

Bandwidth Management and Queuing

This chapter describes how you can use Cisco's bandwidth management and queuing features in conjunction with DLSw+ to enhance the overall performance of your network.

Many enterprises run Cisco networks with a mixture of SNA and client/server protocols. If you anticipate that because of your traffic volume or bandwidth limitations, there will be contention for bandwidth, read this chapter. In general, the queuing techniques described in this chapter (with the exception of DLCI prioritization and policy routing) do not even take effect unless there is congestion in the network.

Even if you decide you need to apply some of these queuing techniques, you may not need them everywhere. The output queuing mechanisms described in this chapter can be applied to an individual interface, allowing you to apply queuing to lower-speed access lines while not applying it to higher-speed trunk lines.

Note: The queuing mechanisms described in this chapter apply only to TCP encapsulation.

Introduction to Cisco IOS Queuing Features

Bandwidth management involves deciding what traffic is highest priority, ensuring that it gets the bandwidth it needs, and deciding how to handle the lower-priority traffic. The Cisco IOS Software offers many options for identifying high-priority traffic: protocol, message size, TCP port number, input interface address, LLC SAP, MAC address, SDLC address with STUN, or LOCADDR.

DLSw+ places all SNA and NetBIOS traffic into TCP packets, making it difficult or impossible to identify the traffic by the above characteristics. For that reason, DLSw+ supports opening four separate TCP connections and places traffic directly from the input queue into one of these four pipes based on priority. At the output interface, you can prioritize among these four TCP connections based on their TCP port number.

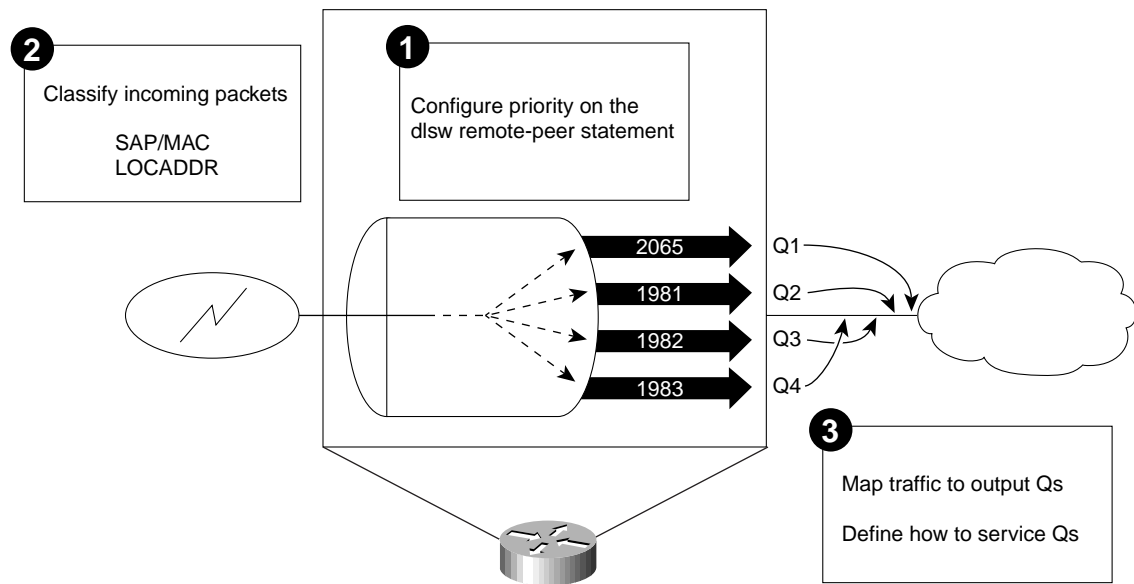
When traffic has been assigned to a queue, the Cisco IOS Software offers several options for servicing the queues. The key techniques for DLSw+ traffic are *custom queuing* and *priority queuing*. In addition, there is *weighted fair queuing* and *DLCI prioritization*. All of these techniques are described in this chapter.

Figure 5-1 describes the tasks required to configure bandwidth management in a Cisco router. There are three steps:

Step 1. If you choose to distinguish within DLSw+ traffic, to prioritize SNA ahead of NetBIOS, or to prioritize interactive terminal traffic over batch print jobs, you need to include the priority keyword in the appropriate dlsw remote-peer command. Including this keyword causes DLSw+ to open four TCP connections (identified by ports 2065, 1981, 1982, and 1983). By default, DLSw+ transmits certain traffic over certain TCP connections.

- Step 2. The next step is to classify packets on the incoming port and assign the traffic to the appropriate TCP connection. This can be done based on SAP, MAC address, or LOCADDR. If you do Step 1, you must also do Step 2 to have any effect on how the bandwidth is allocated. Step 1 opens the TCP pipes. Step 2 assigns traffic to the pipes.
- Step 3. Next, you must assign traffic to the appropriate output queue based on protocol, TCP port number, or message size and then define the queuing technique to be used on the interface (for example, custom queuing or priority queuing). Step 1 and Step 2 may be unnecessary in your environment, but you may still choose to distinguish DLSw+ from other traffic, in which case you need to do Step 3.

Figure 5-1 Tasks Required to Control How Traffic Is Forwarded in Cisco Routers



The queuing of packets only occurs when the total number of outbound packets exceeds the capacity of the outbound link. If a link is not congested, then the router does not need to implement any queuing mechanism, because as soon as it has queued the packet onto the outbound interface, the packet can be sent.

Traffic Classification

The Cisco IOS Software supports packet classification by protocol, by TCP port number, by input interface, by message length, and by extended access list. DLSw+ traffic can be classified ahead of other TCP/IP traffic because by default it always uses TCP port number 2065. To classify traffic within DLSw+, specify the priority keyword in a `dlsw remote-peer` command.

When priority is specified, DLSw+ automatically activates four TCP connections to that remote peer (ports 2065, 1981, 1982, and 1983). Priority needs to be specified only if you need to prioritize between SNA and NetBIOS, or within SNA by LOCADDR, or MAC or SAP pair (known as SAP prioritization). In addition, this granular packet classification is possible only when TCP encapsulation is selected for a specific remote peer. By default DLSw+ assigns certain traffic to specific TCP ports:

- TCP port 2065 defaults to high priority; in the absence of any other configuration, this port carries all circuit administration frames (CUR_cs, ICR_cs, contact SSP frames, disconnect SSP frames, XID, ICR_ex), peer keepalives, and capabilities exchange



- TCP port 1981 defaults to medium priority; in the absence of any other configuration, this port does not carry any traffic
- TCP port 1982 defaults to normal priority; in the absence of any other configuration, this port carries information frames (nonbroadcast datagram frames)
- TCP port 1983 defaults to low priority; in the absence of any other configuration, this port carries broadcast traffic (CUR_ex, Name_query_ex, SSP DATA/DGRM broadcasts)

Note: You can configure specific traffic to go into either port 2065, 1981, or 1983. If you specify priority in the `dlsw remote-peer` command and do nothing else, all data traffic goes in TCP port 1982 and all unspecified traffic goes in TCP port 1982.

You can use classification techniques such as SAP prioritization to change the port assignment of traffic destined for DLSw. However, these techniques have no impact on how the traffic is handled on the output queue. To control how each of the TCP ports is handled on the output queue, you must map the TCP ports to different queue numbers, define the queuing algorithm, and apply that queue list to the output interface.

SAP Prioritization

You can create a priority list that assigns traffic by SAP or MAC address to different TCP ports. You can then apply that list to a LAN interface on a router (support for Ethernet requires Cisco IOS Release 11.0 or later and support for FDDI requires Release 11.2). As traffic enters the router, DLSw+ assigns it to a TCP port and passes it to the appropriate output interface.

To provide a fine granularity in the prioritization of packets, the `priority-list` command allows you to specify any combination of destination SAP (DSAP), source SAP (SSAP), destination MAC (DMAC), and source MAC (SMAC). For example, if you want to prioritize all SNA traffic (SAP 04) over NetBIOS traffic (SAP F0), then only the DSAP or SSAP needs to be specified in the command. In contrast, if you want to give precedence to traffic on a particular LLC2 session, then you must specify all four parameters (DSAP, SSAP, DMAC, SMAC) that uniquely identify a LLC2 session. The command syntax is:

```
sap-priority-list list-number queue-keyword [dsap ds] [ssap ss] [dmac dm] [smac sm]
```

where *list-number* is an arbitrary integer between 1 and 10 that identifies the SAP priority list. The argument *queue-keyword* is a priority queue name or a DLSw+ TCP port name (for example, high, medium, normal, or low).

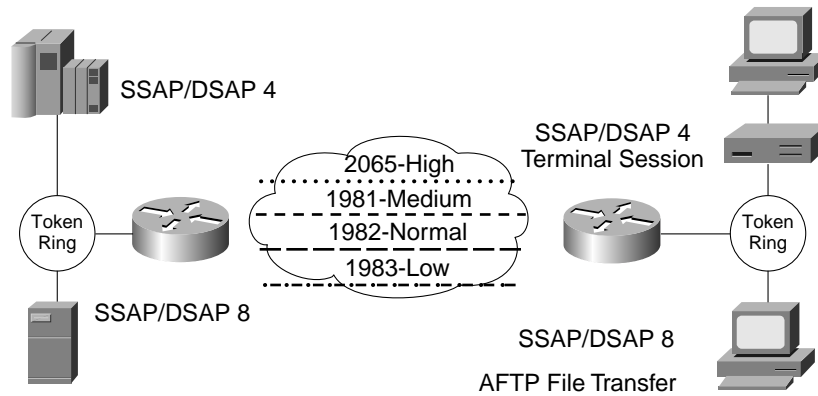
To map a SAP priority list to an Ethernet bridge group (requires Cisco IOS Release 11.0 or later), specify the `sap-priority` keyword on the `dlsw bridge-group` command as follows:

```
dlsw bridge-group group-number sap-priority list
```

where *list* identifies the SAP priority list.

In Figure 5-2, SNA batch and SNA interactive traffic are assigned to different TCP ports so that interactive traffic gets preferential service. This is only possible if batch and interactive traffic have different SSAP and DSAP pairs. In this configuration, traffic from SAP 4 is assigned to TCP port number 2065, and traffic from SAP 8 is assigned to TCP port number 1983. Traffic from all other SAPs is placed in TCP port number 1982 by default. Associating the traffic to different TCP ports allows the router to prioritize one type of traffic over the other types. Classifying packets and queuing them to different TCP ports on the input queue does not determine how the traffic is handled on the output queue. The actual prioritization of the TCP ports on the output queue is handled with other commands that will be described later in this chapter.

Figure 5-2 Traffic Assigned to Different DLSw+ TCP Ports Based on SAP



```
sap-priority-list 1 high ssap 4 dsap 4
sap-priority-list 1 low ssap 8 dsap 8
interface TokenRing0
sap-priority 1
```

Another use of SAP prioritization is to give high priority to traffic destined for a FEP by using an output queuing mechanism in conjunction with the following command:

```
sap-priority-list 10 high dmac 4001.3745.0001
```

SAP prioritization only applies for LAN-attached devices when using TCP encapsulation to connect to remote peers. SAP prioritization cannot be used in conjunction with LOCADDR prioritization. If both are specified, LOCADDR takes precedence.

LOCADDR Prioritization

LOCADDR is the SNA local address assigned by an SNA boundary network node (PU 4/5) to uniquely identify a dependent SNA LU. (For independent LUs, the LOCADDR is assigned dynamically during session establishment and cannot be used to distinguish between application types.) LOCADDR is carried in the SNA format indicator 2 (FID2) headers that are used when a PU 2.0/2.1 communicates with a PU 4/5.

When DLSw+ is used to transport data between PU 2.0/2.1 and PU 4/5, you can prioritize SNA traffic by LOCADDR. To do this, create a priority list that assigns traffic based on LOCADDR to different TCP ports. Then apply that list to a Token Ring or SDLC interface on a router. As traffic enters the router, DLSw+ assigns it to a TCP port and passes it to the appropriate output interface.

To provide fine granularity in the prioritization of packets, the `locaddr-priority-list` command allows you to prioritize individual LUs. For example, this command lets you prioritize interactive devices ahead of printers.

The command syntax is:

```
locaddr-priority-list list-number address-number queue-keyword
```

where *list-number* is an arbitrary integer between 1 and 10 that identifies the priority list. The argument *address-number* uniquely identifies an SNA device.

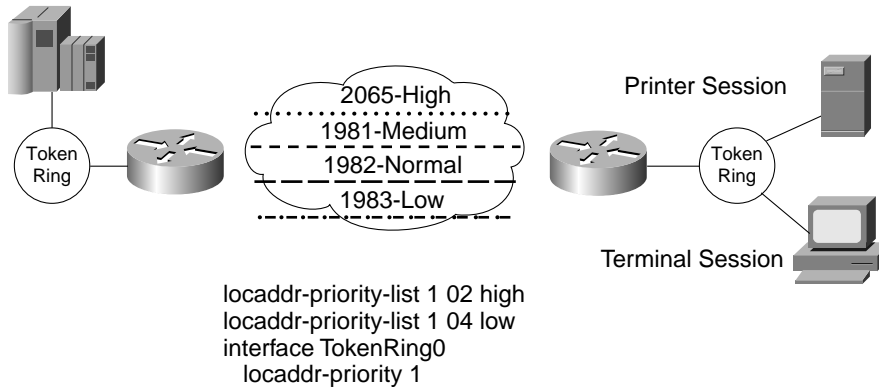
To map a LOCADDR priority list to an Ethernet bridge group (requires Cisco IOS Release 11.0 or later), specify the `locaddr-priority` keyword on the `dls w bridge-group` command as follows:

```
dls w bridge-group group-number locaddr-priority list
```

where *list* identifies the SAP priority list.

In Figure 5-3, the printer (at LOCADDR 4) is assigned to TCP port 1983. A specific terminal or set of terminals can be assigned to TCP port 2065. All other DLSw+ traffic defaults to TCP port 1982. Classifying packets into different TCP ports on the input queue does not determine how the traffic is handled on the output queue. The actual prioritization of the TCP ports on the output queue is handled with other commands that will be described later in this chapter.

Figure 5-3 Traffic Assigned to Different DLSw+ TCP Ports Based on LOCADDR



LOCADDR prioritization applies to dependent LUs attached to DLSw+ via QLLC, SDLC, Token Ring, Ethernet, or FDDI when using TCP encapsulation to communicate with remote peers. LOCADDR prioritization cannot be used in conjunction with SAP prioritization. If both are specified, LOCADDR takes precedence.

SNA ToS

DLSw+ type of service (ToS) is another method for providing granularity and ensuring prioritization for your SNA traffic. It maps SNA Class of Service (COS) to IP ToS, ensuring priority is preserved across the IP network. When DLSw+ is used in conjunction with APPN, SNA ToS maps APPN COS to IP ToS, and preserves SNA COS across an IP network.

When the priority option on the `dlsw remote-peer` command is specified, DLSw+ automatically activates four TCP connections to the remote peer, sets IP Precedence values and assigns traffic to specific ports according to the rules defined in Table .

Table 5-1 TCP Port-to-IP Precedence Default Mapping

TCP Port	DLSw+ Priority Queue	IP Precedence Value	IP Precedence 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 `dls w tos map` command or by using policy-based routing. In the following example, the medium-priority traffic is remapped to immediate IP precedence; normal-priority traffic is mapped to priority IP precedence; and low-priority traffic is mapped to routine IP precedence (the high-priority traffic remains at critical precedence):

```
dls w tos map high 5 medium 2 normal 1 low 0
```

After opening the four pipes and separating the traffic into individual queues, apply weighted fair queuing to service the queues.

When DLSw+ is used in conjunction with APPN, ToS maps APPN COS to IP ToS and preserves SNA COS across an IP network. For example, SNA batch can be prioritized higher than SNA interactive traffic. When the priority keyword is specified on the `dls w remote-peer` command, DLSw+ automatically activates four TCP connections to the remote peer, sets IP precedence values and assigns traffic to specific ports according to the rules defined in Table 5-1.

Table 5-2 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
#CONNECT	Medium	1982	Normal	Flash	3
#Batch	Low	1983	Low	Immediate	2

APPN and DLSw+ must be running in the same router for APPN COS to IP ToS mapping to occur. ToS only applies to TCP and FST encapsulation types. When using FST encapsulation, ToS marks all DLSw+ traffic with IP precedence “network.” The user cannot alter these default mappings.

Queuing Algorithms

The Cisco IOS Software implements four different output queuing algorithms:

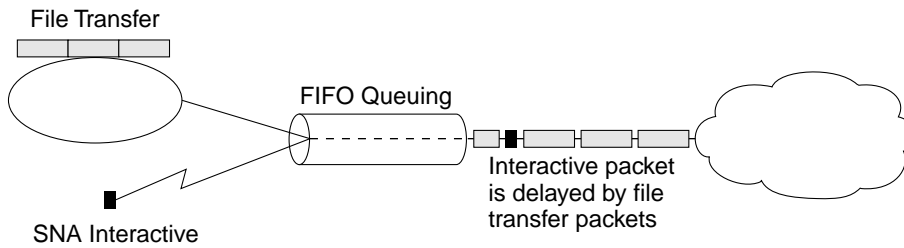
- First in, first out queuing
- Priority queuing
- Custom queuing
- Weighted fair queuing

Each queuing method has advantages and disadvantages. This section describes how each one works and shows configuration examples.

First In, First Out Queuing

This is the simplest and most common interface queuing technique and works well if links are not congested. It is the default queuing mechanism for any interface with more than 2 MB bandwidth. The first packet to be placed on the output interface queue is the first packet to leave the interface (see Figure 5-4). The problem with first in, first out queuing is that when a station starts a file transfer, it can consume all the bandwidth of a link to the detriment of interactive sessions. The phenomenon is referred to as a packet train because one source sends a “train” of packets to its destination and packets from other stations get caught behind the train. First in, first out queuing is effective for large links that have little delay and minimal congestion.

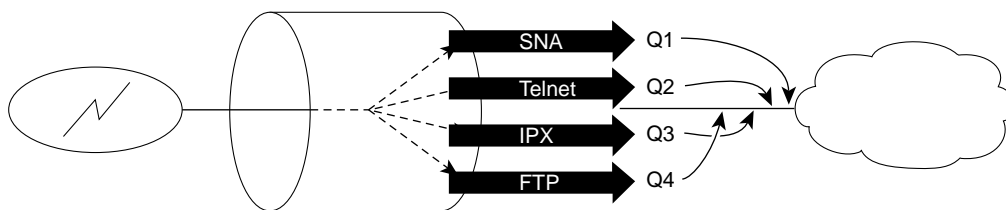
Figure 5-4 Potential Impact of a File Transfer on Interactive Traffic



Priority Queuing

Priority queuing allows network managers to define how they wish traffic to be prioritized in the network. By defining a series of filters based on packet characteristics, traffic is placed into a number of queues; the queue with the highest priority is serviced first, then the lower queues are serviced in sequence (see Figure 5-5). If the highest priority queue is always full, then this queue is continually serviced and packets from the other queues queue up and are dropped. In this queuing algorithm one particular kind of network traffic can dominate all others. Priority queuing assigns traffic to one of four queues: high, medium, normal, and low.

Figure 5-5 Priority Queuing Services Traffic on the Highest Priority Queue First



```

Interface Serial1
ip address 20.0.0.1 255.0.0.0
 priority-group 1
!
priority-list 1 protocol ip high tcp 2065
priority-list 1 protocol ip medium tcp 23
priority-list 1 protocol ipx normal
priority-list 1 protocol ip low tcp 21
    
```

In Figure 5-5, the priority-group command assigns priority list 1 to Serial1. The priority-list command defines the queuing algorithm to be used by queue list 1 and maps the traffic into various queues. Priority queuing is useful when you want to guarantee that the DLSw+ traffic will get through even if it delays other types of traffic. It works best if the DLSw+ traffic is low volume (for example, a small branch with a transaction rate of five to ten transactions per minute), and the number of queues is kept to a minimum (two or three). In this configuration, DLSw+ is in the highest-priority queue, Telnet (TCP port 23) is in the medium queue, IPX is in the normal queue, and FTP (TCP port 21) is in the lowest-priority queue.

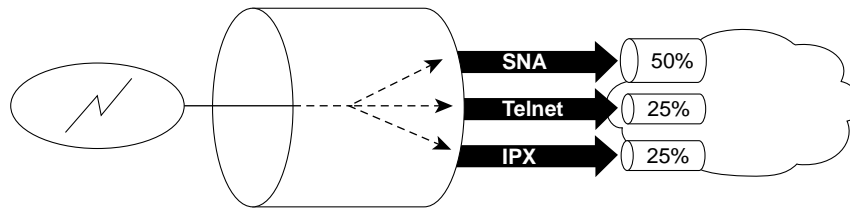
Custom Queuing

Custom queuing, or bandwidth allocation, reserves a portion of the bandwidth of a link for each selected traffic type. To configure custom queuing, the network manager must determine how much bandwidth to reserve for each traffic type. If a particular type of traffic is not using the bandwidth reserved for it, then other traffic types may use the unused bandwidth.

Custom queuing works by cycling through the series of queues in round-robin order and sending the portion of allocated bandwidth for each queue before moving to the next queue. If one queue is empty, the router sends packets from the next queue that has packets ready to send. Queuing of packets is still first in, first out in nature in each classification (unless APPN is running in the router, in which case the queue is ordered by SNA transmission priority), but bandwidth sharing can be achieved between the different classes of traffic.

In Figure 5-6, custom queuing is configured to take 4000 bytes from the SNA queue, 2000 bytes from the Telnet queue, and 2000 bytes from the default queue. This allocates bandwidth in the proportions of 50, 25, and 25 percent. If SNA is not using all its allocated 50 percent of bandwidth, the other queues can utilize this bandwidth until SNA requires it again.

Figure 5-6 Custom Queuing Removes Specified Byte Count of Traffic from Each Queue in Round-Robin Fashion, Allocating the Bandwidth Proportionally Among the Queues



```
Interface Serial0
ip address 20.0.0.1 255.0.0.0
  custom-queue-list 1
  !
queue-list 1 protocol ip 1 tcp 2065
queue-list 1 protocol ip 2 tcp 23
queue-list 1 default 3
queue-list 1 queue 1 byte-count 4000
queue-list 1 queue 2 byte-count 2000
queue-list 1 queue 3 byte-count 2000
```

Custom queuing is commonly used when deploying DLSw+ networks because it allows the network manager to ensure that a guaranteed percentage of the link can be used for SNA, Telnet, and FTP. However, unless the DLSw+ traffic is broken into separate TCP conversations (using SAP or LOCADDR prioritization described earlier), batch SNA transfer or NetBIOS traffic shares the same output queue and may negatively impact interactive SNA response times.

In Cisco IOS Release 11.0, the number of queues available for custom queuing was increased from 10 to 16. The byte counts you should assign to each queue depend upon the bandwidth of the link and the message sizes of the protocols. Byte counts that are too high may adversely skew the performance of custom queuing on low-speed interfaces.

Considerations

When choosing the byte count values for each queue you must consider the following:

- When the byte count value is exceeded, the frame that is currently being transmitted is completely sent. Therefore, if you set the byte count to 100 bytes and the frame size of your protocol is 1024 bytes, then every time this queue is serviced, 1024 bytes are sent, *not* 100 bytes.
- Very large byte counts produce a “jerky” distribution. That is, if you assign 10,000, 15,000, 20,000, and 25,000 to four queues, each protocol is serviced nicely when its queue is the one being serviced, but after it is serviced, it may take some time to get back to that queue.
- Window size also affects the bandwidth distribution. If the window size of a particular protocol is set to one, then that protocol does not place another frame in the queue until it receives an acknowledgment. The custom queuing algorithm moves to the next queue if the byte count is exceeded or there are no frames in that queue. Therefore, with a window size of one, only one frame is sent each time. If your byte count is set to 2 KB and your frame size is 256 bytes, then only 256 bytes are sent each time this queue is serviced.
- You need to know the frame size of each protocol. Some protocols, such as IPX, negotiate the frame size at session startup time.

Determining the Byte Count

To ensure that the actual bandwidth allocation is as close as possible to the desired bandwidth allocation, you must determine the byte count based on each protocol’s frame size. Without doing this, your percentages may not match what you configure.

For example, suppose one protocol has 500-byte frames, another has 300-byte frames, and a third has 100-byte frames. If you want to split the bandwidth evenly across all three protocols, you might choose to specify byte counts of 200, 200, and 200 for each queue. However, that does not result in a 33:33:33 ratio because when the router serviced the first queue, it would send a single 500-byte frame; when it serviced the second queue, it would send a 300-byte frame; and when it serviced the third queue, it would send two 100-byte frames, giving you an effective ratio of 50:30:20. Had you instead specified 1000, 1000, 1000, the router would send two 500-byte frames, five 200-byte frames, and ten 100-byte frames with a bandwidth ratio of exactly 33:33:33.

However, the delay to send 1000 bytes might be too large. Another alternative is to specify 500, 600, 500, which will result in a ratio of 31:38:31 and may be acceptable.

Fortunately, you do not have to use trial and error to determine the correct byte counts. To determine byte counts, follow these steps:

- Step 1. Produce a ratio of all frame sizes, dividing into the largest frame size. For example, assume that the frame size for protocol A was 1086 bytes, for protocol B was 291 bytes, and for protocol C was 831 bytes. The ratios would be:
$$1086/1086: 1086/291: 1086/831$$
- Step 2. Now multiply the results by the percentages of bandwidth you want each protocol to have. In this example we will allocate the following percentages: 20 percent for A, 60 percent for B, and 20 percent for C. This gives us:
$$1086/1086(0.2): 1086/291(0.6): 1086/831(0.2)$$

or

$$.2: 2.239: 0.261$$
- Step 3. Normalize the ratio by dividing each value by the smallest value, that is:
$$.2/.2: 2.239/.2: .261/.2$$

or

$$1:11.2:1.3$$

This is the ratio of the number of frames that must be sent so that the percentage of bandwidth that each protocol uses is approximately in the ratio of 20, 60, and 20 percent.

Step 4. Note that any fraction in any of the ratio values means that an additional frame will be sent. In the example above, the number of frames sent would be one 1086 byte frame, twelve 291-byte frames, and two 831-byte frames, or 1086, 3492, and 1662 bytes, respectively, from each queue. These are the byte counts you would specify in your custom queuing configuration. To determine the bandwidth distribution this represents, first determine the total number of bytes sent after all three queues are serviced:

$$(1 \times 1086) + (12 \times 291) + (2 \times 831) = 1086 + 3492 + 1662 = 6240$$

Then determine the percentage of the 6240 bytes that was sent from each queue:

$$1086/6240, 3492/6240, 1662/6240 = 17.4, 56, \text{ and } 26.6 \text{ percent}$$

As you can see, this is close to the desired ratio of 20:60:20. The resulting bandwidth allocation can be tailored further by multiplying the original ratio of 1:11.2:1.3 by an integer, and trying to get as close to three integer values as possible. For example, if we multiply the ratio by 2, we get 2:22.4:2.6. We would now send two 1086-byte frames, twenty-three 291-byte frames, and three 831 byte frames, or 2172+6693+2493, for a total of 11358 bytes. The resulting ratio is 19:59:22 percent, which is much closer to the desired ratio than we achieved above.

Do not forget that using a very large byte count may cause other problems.

Custom Queuing Configuration

Following is a basic configuration used for custom queuing with SAP prioritization:

```
ssap-priority-list 1 low ssap F0 dsap F0
locaddr-priority-list 1 2 high
locaddr-priority-list 1 3 low
locaddr-priority-list 1 4 medium
source-bridge ring-group 3
dlsw local-peer peer-id 136.222.2.
dlsw remote-peer 0 tcp 136.222.1.1 priority
!
interface Ethernet0
 ip address 128.207.1.152 255.255.255.0
!
interface Serial0
 ip address 136.222.10.2 255.255.255.0
 no keepalive
 custom-queue-list 3
!
interface Serial1
 ip address 136.222.20.2 255.255.255.0
 no keepalive
 custom-queue-list 3
!
interface TokenRing0
 ip address 136.222.2.1 255.255.255.0
 ring-speed 16
 source-bridge active 2 1 3
 source-bridge spanning
 sap-priority 1
 locaddr-priority 1
!
router igrp 100
 network 136.222.0.0
!
router igrp 109
 network 131.108.0.0
!
```

```
queue-list 3 protocol ip 1 tcp 2065
queue-list 3 protocol ip 2 tcp 1981
queue-list 3 protocol ip 3 tcp 1982
queue-list 3 protocol ip 4 tcp 1983
queue-list 3 protocol ip 5
queue-list 3 protocol ipx 6
queue-list 3 default 7
queue-list 3 queue 1 byte-count 1200
queue-list 3 queue 4 byte-count 1200
queue-list 3 queue 5 byte-count 1200
queue-list 3 queue 6 byte-count 1200
queue-list 3 queue 7 byte-count 500
```

The default byte count for queues 2 and 3 is 1500 even though it does not appear in the configuration.

Weighted Fair Queuing

Weighted fair queuing classifies traffic into conversations and applies priority (or weights) to identified traffic to determine how much bandwidth each conversation is allowed relative to other conversations. Conversations are broken into two categories: those requiring large amounts of bandwidth and those requiring a relatively small amount of bandwidth. The goal is to always have bandwidth available for the small bandwidth conversations and allow the large bandwidth conversations to split the rest proportionally to their weights.

Cisco implements bitwise round-robin fair queuing in Cisco IOS Release 11.0 and later. The prime advantage of fair queuing is that it requires no configuration from the network manager because the router automatically classifies packets passing through an interface into conversations, based on the following:

- TCP/UDP port address
- IP source/destination address, protocol type, type of service
- Frame Relay DLCI
- X.25 logical channel number (LCN)
- SRB frame MAC/SAP

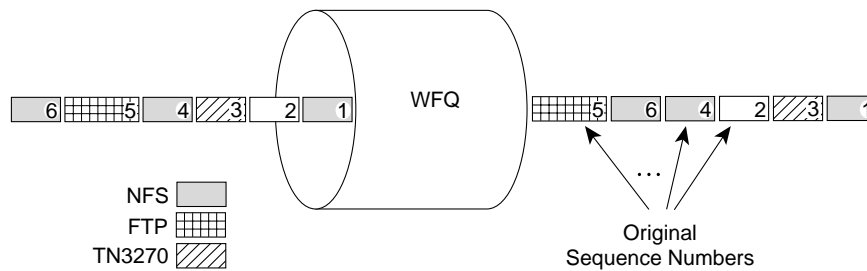
In each case, enough of the packet is checked to break down the streams of packets into separate conversations.

A key disadvantage is that weighted fair queuing does not offer as precise a control over the bandwidth allocation as custom queuing. In addition, in SNA environments, weighted fair queuing typically sees multiple SNA conversations as a single conversation. For example, DLSw+ uses either one or four TCP ports. APPN uses a single LLC2. Hence, instead of SNA interactive sessions moving to the front of the queue, DLSw+ TCP pipes may move to the back of the queue, depending on the number of sessions and quantity of traffic being sent over DLSw+. It is possible, however, to weight certain queues more favorably, which is recommended when using weighted fair queuing in conjunction with DLSw+ or other SNA features. This topic is covered toward the end of this chapter. In general, do not view weighted fair queuing as an alternative to custom queuing or priority queuing in SNA environments, but simply as a better means of handling default queuing when compared to first in, first out.

In weighted fair queuing, packets between active conversations are reordered so that low-volume conversations are moved forward and high-volume conversations are moved toward the tail of the queue. This reordering results in packet trains being broken up and low-volume conversations receiving preferential service. The high-volume conversations share the delay induced by reordering equally, whereby no one conversation is affected more than another.

In Figure 5-7, packets arrive at the router in the order indicated on the left. They are then reordered according to the size and volume of the three conversations so that the packet from conversation 3 (TN3270) is sent second.

Figure 5-7 Weighted Fair Queuing Reorders Packets on the Output Queue, and Packets within a Single Conversation Are not Reordered



The weighting in weighted fair queuing is currently affected by two mechanisms: IP precedence and Frame Relay discard eligible (DE), forward explicit congestion notification (FECN), and backward explicit congestion notification (BECN). The IP precedence field has values between 0 (the default) and 7. As the precedence value increases, the algorithm allocates more bandwidth to that conversation, which allows it to transmit more frequently.

In a Frame Relay network, the presence of congestion is flagged by the FECN and BECN bits. When congestion is flagged, the weights used by the algorithm are altered so that the conversation encountering the congestion transmits less frequently.

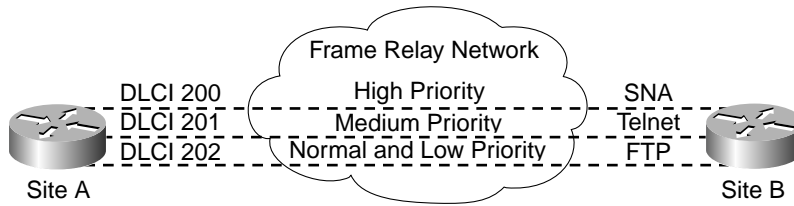
DLCI Prioritization

DLCI prioritization is a process where different traffic types are placed on separate DLCIs so that the Frame Relay network can provide a different CIR for each traffic type. Priority queuing provides bandwidth management control over the access link to the Frame Relay network. Frame Relay switches (for example, the Stratacom IPX, IGX, and BPX/AXIS switches) provide prioritization within the Frame Relay cloud. In other words, the DLCI does not prioritize the traffic, it separates the traffic so that Frame Relay can prioritize it based on the DLCI number. This feature was introduced in Cisco IOS Release 11.0.

In Figure 5-8, SNA traffic is placed on the first DLCI, Telnet is placed on the second DLCI, and all other traffic is placed on the third DLCI. (The first DLCI number corresponds to high, the second to medium, and so on.) Traffic can be differentiated up to four different DLCIs with this feature. CIRs for each DLCI can then be set to a CIR Be and Bc value appropriate to the characteristics of traffic being sent across the DLCI. The following configuration shows how to use DLCI prioritization to place DLSw+ traffic on DLCI 200, Telnet on DLCI 201, and all other traffic on DLCI 202:

```
Interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.200 point-to-point
ip address 20.0.0.1 255.0.0.0
priority-group 2
frame-relay priority-dlci-group 2 200 201 202 202
!
priority-list 2 protocol ip high tcp 2065
priority-list 2 protocol ip medium tcp 23
priority-list 2 default low
```

Figure 5-8 DLCI Prioritization Places SNA, Telnet, and FTP Traffic on Different DLCIs



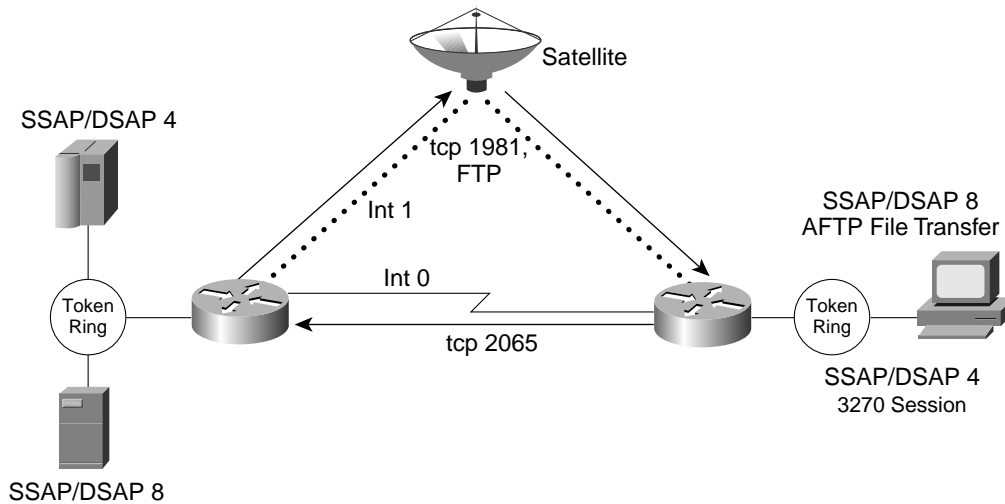
Directing Traffic Flows with Policy Routing

Policy routing is the ability to specify the path that traffic will take through the network or the priority it will receive, based on user-specified parameters.

By using policy routing, a network administrator can control the traffic path, bypassing the normal routing tables. This can be useful where transmission lines between two points have differing characteristics.

In Figure 5-9, there is a low-bandwidth terrestrial link with a low-propagation delay between two points, and a high-bandwidth, high-propagation delay satellite link. The low-bandwidth SNA interactive traffic would be best directed across the terrestrial link, and FTP and SNA file transfers across the satellite link. Policy routing, which was introduced in Cisco IOS Release 11.0, allows you to achieve this.

Figure 5-9 Policy Routing and SAP Prioritization Direct Traffic across Links that Best Meet the Services Requirements of the Traffic



To achieve the result shown in Figure 5-9, use the following configuration:

```

source-bridge ring-group 100
dls local-peer peer-id 4.0.0.4
dls remote-peer 0 tcp 5.0.0.5 priority----->priority keyword opens 4 TCP ports
interface TokenRing0
  ring-speed 16
  sap-priority 1----->maps a sap-priority list to an
  interface
  source-bridge 1 1 100
  source-bridge spanning
ip policy route-map test----->use policy routing for ip traffic from this
ring
sap-priority-list 1 high ssap 4 dsap 8----->assigns terminal sessions to high
sap-priority-list 1 low ssap 8 dsap 8-----> assigns AFTP sessions to low
interface Serial 0
  ip address 20.0.0.1 255.0.0.0
interface Serial 1
  ip address 30.0.0.1 255.0.0.0
  ip local policy route-map test ----->use policy routing for IP originating in this
  rtr
access-list 101 permit tcp any any eq 2065----->permit port 2065 with any ip address
access-list 102 permit tcp any any eq 1981----->AFTP traffic (now in tcp 1981)
access-list 102 permit tcp any eq 20 any ----->FTP traffic
route-map test permit 3----->Defined default path
  set default int serial 0
route-map test permit 2 ----->Define route map "test" 2
  match ip address 102----->all ip addresses that pass filter
  102
  set ip next-hop 30.0.0.7
route-map test permit 1 ----->Define route map "test" 1
  match ip address 101----->all ip addresses that pass filter
  101
  set ip next-hop 20.0.0.6

```

The configuration shows how to use a combination of techniques to prioritize traffic across a WAN. The configuration for policy routing is achieved via route maps. Interface Serial 0 is connected to the terrestrial land line, and Serial 1 is connected to the satellite. Policy routing causes the routing table (which is normally used for forwarding packets) to be ignored and the network administrator's rules to be applied to the forwarding of packets.

You can also use policy routing to determine routing priorities. Policy routing allows you to classify traffic and set the appropriate IP precedence value. In this manner you can sort the network traffic into various types of service at the perimeter of the network and implement those types of service in the core of the network using priority, custom, or weighted fair queuing. As mentioned earlier, the weighting in weighted fair queuing is determined by the value of the IP precedence field. As the precedence value increases, more bandwidth is allocated to that conversation, which allows it to transmit more frequently. This eliminates the need to explicitly classify the traffic at each WAN interface in the core network.

Precedence is a field in the IP header that is used to determine the priority of a packet. Most applications do not set this field so it is typically set to zero. There are eight possible values for the precedence field (see Table 5-3).

Table 5-3 Precedence Field Values

Value	Definition
Network	Match packets with network control precedence ¹ (7)
Internet	Match packets with internetwork control precedence ¹ (6)

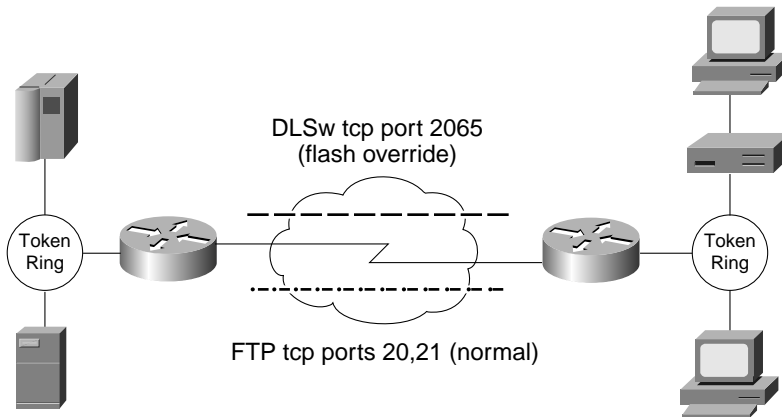
Table 5-3 Precedence Field Values (Continued)

Value	Definition
Critical	Match packets with critical precedence (5)
Flash Override	Match packets with flash override precedence (4)
Flash	Match packets with flash precedence (3)
Immediate	Match packets with immediate precedence (2)
Priority	Match packets with priority precedence (1)
Routine	Match packets with routine precedence (0)

1. Reserve this value for network critical traffic

By modifying the precedence value, you can increase the amount of bandwidth that weighted fair queuing allocates to the conversation. For example, by giving DLSw+ traffic a precedence of critical, as shown in Figure 5-10, the DLSw+ conversation (which is all DLSw+ traffic on a given TCP connection) is given higher priority than an FTP conversation going across the same link.

Figure 5-10 Use Policy Routing to Set the Precedence Bits to Give DLSw+ More Weight



If you are using DLSw+ in a weighted fair queuing environment, it is important to configure DLSw+ with more weight, because a single DLSw+ peer connection carries many discrete conversations. Weighted fair queuing only sees one conversation.

The following configuration uses policy routing with weighted fair queuing to set the precedence bits to give DLSw+ higher priority:

```
source-bridge ring-group 100
dlsw local-peer peer-id 4.0.0.4
dlsw remote-peer 0 tcp 5.0.0.5
interface Serial 0
 ip address 20.0.0.1 255.0.0.0
 ip local policy route-map test-----turns on policy routing
access-list 101 permit tcp any any eq 2065-----allows any ip address w/port 2065
route-map test permit 20
match ip address 101-----all ip addresses that pass filter 101
set ip precedence flash-override
```

RSVP Bandwidth Reservation

Resource Reservation Protocol (RSVP) bandwidth reservation allows DLSw+ to reserve network bandwidth for TCP connections between DLSw+ peers. The user specifies the amount of RSVP reserved bandwidth in the following ways:

- *Globally*—When the user configures the `dlsw rsvp` command, DLSw+ uses these values for initiating RSVP to all its peers. After RSVP is globally enabled, the user must enable RSVP on specific peers. The user can retain the *average-bit-rate* and *maximum-burst* values set in the `dlsw rsvp` command or the user can override these values for any particular peer connection (remote, promiscuous, or peer-on-demand) by configuring the `dlsw remote peer`, `dlsw prom-peer-defaults`, or `dlsw peer-on-demand-defaults` command.
- *Per peer*—When the user configures the `dlsw remote peer tcp` command, DLSw+ configures the RSVP parameters specifically for this peer connection.
- *Type of peer connection*—When the user configures either the `dlsw peer-on-demand-defaults` or `dlsw prom-peer defaults` command, DLSw+ uses the configured RSVP parameters for peer-on-demand and promiscuous peer connections, respectively.

In any of these situations, the user turns off RSVP by setting the *average-bit-rate* or *maximum-burst* values to 0.

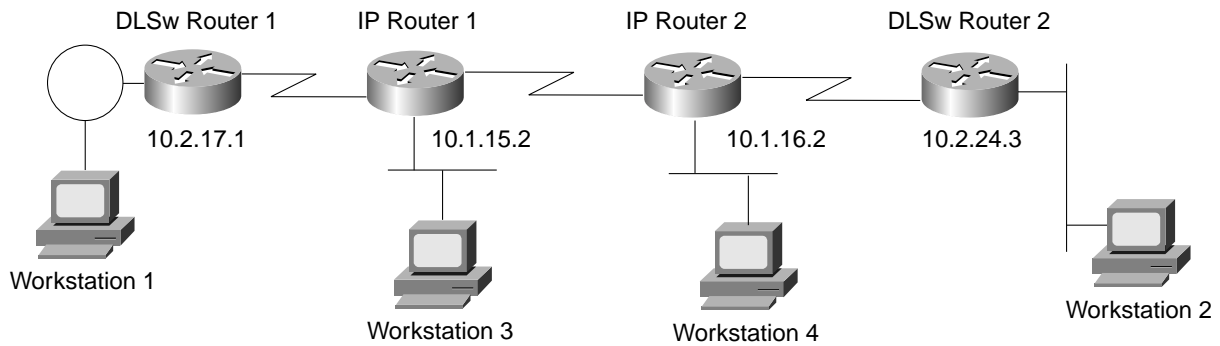
Because RSVP requires both a sender and a receiver, the DLSw+ RSVP bandwidth reservation feature must be implemented on both devices of a DLSw+ connection. However, RSVP does not need to be configured on all devices that are in the IP routed path between two DLSw+ peers. In this type of configuration the devices in the middle must support only IP RSVP; they do not need to be configured for the new DLSw+ RSVP bandwidth reservation feature or of DLSw+. The devices between the peers prioritize the IP packets belonging to the DLSw+ session according to the IP ToS settings. If, however, the devices in the middle do not support IP RSVP, end-to-end bandwidth is not guaranteed.

In the case of priority peers, RSVP bandwidth reservation is done only for the highest priority connection to the peer (TCP port 2065). If the user configures priority queuing and RSVP on the same peer, the user must ensure that the RSVP designated traffic is assigned to the highest priority TCP peer connection.

If the users change the *average-bit-rate* or *maximum-burst* settings without removing the existing RSVP bandwidth reservation, a message warns the users that they are removing the existing reservation and that they need to request a new reservation with new values.

In Figure 5-11, DLSWRTR 1 and DLSWRTR 2 are configured for the DLSw+ RSVP bandwidth reservation feature with an average bit rate of 40 and a maximum-burst rate of 10.

Figure 5-11 Sample Configuration of DLSw+ with RSVP



```
DLSw Router 1
dlsw local-peer peer-id 10.2.17.1
dlsw remote-peer 0 tcp 10.2.24.3
dlsw rsvp 40 10
```

```
DLSw Router 2
dlsw local-peer peer-id 10.2.24.3
dlsw remote-peer 0 tcp 10.2.17.1
dlsw rsvp 40 10
```

The following show commands are useful in verifying the DLSw+ RSVP feature:

- `show ip rsvp request`—Verifies whether the RSVP RESV messages for DLSw+ are sent all the way through the RSVP network to the remote peer
- `show ip rsvp reservation`—Verifies that the RSVP bandwidth reservations are in place for the DLSw+ peers
- `show ip rsvp sender`—Verifies that the RSVP PATH messages for DLSw+ are sent and that the feature is working correctly

To disable the DLSw+ RSVP bandwidth reservation feature for all peers, issue the global configuration `no dlsw rsvp` command. Setting the *average-bit-rate* and *maximum burst* values to 0 in the `dlsw remote-peer tcp`, `dlsw prom-peer defaults`, and `dlsw peer-on-demand defaults` commands turns off RSVP for a particular peer connection. The reservations made by the DLSw+ RSVP commands can be deleted by the global RSVP commands (for example, `no ip rsvp reservation`).

