Design Implementation Guide

Frame Relay

by Adrien Fournier
Technical Marketing: WAN PME

Introduction

Frame Relay networks are growing at an unprecedented rate and are quickly becoming the wide-area networking solution of choice in the Fortune 1000 marketplace and beyond. This document discusses Frame Relay technology, protocol overhead associated with Frame Relay networks, memory usage requirements by platform, and features associated with a Frame Relay network. A companion Excel spreadsheet can be used to determine total overhead for your network based on routing, routed, and bridged protocols. (See Reference 1.)

The use of Frame Relay technology in today's networks varies significantly as to how it is deployed. Many players in this arena offer a multitude of solutions. The focus in this document is primarily on router based Frame Relay networks.

Overview

Frame Relay provides a packet switching data communications capability that is used across the interface between user devices (for example, routers, bridges, and host machines) and network equipment (for example, switching nodes) and operates at Layer 2 of the seven-layer OSI model. The network providing the Frame Relay interface can be either a carrier-provided public network or a network of privately owned equipment serving a single enterprise.

As an interface to a network, Frame Relay protocol is similar to X.25. However, Frame Relay differs significantly from X.25 in its functionality and format. In particular, Frame Relay is a more streamlined protocol, facilitating higher performance and greater efficiency. With the introduction of digital transmission technology, far less error detection and correction was required compared to that of analog technology commonly seen in X.25 networks. The elimination of this redundant error control at this layer and allowing the layers above to handle error detection and retransmission allows greater performance and efficiency to be realized with Frame Relay.

In comparison to a dedicated, leased-line network, the main attraction for migrating to Frame Relay is cost, in terms of both recurring costs and connectivity requirements such as numerous physical ports and channel service units/data service units (CSUs/DSUs). In a competitive market, the delta in cost savings is typically higher.

As an interface between user and network equipment, Frame Relay provides a means for statistically multiplexing many logical data conversations (referred to as virtual circuits [VCs]) over a single physical transmission link. This setup contrasts with systems that use only time-division multiplexing (TDM) techniques for supporting multiple datastreams. Frame Relay's statistical multiplexing provides more flexible and efficient use of available bandwidth.

Basics

Frame Relay Terms

Many terms apply to Frame Relay networking; the following terms are used in this paper:

Link access rate: The clock speed of the connection (local loop) to the Frame Relay network. The clock can be provided by the CSU/DSU or the Frame Relay switch itself. The link access rate is the rate at which data travels into or out of the network, regardless of other settings.

Data-link connection identifier (DLCI): Contained in every Frame Relay header to identify the logical circuit between the customer premises equipment (CPE) and the Frame Relay switch. The switch then maps (through configuration) the DLCIs at both ends to create a permanent virtual circuit (PVC). Each PVC has configuration parameters that usually include committed information rate (CIR), Committed Burst (B_c), and Excess Burst (B_e). They are defined as follows:

CIR: The rate in bits per second that the Frame Relay switch agrees to transfer data. This rate is usually averaged over a minimum increment of the committed rate measurement interval (T_c). The CIR is configurable on the Frame Relay switch on an individual PVC basis, and can be any value between zero and the actual link speed. Service providers' options differ.

B_c: The maximum number of bits that the switch agrees to transfer during any T_c. Typical values of T_c are 0.5 to 2 seconds. The formula for calculating B_c follows (assume a CIR of 32 kbps and a T_c of 2 seconds):

T_c = $\mathbf{B}_c$ / CIR ($\mathbf{B}_c$ is typically a multiple of CIR)
or
$\mathbf{B}_c$ = T_c x CIR
64 kbits = 2 sec x 32 kbps

The significance of this result is that the higher the ratio of B_c to CIR, the longer the switch can handle a sustained burst, which means that larger input buffer resources are needed. Although latency may grow as more frames are buffered, in most cases it is not a problem. The very nature of Frame Relay technology allows for buffering elasticity for handling differing link access rates, as well as being able to handle bursts of data. Vendors' implementations differ, however, as do service providers' options.

B_e: The maximum number of uncommitted bits that the switch attempts to transfer beyond the CIR. B_e is limited by the link access rate. The formula for B_e follows:

([B_c + B_e]/ B_c) x CIR   Link Access Rate
([64+64]/64)x 32) = 64 kbps (This configuration would not work for a link access rate of 56 kbps)

The percentage of successfully transmitted frames in the B_e region is entirely dependent on the provisioning and configuration of the Frame Relay network. Obviously, if the switches are heavily configured or the switch-to-switch network bandwidth is limited, congestion is more likely, with the resultant higher probability of dropped frames.

Discard eligible (DE): This bit in the Frame Relay header (see Figure 1), when set, indicates that the frame is eligible for discard in the event of network congestion. The bit is set when a frame entering the switch is determined to be in excess of the CIR but less than the excess burst limit. If the B_e limit is exceeded, the frame should be dropped in normal operation. The option to not enforce this rule is available in some Frame Relay switches and may be useful where ingress and egress access speeds are equal or where full bandwidth could be used when available. This scenario would mostly apply to private Frame Relay networks.

The DE bit can also be set before entering the Frame Relay switch by CPE equipment such as a router. The Cisco Internetwork Operating System (Cisco IOS™) software allows the setting of the DE bit for packets classified through standard access lists.

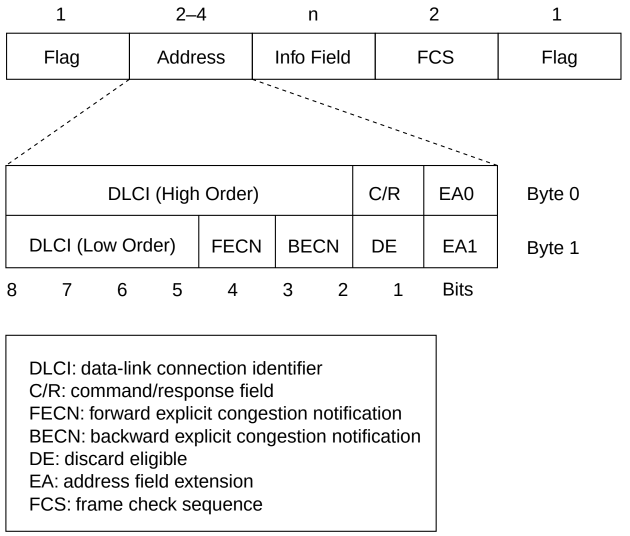
Forward explicit congestion notification (FECN): This bit in the Frame Relay header (see Figure 1) is set by the network Frame Relay switch to notify the end station receiving the frame that the frame has been delivered through a congested path of the network. How the destination end station or intervening router Frame Relay access drive (FRAD) reacts to the notification is of no consequence to the switch. Its sole function is to notify. It's up to the destination end station or intervening router/FRAD to take action if any is required.

Backward explicit congestion notification (BECN): This bit in the Frame Relay header (see Figure 1) is set by the network Frame Relay switch to notify the source (sending) station that congestion exists in the path it is transmitting into. The source end station or intervening router/FRAD should take immediate action to *reduce* data being sent into the network while the congestion condition continues. (See "Traffic Shaping" feature in "Key Features" section.)

To put some of these terms into perspective, consider the following. Data transmitted across the physical link into the Frame Relay switch is transferred at the link access rate. The switch counts the incoming bits on a per-VC basis as B_c bits within time interval T_c. Any bits arriving in excess of the B_c limit are counted as B_e bits, and the frame containing these bits will have the DE bit set. These frames are forwarded if there is no congestion detected in the network. Once the excess burst limit is exceeded, the switch discards new incoming frames. Of course, as the frames on the input queue within the committed burst region are forwarded, the B_c bit counter decreases, allowing room for more committed data. This scheme is sometimes referred to as a "Leaky Bucket" algorithm, which is well suited for handling bursty traffic.

Frame Format

Figure 1 Frame Relay Frame Format (ANSI T1.618)

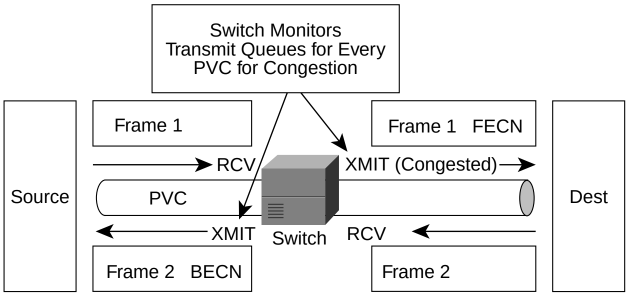


DLCI: data-link connection identifier
C/R: command/response field
FECN: forward explicit congestion notification
BECN: backward explicit congestion notification
DE: discard eligible
EA: address field extension
FCS: frame check sequence

Congestion

Congestion is monitored by the Frame Relay switch(es) in the network. One of the ways monitoring is done is by examining the queue depths of the transmit queues on an individual VC basis (see Figure 2). Once the average queue size exceeds its optimal threshold over some predefined period of time, frames entering this queue will have the FECN bit set. This scenario will continue until the queue depth falls below the suboptimal threshold. Remember, this is only a notification. Ideally, the destination end station’s protocol stack receiving this notification could initiate a throttling mechanism such as a window size reduction signal back to the source end station. Regardless of the existence of this capability, when congestion persists, the Frame Relay switch notifies the source end station or intervening router/FRAD by setting the BECN bit on frames going back to that source.

Figure 2 Congestion on Transmit Queue Causes FECN to be Set on Frames to Destination, and Eventually BECN To Be Set on Frames Back to Source



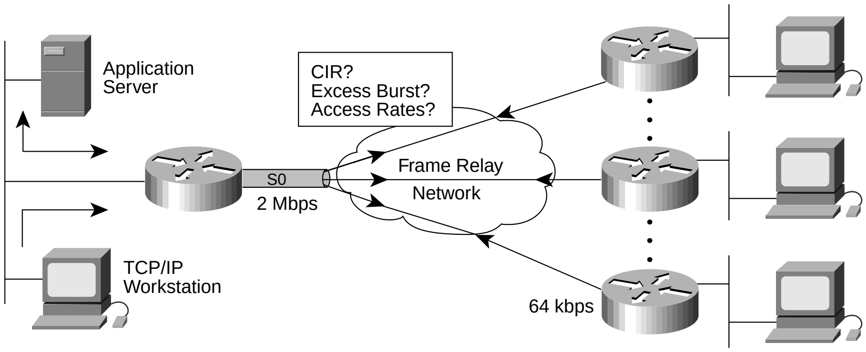
Managing congestion in a Frame Relay network is a cooperative effort between switch vendors, router/FRAD vendors, and end-station protocols. Pushing the capacity limits of Frame Relay networks will certainly require this cooperative effort to produce effective congestion management, which in turn will reduce the cost of running the network. (See “Traffic Shaping” feature in “Key Features” section.)

Planning

Frame Relay Subscription

Most deployed Frame Relay networks are star (or hub and spoke) topologies (see Figure 3), fitting the model of host-based application or central server access. When provisioning Frame Relay services for a topology of this kind, care must be taken to provide adequate bandwidth and resources to handle all the traffic needs of the organization as well as any protocol overhead. Oversubscription in these environments is quite common and, if not carefully planned, can lead to adverse network performance, including poor interactive response times. The underlying problem associated with this poor performance can mostly be attributed to underestimating the actual volume of traffic and traffic patterns that exist on the network. Successful oversubscribed services contain enough bandwidth and buffering capacity to handle the traffic. Using the star (or hub and spoke) topology, let’s examine the subscription issues associated with deploying a simple network.

Figure 3 Star Topology Using a 2-Mbps Link Access Rate at the Central Site with Remote Sites Using a 64-kbps Link Access Rate



The central (hub) site physical link connection typically has link speeds of fractional T1/E1, or T1/E1 (56 kbps to 2.0 Mbps), while common access (spoke) link speeds are 56/64 kbps. This area is where the question of over/undersubscription comes into play. Let’s consider the following scenarios:

1) Subscribing to physical link access rates:

Aggregate the available bandwidth of all the access (spoke) sites into the central (hub) site under the following conditions:

- Central site link rate of 2 Mbps
- 30 X access sites, each with a link rate of 64 kbps
- If all access sites burst to 100 percent, the central hub site can theoretically handle all traffic

With knowledge of overall bandwidth requirements and sufficient router resources (memory), but little knowledge of what the protocol mix is, and the link overhead used, this level of subscription is safe. Many will argue that it is overkill. A major factor to consider when subscribing at this level is network congestion. If all available bandwidth is to be used, then CIR should be high, if not equal to, access link speed, and B_c should be set to allow for sustained bursts. Of course, the cost of implementing this setup can be quite high. Depending on the service provider, the costs may be close to those of leased-line connectivity.

2) Subscribing to a moderate CIR:

Aggregate the CIRs of all VCs into the central hub site's bandwidth capacity under the following conditions (assume a CIR of 32 kbps on each VC):

- Central site link rate of 2 Mbps
- 60 X access sites, each with link rate of 64 kbps and CIR of 32 kbps
- If all access sites burst over their respective CIRs, congestion will occur at the egress point of the Frame Relay network, in this case the switch interface to the central hub site; frames will be dropped
- If the central hub site bursts beyond all the individual CIRs, congestion occurs at the egress point of the access spoke interfaces of the Frame Relay network and the ingress point of the Frame Relay network is potentially overwhelmed; remember that the central hub connection is a single physical interface multiplexing multiple VCs, in this case 60 PVCs

Of course, this scenario requires that the CIR is chosen correctly based on knowledge of bandwidth requirements for the traffic patterns and protocols used. Choosing a CIR of 32 kbps should mean that the requirement is to have a guaranteed bandwidth available to pass at least 32 kbps worth of data, with the flexibility of bursting above this rate, but not for a sustained period of time. This flexibility is what makes Frame Relay technology suitable for bursty traffic applications. It is unlikely in this scenario that all access sites would burst at the same time, although certain traffic patterns may indeed be occurring, such as broadcasts. (See User Traffic Mix and Patterns.) If bandwidth requirements are very low (that is, small amounts of single protocol traffic and no broadcasts), then the number of access spoke sites that you aggregate into the central hub site can be increased. This situation is oversubscription. The issues now move into the router and how many VCs it can effectively handle.

3) Subscribing with zero CIR:

Some service providers offer a zero CIR provisioning, which typically includes a separate service level agreement (that is, if you are dropping packets consistently, they will do something about it). To offer this kind of provisioning, the service providers design the Frame Relay network with sufficient resources to handle the traffic loads (bandwidth and buffers). The pricing for zero CIR is typically very attractive. The use of the Frame Relay traffic shaping features in Cisco IOS V.11.2 software to control the amount of data flowing into the network can be of great value when provisioning for zero CIR. If frame drops occur, traffic entering the Frame Relay network can be throttled back.

- The same parameter requirements exist here as for scenario 2; the major difference here is that all frames entering the network are beyond B_c; however, the switches are typically configured to not drop frames until there is significant congestion

Consider User Traffic Mix and Patterns

Assume a network in a banking environment, where each branch supports a router that is Frame Relay connected to a central hub router. Each branch supports a LAN segment that connects to branch PCs and a serial link connecting the automated teller machines (ATMs). The applications on the PCs are primarily interactive (query/response) but are also used for file transfer, the latter for uploading the day's business up to central site. Since this uploading is done after hours, interactive response times during the daytime meet or exceed requirements. The file transfers can be quite lengthy, especially on major payroll days. A peak usage also occurs on the ATMs on these days, and the lineups can be long. If the file transfers are running during this peak period, ATM response times will be adversely affected. ATMs have quite low bandwidth demands because the frames generated are (relatively) quite small and infrequent. The other consideration is, What's happening at the central hub site? Many sites in the same time zones are all transferring data in, and are likely exceeding the excess burst rates on all PVCs. Frames are being dropped and retransmissions are frequent.

When subscribing Frame Relay circuits, look for worst-case scenarios and their risks, and plan accordingly. In this example, custom queuing on a per-VC basis (Cisco IOS V.11.2 software) could be configured to prioritize interactive traffic over file transfers. Alternatively, provision separate VCs for interactive traffic, subscribe additional CIR and excess burst on a single VC, or stagger scheduled times of file transfers.

KNOW YOUR TRAFFIC!!!

Router DLCI Capacity

How many DLCIs can one configure per physical interface? How many DLCIs can one configure in a specific router? These two questions are frequently asked. Disappointingly, the answer is, “It depends.” (One of the most frequently used *answers).* We will first review the technical limits, which are beyond the practical maximum. Then we'll look at the factors associated with practical limitations.

Technical Limits

DLCI address space: Approximately 1000 DLCIs can be configured on a single physical link, given a 10-bit address. Because certain DLCIs are reserved (vendor implementation dependent), the maximum is about 1000.

Local Management Interface (LMI) status update: The LMI protocol (ANSI Annex D, and ITU-T standards also) requires that all PVC status reports fit into a single packet and generally limits the number of DLCIs to less than 800, depending on the maximum transmission unit (MTU) size.

```
(MTU - 20)/5 = Max DLCIs (approximate)
(4000 - 20) / 5 = 796 DLCIs, where 20 = Frame Relay and LMI header
```

Default MTU on serial interfaces is 1500 bytes, yielding a maximum of 296 DLCIs per interface.

Note: These numbers vary slightly, depending on LMI type.

Practical Limitations

User Data: How much and what kind of user data is expected to travel across each VC? The answer to this question is necessary to determine initial provisioning of the Frame Relay service. File transfers will quickly consume CIR and excess burst limits. If the requirement is to configure many DLCIs into a physical link, then broadcasts should be eliminated and multiple protocols avoided.

Broadcast traffic: Two implications are evident: one-bandwidth consumption for broadcast traffic and two-broadcast replication in the central hub router. To avoid broadcast traffic:

- Use default or static routes
- Filter service access points (SAPs)

See next section for details on broadcast traffic analysis.

Memory constraints: What else is running on the router? Routing protocols? How big is the routing table? Is there enough memory for all the DLCIs? See “Memory Utilization for Frame Relay Configured Resources” section.

Processor constraints: Avoid the use of processor intensive features such as compression, prioritization, or process switched protocols when maximizing the number of VCs in a router. From Cisco IOS Release 11.0 software and beyond, many of these features have moved to the fast switching path.

Guidelines: Efficiency of Frame Relay software can be compromised when stretching the limits. The *general* guidelines for maximum numbers of DLCIs per interface (as per the Internet Design Guide) are between 10 and 50, depending on the traffic and constraints listed previously. If you choose to go beyond this number, consider the following guidelines and measure as you go. Making guesses as to what you think your traffic is and what it actually is can cost an extra router or two.

1) Determine the user traffic requirements. From this estimate, determine by how much you can safely oversubscribe the central hub link. Consider the 2-Mbps link subscription scenarios discussed previously and do not exceed a 3:1 oversubscription. That's 120 DLCIs, where each VC's normal traffic level seldom exceeds 16 kbps. This scenario precludes the use of file transfer on these VCs. Typical low traffic applications could be ATMs, paging systems, or Simple Network Management Protocol (SNMP) updates. If you don't know what your traffic patterns plus link overhead will be, don't oversubscribe.

2) Eliminate all broadcasts on the Frame Relay link. If you *must* run a routing protocol, carefully plan to use route summarization, and minimize routing table size by advertising, for example, a single default route.

3) Maximum DLCIs per router platform guideline, based on extrapolation from empirical data established on a Cisco 7000 router platform:

- Cisco 2500: 1 X T1/E1 link @ 60 DLCIs per interface = 60 total
- Cisco 4000: 1 X T1/E1 link @ 120 DLCIs per interface = 120 total
- Cisco 4500: 3 X T1/E1 links @ 120 DLCIs per interface = 360 total
- Cisco 4700: 4 X T1/E1 links @ 120 DLCIs per interface = 480 total
- Cisco 7000: 4 X T1/E1/T3/E3 links @ 120 DLCIs per interface = 480 total
- Cisco 7200: 5 X T1/E1/T3/E3 links @ 120 DLCIs per interface = 600 total
- Cisco 7500: 6 X T1/E1/T3/E3 links @ 120 DLCIs per interface = 720 total

Note: These numbers are guidelines only, and assume that all traffic is fast switched.

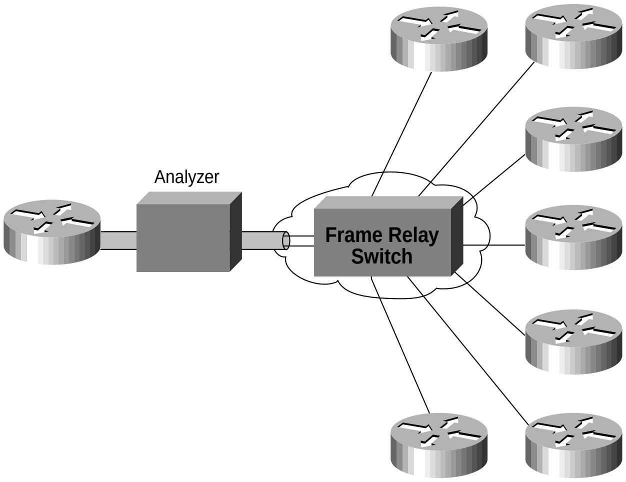
Broadcast Traffic Analysis

A major concern in router-based Frame Relay networks is bandwidth consumption by broadcast traffic because of packet replication on a single physical interface. This consumption is most common at the central site, where concentration of VCs onto physical links is much higher than, say, at the access or distribution sites. To analyze your own interface overhead, use the Excel spreadsheet mentioned in Reference 1.

Using a simple hub and spoke topology (see Figure 4), analysis of WAN traffic was performed on the hub (core) router link that connects to the Frame Relay switch. Multiple protocols were configured on the routers and the resulting overhead data was observed. Analysis of the traffic consisted of broadcasts from:

- Routing protocols
- Routed protocols
- Transparent bridging
- Remote source-route bridging (SRB)

Figure 4   WAN Traffic Analysis Testbed



Routing Updates

Routing updates in a Frame Relay network can have significant impact on performance because routing updates go out on a per-PVC (DLCI) basis. Therefore, when you configure multiple DLCIs on a single WAN Frame Relay interface, routing updates are replicated for each DLCI. Although distance vector routing protocols such as Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP®) are easily predictable in that updates go out at regular intervals, the same is not true of link state protocols such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). Enhanced IGRP, an advanced distance vector routing protocol, also falls in the same behavior category as OSPF. Although Enhanced IGRP is not a link state protocol, routing updates are propagated in a similar way as a link state protocol, that is, flooding occurs when adjacencies are lost. (See “Routing with OSPF” and “Routing with Enhanced IGRP” sections.)

The following subsections give a breakdown on a per-protocol basis of the overhead consumed when these protocols are configured. To calculate overhead for a particular environment, use the Excel spreadsheet referenced at the end of this document.

Routing Information Protocol

RIP updates flow every 30 seconds. Each RIP packet can contain up to 25 route entries, for a total of 536 bytes; 36 bytes of this total are header information, and each route entry is 20 bytes. Therefore, if you advertised 1000 routes over a Frame Relay link configured for 50 DLCIs, you would end up with 1 MB of routing update data every 30 seconds, or 285 kbps of bandwidth consumed. On a T1 link, this bandwidth represents 18.7 percent of the bandwidth, with each update duration being 5.6 seconds. This amount of overhead is considerable, and it is borderline acceptable, but CIR would have to be in the region of the access speed. Obviously, anything less than a T1 would incur too much overhead.

1000/25 = 40 packets X 36 = 1440 header bytes

1000 X 20 bytes = 20,000 bytes of route entries

Total 21,440 bytes X 50 DLCIs = 1072 MB of RIP updates every 30 seconds

1,072,000 bytes / 30 sec X 8 bits = 285 kbps

IGRP

IGRP updates flow every 90 seconds (configurable). Each IGRP packet can contain 104 route entries, for a total of 1492 bytes, 38 of which are header information, and each route entry is 14 bytes. If you advertised 1000 routes over a Frame Relay link configured with 50 DLCIs, you would end up with approximately 720 KB of routing update data every 90 seconds, or 64 kbps of bandwidth consumed. On a T1 link, this bandwidth would represent 4.2 percent of the bandwidth, with each update duration being 3.7 seconds. This overhead is an acceptable amount.

1000/104 = 9 packets X 38 = 342 header bytes

1000 X 14 = 14,000 bytes of route entries

Total = 14,342 bytes X 50 DLCIs = 717 KB of IGRP updates every 90 seconds

717,000 bytes / 90 X 8 bits = 63.7 kbps

Routed Protocols

AppleTalk (Extended)

Routing Table Maintenance Protocol (RTMP) routing updates occur every 10 seconds (configurable). Each RTMP packet can contain up to 94 extended route entries, for a total of 564 bytes, 23 bytes of header information, and each route entry is 6 bytes. If you advertised 1000 AppleTalk networks over a Frame Relay link configured for 50 DLCIs, you would end up with approximately 313 KB of RTMP updates every 10 seconds, or 250 kbps of bandwidth consumed. To remain within an acceptable level of overhead (15 percent or less), a T1 rate would be required.

1000/94 = 11 packets X 23 bytes = 253 header bytes

1000 X 6 = 6000 bytes of route entries

Total = 6253 X 50 DLCIs = 313 KB of RTMP updates every 10 seconds

313,000 / 10 sec X 8 bits = 250 kbps

DECnet

DECnet routing updates occur every 40 seconds (configurable). Each DECnet routing packet can contain up to 368 route entries for a total of 1490 bytes, 25 bytes of header information, and each route entry is 4 bytes. If you advertised 1000 DECnet routes over a Frame Relay link configured for 50 DLCIs, you would end up with 203 KB of DECnet updates every 40 seconds, or 40.8 kbps of bandwidth consumed. To remain within an acceptable level of overhead (15 percent or less), a rate of 256 kbps would be required.

1000/368 = 3 packets X 25 bytes = 75 bytes of header

1000 X 4 = 4000 bytes of route entries

Total = 4075 X 50 DLCIs = 203,750 bytes of DECnet updates every 40 seconds

203,750 / 40 sec X 8 bits = 40.8 kbps

IPX RIP

IPX RIP packet updates occur every 60 seconds (configurable). Each IPX RIP packet can contain up to 50 route entries for a total of 536 bytes, 38 bytes of header information, and each route entry is 8 bytes. If you advertised 1000 IPX routes over a Frame Relay link configured for 50 DLCIs, you would end up with 536 KB of IPX updates every 60 seconds, or 58.4 kbps of bandwidth consumed. To remain within an acceptable level of overhead (15 percent or less), a rate of 512 kbps would be required.

1000/50 = 20 packets X 38 bytes = 760 bytes of header

1000 X 8 = 8000 bytes of route entries

Total = 8760 X 50 DLCIs = 438,000 bytes of IPX updates every 60 seconds

438,000 / 60 sec X 8 bits = 58.4 kbps

Novell's IPX Service Advertisement Protocol

IPX SAP packet updates occur every 60 seconds (configurable). Each IPX SAP packet can contain up to seven advertisement entries for a total of 536 bytes, 38 bytes of header information, and each advertisement entry is 64 bytes. If you broadcast 1000 IPX advertisements over a Frame Relay link configured for 50 DLCIs, you would end up with 536 KB of IPX updates every 60 seconds, or 58.4 kbps of bandwidth consumed. To remain within an acceptable level of overhead (15 percent or less), a rate of greater than 2 Mbps would be required. Obviously, SAP filtering would be required in this scenario. Compared to all other protocols mentioned herein, IPX SAP updates require the most bandwidth.

1000/7 = 143 packets X 38 bytes = 5434 bytes of header

1000 X 64 = 64,000 bytes of route entries

Total = 69,434 X 50 DLCIs = 3,471,700 bytes of IPX service advertisements every 60 seconds

3,471,700 / 60 sec X 8 bits = 462 kbps

VINES

VINES Routing Table Protocol (RTP) packet updates occur every 90 seconds (configurable). Each RTP packet can contain up to 104 route entries for a total of 1492 bytes, 28 bytes of header information, and each route entry is 8 bytes. If you advertised 1000 VINES routes over a Frame Relay link configured for 50 DLCIs, you would end up with 415 KB of VINES updates every 90 seconds. Using the same math as in previous examples gives you 37 kbps of bandwidth consumed. To remain within an acceptable level of overhead (15 percent or less), a rate of 256 kbps would be required.

1000/104 = 10 packets X 28 bytes = 280 bytes of header

1000 X 8 = 8000 bytes of route entries

Total = 8280 X 50 DLCIs = 414,000 bytes of IPX updates every 60 seconds

414,000 / 90 sec X 8 bits = 37 kbps

XNS

XNS RIP packet updates occur every 30 seconds (configurable). Each XNS RIP packet can contain up to 25 route entries for a total of 536 bytes, 46 bytes of header information, and each route entry is 20 bytes. If you advertised 1000 XNS routes over a Frame Relay link configured for 50 DLCIs, you would end up with 1.08 MB of XNS updates every 30 seconds. Using the same math as in previous examples gives you 291 kbps of bandwidth consumed. To remain within an acceptable level of overhead (15 percent or less), a rate of 2 mbps would be required.

1000/25 = 40 packets X 46 = 1840 header bytes

1000 X 20 = 20,000 bytes of route entries

Total = 21,840 X 50 DLCIs = 1,092,000 bytes of IPX updates every 30 seconds

1,092,000 / 30 secs X 8 bits = 291 kbps

Inverse Address Resolution Protocol Traffic at Startup

Address resolution is done on a DLCI basis for each protocol at startup time. It takes three packets to resolve addresses for each protocol on each DLCI. If you configured 50 DLCIs, each carrying three protocols (say IP, IPX, and AppleTalk), then 450 packets would flow, ranging between 30 and 46 bytes each, depending on the protocol address length. Inverse Address Resolution Protocol (ARP) is turned on by default when Frame Relay is configured on a link. The target hardware address used is the Frame Relay address field, which includes the DLCI. This traffic is bursty for a short period of time and could cause minor congestion during startup, say powering everything up on a Monday morning or following a power outage. Once addresses are resolved, no further broadcasts are expected. In the event of an unstable environment where there are line transitions, a repetition of the inverse ARP process will occur.

Bridging

Transparent Bridging Spanning-Tree Protocol

Configuration messages called bridge protocol data units (BPDUs) used in the spanning-tree protocols supported in Cisco bridge/routers flow at regular intervals between bridges and constitute a significant amount of traffic because of their frequent occurrence. There are two types of spanning-tree protocols in transparent bridging. First introduced by the Digital Equipment Corporation (DEC), the algorithm was subsequently revised by the IEEE 802 committee and published in the IEEE 802.1d specification. The DEC Spanning-Tree Protocol issues BPDUs at one-second intervals, while the IEEE issues BPDUs at two-second intervals. Each packet is 41 bytes, which includes a 35-byte configuration BPDU message, a 2-byte Frame Relay header, 2-byte Ethertype, and a 2-byte FCS.

Remote SRB

Remote SRB configured with TCP/IP encapsulation occurs on a router-to-router peer basis. Therefore, in a typical hub and spoke Frame Relay network configuration, the hub router will be peered to each of the spoke site routers. TCP/IP keepalive request/response packets will flow between these peers every 60 seconds.

Data Link Switching

Data-link switching (DLSw) keepalives flow every 30 seconds to remote peers. The keepalive interval is configurable per remote peer on `dlsw remote-peer` configuration command. The overhead is 12 bytes, irrespective of encapsulation used; however, encapsulation of the underlying media will change the actual frame size.

Routing with OSPF

Introduction

Overhead associated with OSPF is not as obvious and predictable as that with traditional distance vector routing protocols. The unpredictability comes from whether or not the OSPF network links are stable. If all adjacencies to a Frame Relay router are stable, only neighbor hello packets (keepalives) will flow, which is comparatively much less overhead than that incurred with a distance vector protocol (RIP, IGRP). If, however, routes (adjacencies) are unstable, link-state flooding will occur, and bandwidth can quickly be consumed. OSPF also is very processor intensive when running the Dijkstra algorithm, used for computing routes. When configuring OSPF in a Frame Relay environment, consider the following subsections, summarized from the *OSPF Design Guide.* (See Reference 2.)

Adjacencies on Nonbroadcast Multiaccess (NBMA)

In earlier releases of Cisco IOS software, special care had to be taken when configuring OSPF over multiaccess nonbroadcast medias such as Frame Relay, X.25, and ATM. The OSPF Protocol considers these media like any other broadcast media such as Ethernet. NBMA clouds are typically built in a hub and spoke topology. PVCs or switched virtual circuits (SVCs) are laid out in a partial mesh and the physical topology does not provide the multiaccess that OSPF believes is out there. For the case of point-to-point serial interfaces, OSPF will always form an adjacency between the neighbors. OSPF adjacencies exchange database information. In order to minimize the amount of information exchanged on a particular segment, OSPF elects one router to be a designated router (DR), and one router to be a backup designated router (BDR) on each multiaccess segment. The BDR is elected as a backup mechanism in case the DR goes down. The idea behind this setup is that routers have a central point of contact for information exchange. The selection of the DR became an issue because the DR and BDR needed to have full physical connectivity with all routers that exist on the cloud. Also, because of the lack of broadcast capabilities, the DR and BDR needed to have a static list of all other routers attached to the cloud. This setup was achieved using the neighbor command:

```
neighbor ip-address [priority number] [poll-interval seconds]
```

In more recent releases different methods can be used to avoid the complications of configuring static neighbors and having specific routers becoming DRs or BDRs on the nonbroadcast cloud. Specifying which method to use is influenced by whether we are starting the network from scratch or rectifying an already existing design.

Point-to-Point Subinterfaces

A subinterface is a logical way of defining an interface. The same physical interface can be split into multiple logical interfaces, with each subinterface being defined as point to point. This scenario was originally created in order to better handle issues caused by split horizon over NBMA and vector based routing protocols. A point-to-point subinterface has the properties of any physical point-to-point interface. As far as OSPF is concerned, an adjacency is always formed over a point-to-point subinterface with no DR or BDR election. OSPF will consider the cloud as a set of point-to-point links rather than one multiaccess network. The only drawback for the point to point is that each segment will belong to a different subnet. This scenario might not be acceptable since some administrators have already assigned one IP subnet for the whole cloud.

IP Unnumbered

Another workaround is to use IP unnumbered interfaces on the cloud. This scenario also might be a problem for some administrators who manage the WAN based on IP addresses of the serial lines.

Selecting Interface Network Types

The command used to set the network type of an OSPF interface follows:

```
ip ospf network {broadcast | non-broadcast | point-to-multipoint}
```

Point-to-Multipoint Interfaces

An OSPF point-to-multipoint interface is defined as a numbered point-to-point interface having one or more neighbors. This concept takes the previously discussed point-to-point concept one step further. Administrators do not have to worry about having multiple subnets for each point-to-point link. The cloud is configured as one subnet. This setup should work well for people who are migrating into the point-to-point concept with no change in IP addressing on the cloud. Also, they would not have to worry about DRs and neighbor statements. OSPF point to multipoint works by exchanging additional link-state updates that contain numerous information elements that describe connectivity to the neighboring routers.

Broadcast Interfaces

This approach is a workaround for using the “neighbor” command, which statically lists all existing neighbors. The interface will be logically set to broadcast and will behave as if the router is connected to a LAN. DR and BDR election will still be performed, so special care should be taken to assure either a full mesh topology or a static selection of the DR based on the interface priority.

Dealing with Large OSPF Networks

Careful planning is required for larger networks in the area of addressing and bandwidth constraints. Use of variable-length subnet masks (VLSMs) and OSPF route summarization can respectively deal with these issues.

VLSM

The idea behind VLSMs is to offer more flexibility in dealing with dividing a major net into multiple subnets and still being able to maintain an adequate number of hosts in each subnet. Without VLSM only one subnet mask can be applied to a major network, restricting the number of hosts given the number of subnets required. If we pick the mask such that we have enough subnets, we wouldn’t be able to allocate enough hosts in each subnet. The same is true for the hosts; a mask that allows enough hosts might not provide enough subnet space. Using VLSM saves on available address space. Refer to the *OSPF Design Guide* (Reference 2) for more information.

Route Summarization

Summarizing is the consolidation of multiple routes into one single advertisement. This summarization is normally done at the boundaries of area border routers (ABRs). Although summarization could be configured between any two areas, it is better to summarize in the direction of the backbone. This way the backbone receives all the aggregate addresses and in turn will inject them, already summarized, into other areas. Clearly, reducing the amount of information contained in route advertisements will make more efficient use of the available bandwidth. Refer to the *OSPF Design Guide* (Reference 2) for more information.

Full Mesh versus Partial Mesh

NBMA clouds such as Frame Relay or X.25 are always a challenge. The combination of low bandwidth and too many link states is a recipe for problems. A partial mesh topology has proven to behave much better than a full mesh. A carefully laid out point-to-point or point-to-multipoint network works much better than multipoint networks that have to deal with DR issues.

Routing with Enhanced IGRP

Introduction

IGRP was significantly enhanced in Cisco IOS software releases 10.3(11), 11.0(8), 11.1(3), and all later releases. The implementation was changed to improve the performance on low speed networks (including Frame Relay) and in configurations with many neighbors. For the most part, the changes are transparent. Most existing configurations should continue to operate as before. However, in order to take advantage of the improvements for low speed links and Frame Relay networks, it is important to properly configure the bandwidth on each interface on which Enhanced IGRP is running. Although the enhanced implementation will interoperate with the earlier version, the full benefits of the enhancements may not be realized until the entire network is upgraded. (See Reference 3.)

Bandwidth Control

The enhanced implementation uses the configured interface bandwidth in order to determine how much Enhanced IGRP data to transmit in a given amount of time. By default, Enhanced IGRP will limit itself to using no more than 50 percent of the available bandwidth. The primary benefit of controlling Enhanced IGRP's bandwidth usage is to avoid losing Enhanced IGRP packets, which could occur when Enhanced IGRP generates data faster than the line can absorb it. This benefit particularly enhances Frame Relay networks, where the access line bandwidth and the PVC capacity may be very different. A secondary benefit is to allow the network administrator to ensure that some bandwidth remains for passing user data, even when Enhanced IGRP is very busy.

Configuration Commands

The amount of bandwidth is controlled by two interface subcommands:

```
bandwidth <nnn>
```

and one of the following:

```
ip bandwidth-percent Enhanced IGRP <AS-number> <ppp>
appletalk Enhanced IGRP-bandwidth-percent <ppp>
ipx bandwidth-percent Enhanced IGRP <AS-number> <ppp>
```

for IP, AppleTalk, and IPX Enhanced IGRP, respectively.

The `bandwidth-percent` command tells Enhanced IGRP what percentage of the configured bandwidth it may use. The default is 50 percent. Since the bandwidth command is also used to set the routing protocol metric, it may be set to a particular value for policy reasons. The `bandwidth-percent` command can have values greater than 100 if the bandwidth is configured artificially low because of such policy reasons.

Configuration Problems

If the bandwidth is configured to be a small value relative to the actual link speed, the enhanced implementation may converge at a slower rate than the earlier implementation. If the value is small enough and there are enough routes in the system, convergence may be so slow that it triggers “stuck in active” detection, which may prevent the network from ever converging. This state is evidenced by repeated messages of the form:

```
%DUAL-3-SIA: Route XXX stuck-in-active state in IP-Enhanced IGRP YY. Cleaning up
```

The workaround for this problem is to raise the value of the “active” timer for Enhanced IGRP by configuring:

```
router Enhanced IGRP
timers active-time
```

The default value in the enhanced code is three minutes; in earlier releases, the default is one minute. This value would need to be raised throughout the network. If the bandwidth is configured to be too high (greater than the actual available bandwidth), the loss of Enhanced IGRP packets may occur. The packets will be retransmitted, but this retransmission may degrade convergence. The convergence in this case will be no slower than the earlier implementation, however.

Configuration Guidelines

The following guidelines give recommendations for configuration on NBMA interfaces. The recommendations are described in terms of configuring the interface bandwidth parameter (with Enhanced IGRP being able to use 50 percent of that bandwidth by default). If the interface bandwidth configuration cannot be changed because of routing policy considerations, or for any other reason, the `bandwidth-percent` command should be used to control the Enhanced IGRP bandwidth. On low speed interfaces, raising the available bandwidth for Enhanced IGRP above the default of 50 percent is advisable in order to improve convergence.

NBMA Interfaces (Frame Relay, X.25, ATM)

It is particularly critical to configure NBMA interfaces correctly, because otherwise many Enhanced IGRP packets may be lost in the packet-switched network. There are three basic rules:

- The traffic that Enhanced IGRP is allowed to send on a single VC cannot exceed the capacity of that VC
- The total Enhanced IGRP traffic for all VCs cannot exceed the access line speed of the interface
- The bandwidth allowed for Enhanced IGRP on each VC must be the same in each direction

The scenarios for NBMA interfaces include:

- Pure multipoint configuration (no subinterfaces)
- Pure point-to-point configuration (each VC on a separate subinterface)
- Hybrid configuration (point-to-point and multipoint subinterfaces)

Each is examined separately.

Pure Multipoint Configuration (No Subinterfaces)

In this configuration Enhanced IGRP will divide the configured bandwidth evenly across each VC. You must ensure that this will not overload each VC. For example, if you have a T1 access line with four 56k VCs, you should configure the bandwidth to be 224 kbps (4 x 56k) in order to avoid dropping packets. If the total bandwidth of the VCs equals or exceeds the access line speed, configure the bandwidth to equal the access line speed. Note that if the VCs are of different capacities, the bandwidth must be set to account for the lowest capacity VC.

For instance, if a T1 access line has three 256k VCs and one 56k VC, the bandwidth should be set to 224 kbps (4 x 56k). In such configurations, putting at least the slow VC onto a point-to-point subinterface is strongly recommended (so that the bandwidth can be raised on the others).

Pure Point-to-Point Configuration (Each VC on a Separate Subinterface)

This configuration allows maximum control, since the bandwidth can be configured separately on each subinterface, and is the best configuration if the VCs have different capacities. Each subinterface bandwidth should be configured to be no greater than the available bandwidth on the associated VC, and the total bandwidth for all subinterfaces cannot exceed the available access line bandwidth. If the interface is oversubscribed, the access line bandwidth must be divided across each of the subinterfaces. For instance, if a T1 access line (1544 kbps) has ten VCs with a capacity of 256 kbps, the bandwidth on each subinterface should be configured to be 154 kbps (1544/10).

Hybrid Configuration (Point-to-Point and Multipoint Subinterfaces)

Hybrid configurations should use combinations of the two individual strategies, while ensuring that the three basic rules are followed.

Oversubscribed Hub and Spoke Frame Relay Configuration (Subinterfaces)

A fairly common configuration in networks with light amounts of transaction traffic is a hub and spoke configuration on which the access line to the hub is oversubscribed (since there is not usually enough data traffic to cause this oversubscription to be a problem). In this scenario, assume a 256-kbps access line to the hub, with 56-kbps access lines to each of ten spoke sites. IP Enhanced IGRP process 123 is configured.

Because a maximum of 256 kbps are available, we cannot allow any individual PVC to handle more than 25 kbps (256/10). Since this data rate is fairly low, and we don't expect very much user data traffic, we can allow Enhanced IGRP to use up to 90 percent of the bandwidth.

The hub configuration would look like:

```
interface Serial 0
encapsulation frame-relay
interface Serial 0.1 point-to-point
bandwidth 25
ip bandwidth-percent Enhanced IGRP 123 90
interface Serial 0.2 point-to-point
bandwidth 25
ip bandwidth-percent Enhanced IGRP 123 90
...
```

Each spoke router must be configured to limit Enhanced IGRP traffic to the same rate as that of the hub in order to satisfy the third rule given previously. The spoke configuration would look like:

```
interface Serial 0
encapsulation frame-relay
interface Serial 0.1 point-to-point
bandwidth 25
ip bandwidth-percent Enhanced IGRP 123 90
```

Note that Enhanced IGRP will not use more than 22.5 kbps (90 percent of 25k) on this interface, even though its capacity is 56 kbps. This configuration will not affect user data capacity, which will still be able to use the entire 56 kbps.

Alternatively, if it is desired to set the interface bandwidth to reflect the PVC capacity, you can adjust the bandwidth percentage for Enhanced IGRP. In this example, the desired bandwidth for Enhanced IGRP is (256k/10) x 0.9 = 23.04k; the bandwidth percentage would be 23.04k/56k = 0.41 (41 percent). So the same effect would be accomplished by configuring:

```
interface Serial 0.1 point-to-point
bandwidth 56
ip bandwidth-percent Enhanced IGRP 123 41
```

Memory Utilization for Frame Relay Configured Resources

Memory consumption for Frame Relay resources occurs in three areas:

- Each DLCI: 216 bytes
- Each map statement: 96 bytes (or dynamically built map)
- Each IDB (hardware interface + encap Frame Relay): 5040 + 8346 = 13,386 bytes
- Each IDB (software subinterface): 2260 bytes

Example: 2501 using 2 Frame Relay interfaces, each with 4 subinterfaces, total of 8 DLCIs, and associated maps
2-interface IDB x 13,386 = 26,772 physical interfaces
8-subinterface IDB x 2260 = 18,080 subinterfaces
8 DLCIs x 216 = 1728 DLCIs
8 map statements x 96 = 768 map statements or dynamics
Total = 47,348 bytes of RAM used

Note: The values used here are based on Cisco IOS Release 11.1 software. Other releases will vary slightly.

Key Cisco Features and When To Use

Following is a discussion of key features that can be used to solve problems associated with designing scalable Frame Relay networks. For a full description of these features, refer to the published documentation available through Cisco Connection Online (CCO) or in training documentation.

Traffic Shaping over Frame Relay

Introduction

Congestion management in Frame Relay networks has been a challenge for some time now. Frame Relay technology does have congestion notification mechanisms built into the specifications, and most switch vendors implement them. However, the notifications were mainly intended for end systems, which are most often the source of the congestion condition to begin with. The intervening router connecting the end stations to the Frame Relay network have played a largely passive role in Frame Relay switch congestion, except, of course that they promote the congestion notifications to protocols having a congestion indicator, namely DECnet IV and OSI Connectionless Network Service (CLNS). Backward explicit congestion notification (BECN) support for Systems Network Architecture (SNA) was implemented by reducing the window size in the Logical Link Control (LLC) header. Most other protocols, however, are simply not notifiable, in that their protocol headers contain no congestion indication. This driving need for more control in this area has led to the development of a congestion management feature that also includes the prioritization of data going into a Frame Relay network.

Description

The traffic shaping over Frame Relay feature is available in Cisco IOS Release 11.2 software. This feature allows the router to regulate and prioritize the transmission of frames on a per-VC basis to the network as well as react to congestion notification from the Frame Relay network. Traffic shaping for Frame Relay can be broken down into three main components:

- Rate enforcement on a per-VC basis: Defines and enforces a rate on the VC at which the router will send traffic into the network
- Generalized BECN support on a per-VC basis: Enables router to dynamically fluctuate the rate at which it sends packets, depending on the BECNs it receives; for example, if the router begins receiving a significant number of BECNs, it will reduce the frame transmit rate; as BECNs become more intermittent, the router will increase the frame transmission rate
- VC queuing (custom, priority, and first in first out [FIFO]): For circuits carrying more than one protocol, queuing can now be applied on a per-VC basis; this setup can be accomplished by configuring queuing as in earlier releases, and then applying either the queue list for custom queuing or priority group for priority queuing to the map class command used in traffic shaping (see following discussion); queuing must be defined the same on sending and receiving routers for proper operation

Additional overall feedback is provided to the traffic shaping algorithm by monitoring the queue depth of the physical interface.

This functionality will apply to both PVCs and SVCs.

The rate enforcement algorithm used incorporates a two-stage queuing process. The first level is where the queuing on a VC basis, using either custom, priority, or the default first come first served mechanism, is implemented. The output of these first-stage queues feeds into a single interface level queue. Traffic is metered at the output of the per-VC queues based on the configuration parameters specified for each of these queues.

Weighted fair queuing (WFQ) and traffic shaping over Frame Relay are mutually exclusive.

Configurable Parameters

For each Frame Relay VC, the user may configure the following parameters:

- CIR—Committed information rate
- B_c—Committed burst size
- B_e—Excess burst size
- Q—Queuing algorithm to be used within the VC

Facilities are provided so that a user may configure a default profile for all VCs at the interface or subinterface level, which can be overridden at the individual VC level if required.

End-User Interface

```
[no] frame-relay traffic-shaping
```

Interface level command turns on traffic shaping and per-VC queuing.

```
[no] frame-relay traffic-rate <average> [<peak>]
```

Map class subcommand. <average> is average rate, equivalent to CIR in bps. <peak> is peak rate, equivalent to $CIR + B_e/t = CIR(1+B_e/B_c)$. If peak is omitted, the default value used is derived from the interface bandwidth parameter.

Note: If Enhanced IGRP (new) is configured, then it is advisable to configure the <peak> parameter, as opposed to letting it default. Enhanced IGRP can also be dependent on the interface bandwidth parameter.

Other related commands are:

```
frame-relay custom-queue-list <list #>
frame-relay priority-group <list #>
frame-relay becn-response-enable
frame-relay class <map-class-name>
show frame-relay pvc (Extended to include specified parameters and queuing used)
```

Note: See Cisco IOS V.11.2 software training documentation for configuration examples at http://wwwin/Mkt/KP/SSCT_CAT/Support/xfrtrafsh.pdf.

Problems Solved

Prioritization of packets on a per-VC basis solves a major problem when multiple protocols are configured on the same DLCI. The ability to prioritize, for example, SNA data over IP and other protocols in order to preserve session response time for legacy applications allows new IP-based applications to safely share the same resources.

Frame Relay Traffic entering a Frame Relay network does so at the link access rate, regardless of any parameters set on the switch such as CIR, B_e, or Ce. Of course the rate and volume of traffic entering the switch will be monitored at the input and decisions will be made on what to do with the traffic based on these parameters. The incoming packets either can be propagated into the network, which is the case for all traffic entering within CIR, or the packets can be marked as discard eligible for those packets entering between CIR and the B_e limit, and thus subject to being dropped if network congestion exists, or the packets simply get dropped at the input if B_e is exceeded. The Frame Relay traffic shaping rate enforcement feature provides control over how much data is sent into the network. It can allow, for example, packets to enter the Frame Relay network within CIR and thus have a guarantee of being propagated through. It can further ensure that traffic enters the network within the B_e limit so that immediate drops do not occur.

For zero CIR service offered by some service providers, the traffic shaping rate enforcement feature could provide a level of control at the router. If you are experiencing drops in the zero CIR environment, some parameter experimentation and monitoring may be in order to identify at what level the Frame Relay switch begins to drop data. Finding the correct rate enforcement parameters for this particular environment can result in maximizing the available resources while minimizing the amount of dropped frames.

In the event of congestion within the Frame Relay network where BECN is provided to the source (in this case the router) a further throttling of data that is network-bound will occur. The problem solved here is that the congestion in the Frame Relay network now has a much better chance of decongesting quickly, without dropping as many DE packets as it might have.

CONFIGURATION EXAMPLE: Router Frame Relay 45 is a hub router, and Frame Relay 3 is a spoke router.

Frame Relay 45 Router Configuration

Current configuration:

```
!
version 11.2
!
hostname FR45
!
!
interface Ethernet0
ip address 192.150.42.61 255.255.255.248
media-type 10BaseT
!
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay traffic-shaping                **Enable traffic shaping on interface**
!
interface Serial0.1 point-to-point
ip address 171.68.157.113 255.255.255.240
ipx network AB449D80
frame-relay class 32cir                    **Here, the map class defined as follows is assigned to subinterface**
frame-relay interface-dlci 101 broadcast
!
```

Frame Relay 45 Router Configuration

```
interface Serial0.2 point-to-point
ip address 171.68.157.129 255.255.255.240
frame-relay class 16cir                **Here, the map class defined as follows is assigned to subinterface**
frame-relay interface-dlci 102 broadcast
!
interface Serial0.3 point-to-point
ip address 171.68.157.145 255.255.255.240
frame-relay class bc64                 **Here, the map class defined as follows is assigned to subinterface**
frame-relay interface-dlci 103 broadcast
!
!
router Enhanced IGRP 44
network 171.68.0.0
!
ip route 171.69.1.129 255.255.255.255 192.150.42.62
!
map-class frame-relay 32cir
frame-relay traffic-rate 32000 64000   **Here, the average and peak rates are set to the VCs CIR, and $B_e$**
frame-relay custom-queue-list 1        **Here, a custom queue list is also assigned to this map class**
!
map-class frame-relay 16cir
frame-relay traffic-rate 16000 64000   **Here, the average and peak rates are set to the VCs CIR, and $B_e$**
!
map-class frame-relay bc64
frame-relay cir in 32000               **Here, specific control of parameters is possible on a bidirectional basis**
frame-relay cir out 32000
frame-relay bc in 32000
frame-relay bc out 64000
frame-relay be in 64000
frame-relay be out 64000
!
queue-list 1 protocol ip 1
queue-list 1 protocol ipx 2
queue-list 1 queue 1 byte-count 4200
queue-list 1 queue 2 byte-count 1400
!
!
end
```

FR3 Configuration

Current configuration:

```
!
version 11.2
!
hostname FR3
!
enable password cisco
!
ipx routing 0000.0c18.d70c
!
interface Ethernet0
ip address 198.19.1.1 255.255.255.0
ip helper-address 171.68.159.82
ipx network C6130301
!
!
interface Serial0
ip address 171.68.157.146 255.255.255.240
encapsulation frame-relay
ipx network AB449D90
frame-relay traffic-shaping
frame-relay class 32cir                  **Here, the map class defined as follows is assigned to interface**
!
router Enhanced IGRP 44
network 171.68.0.0|
!
!
no ip classless
ip route 171.69.1.129 255.255.255.255 192.150.42.161
!
map-class frame-relay 32cir
frame-relay traffic-rate 32000 64000     **Here, the average and peak rates match those in the hub router**
frame-relay custom-queue-list 1          **Here, the custom queue list matches that in the hub router**
!
queue-list 1 protocol ip 1
queue-list 1 protocol ipx 2
queue-list 1 queue 1 byte-count 4200
queue-list 1 queue 2 byte-count 1400
!
```

FR3 Configuration

```
!
end
```

Dial Backup per DLCI

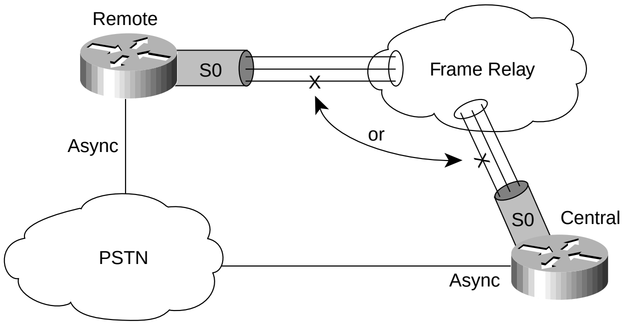
To back up an individual DLCI requires that a separate interface be used. This feature works by detecting that the VC is no longer available, and this detection is learned through either the Local Management Interface (LMI) status updates from the switch or from not receiving keepalives on the physical link; the latter case would apply if the physical link goes down. In a typical hub and spoke scenario, the spoke or access site should initiate the backup, easing resources required at the central site. The backup can be done through a separate serial interface, an async port, or over an Integrated Services Digital Network (ISDN) link. For ISDN, some service providers offer media support backup into a Frame Relay switch port. This setup will become more popular with the availability of SVCs. Otherwise, backup is typically accomplished by bypassing the entire Frame Relay network and connecting directly to the central hub site.

If the serial interface option is used, the interface could be brought up; this action would in turn initiate a switched synchronous connection to be made on the serial interface of the access device. A dedicated serial port at the hub site or a switched connection into a Multichannel Interface Processor (MIP) would be required. This strategy could be expensive in resources.

A good strategy if the ISDN option is employed is to use Primary Rate Interface (PRI) ISDN at the central site. Concentration of multiple ISDN circuits in a single router is achievable and more cost effective.

If the async interface is used, then dialing into a terminal access server located at the hub or central site works well. A configuration example using the async line is shown in Figure 5:

Figure 5 Dial Backup Initiated from Spoke Site Router upon Detection of VC Failure either Locally or Remotely



CONFIGURATION EXAMPLE: Spoke router dial backup using async interface when DLCI 207 goes down

Example

Current configuration:

```
!
version 11.1
!
hostname fr7
!
enable password cisco
!
```

Example

```
!
interface Ethernet0
ip address 8.51.12.33 255.255.255.224
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
!
interface Serial0.2 point-to-point   **Note that a subinterface is used when backup is defined.**
backup delay 0 0                     **A failed DLCI causes the subinterface to go down, thus **
backup interface Async1              **triggering the backup interface async 1 to be activated. **
ip address 171.68.157.226 255.255.255.240
frame-relay interface-dlci 207 broadcast
!
interface Async1
ip address 172.58.157.112 255.255.255.0
dialer in-band
dialer string 18005551212
!
router Enhanced IGRP 44
network 171.68.0.0
!
!
line con 0
exec-timeout 0 0
password cisco
line aux 0
line vty 0 4
password cisco
login
!
end
```

Broadcast Queue

Broadcast queue is a major feature for use in medium to large IP or IPX networks where routing and SAP broadcasts must flow across the Frame Relay network. The broadcast queue is managed independently of the normal interface queue, has its own buffers, and has a configurable size and service rate (see Figure 6). This broadcast queue is not used for bridging spanning-tree updates (BPDUs) because of timing sensitivities. These packets will flow through the normal queues. The interface command to enable broadcast queue follows:

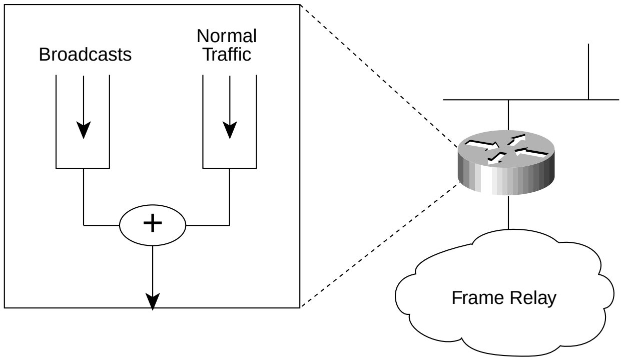
`frame-relay broadcast-queue` *size byte-rate packet-rate*

A broadcast queue is given a maximum transmission rate (throughput) limit measured in bytes per second and packets per second. The queue is serviced to ensure that only this maximum is provided. The broadcast queue has priority when transmitting at a rate below the configured maximum, and hence has a guaranteed minimum bandwidth allocation. The two transmission rate limits are intended to avoid flooding the interface with broadcasts. The actual limit in any second is the first rate limit that is reached. Given the transmission rate restriction, additional buffering is required to store broadcast packets. The broadcast queue is configurable to store large numbers of broadcast packets. The queue size should be set to avoid loss of broadcast routing update packets. The exact size depends on the protocol being used and the number of packets required for each update. To be safe, the queue size should be set so that one complete routing update from each protocol and for each DLCI can be stored. As a general rule, start with 20 packets per DLCI. Also as a general rule, the byte rate should be less than both of the following:

- N/4 times the minimum remote access rate (measured in bytes per second), where N is the number of DLCIs to which the broadcast must be replicated
- 1/4 the local access rate (measured in bytes per second)

The packet rate is not critical if the byte rate is set conservatively. As a general rule, the packet rate should be set assuming 250-byte packets. The defaults are 64 queue size, 256,000 bytes per second (2,048,000 bps), and 36 pps.

Figure 6 Broadcast Queue Separate from Normal Traffic



CONFIGURATION EXAMPLE: Broadcast queue set for interface serial 0

Example

Current configuration:

```
!
version 11.2
!
hostname FR45
!
!
interface Ethernet0
ip address 192.150.42.61 255.255.255.248
media-type 10BaseT
!
interface Serial0
description T1 HUB LINK TO TELCO
```

Example

```
no ip address
encapsulation frame-relay
no fair-queue
frame-relay traffic-shaping
frame-relay broadcast-queue 80 48000 160     ** Queue size = 80, byte rate = 48 kbps (384 kbps), packet rate = 160 pps**
!
interface Serial0.1 point-to-point
ip address 171.68.157.113 255.255.255.240
frame-relay interface-dlci 101 broadcast
!
interface Serial0.2 point-to-point
ip address 171.68.157.129 255.255.255.240
frame-relay interface-dlci 102 broadcast
!
router Enhanced IGRP 44
network 171.68.0.0
!
end
```

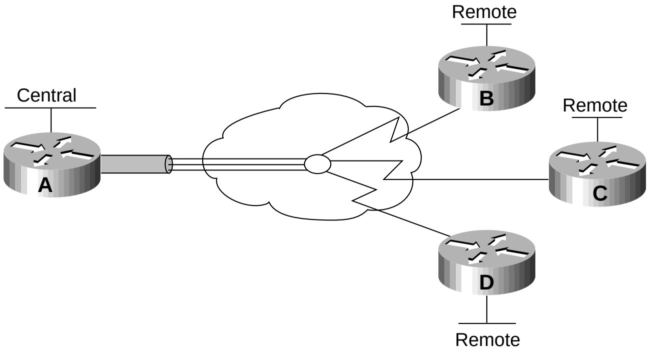
Subinterfaces

Frame Relay networks provide multiple point-to-point links, or permanent virtual circuits (PVCs), through the same physical serial interface. Subinterfaces allow blocks of one or more VCs to be treated as separate subnetworks. A subinterface with a single VC is modeled as a point-to-point link. A subinterface with multiple VCs is modeled as a LAN. Protocols such as IP, IPX, and bridging view each subinterface as a separate interface with its own address and protocol assignments. In the following example, subinterface 1 models a point-to-point subnet and subinterface 2 models a broadcast subnet:

```
interface serial 0
encapsulation frame-relay
!
interface serial 0.1 point-to-point
ip address 10.0.1.1 255.255.255.0
frame-relay interface-dlci 42
!
interface serial 0.2 multipoint
ip address 10.0.2.1 255.255.255.0
frame-relay map 10.0.2.1 255.255.255.0 17 broadcast
frame-relay map 10.0.2.2 255.255.255.0 18
```

Subinterfaces provide a mechanism for supporting partially meshed Frame Relay networks. In the past, a single network number (such as an IP subnet or an IPX network number) was assigned to an entire Frame Relay or X.25 network. Most protocols assume transitivity on a logical network; that is, if station A can talk to station B, and station A can talk to station C, then station B should be able to talk to station C directly (see Figure 7). This scenario is true on LANs, but has not been true on public switched networks unless they were fully meshed. Additionally, certain protocols such as IPX and AppleTalk could not be supported on partially meshed networks because they require "split horizon," in which a routing protocol packet received on an interface cannot be transmitted out the same interface even if the packet is received and transmitted on different VCs. Subinterfaces address these limitations by providing a way to subdivide a partially meshed Frame Relay network into a number of smaller, fully meshed (or point-to-point) subnetworks.

Figure 7 Routers B, C, and D Can Have Connectivity with Each Other through Router A Using Subinterfaces



CONFIGURATION EXAMPLE: Point-to-point subinterfaces configured at hub site

Example

Current configuration:

```
!
version 11.2
!
hostname Router-A
!
!
interface Ethernet0
ip address 192.150.42.61 255.255.255.248
media-type 10BaseT
!
interface Serial0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay traffic-shaping
frame-relay broadcast-queue 64 48000 36
!
```

Example

```
interface Serial0.1 point-to-point      **To router B**
ip address 171.68.157.113 255.255.255.240
frame-relay interface-dlci 101 broadcast
!
interface Serial0.2 point-to-point      **To router C**
ip address 171.68.157.129 255.255.255.240
frame-relay interface-dlci 102 broadcast
!
interface Serial0.3 point-to-point      **To router D**
ip address 171.68.157.145 255.255.255.240
frame-relay interface-dlci 103 broadcast
!
!
router Enhanced IGRP 44                 **When using a routing protocol, partial mesh connectivity is attained among routers A,
                                        B, C, and D with very little configuration.**
network 171.68.0.0
!
!
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
login
!
end
```

Inverse ARP

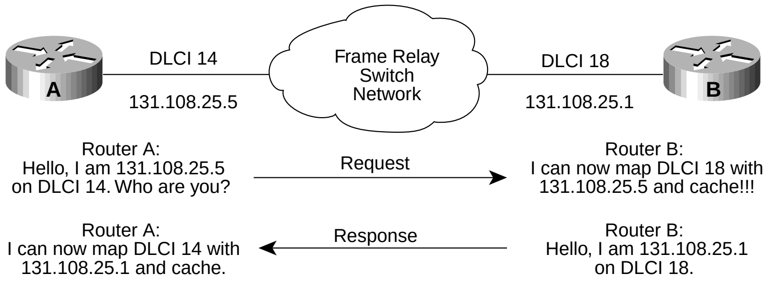
Inverse ARP allows a router running Frame Relay to dynamically discover the protocol address of a device associated with the VC. The use of inverse ARP greatly reduces the configuration task of specifying map statements and includes the following protocols:

IP, IPX, XNS, AppleTalk, DECnet and VINES. Inverse ARP is on by default.

This implementation of inverse ARP is based on RFC 1293. It allows a router or access server running Frame Relay to discover the protocol address of a device associated with the VC. In Frame Relay, PVCs are identified by a DLCI, which is the equivalent of a hardware address. By exchanging signaling messages, a network announces a new VC, and with inverse ARP, the protocol address at the other side of the circuit can be discovered (see Figure 8). The following example sets inverse ARP on an interface running IP:

```
interface serial 0
frame-relay inverse-arp ip 100
```

Figure 8 Frame Relay Inverse ARP Exchange



AutoInstall

This feature allows for simple router installation at a remote site from a centralized management location. The central location connects to the remote router via the Frame Relay link and downloads a configuration file. This downloading results in considerable time and cost savings since users at remote sites do not need any specialized training to install a router. The central management location has complete control of the process, and the remote user only needs to power up the remote router, configure encapsulation as Frame Relay, and connect the serial line.

AutoInstall Steps

1) New unconfigured (write erase) router's lowest-numbered serial interface is configured for encapsulation Frame Relay and is connected to Frame Relay network. LMI will resolve what the DLCI is.

2) The new router will now find an IP address via BOOTP. This scenario requires a gateway router at the other end and a Trivial File Transfer Protocol (TFTP) server. The gateway router must be configured to map DLCI to IP addresses.

3) The gateway router acts as a BOOTP server, with an IP helper address pointing to a TFTP server.

4) The new router, upon receipt of the IP address for serial interface and the TFTP server address provided by the BOOTP server, will now automatically send a TFTP request for the configuration file. The gateway router forwards the request to the TFTP server.

5) The TFTP server sends *network-confg* to requester, which in turn tries to resolve Hostname.

6) If Hostname is resolved, *hostname-confg* is sent and we're done. If not…

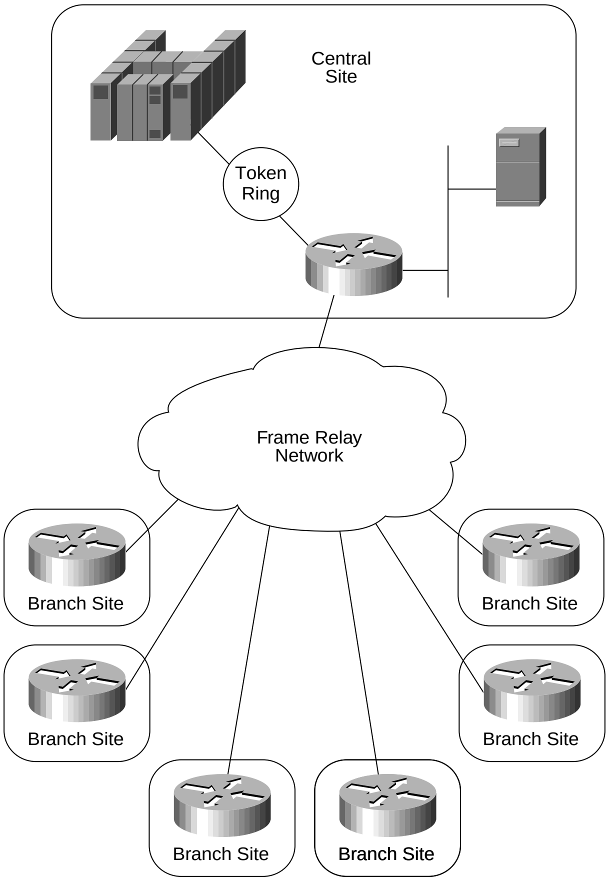
7) TFTP server sends *router-confg*, a basic default configuration.

Design Topologies and Strategies

Star Topology (Hub and Spoke)

The topology most referred to in this design guide, the star topology, is the most common Frame Relay network design found today. It accounts for more than 65 percent of all Frame Relay networks, and spans from as few as two remote sites to several hundred remote sites. This topology can be scaled somewhat by using multiple central site router interfaces or multiple central site routers to split the network into multiple segments. In a pure hub and spoke topology, there are no spoke-to-spoke connectivity requirements. Occasional spoke-to-spoke connectivity can be achieved by the use of SVCs available in Cisco IOS Release 11.2 software. To scale to larger networks, a hierarchical design is recommended. (See Figure 9.)

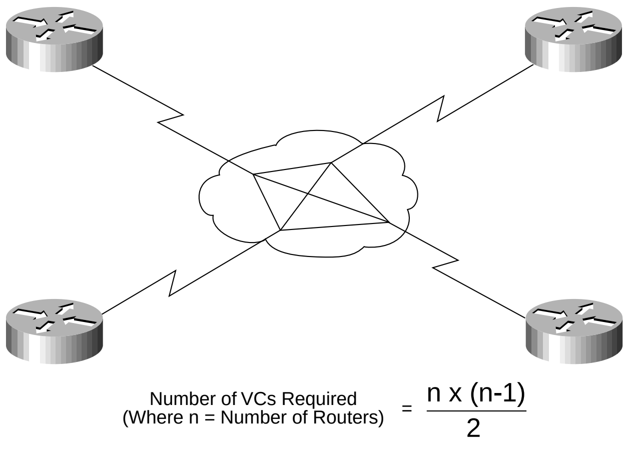
Figure 9 Multiple Branch (Spoke) Sites Communicate to Single Central (Hub) Site



Full Mesh Topology

A full mesh topology is required for full connectivity requirements. In a Frame Relay environment, VCs must be provisioned to all destination routers from each router. In a large, full mesh network, this provisioning can be costly in PVCs, but not as costly as leased-line circuits, which would require sufficient router interface ports as well as CSU/DSUs for each interface. Full mesh connectivity is typically required when designing the core portion of a wide-area network, where a distributed server design exists. To calculate how many VCs are required for full connectivity, see Figure 10.

Figure 10 Full Mesh Topology for Any-to-Any Connectivity



Partial Mesh Topology with Redundant Central Site

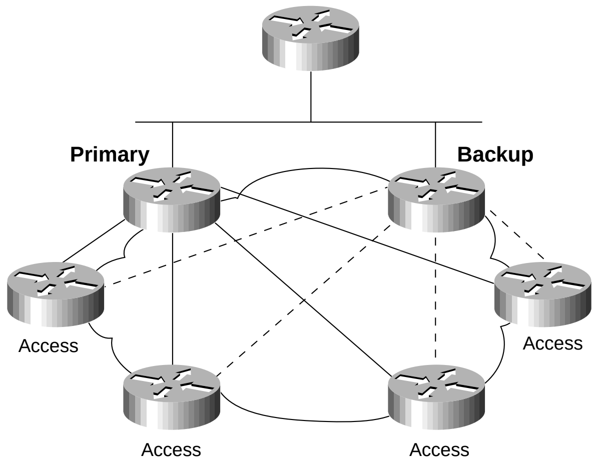
Partial mesh topologies incorporate a basic hub and spoke design where a central site exists, but where connectivity between spoke sites is also required. This design is common in the retail environments where both centralized warehousing and store-to-store inventories can be accessed from any location. The use of point-to-point subinterfaces facilitates this kind of design by providing any-to-any connectivity through the central hub router.

In the example that follows is a partial mesh design that also has redundant centralized routers for backup (see Figure 11).

Different strategies could be used in this example:

- Have spoke site routers connected to both primary and backup hub routers through separate PVCs and either configure Hot Standby Router Protocol (HSRP) on primary and backup or run both PVCs to load share when both are active, while providing backup if one should fail
- Trigger ISDN dial backup to occur at the access site when either the Frame Relay network or the primary router goes down; taking this a step further, with dual hub routers, split the VC connections to all the spoke sites between the two hub routers, and back up each hub router’s VCs to the alternate hub router using ISDN PRI
- Trigger Frame Relay SVC connection to backup router

Figure 11 Each Access Site Has Two Paths to Redundant Central Router



Hierarchical Mesh

A hierarchical design is required for building large Frame Relay networks or integrating separate networks into a larger single network as is the case in business mergers. The idea is to split the network into different layers, where each layer has distinct connectivity functions (see Figure 12). The three layers follow:

Core Layer

The core makes up the corporate backbone whereby dispersed central or large regional offices are interconnected through high-speed links in a full mesh topology. From a router perspective, this scenario means that certain interfaces will be dedicated to connecting to other core routers and local servers, as well as connecting to the distribution layer, which may or may not be collocated.

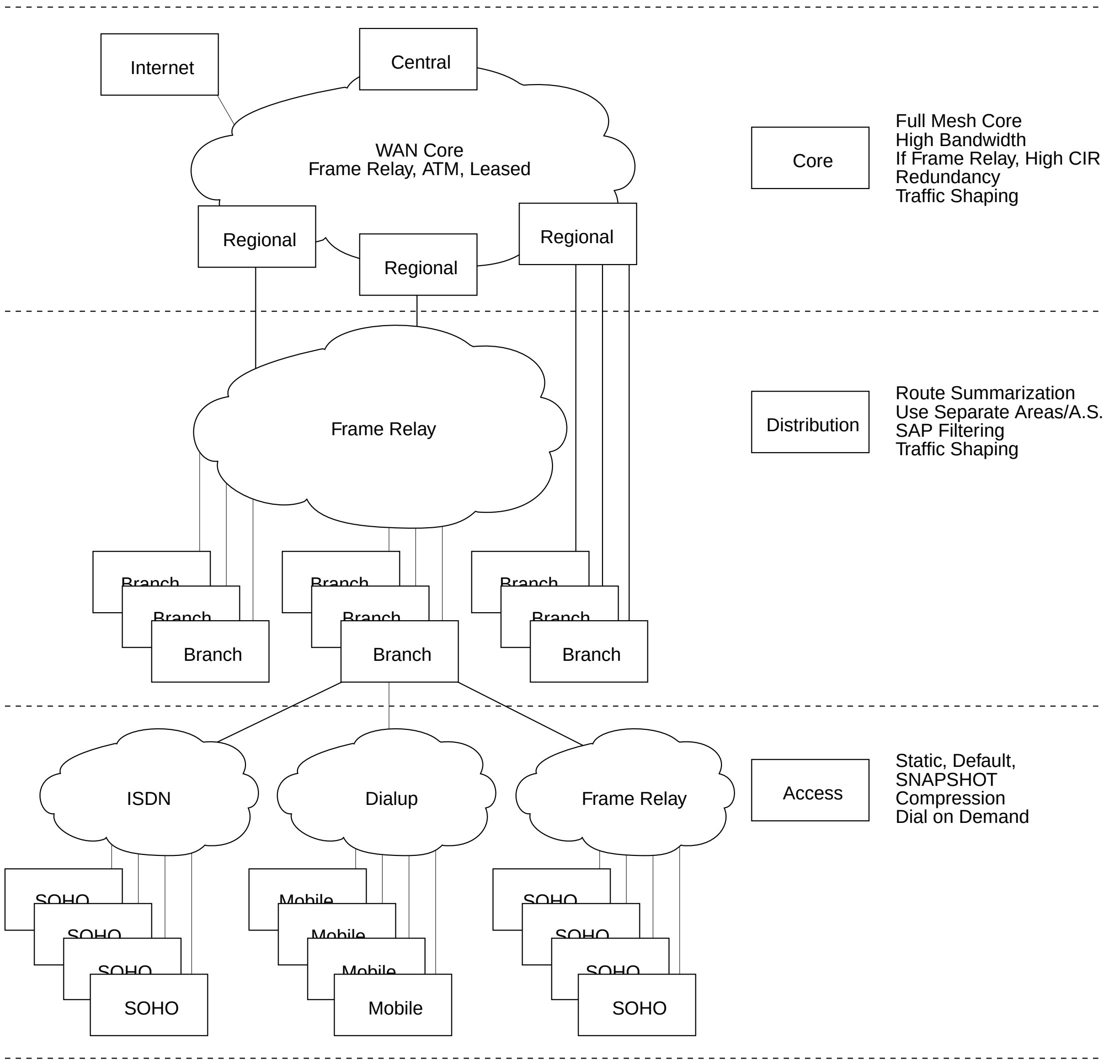
Distribution Layer

The distribution layer will be the connection point to the core and access sites. The distribution layer can provide isolation of unnecessary broadcast traffic entering the core layer, as well as enabling bandwidth, security, and other processor intensive features to the access layer. Converting a pure hub and spoke topology to a hierarchical mesh would imply adding distribution routers at the central (core) sight and isolating the core routers from the access (spoke) sites.

Access Layer

The access layer is primarily where end stations will connect, but extensions beyond this layer (access sublayers) are quite common. The connection to the distribution layer is often relatively slow speed, so efficient use of bandwidth is important. Data/payload compression and local acknowledgment are features that can be employed effectively at this level. If a routing protocol is desired to run down to the access layer, then consider SNAPSHOT, or ensure that routing advertisements are small.

Figure 12   Hierarchical Design Separates Network Segments



Conclusion

Designing successful Frame Relay networks requires planning and efficient use of features. With the implementation variations that exist among Frame Relay equipment vendors and the many differing options provided by each service provider, predicting performance is difficult at best. Knowledge of what your traffic is and how much of it needs to be moved (such as SAP or broadcast filtering) is a good start in the planning process. Narrowing down your choice of network and technologies will be easier with this knowledge.

As Frame Relay and related technologies such as ATM mature, the emphasis is being focused on interworking, bandwidth efficiency, and quality of service. Integrating data, voice, and video is the key driver for this focus, and it is in these areas in which the greatest improvements will be seen.

Appendix

Frame Relay Related Standards Supported in the Cisco IOS Software

RFCs

- RFC 1490—Multiprotocol encapsulation
- RFC 1315—Frame Relay MIB
- RFC 1293—Frame Relay inverse ARP
- RFC 1144—TCP/IP header compression

FRF Implementation Agreements

- FRF 1.1—User-Network Interface (UNI)
- FRF 2.1—Frame Relay Network-to-Network Interface (NNI)
- FRF 3.1—Multiprotocol encapsulation
- FRF 4—SVCs
- FRF 6—Frame Relay service customer network management (MIB)

Other Standards

- Gang of four LMI
- Q.922 Annex A
- ANSI T1.617 Annex D
- ANSI T1.618, T1.606
- ITU-T Q.933, Q.922

References

1) *Frame Relay Design Tool Excel Spreadsheet*
http://www.cisco.com/warp/customer/790/General/frtoo_pc.xls

2) *OSPF Design Guide*
http://www.cisco.com/warp/public/104/1.html

3) *Configuration Notes for the Enhanced Implementation of Enhanced IGRP*
http://cio.cisco.com/warp/customer/103/12.html

Cisco Systems
Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
World Wide Web URL:
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems Europe
s.a.r.l.
Parc Evolic-Batiment
L1/L2
16, Avenue du Quebec
BP 706-Villebon
91961 Courtaboeuf Cedex
France
Tel: 33 1 6918 61 00
Fax: 33 1 6928 83 26

Intercontinental Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
Tel: 408 526-7660
Fax: 408 526-4646

Latin American Headquarters
Cisco Systems, Inc.
790 N.W. 107th Avenue
Suite 102
Miami, FL 33172
Tel: 305 228-1200
Fax: 305 222-8456

Japanese Headquarters
Nihon Cisco Systems K.K.
Fuji Building
3-2-3 Marunouchi
Chiyoda-ku, Tokyo 100
Japan
Tel: 81 3 5219 6000
Fax: 81 3 5219 6010

Cisco Systems has over 190 offices in the following countries. Addresses, phone numbers, and fax numbers are listed on the Cisco Connection Online Web site at http://www.cisco.com.

Argentina • Australia • Austria • Belgium • Brazil • Canada • Chile • China (PRC) • Colombia • Costa Rica • Denmark • Finland • France • Germany Hong Kong • India • Indonesia • Ireland • Italy • Japan • Korea • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Philippines Portugal • Singapore • South Africa • Spain • Sweden • Switzerland • Taiwan, ROC • Thailand • United Arab Emirates • United Kingdom • Venezuela