

Maintaining SNA Response Times

Maintaining consistent and low SNA response times is the “holy grail” of the SNA Internetworking quest. The requirement is frequently called meeting service-level agreements or protecting mission-critical data. The origins of this requirement are often technical, historical, political, or some combination of the three. SNA response time requirements vary greatly between customers. This requirement does not limit itself to SNA applications. A plan for network migration to Frame Relay often requires equal attention be paid to protecting other mission-critical traffic such as a stock trading system. However, the techniques discussed in this guide are flexible and powerful enough to be adapted to the vast majority of customer settings, applications, and protocols.

The bulk of existing SNA networks are traditional hierarchical networks in which many remote sites access a single central site. Classic hub-and-spoke or star Frame Relay network topologies are common. Other design topologies, including variations on fully meshed or partially meshed networks, exist but are rare in comparison.

Latest industry studies show that approximately 95 percent of all public Frame Relay networks are star topologies, and we believe that this is also true for enterprise networks. The majority of line speeds are 64 kbps or lower. This is likely to create a common set of problems that are likely to occur as legacy SNA networks continue to migrate to the classic Frame Relay star topologies.

In this chapter, we will look at the most prevalent problems that degrade response times and techniques to resolve them. Much is written about the factors that contribute toward degrading response times. Most of the technical literature concentrates on transmission theory, error rates, retransmission algorithms, and queuing delays from a Layer 2 perspective. We will concentrate primarily on queuing delays in the router from a Layer 3 perspective. In this discussion, we will take the 80:20 approach to resolving problems and discuss the 20 percent of the issues that can resolve 80 percent of the response time problems.

Issues that Negatively Impact Response Times

This section discusses the issues that have a negative impact on response times.

Propagation Delay

Propagation delay is the amount of time it takes for a signal to propagate across the media on which it is transmitted. Propagation delay is fixed and subject to distance factors. A complete discussion of this is beyond the scope of this document. Coast-to-coast propagation delay typically accounts for approximately 65-75 msec. SNA user response times are on the order of second intervals. Propagation delay is insignificant relative to the overall user perceived response times.

Transmission Delay

Transmission delay, or queuing delay, is the amount of time required to put data on the wire. It is sometimes referred to as serialization delay for a single packet. The major issue that impacts transmission time is the maximum transmission unit (MTU). For example, it takes almost .25 seconds to transmit 1500 bytes on a 64-kbps link. A queue build-up of ten or twenty 1500-byte packets would take two or three seconds to transmit. If an SNA packet was scheduled behind all this traffic, it would add at least that much time to the response times. Queuing delays during periods of congestion are a major factor that impacts the response times perceived by end users. The sections “Mixing Batch and Interactive Traffic” and “Mismatched Access Speeds” address resolving these delays.

Encapsulation overhead also contributes to transmission delays though not significantly. The issue of encapsulation overhead receives far too much attention. For example, DLSw+ TCP/IP encapsulation adds about 56 bytes of transmission overhead. It takes .007 seconds to transmit 56 bytes on a 64-kbps link. Not many users will notice this response time delay.

Latency

Latency is the delay associated with making a switching decision. CPU overhead is a contributing factor in latency. CPU overhead is the amount of time it takes the router to encapsulate and switch a packet (or frame). These factors contribute to latency in the Frame Relay backbone network on switches as well, but in smaller amounts.

Mixing Batch and Interactive Traffic

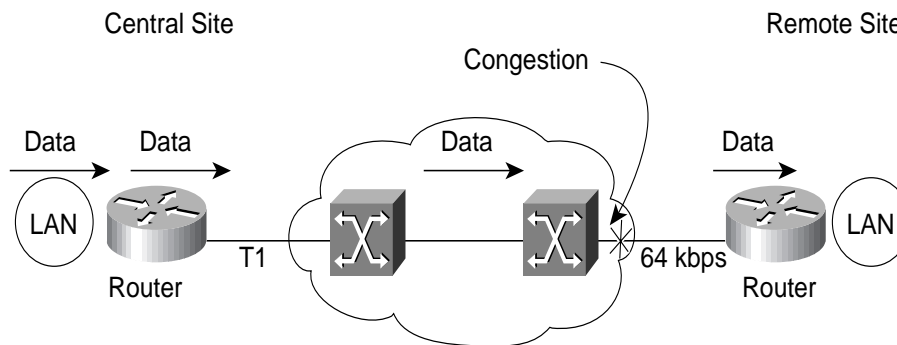
To preserve SNA response times, interactive SNA traffic must be separated from batch traffic and the interactive traffic must be prioritized. The initial reaction when addressing response time delays is to separate SNA from IP. For the most part, this would be correct. However, not all SNA traffic is interactive and not all batch traffic is IP (SNA can perform file transfers and IP has interactive applications). It is not the mix of protocols that causes the delays but the mix of batch and interactive traffic. Batch traffic interferes with interactive traffic response times.

To improve response times, you must separate batch traffic from interactive traffic and use a method of traffic prioritization to meet response time requirements. For example, if INDSFILE is present in a network, then even if SNA is high priority, the batch SNA traffic will negatively impact the response times for interactive SNA traffic.

Mismatched Access Speeds

Mismatched access speeds introduce a subtle problem that impacts just about all Frame Relay networks in production today. For example, consider traffic going from the central site to a remote site (see Figure 2-1). If a remote site is connected to a 64-kbps access speed line and the central site is connected to a T1 rate access speed line, traffic being transmitted by the router at the central site is clocked out of the Frame Relay serial port at T1 speeds. Thus, the likelihood of congestion appearing on the T1 interface is small. Because the data is traveling from T1 speeds to 64 kbps, a bottleneck will occur at some point. The bottleneck is encountered not at the T1 port when it exits the router, but at the 64-kbps egress port when it exits the Frame Relay switch. During periods of congestion at the Frame Relay switch egress port, frames are not prioritized and, because of excessive queue depths, transmission delays will be common. During periods of extreme congestion and buffer shortages, frames are randomly dropped at the egress port. If one of the randomly dropped frames is an SNA packet, this will cause a retransmit that will wreak havoc on response times.

Figure 2-1 Point of Congestion Caused by Mismatched Access Speeds



Techniques to Improve Response Times

The previous section talked about the issues that contribute to the degradation of response times. In this section, we talk about some generic techniques and some specific features of Cisco IOS software that can be used to improve SNA response times.

Reducing Transmission Delay

As previously mentioned, transmission delay is largely a constant based on the end-to-end propagation delay and the clock speed of the access port. The only way a user has to reduce transmission delay is to create smaller packets or increase clock rates. In many cases, when migrating an SNA network from a legacy SDLC 9.6-kbps or 19.2-kbps environment, significant improvements can be realized from the increase in clock speeds that Frame Relay provides.

Another alternative to lowering transmission delay is to reduce the IP MTU on the Frame Relay port. This improves response times by reducing the overall queuing delays. However, this can have a severe impact on the network by causing excessive IP fragmentation. Fragmenting not only causes CPU overhead to perform the fragmenting operation, but it also significantly increases the number of packets in transit on an enterprise network. Reducing the MTU from 1500 to 250 bytes effectively increases packet rates five-fold. This is not a major impact at remote sites where there are low rates of traffic, but at the central site aggregation points this can have a serious impact.

IP MTU path discovery solves the IP fragmentation problem, but it does not solve the problem of increased packet rates at central points of aggregation.

The solution to this dilemma is to fragment the frame just at the points where there are slow transmission speeds and to reassemble the fragments before forwarding the frame into the enterprise network, in other words a Layer 2 fragmenting scheme. The idea is to fragment the large frames, the so-called jumbo-grams, and interleave the shorter high-priority frames between the fragments. Dropping the MTU size is a Layer 3 fragmenting technique discussed above. There are two ways to fragment at Layer 2: FRF.12 and MLPPP. At this time, Cisco IOS software does not support FRF 12, but it will in a future release. Some Layer 2 multiservice Cisco products do support a variation of FRF 12 (for example, the Cisco 3810) to solve a similar problem for voice prioritization. At this time, the 3810 does not support any other Layer 3 QoS features (that is PQ, CQ, WFQ, and so forth). Cisco does support MLPPP in combination with weighted fair queueing (WFQ) and Resource Reservation Protocol (RSVP) on leased-line Point-to-Point Protocol (PPP) links, but not over Frame Relay (support for this is targeted in Release 12.0(4)T).

Another way to reduce the packet size is to compress the packet using Frame Relay payload compression or TCP header compression. Payload compression is a CPU-intensive process because it must compute mathematically complex operations on every bit of every byte transmitted. Keeping up with this demand at high packet rates is difficult. To make matters worse, the locations where compression is needed most are at remote sites where there are very slow lines and low-cost machines. Fortunately, the low-speed lines create a bottleneck that allows low-cost machines to keep up with the CPU processing demands. Care must be taken when considering using compression based on the platform in use. For example, a Cisco 2500 compressing traffic at 256-kbps rates runs at 99 percent CPU utilization. This situation must be avoided, so the recommendations on which speed to run on which platforms are as follows:

- Cisco 2500: less than 128 kbps
- Cisco 4000: less than 256 kbps
- Cisco 4500: less than 800 kbps
- Cisco 4700: less than T1 or E1 speeds
- Cisco 7x00: use compression service adapter (CSA)

Because the CPU cost is on a per-bit basis, the speeds listed are aggregate speeds. For example, on a Cisco 4700 one can compress on three 512-kbps links roughly totaling a single T1.

There are two forms of Frame Relay payload compression available. The oldest form is packet-by-packet compression, which is licensed from STAC Corporation. It implements a compression methodology that looks for repeated patterns on each packet. Thus, the larger the packet, the better the compression ratios as the likelihood of finding repeated characters increases.

The other form of payload compression is FRF 9, which is a standards-based compression method approved by the Frame Relay forum. FRF 9 is a dictionary-based compression methodology that uses a sliding window of the most recently transmitted data (the dictionary) to find repeated patterns. The dictionary is often much larger than a single packet and thus it can achieve much higher compression ratios (typically 2:1 ratios).

TCP header compression is another form of compression that is not as CPU intensive because it compresses only the 40-byte TCP/IP header. The benefits realized using TCP header compression are not of great significance if the packet is large because the ratio of payload to header is small. If packets are small, then header compression is much more beneficial. There is no known performance data available for TCP header compression.

Techniques for Separating Batch Traffic from Interactive Traffic

This section is not specific to Frame Relay environments. Many of the features and configuration samples discussed here are found in *DLSw+ Design and Implementation Guide*. This section provides additional information on design issues and, because DLSw+ is being implemented in large scale in Frame Relay environments, stresses the importance of Frame Relay.

To protect SNA response times, SNA traffic must be isolated from other ambient traffic present at a bottleneck. In general, batch traffic must be separated from interactive traffic. In some cases, this is as simple as separating SNA from IP, and will work in most circumstances. However, a more general solution is to separate all interactive traffic from the batch traffic. Interactive traffic can be in the form of SNA 3270, Telnet, Hypertext Transfer Protocol (HTTP), or other custom applications. Batch traffic can be either SNA 3270 (IND\$FILE) or some form LU 6.2 or LU 0 file transfer application, FTP, NetBIOS, HTTP (yes, it can be considered batch or interactive), or any other custom application.

By classifying all interactive traffic together, bandwidth demands should be very meager in that traffic classification. Mixing interactive SNA 3270 with Telnet should not be a major issue because the demands of Telnet are minimal (mostly single-byte frames). There is also the option to create another class of service for Telnet if requirements call for it.

The more difficult task is to separate SNA batch traffic from SNA interactive traffic. Where dependent LUs are used the only means the router has to separate traffic is the SNA Network Addressable Unit (NAU), which is otherwise referred to as the LU address or the LOCADDR. Architecturally, SNA refers to this class of LUs as “dependent” because of their dependency on the VTAM SSCP. ALL LU type 2, all LU type 0, and some LU type 6.2 applications are dependent.

The separation of batch from interactive is further complicated if LU 6.2 parallel sessions is used because the NAU or LU address is dynamically allocated. Parallel session support is referred to as “independent” LUs because they do not require the assistance of the VTAM SSCP; it is APPN peer-to-peer communications. SNA class of service (CoS) to DLSw+ type of service (ToS), introduced in Cisco IOS Release 11.3, resolves this problem. This works by mapping SNA CoS to IP ToS (or precedence). SNA CoS to DLSw+ ToS can be used in conjunction with DLUR to migrate old dependent LUs to APPN. However, to map SNA CoS to DLSw+ ToS requires the use of APPN and DLSw+ on the same router and APPN in the enterprise network.

Features to separate SNA batch and interactive traffic without using APPN CoS to DLSw+ ToS exist in Cisco IOS. SNA batch traffic is typically a printer data stream or file transfer. Using the LU address, the router can redirect batch traffic from a particular LU (for example, LOCADDR 04) to a specific DLSw+ TCP port (1980) and redirect interactive traffic (LOCADDR 02) on a different DLSw+ TCP port (2065). Once the traffic is separated on different TCP flows, it is simply a matter of using any one of the various queuing techniques to establish packet scheduling priorities based on the DLSw+ TCP ports.

Special considerations must be made if INDSFILE is present in a network. It is important to highlight that the use of INDSFILE degrades SNA response times even in the legacy native SNA environments. INDSFILE is a very old, very inelegant method of transferring data. It essentially uses 3270 screens to batch move data. 3270 screens were not designed for this purpose. In networks today there is little excuse for using INDSFILE for data transfer. If INDSFILE is in use, there is no way to stop a user from starting a file transfer on any SNA 3270 session that could just as well be interactive. The best advice for customers trying to preserve SNA response times is to recommend they discontinue the use of INDSFILE and move to a more advanced method of file transfer. Even if a customer uses LU 6.2 for file transfers, SNA 3270 traffic can be identified in the dependent LU address space and be assigned to a DLSw+ TCP port. Then set the default for the remaining independent LU batch traffic to a different DLSw+ TCP port and prioritize at will.

Opening Multiple DLSw+ Ports

The first step in separating traffic on different flows is to create the flows on which to put the traffic. This is done by adding the priority keyword on the **dlsw remote-peer** command as follows:

```
dlsw remote-peer 0 tcp 1.1.1.1 priority
```

The priority keyword instructs DLSw+ to open four sessions to the remote peers on ports 2065, 1983, 1982, and 1981. By convention alone, these ports are given names: high, normal, medium, and low. This convention is not widely known, but it is critical to understanding how traffic is mapped to TCP port numbers. In reality, not much prioritization occurs until a queuing method is put in place on an output interface. The Cisco IOS software TCP driver checks port queues in descending numeric order. In fact, Cisco IOS software queuing methods can override

this convention by mapping the high port (2065) to a low service class using priority queuing (there's really no need to do this in practice though). It is important to understand the naming conventions and the mapping of names to port numbers because it is not apparent from looking at the configuration commands.

This process will increase the number of aggregate ports that are open and active by a factor of four. Although the total number of TCP sessions is not considered an important factor when sizing for performance, the total count of TCP sessions at the central site needs to be watched. There are some scaling issues when session counts reach large numbers (600 sessions).

Mapping Traffic by SAP to DLSw+ Ports

On workstations at the LAN, instances of Layer 3 or higher protocol stacks access link layer (or data-link control) services through a SAP. The SAP serves two purposes, which can be confusing. It identifies the protocol and addresses the link. The IEEE 802 committee has administered a number space for SAPs that appear on LANs. For example, SNA has reserved SAP 0x04, 0x08, and 0x0C, and NetBIOS has SAP 0xF0. Unfortunately, the SAP addressing was too small (four bits source and destination) to accommodate all protocol types migrated from Ethernet. Thus, a special SAP (0xAA) was reserved to extend the protocol identification. SAP 0xAA indicates that another header appears (called the SNAP header for sub-network access point) to represent the Ethernet protocol type field. For example, IP uses the SNAP SAP (0xAA) and within the SNAP header specifies the familiar 0x0800 protocol ID.

If you need to separate traffic by SAP, then the SAP prioritization feature can be used. Note that SAP prioritization does not understand SNAP headers. Most protocols in the SNAP field are routable. It is most common to separate SNA from NetBIOS using SAP prioritization. To do this, you need to first build the following list:

```
sap-priority-list 1 high ssap 04 dsap 04
sap-priority-list 1 low ssap F0 dsap F0
```

These commands define a global sap-priority-list and map SNA traffic to DLSw+ port 2065 and NetBIOS traffic to DLSw+ port 1981. You need to know that the high and low keywords map to these DLSw+ ports.

Then the **sap priority-list** command must be applied to an input LAN interface or applied to a bridging domain. The input LAN interface can be a Token Ring interface. The bridging domain can be a DLSw+ bridge group for Ethernet.

```
interface tokenring 0
sap-priority 1
```

or

```
dls w bridge-group 1 sap-priority 1
```

This process applies to any SAP that can be bridged. A packet with a SAP belonging to a routable protocol (such as SAP 0xAA) will be routed before it is evaluated by the **sap priority-list** command. As a result, this feature can be applied to other protocols and services identified by LLC SAP.

Mapping Traffic by LU to DLSw+ Ports

The feature to separate traffic by LU is called LOCADDR prioritization and uses the **locaddr priority-list** command. The configuration syntax for this command is as follows:

```
locaddr priority-list 1 02 high
locaddr priority-list 1 05 low
```

This configuration maps every LU addressed 02 to DLSw+ port 2065 and every LU addressed 05 to port 1981.



Next, the **locaddr priority-list** command must be applied to an input LAN interface or bridging domain and can be either Token Ring or a DLSw+ bridge group for Ethernet.

```
interface tokenring 0
locaddr-priority 1

or

dlsw bridge-group 1 locaddr-priority 1
```

Map SNA CoS to IP ToS

The process of mapping SNA CoS to IP ToS involves two steps. First, IP precedence is automatically established by virtue of using the priority keyword in the **dlsw remote-peer** command. Second, APPN CoS to IP ToS is automatically established by using APPN on the router and using DLSw+ to bridge SNA traffic on the same router.

In the example above, SNA or NetBIOS is being bridged using DLSw+ and the priority keyword is used and DLSw+ opens the four sessions on ports 1981, 1982, 1983, and 2065. In Release 11.3 and later, DLSw+ also assigns default IP precedent values to these TCP session ports, as shown in Table 2-1.

Table 2-1 TCP Port-to-IP Precedence Default Mapping

TPC Port	Priority Queue	IP Precedence	Precedence Numeric Value
2065	High	Critical	5
1981	Medium	Flash override	4
1982	Normal	Flash	3
1983	Low	Immediate	2

The default precedence values can be overridden using the **dlsw tos map** command or using policy-based routing. (See “Weighted Fair Queuing”.) The **dlsw tos map** command looks like the following:

```
dlsw tos map high 5 medium 2 normal 1 low 0
```

This example remaps medium priority traffic to immediate precedence, normal priority traffic to priority precedence, and low priority traffic to routine precedence. High-priority traffic remains at critical precedence.

The other method of mapping SNA CoS to IP ToS is when APPN is being routed and uses the DLSw+ VDLC interface as a link layer connection. This is not the same as bridging APPN over DLSw+. In this situation, the router is a full APPN Network Node routing SNA. The APPN Network Node participates fully in the session establishment process where it has easy access to session priority. When the Network Node accesses link layer services, the APPN CoS (APPN transmission priority) is mapped to a DLSw+ port. Table 2-2 lists the default mappings. There are conveniently four APPN transmission priorities and four DLSw+ priority ports as a result of the DLSw+ **priority** keyword.

Table 2-2 APPN CoS to IP ToS Mapping

APPN Mode Names	SNA Transmission Priority	TCP Port	Priority Queue	IP Precedence	Precedence Numeric Value
CPSNASVCMGR	Network	2065	High	Critical	5
#INTER	High	1981	Medium	Flash override	4

APPN Mode Names	SNA Transmission Priority	TCP Port	Priority Queue	IP Precedence	Precedence Numeric Value
#CONNECT	Medium	1982	Normal	Flash	3
#BATCH	Low	1983	Low	Immediate	2

Currently, there is no way to change these default mappings. For example, the mode CPSNASCVMG is assigned network transmission priority mapped to TCP port 2065, the mode #INTER is given high priority that is mapped to port 1981 at IP precedence flash override, #CONNECT is given medium priority, and #BATCH is given low priority. In addition BIND commands and IPM (pacing messages) are given their network transmission priority by APPN. See *APPN Design and Implementation Guide* and the *APPN Architecture Reference (SC-3422)* for more information on APPN transmission priorities and COS.

Choosing a Traffic Prioritization Method

After separating interactive and batch traffic into individual streams, a prioritization method needs to be chosen. Cisco IOS software has a large and growing list of prioritization options, including priority queuing (PQ), custom queuing (CQ), weighted fair queuing (WFQ), and weighted random early detection (WRED).

Prioritization methods collectively can be called “queuing methods,” “output queuing,” or “fancy queuing.” One important factor that is often overlooked is that there must be congestion on an output interface for traffic prioritization to take effect. By strict definition, “congestion” is when there are one or more packets on the interface output queue. Starting in the Release 11.0 time frame, these prioritization methods were fast-switched. At the first sign of congestion, when the hardware FIFO buffer on the serial chip is full or the CBUS transmit queue reaches a certain threshold, the fast switching software copies the packet into a system buffer and queues it on the output queue. The traffic waiting on this output queue gets the prioritization method applied to it. If the fast-switching software detects no congestion, then traffic is fast-switched.

Priority Queuing

This method of traffic prioritization is the oldest available. PQ has four classes of service: high, medium, normal, and low. Traffic is assigned to a service class using the **priority-list** command. An example of this command is as follows:

```
access-list 101 permit tcp any any eq 2065
access-list 101 permit tcp any eq 2065 any

priority-list 1 protocol ip high list 101
```

This example configures an extended access list to identify DLSw+ port 2065 traffic and puts it in the high PQ service class. All remaining traffic defaults to the normal service class queue. PQ will service each class queue in the order of priority in a round robin fashion until it is empty. The risk with using PQ is that the higher priority queues are given all bandwidth if it is required. If high-priority traffic is always present, no other service classes will get service and all other traffic is starved. This starvation issue historically has been the source of many network problems.

In the configuration example above, SNA traffic could potentially get all the bandwidth and preempt any other traffic from being transmitted. However, testing has shown that PQ can provide optimal response times (around .60 second response times on a 64-kbps link with 50 other file transfers active). PQ can be used effectively if the network designer is certain that SNA traffic will only be interactive with reasonable numbers of users sharing the link bandwidth. Ten users on a 64-kbps link is reasonable. 300 users on a 64-kbps link is not.

The final step in implementing PQ is to place the priority list on an outbound interface as follows:

```
interface serial0
priority-group 1
```

Custom Queuing

Because of the bandwidth starvation issues associated with using PQ, CQ was introduced to provide better transmission fairness and more service classes for sharing link bandwidth. Up to 16 service-class queues can be defined. The user can configure service levels for each queue by defining a byte count that limits the amount of bandwidth given to a service class. Just like PQ, traffic is designated for a particular service class by protocol or by using an access lists as follows:

```
access-list 101 permit tcp any any eq 2065
access-list 101 permit tcp any any eq 2065 any
queue-list 1 protocol ip 1 list 101
queue-list 1 default 2
queue-list 1 queue 1 byte-count 2000
queue-list 1 queue 2 byte-count 8000
```

In this configuration, we have defined an access list to identify DLSw+ port 2065 traffic and place it in queue number 1, which has a byte count of 2000. All other traffic (including IP) will go to default queue number 2, which has a byte count of 8000. In this example, bandwidth is allocated in an SNA:IP ratio of 2:8, or 20 percent SNA and 80 percent IP (or other).

Using a byte count and describing CQ in terms of bandwidth allocation is unusual. CQ does not fragment packets, so it cannot stop transmitting in the middle of one. Thus, when defining queue byte counts, you must consider the MTU of the packets.

In the example above, if SNA has a maximum frame size of 1024 and IP the default MTU of 1500, then CQ would send two SNA frames followed by six IP frames. This achieves the desired bandwidth allocation. However, it also highlights one of the drawbacks of CQ, which is that CQ creates packet trains within the class and does not interleave packets well. The amount of time it takes to transmit six 1500-byte IP frames on a 64-kbps link is approximately 1.2 seconds. If the requirement were to maintain a subsecond response time, it would not be achieved.

If optimal response times are desired, it may be best to define byte counts in a way that the maximum size packet train is a single packet. This configuration is shown in the following example:

```
queue-list 1 queue 1 byte-count 1000
queue-list 1 queue 2 byte-count 1000
```

This configuration results in good packet interleaving but potentially gives SNA 50 percent of the bandwidth. In most circumstances SNA will never reach 50 percent bandwidth utilization because of its interactive nature. If SNA file transfers are present, they must be separated into their own queue. An SNA file transfer will demand the 50 percent utilization creating problems for interactive SNA users as well.

In summary, to achieve optimal response times, there is a risk/reward trade off. Using PQ you risk sacrificing the entire bandwidth (100 percent) to priority traffic. With CQ you risk losing half (50 percent) of the bandwidth.

The final step in configuring CQ is to place the CQ list on a serial interface as follows:

```
interface serial0
custom-queue-list 1
```

Weighted Fair Queuing

WFQ is a sophisticated queuing method. It is often referred to as a flow-based, as opposed to class-based, queuing method because it queues traffic per session (TCP or UDP port, and so on). Because of flow-based queuing, scaling WFQ on broadband trunks (speeds greater than E1) may be too CPU intensive. The distributed versatile interface processor (VIP) implementations are more effective in broadband trunk or LAN interface scenarios. Recently, distributed VIP implementations have introduced some class-based fair-queuing methods. However, these are not discussed in this document.

WFQ dynamically determines flow behavior in real-time and favors traffic with an interactive nature over other more aggressive traffic such as batch transfers. At first glance, this looks promising for SNA. However, although DLSw+ represents many interactive flows from many users, it is considered a single flow by WFQ. The concern is that many interactive SNA users will make the DLSw+ session appear bandwidth hungry and WFQ will penalize it. This is considered a major issue with using WFQ and DLSw+, but it is only a minor issue. Recall the interactive nature of SNA 3270 users. Even with 10 or 20 users over the same DLSw+ session, transaction rates are on the order of transactions per minute, which will result in packet rates of the same order. Under normal circumstances, this transaction rate will never reach levels that one would consider high bandwidth.

A greater threat to SNA response times has to do with TCP congestion management (slow-start and back-off). TCP congestion management uses a windowed protocol that automatically adjusts to the bottlenecks along the session path. In extreme circumstances (for example, sharing a 64-kbps link with 20 or 30 FTPs) TCP windows can be reduced to sizes that require every TCP packet to be acknowledged even during large file transfers (in other words, a window size of one). This situation very closely resembles an interactive session that has the adverse effect of creating artificial competition with interactive SNA DLSw+ traffic. Under these conditions WFQ cannot distinguish between the SNA interactive traffic and the TCP/IP batch traffic. As a result, administrative intervention is required to give SNA traffic differentiated service.

WFQ needs some method to distinguish SNA interactive traffic (encapsulated in TCP/IP) from other IP traffic during periods of extreme congestion. One method that can be used is to modify the weight of the DLSw+ traffic by using the precedence bits in the IP header. When WFQ determines the packet scheduling order, the lower the weight, the higher the packet priority. The computed weight is a function of the frame length (or transmission time of completion) and its position in its conversation queue. The length of the packet is reduced using the precedence bits in the IP packet ToS field. Thus, the only advantage SNA can hope to have over other traffic lies in gaining priority (lower weight) from the precedence bits. There are several mechanisms used to set precedence on SNA traffic.

Setting precedence on DLSw+ packets is done by using policy-based routing or by relying on DLSw+. (See “Map SNA CoS to IP ToS.”) Setting precedence on DLSw+ packets using policy-based routing can be done in the following manner:

```
ip local policy route-map SNADLSW

access-list 101 permit tcp any any eq 2065
access-list 101 permit tcp any eq 2065 any

route-map snadlsw permit 10
match ip address 101
set ip precedence critical
```

WFQ is enabled on a physical interface by default. No configuration is required.

Note: WFQ must be used in combination with traffic shaping over Frame Relay to be most effective.



Weighted Random Early Detection

WRED is one of the newer and more sophisticated queuing methods. It is considered a congestion avoidance method because it drops traffic based on mean queue depths instead of tail dropping as do the previous queuing methods. WRED is also considered a class-based queuing method because it deals with traffic based on class definitions. There are nine WRED classes. There is a class for each precedence level plus one for RSVP traffic. Currently, WRED works only for IP traffic. Traffic is placed on a single queue for transmission. Packets are selected for discard based on probabilities (thus the term random). The probability computations used for packet selection are a function of the packet precedence and mean queue depth on the output interface. The probability of a packet discard increases as the precedence decreases and as the mean queue depth increases.

By randomly selecting packets for discard based on probabilities, instead of tail-dropping when queues overflow, WRED resolves a phenomenon called global synchronization and more fairly discards among sessions using the link. Global synchronization can occur when simultaneously tail-dropping many packets across many sessions makes TCP back-off algorithms kick-in at the same time. TCP slow-start ramps up only to repeat the same pattern. This phenomenon has been observed on the Internet. Most enterprise networks today are not subject to this phenomenon, but it is possible in theory.

WRED works in conjunction with the precedence bits to differentiate service the same way WFQ does. Giving DLSw+ higher IP precedence reduces the probability that an SNA packet will be discarded, avoiding retransmissions that negatively impact response times. Note that SNA traffic is not scheduled at a higher priority using WRED. SNA is simply less likely to be dropped.

WRED is not recommended on slow links when using SNA because the single queue implementation can cause some queuing delays for SNA traffic and WRED does no traffic prioritization or packet sorting. WRED is more suitable for broadband trunks because the high speeds provide better algorithmic scaling properties and lessen the problems associated with long queuing delays.

Use the **random-detect** command to enable WRED as follows:

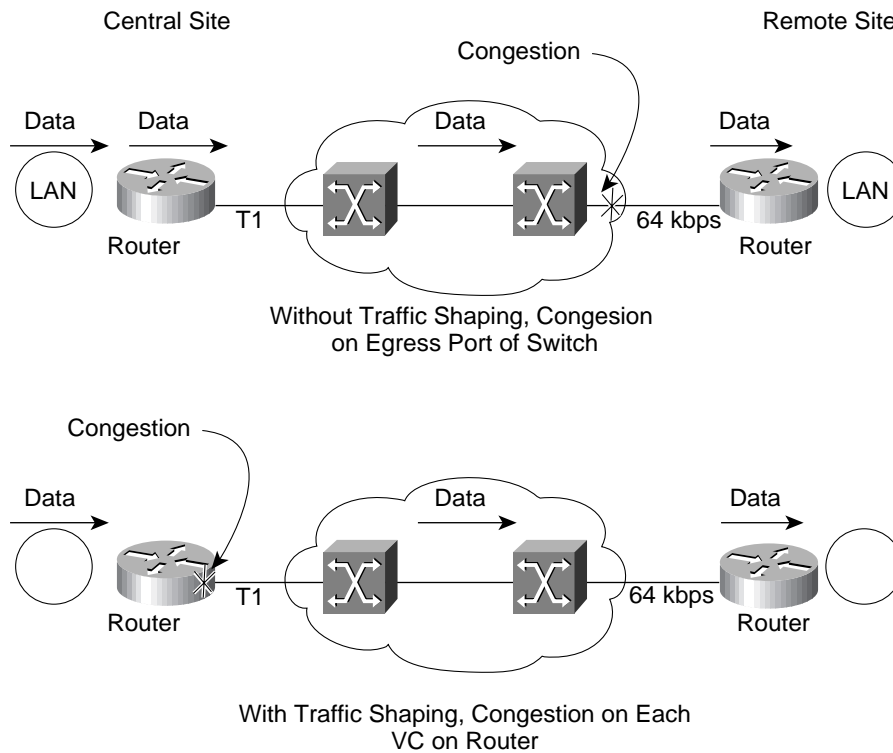
```
interface serial0
random-detect 9
```

Traffic Shaping

Frame Relay networks without traffic shaping cannot effectively prioritize traffic.

For an overview of congestion management in Frame Relay see the Technology Overview chapter. From that chapter, recall that mismatched access speeds can have a negative effect on SNA response times. Congestion occurs on the egress port of the Frame Relay switch where little traffic prioritization takes place. The solution to this problem is to use traffic shaping on the router and move the point of congestion to the router (see Figure 2-2). The concept is simple. Moving congestion from the Frame Relay switch to the router allows all of the above traffic prioritization methods to take place.

Figure 2-2 Effects of Traffic Shaping



An important clarification to make is that traffic shaping in a Frame Relay environment takes place on each DLCI in the router. So, there is a traffic-shaping output queue for every Frame Relay virtual circuit. Traffic shaping applied to each DLCI creates congestion (when traffic rates are higher than the traffic-shaping rates defined), and the congestion effectively creates an output queue for each DLCI. This per DLCI output queue is called the traffic-shaping queue. An output queuing method (such as PQ, CQ, WFQ, or the default FIFO) is applied to the traffic-shaping queue. Therefore, a traffic-shaping queue for each DLCI requires additional system buffer memory to accommodate the additional queuing. In situations where there are large numbers of DLCIs, buffer tuning may be necessary and additional I/O memory may be required on low-end systems that allocate buffer pool from I/O memory.

Another important issue to keep in mind when using traffic shaping in network designs is to ensure that traffic entering the Frame Relay network from the router, on the ingress port of the Frame Relay switch, is never faster than the traffic shaping taking place inside the Frame Relay network. For example, if a router is shaping at 70 kbps and the Frame Relay network is shaping at 64 kbps somewhere across the connection path inside the Frame Relay network there will be queuing delays. Also, it has been observed that Cisco IOS traffic shaping and ForeSight traffic shaping on Cisco WAN switches do not match. Even if both rates are set at 64 kbps, Cisco IOS software seems to send traffic faster than ForeSight. If an adjustment is not made for this difference, then an ingress queue buildup will result on the WAN switch running ForeSight. Packets will not be prioritized properly by the WAN switching equipment, so steps must be taken to ensure that the traffic rate entering the Frame Relay network is always lower than the rate made available by the Frame Relay network. This means traffic-shaping rates should generally be slightly lower on the router than the way they are defined on WAN switches.

There are two kinds of traffic shaping: generic traffic shaping (GTS) and Frame Relay traffic shaping (FRTS).

Generic Traffic Shaping

GTS can be applied only to a physical interface port or subinterface (in other words, internal to IOS an Interface Descriptor Block [IDB] style interface). In a Frame Relay environment, GTS should be used only if it is applied to a point-to-point subinterface. If there is a single DLCI present on each subinterface, it is equivalent to having traffic shaping applied to each DLCI. If the design calls for multipoint DLCIs, then GTS can be applied only to the IDB that holds them (on the major physical interface or subinterface), which is not very effective in a Frame Relay environment. GTS cannot accommodate per-DLCI traffic shaping when using multipoint (FRTS must be used instead).

GTS Traffic Prioritization

Only WFQ is supported on the GTS traffic-shaping queue. GTS also supports RSVP. There is no way to disable WFQ, so FIFO queuing is not supported. Regardless of limitations, GTS and WFQ may be the solution for some designs if the requirements are simple. Configure GTS as follows:

```
interface serial0.1 point-to-point
traffic-shape rate 62000 6000 6000 1000
traffic-shape fecn-adapt
frame-relay interface-dlci 801
```

In addition, GTS supports RSVP flows.

GTS Rate Adaptation

When GTS is applied to an interface with Frame Relay encapsulation, it will implement the rate adaptation algorithm discussed in the “Explicit Congestion Notification” section of the Technology Overview chapter. GTS also responds to FECN frames with a test frame. The commands to enable these features are as follows:

```
interface serial0.1 point-to-point
traffic-shape adaptive 32000
traffic-shape fecn-adapt
```

The adaptive traffic-shaping command will set the lowest rate that GTS will shape in the presence of severe congestion. The severity of the congestion is a function of the rate in which BECN frames are received.

Frame Relay Traffic Shaping

FRTS is also referred to as CIR rate enforcement. It can be used on single DLCI point-to-point subinterfaces or applied to specific DLCIs in a multipoint interface or subinterface. FRTS provides maximum flexibility by defining a map-class and configuring all the relevant traffic-shaping parameters for each DLCI in a multipoint interface or subinterface. The map-class can be applied to a group of multipoint DLCIs on an interface. There is also a provision for overriding a map-class for a particular DLCI via the **interface-dlci** command.

The following configuration shows how to apply a Frame Relay map-class to multipoint DLCIs and how to override the map on an individual DLCI for that interface (DLCI 401):

```
interface serial1/0
  encapsulation frame-relay ietf
  frame-relay traffic-shaping
  frame-relay class 16kcir
  frame-relay lmi-type ansi

interface serial1/0.1 multipoint
  frame-relay interface-dlci 401
  class 32kcir
  frame-relay interface-dlci 402
  frame-relay interface-dlci 403
  frame-relay interface-dlci 404
  frame-relay interface-dlci 405
  frame-relay interface-dlci 406
  frame-relay interface-dlci 407
  frame-relay interface-dlci 408
```

In the above configuration, DLCI 401 gets assigned the 32kcir map-class. All other DLCIs pick up the default map-class 16kcir from the major interface. The following configuration shows how to define the map-class statements:

```
map-class frame-relay 32kcir
  frame-relay cir 32000
  frame-relay bc 2000
  frame-relay be 2000
map-class frame-relay 16kcir
  frame-relay cir 16000
  frame-relay bc 2000
  frame-relay be 2000
```

Frame Relay Traffic Shaping and Traffic Prioritization

FRTS also has a caveat with regard to traffic prioritization. Frame Relay traffic shaping supports FIFO, PQ, and CQ, but not WFQ. Cisco IOS software Release 12.0(4)T will support WFQ (but not RSVP). However, PQ, CQ, and FRTS in combination with multipoint DLCIs can be accomplished. In most large-scale Frame Relay implementations, FRTS is a requirement.

To enable PQ or CQ with FRTS, the PQ or CQ list is applied to the map-class. The following configuration shows how to apply PQ and CQ to a map-class:

```
map-class frame-relay 16kcir
  frame-relay cir 16000
  frame-relay bc 2000
  frame-relay be 2000
  frame-relay priority-group 1

map-class frame-relay 16kcir
  frame-relay cir 16000
  frame-relay bc 2000
  frame-relay be 2000
  frame-relay custom-queue-list 1
```

Using traffic shaping and traffic prioritization methods on the routers is one way to prioritize traffic over a single Frame Relay virtual circuit or DLCI. It is practical because all traffic can be supported on single virtual circuit to each remote site. This is a low-cost solution. However, there are other methods to accomplish the same or similar prioritization objective. The next two sections discuss these alternate prioritization techniques.

Separating Traffic by DLCI

Traffic can be separated on different DLCIs using a feature called DLCI prioritization. This feature places the burden of providing QoS on the Frame Relay network instead of the router. The router does no prioritization of traffic. It is responsible for separating the traffic on different DLCIs. The Frame Relay switch does the prioritization based on DLCI. By doing this, users are relying on the Frame Relay QoS characteristics for the virtual circuit to meet service-level agreements and relying on the WAN switch to prioritize traffic. This feature in Cisco WAN switches is called priority PVC.

In many instances DLCI prioritization is a viable solution. The major deficiency with this solution is the extra cost of additional virtual circuits. In large Frame Relay networks, there is a scalability issue because of this as well. Also, the traffic prioritization mechanisms employed by Frame Relay switches are often crude. The switch creates two traffic priority service classes on the egress port of the Frame Relay switch: high and low. Cisco WAN switches using ForeSight can reduce the possibility of congestion, but when congestion does occur at the egress port, the best the switch can do is give the high-priority PVC a 10:1 ratio of service over the low-priority service class. For the most part, SNA will never need to use all 10 frames allocated to it on the high-priority virtual circuit. However, if severe congestion continues and frame buffers are depleted at the egress point, then SNA frames will be dropped with equal priority as the low-priority traffic.

To configure DLCI prioritization, the router reuses the same **priority-list** command as PQ. The difference is that the priority list is not placed on an outbound interface, but is placed in an interface command that determines which DLCI is designated as the high-priority DLCI, the medium-priority DLCI, the normal-priority DLCI, and the low-priority DLCI.

An example of DLCI prioritization is configured as follows:

```
interface serial 0
  ip address 1.1.1.1 255.255.255.0
  encapsulation frame-relay
  frame-relay priority-dlci-group 2 401 403 403 403

  access-list 101 permit tcp any any eq 2065
  access-list 101 permit tcp any any eq 2065 any

  priority-list 2 protocol ip high list 101
```

In this example, DLCI 401 on the **frame-relay priority-dlci-group** command is given high-priority status. The IP extended access list, access-list 101, puts DLSw+ port 2065 traffic in the high-priority service class. Though not doing any actual prioritization, this configuration sequence establishes the relationship required to place DLSw+ port 2065 traffic into DLCI 401. DLCI 403 gets other traffic assigned to it and the Frame Relay switch must meet the QoS guarantees for the user.

Separating Traffic using Policy Routing

Policy-based routing (PBR) can also be used to direct the flow of traffic over Frame Relay. If Frame Relay point-to-point subinterfaces are in use, then PBR can be used to set the next interface to route the packet. Though no specific testing was done to verify, it is feasible to use policy routing to direct traffic over multipoint Frame Relay DLCIs by setting the next-hop address to the IP address in the Frame Relay map. The idea is the same as DLCI prioritization—direct incoming traffic to different DLCIs so the Frame Relay network can handle the QoS guarantees defined for each virtual circuit.

PBR is the preferred method to separate traffic on different DLCIs. PRB is fast switched (in Release 11.3 and later) and is easier to follow. There are some performance issues with PBR so care must be taken if it is to be used at high speeds. PBR will eventually be integrated with Cisco Express Forwarding (CEF) to address performance concerns. A sample configuration for PBR is as follows:

```
ip local policy route-map snadlsw

interface serial1/0
  encapsulation frame-relay ietf
  frame-relay lmi-type ansi

interface serial1/0.1 point-to-point
  ip address 1.1.1.1 255.255.255.0
  frame-relay interface-dlci 401

interface serial1/0.2 point-to-point
  ip address 1.1.2.1 255.255.255.0
  frame-relay interface-dlci 402

access-list 101 permit tcp any any eq 2065
access-list 101 permit tcp any eq 2065 any

route-map snadlsw permit 10
  match ip address 101
  set next-interface serial1/0.1

route-map snadlsw permit 20
  set default next-interface serial1/0.2
```

In this example, DLSw+ port 2065 traffic goes out on subinterface serial1/0.1, which is assigned DLCI 401. All other traffic by default goes out subinterface serial1/0.2, which is assigned DLCI number 402.