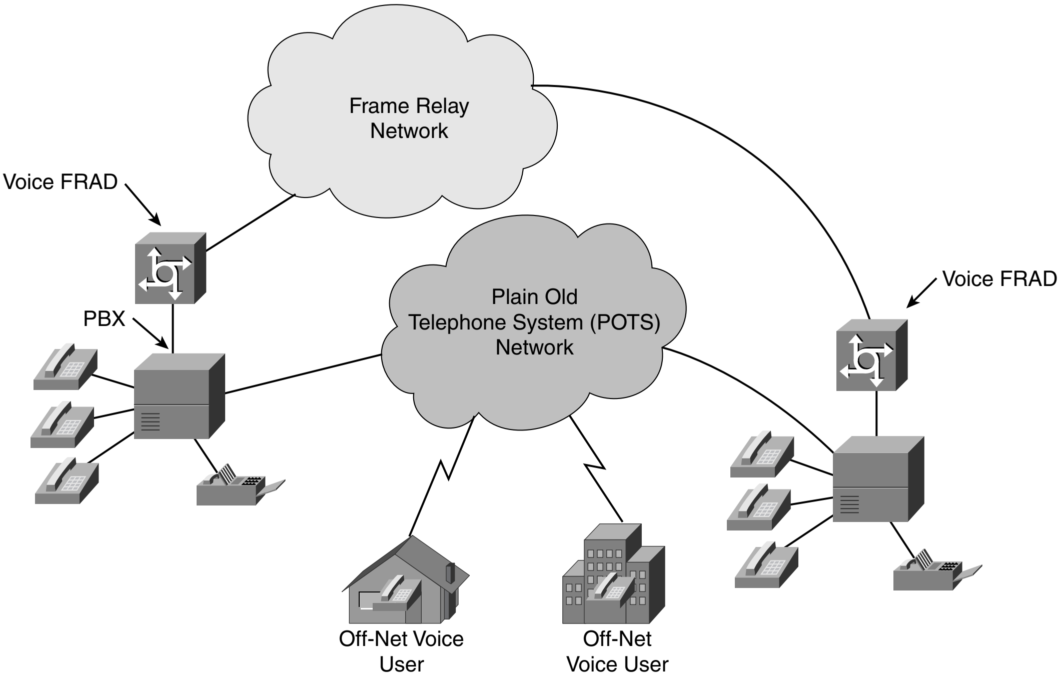
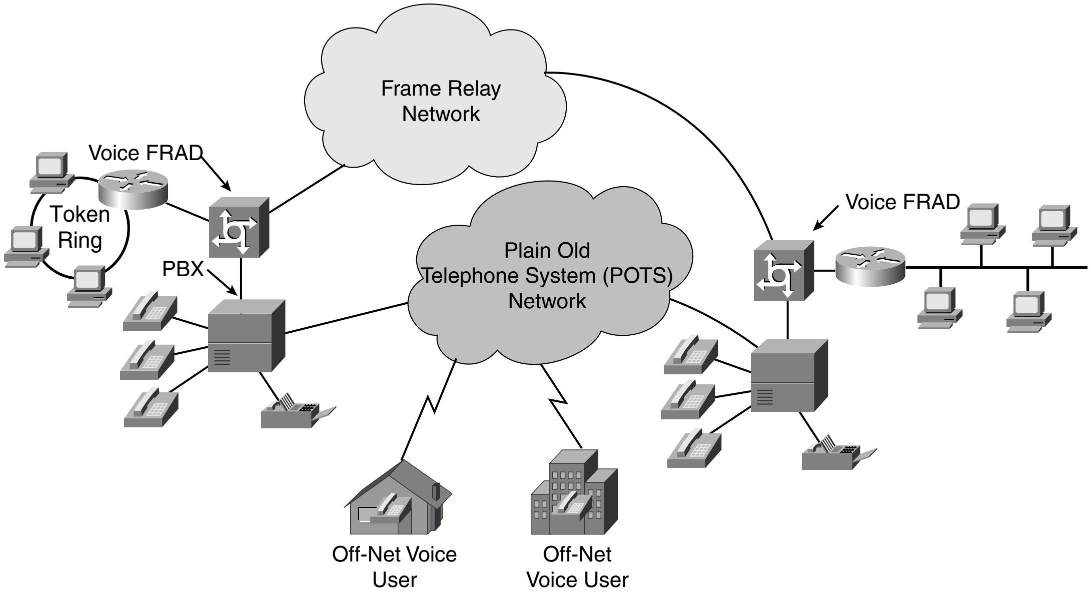


Figure 15-27 illustrates how voice and data can be merged to more effectively utilize WAN resources by the addition of a router to support data communication between enterprise sites.

Figure 15-27 *Frame Relay with Voice, Data, and Fax Communications*



Frame Relay Traffic Shaping

Traffic shaping supports the controlling of the traffic going out of an interface. This control matches the flow of traffic to the speed of the remote destination (or target) interface and ensures that the traffic conforms to policies contracted for the interface. Traffic adhering to a particular profile can be shaped to meet downstream requirements, eliminating bottlenecks in topologies with data-rate mismatches.

The primary reasons for using traffic shaping are to control access to available bandwidth, to ensure that traffic conforms to the policies established for the available bandwidth, and to regulate the flow of traffic to avoid congestion. Congestion can occur when the sent traffic exceeds the access speed of its destination (target) interface across a VC.

Following are some examples of when to use traffic shaping:

- To control access to bandwidth when policy dictates that the rate of a given interface should not, on the average, exceed a certain rate, even though the access rate exceeds the speed.

- To configure traffic shaping on an interface if you have a network with differing access rates. Suppose that one end of the link in a Frame Relay network runs at 256 kbps and the other end of the link runs at 128 kbps. Sending packets at 256 kbps could cause failure of the applications that are using the link.

NOTE Regarding a similar, more complicated case, a link-layer network giving indications of congestion that has differing access rates on different attached DTE; the network might be able to deliver more transit speed to a given DTE device at one time than another. (This scenario warrants that the token bucket be derived, and then its rate maintained.)

- To partition the T1 or T3 links into smaller channels in a subrate service scenario.

Traffic shaping prevents packet loss. The use of traffic shaping is especially important in Frame Relay networks because the switch cannot determine which frames take precedence and therefore which frames should be dropped when congestion occurs. It is important for real-time traffic, such as VoFR, that latency be bounded, thereby bounding the amount of traffic and traffic loss in the data link network at any given time by keeping the data in the router that is making the guarantees. Retaining the data in the router allows the router to prioritize traffic according to the guarantees that the router is making.

Traffic shaping limits the rate of transmission of data, limiting the data transfer to one of the following:

- A specific configured rate
- A derived rate based on the level of congestion

The transfer rate depends on three components that constitute the token bucket: burst size, mean rate, measurement (time) interval.

The mean rate is equal to the burst size divided by the interval, as demonstrated by the following equation:

$$\text{Mean rate} = \text{Burst Size } (B_C + B_E) / \text{Time Interval } (T_C)$$

When traffic shaping is enabled, a maximum burst size can be sent during every time interval. However, within the interval, the bit rate might be faster than the mean rate at any given time.

B_E size is an additional variable that applies to traffic shaping. The excess burst size corresponds to the number of noncommitted bits—those bits outside the CIR—that are still accepted by the Frame Relay switch but marked as DE.

The B_E size allows more than the burst size to be sent during a time interval. The switch will allow the frames that belong to the excess burst to go through, but it will mark them by setting the DE bit. The switch configuration determines whether the frames are sent.

When B_E size equals 0 ($B_E = 0$) the interface sends no more than the burst size every interval, realizing an average rate no higher than the mean rate. When B_E size is greater than 0 ($B_E > 0$) the interface can send as many as $B_C + B_E$ bits in a burst, if the maximum amount was not sent in a previous time period. When less than the burst size is sent during an interval, the remaining number of bits, up to the B_E size, can be used to send more than the burst size in a later interval.

Frame Relay DE Bit

Frame Relay frames can be specified regarding which have low priority or low time sensitivity. These frames will be the first to be dropped when a Frame Relay switch is congested.

The DE bit is the mechanism that allows a Frame Relay switch to identify such frames to be dropped or discarded.

DE lists and groups can be managed in the following manner:

- DE lists can be specified that identify the characteristics of frames to be eligible for discarding.
- DE groups can be specified to identify the affected DLCI.
- DE lists can also be specified based on the protocol or the interface, and on characteristics such as fragmentation of the packet, a specific TCP or User Datagram Protocol (UDP) port, an access list number, or a packet size.

Differences Between Traffic-Shaping Mechanisms

Generic traffic shaping (GTS), class-based shaping, distributed traffic shaping (DTS), and Frame Relay traffic shaping (FRTS) are similar in implementation, share the same code and data structures, differ in regard to their CLIs, and differ in the queue types used.

Following are some examples in which these mechanisms differ:

- For GTS, the shaping queue is a weighted fair queue. For FRTS, the queue can be a weighted fair queue (configured by the **frame-relay fair-queue** command), a strict priority queue with WFQ (configured by the **frame-relay ip rtp priority** command in addition to the **frame-relay fair-queue** command), custom queuing (CQ), priority queuing (PQ), or first-in, first-out (FIFO). See Table 15-16 for detailed differences.
- For class-based shaping, GTS can be configured on a class, rather than only on an access control list (ACL). To do so, you must first define traffic classes based on match criteria including protocols, access control lists (ACLs), and input interfaces. Traffic shaping can be applied to each defined class.

- FRTS supports shaping on a per-DLCI basis; GTS and DTS are configurable per interface or subinterface.
- DTS supports traffic shaping based on a variety of match criteria, including user-defined classes, and DSCP.

Table 15-16 *Differences Between Shaping Mechanisms*

Mechanism	GTS	Class-Based	DTS	FRTS
Command-Line Interface	Applies parameters per subinterface Traffic group command supported	Applies parameters per interface or per class	Applies parameters per interface or subinterface	Classes of parameters Applies parameters to all VCs on an interface through inheritance mechanism No traffic group command
Queues Supported	Weighted fair queuing (WFQ) per subinterface	Class-based weighted fair queuing (CBWFQ) inside GTS	WFQ, strict priority queue with WFQ, CQ, PQ, first come, first served (FCFS) per VC	WFQ, strict priority queue with WFQ, CQ, PQ, FCFS per VC

GTS can be configured to behave the same as FRTS by allocating one DLCI per subinterface and using GTS plus BECN support. The behavior of the two is then the same with the exception of the different shaping queues used.

FRTS, like GTS, can eliminate bottlenecks in Frame Relay networks that have high-speed connections at the central site and low-speed connections at branch sites. Rate enforcement can be configured as a peak rate configured to limit outbound traffic to limit the rate at which data is sent on the VC at the central site.

FRTS can be used to configure rate enforcement to either the CIR or some other defined value, such as the excess information rate, on a per-VC basis. The ability to allow the transmission speed that the router uses to be controlled by criteria other than line speed—the CIR or excess information rate—provides a mechanism for sharing media by multiple VCs. Bandwidth can be allocated to each VC, creating a virtual time-division multiplexing (TDM) network.

PQ, CQ, and WFQ can be defined at the VC or subinterface level. These queuing methods allow for finer granularity in the prioritization and queuing of traffic, providing more control over the traffic flow on an individual VC. If CQ is combined with the per-VC queuing and rate enforcement capabilities, Frame Relay VCs can carry multiple traffic types such as IP, SNA, and IPX with bandwidth guaranteed for each traffic type.

FRTS can dynamically throttle traffic by using information that is contained in the BECN-tagged frames that are received from the network. With BECN-based throttling, frames are held in the router buffers to reduce the data flow from the router into the Frame Relay network. Throttling is done on a per-VC basis, and the transmission rate is adjusted based on the number of BECN-tagged frames received.

Derived Rates

FECNs and BECNs indicate congestion in a Frame Relay WAN and are specified by bits within a Frame Relay frame. FECN and BECN operation is as follows:

- FECNs—These are generated when data is sent out of a congested interface. FECNs indicate to a Frame Relay device that congestion was encountered along the transmission path to the destination. Traffic is marked with BECN if the queue for the opposite direction is full enough to trigger FECNs at the current time.
- BECNs—These notify the sending Frame Relay device to decrease the transmission rate. If the traffic is one-way only (such as multicast traffic), there is no reverse traffic with BECNs to notify the sending device to slow down. When a Frame Relay device receives a FECN, it first determines whether it is sending data. If the Frame Relay device is sending data along the return path of the FECN, this data will be marked with a BECN on its way to the other Frame Relay device. If the Frame Relay device is not sending data, it can send a Q.922 “TEST RESPONSE” message with the BECN bit set.

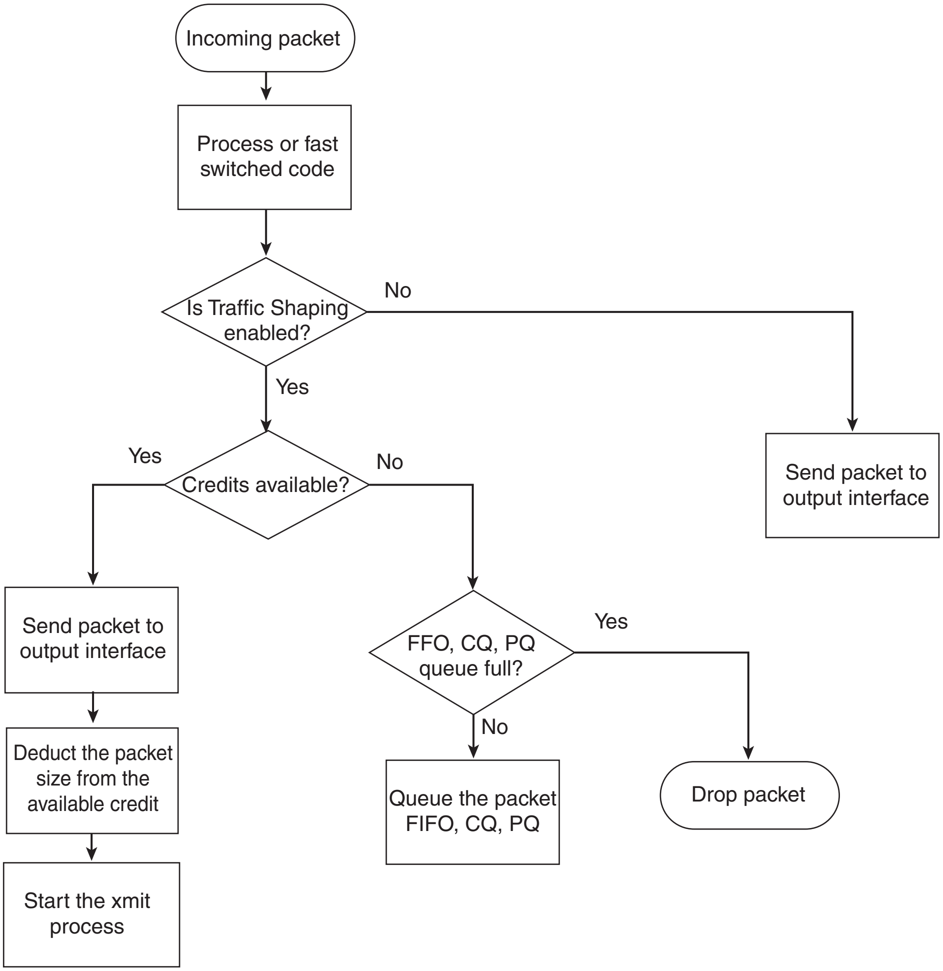
When an interface that is configured with traffic shaping receives a BECN, it immediately decreases, or throttles down, its maximum rate by a significant amount. If, after several intervals, the [throttled] interface has not received another BECN and traffic is waiting in the queue, the maximum rate slightly increases. This dynamically adjusted maximum rate is called the *derived rate*, which will always be between the upper bound and the lower bound that is configured on the interface.

Traffic Shaping Restrictions

FRTS applies only to Frame Relay PVCs and SVCs.

Figure 15-28 represents the traffic shaping process flow upon receipt of a frame for transmission.

Figure 15-28 *Traffic Shaping Flowchart*


Incoming packet
Process or fast switched code
Is Traffic Shaping enabled?
No
Yes
Send packet to output interface
Yes
Credits available?
No
Send packet to output interface
Deduct the packet size from the available credit
Start the xmit process
FFO, CQ, PQ queue full?
Yes
No
Queue the packet FIFO, CQ, PQ
Drop packet

Traffic Policing and Shaping

Cisco IOS QoS offers two types of traffic regulation mechanisms: policing and shaping.

The rate-limiting features of committed access rate (CAR) and the traffic policing feature provide the functionality for policing traffic.

The features of GTS, class-based shaping, DTS, and FRTS provide the functionality for shaping traffic.

These features can be deployed throughout the network to ensure that a frame, packet, or other data source adheres to a stipulated contract. These features can also be used to determine the QoS with which to render the packet. Both policing and shaping mechanisms use the traffic descriptor for a packet—indicated by the classification of the packet—to ensure adherence and service.

Traffic policers and shapers identify traffic descriptor violations in an identical manner. These policers and shapers differ in how they respond to violations, for example:

- A policer drops traffic. For example, the CAR rate-limiting policer will either drop the packet or rewrite its IP precedence, resetting the type of service bits in the packet header.
- A shaper delays excess traffic using a buffer, or queuing mechanism, to hold packets and shape the flow when the data rate of the source is higher than expected. For example, GTS and class-based shaping use a weighted fair queue to delay packets to shape the flow. DTS and FRTS use either a priority queue, a custom queue, or a FIFO queue for the same, depending on how the queue is configured.

Traffic shaping and policing can work in tandem. For example, a good traffic-shaping scheme should make it easy for nodes that are inside the network to detect misbehaving flows. This activity is sometimes called "policing the traffic of the flow."

Because policing and shaping each use the token bucket mechanism, token buckets will be discussed in the next section.

Token Bucket

A token bucket is a formal definition of a rate of transfer with three components: burst size, mean rate, and a time interval (T_C). The mean rate is generally represented as bits per second, and any two values can be derived from the third, as shown by the following formula:

mean rate = burst size / time interval

where

- Mean rate—Also called the CIR. The mean rate specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the committed burst (B_c) size. The burst size specifies in bits (or bytes) per burst how much traffic can be sent within a given unit of time without creating scheduling concerns. For a shaper, such as GTS, burst size specifies bits per burst; for a policer, such as CAR, burst size specifies bytes per burst.
- Time interval (TC)—Also called the measurement interval. The time interval specifies the time in seconds per burst. By definition, over any integral multiple of the interval, the bit rate of the interface will not exceed the mean rate. The bit rate, however, might be arbitrarily fast within the interval.

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic policer, such as CAR, or a traffic shaper, such as FRTS or GTS. A token bucket has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator. (CAR, FRTS, and GTS do not implement either a true token bucket or a true leaky bucket.)

In the token bucket metaphor, the following occurs:

- Tokens are put into the bucket at a certain rate. The bucket has a specified capacity.
- If the bucket fills to capacity, newly arriving tokens are discarded.
- Each token has permission for the source to send a certain number of bits into the network.
- To send a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.
- If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens (in the case of GTS), or the packet is discarded or marked down (in the case of CAR).
- If the bucket is already full of tokens, incoming tokens overflow and are not available to future packets. Thus, at any time, the largest burst that a source can send into the network is roughly proportional to the size of the bucket.

The token bucket mechanism used for traffic shaping has both a token bucket and a data buffer, or queue. If the token bucket mechanism for traffic shaping did not have a data buffer, it would be a traffic policer.

The following applies for traffic shaping:

- Packets that arrive that cannot be sent immediately are delayed in the data buffer.
- A token bucket permits burstiness, but also bounds it.
- Traffic shaping guarantees that the burstiness is bounded so that the flow will never send more quickly than the capacity of the token bucket plus the time interval multiplied by the established rate at which tokens are placed in the bucket.
- Traffic shaping guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.

Traffic Policing with CAR

CAR is a rate-limiting feature for policing traffic, in addition to its packet classification feature. The rate-limiting feature of CAR manages the access bandwidth policy for a network by ensuring that traffic that falls within specified rate parameters is sent, while dropping packets that exceed the acceptable amount of traffic or sending them with a different priority. CAR's exceed action is to either drop or mark down the packet's priority.

The CAR rate-limiting function performs the following:

- Controls the maximum rate of traffic sent or received on an interface.
- Defines Layer 3 aggregate or granular incoming or outgoing (ingress or egress) bandwidth rate limits and specifies traffic-handling policies when the traffic either conforms to or exceeds the specified rate limits.

CAR bandwidth rate limits perform one of two functions:

- Aggregate—Aggregate bandwidth rate limits match all of the packets on an interface or subinterface.
- Granular—Granular bandwidth rate limits match a particular type of traffic based on precedence, MAC address, or other parameters.

CAR is often configured on interfaces at the edge of a network to limit traffic into or out of a network.

CAR Operation

CAR examines traffic received on an interface or a subset of that traffic selected by access list criteria. CAR compares the rate of the traffic to a configured token bucket and takes action based on the result. For example, CAR will drop the packet or rewrite the IP precedence by resetting the type of service (ToS) bits.

CAR can be configured to send, drop, or set precedence.

Aspects of CAR rate limiting include the following:

- Matching criteria
- Rate limits
- Conform and exceed actions
- Multiple rate policies

CAR utilizes a token bucket measurement, operating as follows:

- Tokens are inserted into the bucket at the committed rate.
- The depth of the bucket is the burst size.
- Traffic arriving at the bucket when sufficient tokens are available is said to conform, and the corresponding number of tokens is removed from the bucket.
- If a sufficient number of tokens is not available, then the traffic is said to exceed.

CAR Traffic-Matching Criteria Traffic matching involves the identification of interesting traffic for rate limiting, precedence setting, or both. Rate policies can be associated with one of the following qualities:

- Incoming interface
- All IP traffic
- IP precedence (defined by a rate-limit access list)
- MAC address (defined by a rate-limit access list)
- IP access list (standard and extended)

CAR provides configurable actions, such as send, drop, or set precedence when traffic conforms to or exceeds the rate limit.

NOTE Matching to IP access lists is more processor intensive than matching based on other criteria.

Rate Limits CAR propagates bursts and performs no smoothing or shaping of traffic; therefore, it performs no buffering and adds no delay. CAR is optimized (but not limited) to run on high-speed links, such as DS3 or higher, and in distributed mode on Versatile Interface Processors (VIPs) on the Cisco 7500 series.

CAR rate limits can be implemented either on input or output interfaces or subinterfaces, including those found with Frame Relay and ATM implementations.

Rate limits define which packets conform to or exceed the defined rate based on the following three parameters:

- Average rate—Determines the long-term average transmission rate. Traffic that falls under this rate will always conform.
- Normal burst size—Determines how large traffic bursts can be before some traffic exceeds the rate limit.
- Excess Burst size (BE)—Determines how large traffic bursts can be before all traffic exceeds the rate limit. Traffic that falls between the normal burst size and the Excess Burst size exceeds the rate limit with a probability that increases as the burst size increases.

The tokens in a token bucket are replenished at regular intervals, in accordance with the configured committed rate. The maximum number of tokens that a bucket can contain is determined by the token bucket's normal burst size configuration.

When the CAR rate limit is applied to a packet, CAR removes from the bucket tokens that are equivalent in number to the byte size of the packet. If a packet arrives and the byte size of the packet is greater than the number of tokens available in the token bucket, extended burst capability is engaged (if it is configured).

Setting the extended burst value greater than the normal burst value configures extended burst. Setting the extended burst value equal to the normal burst value excludes the extended burst capability. If extended burst is not configured, the CAR exceed action takes effect because a sufficient number of tokens are not available.

When extended burst is configured, the flow is allowed to borrow the needed tokens to allow the packet to be sent. This capability exists to avoid tail-drop behavior, and, instead, engage behavior like that of random early detection (RED).

Extended burst operates in the following fashion:

- If a packet arrives and needs to borrow *n* number of tokens because the token bucket contains fewer tokens than its packet size requires, then CAR compares the following two values:

  — Extended burst parameter value

  — Compounded debt, which is computed as the sum over all ai:

  a indicates the actual debt value of the flow after packet i is sent. Actual debt is simply a count of how many tokens the flow has currently borrowed.

  i indicates the packet that attempts to borrow tokens since the last time a packet was dropped.

- If the compounded debt is greater than the extended burst value, the exceed action of CAR takes effect. After a packet is dropped, the compounded debt is effectively set to 0. CAR computes a new compounded debt value equal to the actual debt for the next packet that needs to borrow tokens.
- If the actual debt is greater than the extended limit, all packets are dropped until the actual debt is reduced through accumulation of tokens in the token bucket.

Dropped packets do not count against a rate or burst limit. In other words, when a packet is dropped, no tokens are removed from the token bucket.

NOTE Although the entire compounded debt is forgiven when a packet is dropped, the actual debt is not forgiven, and the next packet to arrive to insufficient tokens is immediately assigned a new compounded debt value equal to the current actual debt. In this way, actual debt can continue to grow until it is so large that no compounding is needed to cause a packet to be dropped. In effect, the compounded debt is not really forgiven. This scenario leads to excessive drops on streams that continually exceed normal burst.

Cisco recommends the following values for the normal and extended burst parameters:

normal burst = configured rate × (1 byte)/(8 bits) × 1.5 seconds

extended burst = 2 × normal burst

Now look at an example that shows how the compounded debt is forgiven, but the actual debt accumulates.

In this example, the following parameters are assumed:

- Token rate is 1 data unit per time unit
- Normal burst size is 2 data units
- Extended burst size is 4 data units
- 2 data units arrive per time unit

After two time units, the stream has used up its normal burst and must begin borrowing one data unit per time unit, beginning at time unit 3:

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
1       2               0               0
2       2               0               0
3       2               1               1
4       2               2               3
5       2               3 (temporary)   6 (temporary)
```

The following actions occur at this time:

- A packet is dropped because the new compounded debt (6) would exceed the extended burst limit (4).
- When the packet is dropped, the compounded debt effectively becomes 0, and the actual debt is 2. (The values 3 and 6 were only temporary and do not remain valid if a packet is dropped.)
- The final values for time unit 5 follow. The stream begins borrowing again at time unit 6.

```
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
5       2               2               0
6       2               3               3
7       2               4 (temporary)   7 (temporary)

At time unit 6, another packet is dropped and the debt values are
 adjusted accordingly.
Time    DU arrivals     Actual Debt     Compounded Debt
-------------------------------------------------------
7       2               3               0
```

Conform and Exceed Actions Because CAR utilizes a token bucket, CAR can pass temporary bursts that exceed the rate limit as long as tokens are available.

After a packet has been classified as conforming to or exceeding a particular rate limit, the router performs one of the following actions:

- Transmit—The packet is sent.
- Drop—The packet is discarded.
- Set precedence and transmit—The IP precedence (ToS) bits in the packet header are rewritten. The packet is then sent. You can use this action to either color (set precedence) or recolor (modify existing packet precedence) the packet.
- Continue—The packet is evaluated using the next rate policy in a chain of rate limits. If another rate policy does not exist, the packet is sent.
- Set precedence and continue—Set the IP precedence bits to a specified value and then evaluate the next rate policy in the chain of rate limits.

For VIP-based platforms, two more actions are possible:

- Set QoS group and transmit—The packet is assigned to a QoS group and sent.
- Set QoS group and continue—The packet is assigned to a QoS group and then evaluated using the next rate policy. If another rate policy does not exist, the packet is sent.

Multiple Rate Policies A single CAR rate policy includes information about the rate limit, conform actions, and exceed actions. Each interface can have multiple CAR rate policies corresponding to different types of traffic. For example, low-priority traffic might be limited to a lower rate than high-priority traffic. When multiple rate policies exist, the router examines each policy in the order entered until the packet matches. If no match is found, the default action is to send.

Rate policies can be independent or cascading:

- Independent—Each rate policy deals with a different type of traffic.
- Cascading—A packet might be compared to multiple different rate policies in succession.

Cascading of rate policies supports a series of rate limits to be applied to packets to specify more granular policies. For example, total traffic could be rate limited on an access link to a specified subrate bandwidth and then rate limit World Wide Web traffic on the same link to a given proportion of the subrate limit.

Match packets could be rate limited against an ordered sequence of policies until an applicable rate limit is encountered. For example, as the sequence of policies is applied, the rate limiting of several MAC addresses with different bandwidth allocations can occur at an exchange point.

Up to 100 rate policies can be configured on a subinterface.

CAR Restrictions

CAR and VIP-distributed CAR can only be used with IP traffic. Non-IP traffic is not rate limited.

CAR or VIP-distributed CAR can be configured on an interface or subinterface, with the exception of the following interface types:

- Fast EtherChannel
- Tunnel
- PRI
- Any interface that does not support Cisco Express Forwarding (CEF)

CAR is supported only on ATM subinterfaces with the following encapsulations: aal5snap, aal5mux, and aal5nlpid.

NOTE CAR provides rate limiting and does not guarantee bandwidth. CAR should be used with other QoS features, such as distributed weighted fair queuing (DWFQ), if premium bandwidth assurances are required.

Summary

Frame Relay is a Layer 2 (data link) wide-area network (WAN) protocol that works at both Layer 1 (physical) and Layer 2 (data link) of the OSI model. Although Frame Relay services were initially designed to operate over ISDN service, the more common deployment today involves dedicated access to WAN resources.

Frame Relay networks are typically deployed as a cost-effective replacement for point-to-point private line, or leased line, services. Whereas point-to-point customers incur a monthly fee for local access and long-haul connections, Frame Relay customers incur the same monthly fee for local access, but only a fraction of the long-haul connection fee associated with point-to-point private line services.

Frame Relay was standardized by two standards bodies—internationally by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) and domestically by ANSI (American National Standards Institute).

Frame Relay is a packet-switched technology, meaning that each network end user, or end node, will share backbone network resources, such as bandwidth. Connectivity between these end nodes is accomplished with the use of Frame Relay virtual circuits (VCs).

Frame Relay WAN service primarily comprises four functional components:

- Customer premise Frame Relay access device (FRAD).
- Local access loop to the service provider network.
- Frame Relay switch access port. Link Management Interface parameters are defined here.
- Frame Relay VC parameters to each end site.

Frame Relay is a connection-oriented service, operating the data link layer (Layer 2) of the OSI model. A data-link connection identifier (DLCI) is used to identify this dedicated communication path between two end nodes. This path, or VC, is a bidirectional logical connection across the WAN between two end node DTE devices.

DLCIs are of local significance, unless an agreement has been made with the network service provider to deploy global DLCIs. Local significance means that DLCIs are of use only to the local Frame Relay network device. Frame Relay DLCIs are analogous to an organization's telephone network utilizing speed-dial functions.

Two types of Frame Relay VCs exist:

- Permanent virtual circuits (PVCs)—These are permanently established, requiring no call setup, and utilize DLCIs for endpoint addressing.
- Switched virtual circuits (SVCs)—These are established as needed, requiring call setup procedures and utilizing X.121 or E.164 addresses for endpoint addressing.

Two types of congestion-notification mechanisms are implemented with Frame Relay:

- Forward explicit congestion notification (FECN)—The FECN bit is set by a Frame Relay network to inform the Frame Relay networking device receiving the frame that congestion was experienced in the path from origination to destination. Frame relay network devices that receive frames with the FECN bit will act as directed by the upper-layer protocols in operation. The upper-layer protocols will initiate flow-control operations, depending on which upper-layer protocols are implemented. This flow-control action is typically the throttling back of data transmission, although some implementations can be designated to ignore the FECN bit and take no action.
- Backward explicit congestion notification (BECN)—Much like the FECN bit, the BECN bit is set by a Frame Relay network to inform the DTE that is receiving the frame that congestion was experienced in the path traveling in the opposite direction of frames. The upper-layer protocols will initiate flow-control operations, depending on which upper-layer protocols are implemented. This flow-control action is typically the throttling back of data transmission, although some implementations can be designated to ignore the BECN bit and take no action.

Frame Relay VCs, both permanent and switched, have three configurable parameters that must be agreed upon between each end node and the Frame Relay network provider:

- Committed information rate (CIR)—This is the amount of bandwidth that will be delivered as "best-effort" across the Frame Relay backbone network.
- Discard eligibility (DE)—This is a bit in the frame header that indicates whether that frame can be discarded if congestion is encountered during transmission.
- Virtual circuit identifier
    - — Data-link connection identifiers (DLCIs) for PVCs—Although DLCI values can be 10, 16, or 23 bits in length, 10-bit DLCIs have become the de facto standard for Frame Relay WAN implementations.
    - — X.121/E.164 addressing for SVCs—X.121 is a hierarchical addressing scheme that was originally designed to number X.25 DTEs. E.164 is a hierarchical global telecommunications numbering plan, similar to the North American Number Plan (NANP, 1-NPA-Nxx-xxxx).

The formulae in Table 15-17 can be used to determine the number of VCs required to enable each associated network topology.

Table 15-17 *Summary of Network Topology Formulae*

Network Topology	**Formula**[*]
Fully meshed	$[(N \times (N-1)) / 2]$
Partial-mesh	(Approximation) $[N^2 / \sqrt{(N-1)}]$ (Guideline) $[((N \times (N-1)) / 2) \geq X \geq (N-1)]$
Hub-and-Spoke	$[N-1]$

[*] Note: N is the number of locations.

Frame Relay uses the cyclic redundancy check (CRC) method for error detection. Frame Relay has no inherent error correction mechanisms, leaving error correction to the management and control of the upper-layer protocols.

Local Management Interface (LMI) is a set of enhancements to the basic Frame Relay specification. LMI includes support for keepalive mechanisms, verifying the flow of data; multicast mechanisms, providing the network server with local and multicast DLCI information; global addressing, giving DLCIs global rather than local significance; and status mechanisms, providing ongoing status reports on the switch-known DLCIs.

Three types of LMI are found in Frame Relay network implementations:

- ANSI T1.617 (Annex D)—The maximum number of connections (PVCs) supported is limited to 976. LMI type ANSI T1.627 (Annex D) uses DLCI 0 to carry local (link) management information.

- ITU-T Q.933 (Annex A)—Like LMI type Annex-D, the maximum number of connections (PVCs) supported is limited to 976. LMI type ITU-T Q.933 (Annex A) also uses DLCI 0 to carry local (link) management information.
- LMI (Original)—The maximum number of connections (PVCs) supported is limited to 992. LMI type LMI uses DLCI 1023 to carry local (link) management information.

Frame Relay is a versatile transport mechanism, traditionally supporting four networking applications:

- TCP/IP Suite
- Novell IPX Suite
- IBM SNA Suite
- Voice over Frame Relay (VoFr)

Internet Protocol (IP) is a best-effort delivery protocol, relying on the transmission-control mechanisms (packet acknowledgement and sequencing) that are supported by TCP. IP datagrams, or *packets*, are routed from source to destination based on the address information found in the packet header. IP traffic is typically bursty in nature, making it an ideal network-layer protocol for Frame Relay WANs.

Novell IPX implementations over Frame Relay are similar to IP network implementation. Whereas a TCP/IP implementation would require the mapping of Layer 3 IP addresses to a DLCI, Novell IPX implementations require the mapping of the Layer 3 IPX addresses to a DLCI. Special consideration needs to be made with IPX over Frame Relay implementations regarding the impact of Novell RIP and SAP message traffic to a Frame Relay internetwork.

Migration of a legacy SNA network from a point-to-point infrastructure to a more economical and manageable Frame Relay infrastructure is attractive; however, some challenges exist when SNA traffic is sent across Frame Relay connections. IBM SNA was designed to operate across reliable communication links that support predictable response times. The challenge that arises with Frame Relay network implementations is that Frame Relay service tends to have unpredictable and variable response times, for which SNA was not designed to interoperate or able to manage within its traditional design.

Voice over Frame Relay (VoFr) has recently enjoyed the general acceptance of any efficient and cost-effective technology. In the traditional plain old telephone service (POTS) network, a conventional (with no compression) voice call is encoded, as defined by the ITU pulse code modulation (PCM) standard, and utilizes 64 kbps of bandwidth. Several compression methods have been developed and deployed that reduce the bandwidth required by a voice call to as little as 4 kbps, thereby allowing more voice calls to be carried over a single Frame Relay serial interface (or subinterface PVC).