



Chapter Goals

- Introduce QoS concepts.
- Define QoS tools.
- Discuss QoS tools capabilities.
- Discuss examples of QoS tool usage.

Quality of Service Networking

Introduction

Quality of Service (QoS) refers to the capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet and 802.1 networks, SONET, and IP-routed networks that may use any or all of these underlying technologies. The primary goal of QoS is to provide priority including dedicated bandwidth, controlled jitter and latency (required by some real-time and interactive traffic), and improved loss characteristics. Also important is making sure that providing priority for one or more flows does not make other flows fail. QoS technologies provide the elemental building blocks that will be used for future business applications in campus, WAN, and service provider networks. This chapter outlines the features and benefits of the QoS provided by the Cisco IOS QoS.



Note

A flow can be defined in a number of ways. One common way refers to a combination of source and destination addresses, source and destination socket numbers, and the session identifier. It can also be defined more broadly as any packet from a certain application or from an incoming interface. Recent identification tools have allowed the definition of a flow to be performed more precisely (for instance, to the URL or MIME type inside an HTTP packet). Within this chapter, references to a *flow* could be any one of these definitions.

The Cisco IOS QoS software enables complex networks to control and predictably service a variety of networked applications and traffic types. Almost any network can take advantage of QoS for optimum efficiency, whether it is a small corporate network, an Internet service provider, or an enterprise network. The Cisco IOS QoS software provides these benefits:

- **Control over resources**—You have control over which resources (bandwidth, equipment, wide-area facilities, and so on) are being used. For example, you can limit the bandwidth consumed over a backbone link by FTP transfers or give priority to an important database access.

- **More efficient use of network resources**—Using Cisco’s network analysis management and accounting tools, you will know what your network is being used for and that you are servicing the most important traffic to your business.
- **Tailored services**—The control and visibility provided by QoS enables Internet service providers to offer carefully tailored grades of service differentiation to their customers.
- **Coexistence of mission-critical applications**—Cisco’s QoS technologies make certain that your WAN is used efficiently by mission-critical applications that are most important to your business, that bandwidth and minimum delays required by time-sensitive multimedia and voice applications are available, and that other applications using the link get their fair service without interfering with mission-critical traffic.
- **Foundation for a fully integrated network in the future**—Implementing Cisco QoS technologies in your network now is a good first step toward the fully integrated multimedia network needed in the near future.

QoS Concepts

Fundamentally, QoS enables you to provide better service to certain flows. This is done by either raising the priority of a flow or limiting the priority of another flow. When using congestion-management tools, you try to raise the priority of a flow by queuing and servicing queues in different ways. The queue management tool used for congestion avoidance raises priority by dropping lower-priority flows before higher-priority flows. Policing and shaping provide priority to a flow by limiting the throughput of other flows. Link efficiency tools limit large flows to show a preference for small flows.

Cisco IOS QoS is a tool box, and many tools can accomplish the same result. A simple analogy comes from the need to tighten a bolt: You can tighten a bolt with pliers or with a wrench. Both are equally effective, but these are different tools. This is the same with QoS tools. You will find that results can be accomplished using different QoS tools. Which one to use depends on the traffic. You wouldn’t pick a tool without knowing what you were trying to do, would you? If the job is to drive a nail, you do not bring a screwdriver.

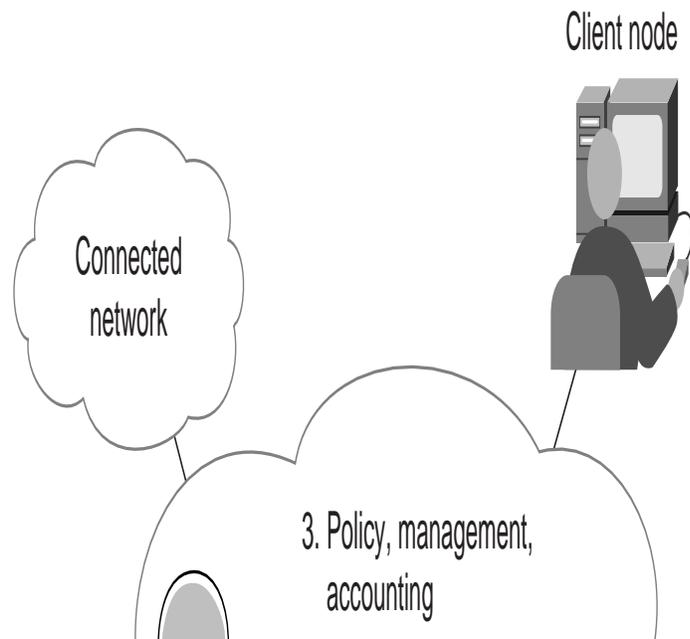
QoS tools can help alleviate most congestion problems. However, many times there is just too much traffic for the bandwidth supplied. In such cases, QoS is merely a bandage. A simple analogy comes from pouring syrup into a bottle. Syrup can be poured from one container into another container at or below the size of the spout. If the amount poured is greater than the size of the spout, syrup is wasted. However, you can use a funnel to catch syrup pouring at a rate greater than the size of the spout. This allows you to pour more than what the spout can take, while still not wasting the syrup. However, consistent overpouring will eventually fill and overflow the funnel.

Basic QoS Architecture

The basic architecture introduces the three fundamental pieces for QoS implementation (see Figure 49-1):

- QoS identification and marking techniques for coordinating QoS from end to end between network elements
- QoS within a single network element (for example, queuing, scheduling, and traffic-shaping tools)
- QoS policy, management, and accounting functions to control and administer end-to-end traffic across a network

Figure 49-1 A Basic QoS Implementation Has Three Main Components



QoS Identification and Marking

Identification and marking is accomplished through classification and reservation.

Classification

To provide preferential service to a type of traffic, it must first be identified. Second, the packet may or may not be marked. These two tasks make up classification. When the packet is identified but not marked, classification is said to be on a per-hop basis. This is when the classification pertains only to the device that it is on, not passed to the next router. This happens with priority queuing (PQ) and custom queuing (CQ). When packets are marked for network-wide use, IP precedence bits can be set (see the section “IP Precedence: Signaling Differentiated QoS”).

Common methods of identifying flows include access control lists (ACLs), policy-based routing, committed access rate (CAR), and network-based application recognition (NBAR).

QoS Within a Single Network Element

Congestion management, queue management, link efficiency, and shaping/policing tools provide QoS within a single network element.

Congestion Management

Because of the bursty nature of voice/video/data traffic, sometimes the amount of traffic exceeds the speed of a link. At this point, what will the router do? Will it buffer traffic in a single queue and let the first packet in be the first packet out? Or, will it put packets into different queues and service certain queues more often? Congestion-management tools address these questions. Tools include priority queuing (PQ), custom queuing (CQ), weighted fair queuing (WFQ), and class-based weighted fair queuing (CBWFQ).

Queue Management

Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and will be dropped. This is a tail drop. The issue with tail drops is that the router cannot prevent this packet from being dropped (even if it is a high-priority packet). So, a mechanism is necessary to do two things:

1. Try to make sure that the queue does not fill up, so that there is room for high-priority packets
2. Allow some sort of criteria for dropping packets that are of lower priority before dropping higher-priority packets

Weighted early random detect (WRED) provides both of these mechanisms.

Link Efficiency

Many times low-speed links present an issue for smaller packets. For example, the serialization delay of a 1500-byte packet on a 56-kbps link is 214 milliseconds. If a voice packet were to get behind this big packet, the delay budget for voice would be exceeded even before the packet left the router! Link fragmentation and interleave allow this large packet to be segmented into smaller packets interleaving the voice packet. Interleaving is as important as the fragmentation. There is no reason to fragment the packet and have the voice packet go behind all the fragmented packets.



Note

Serialization delay is the time that it takes to put a packet on the link. For the example just given, these mathematics apply:

Packet size: 1500-byte packet \times 8 bits/byte = 12,000 bits

Line rate: 56,000 bps

Result: 12,000 bits/56,000bps = .214 sec or 214 msec

Another efficiency is the elimination of too many overhead bits. For example, RTP headers have a 40-byte header. With a payload of as little as 20 bytes, the overhead can be twice that of the payload in some cases. RTP header compression (also known as Compressed Real-Time Protocol header) reduces the header to a more manageable size.

Traffic Shaping and Policing

Shaping is used to create a traffic flow that limits the full bandwidth potential of the flow(s). This is used many times to prevent the overflow problem mentioned in the introduction. For instance, many network topologies use Frame Relay in a hub-and-spoke design. In this case, the central site normally has a

high-bandwidth link (say, T1), while remote sites have a low-bandwidth link in comparison (say, 384 Kbps). In this case, it is possible for traffic from the central site to overflow the low bandwidth link at the other end. Shaping is a perfect way to pace traffic closer to 384 Kbps to avoid the overflow of the remote link. Traffic above the configured rate is buffered for transmission later to maintain the rate configured.

Policing is similar to shaping, but it differs in one very important way: Traffic that exceeds the configured rate is not buffered (and normally is discarded).

**Note**

Cisco's implementation of policing (committed access rate [CAR]) allows a number of actions besides discard to be performed. However, policing normally refers to the discard of traffic above a configured rate.

QoS Management

QoS management helps to set and evaluate QoS policies and goals. A common methodology entails the following steps:

-
- Step 1** Baseline the network with devices such as RMON probes. This helps in determining the traffic characteristics of the network. Also, applications targeted for QoS should be baselined (usually in terms of response time).
 - Step 2** Deploy QoS techniques when the traffic characteristics have been obtained and an application(s) has been targeted for increased QoS.
 - Step 3** Evaluate the results by testing the response of the targeted applications to see whether the QoS goals have been reached.

For ease of deployment, you can use Cisco's Quality of Service Policy Manager (QPM) and Quality of Service Device Manager (QDM). For verification of service levels, you can use Cisco's Internetwork Performance Monitor (IPM).

You must consider that in an ever changing network environment, QoS is not a one-time deployment, but an ongoing, essential part of network design.

End-to-End QoS Levels

Service levels refer to the actual end-to-end QoS capabilities, meaning the capability of a network to deliver service needed by specific network traffic from end to end or edge to edge. The services differ in their level of *QoS strictness*, which describes how tightly the service can be bound by specific bandwidth, delay, jitter, and loss characteristics.

Three basic levels of end-to-end QoS can be provided across a heterogeneous network, as shown in Figure 49-2:

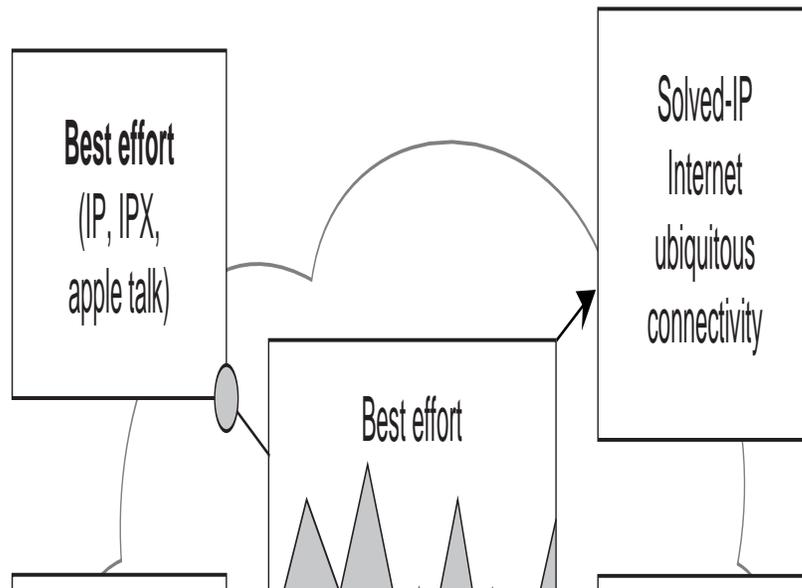
- **Best-effort service**—Also known as lack of QoS, best-effort service is basic connectivity with no guarantees. This is best characterized by FIFO queues, which have no differentiation between flows.
- **Differentiated service (also called soft QoS)**—Some traffic is treated better than the rest (faster handling, more average bandwidth, and lower average loss rate). This is a statistical preference, not a hard and fast guarantee. This is provided by classification of traffic and the use of QoS tools such as PQ, CQ, WFQ, and WRED (all discussed later in this chapter).

- **Guaranteed service (also called hard QoS)**—This is an absolute reservation of network resources for specific traffic. This is provided through QoS tools RSVP and CBWFQ (discussed later in this chapter).

Deciding which type of service is appropriate to deploy in the network depends on several factors:

- The application or problem that the customer is trying to solve. Each of the three types of service is appropriate for certain applications. This does not imply that a customer must migrate to differentiated and then to guaranteed service (although many probably eventually will). A differentiated service—or even a best-effort service—may be appropriate, depending on the customer application requirements.
- The rate at which customers can realistically upgrade their infrastructures. There is a natural upgrade path from the technology needed to provide differentiated services to that needed to provide guaranteed services, which is a superset of that needed for differentiated services.
- The cost of implementing and deploying guaranteed service is likely to be more than that for a differentiated service.

Figure 49-2 The Three Levels of End-to-End QoS Are Best-Effort Service, Differentiated Service, and Guaranteed Service



Classification—Identifying Flows

To provide priority to certain flows, the flow must first be identified and (if desired) marked. These two tasks are commonly referred to as just *classification*.

Historically, identification was done using access control lists (ACLs). ACLs identify traffic for congestion-management tools such as PQ and CQ. Because PQ and CQ are placed on routers on a hop-by-hop basis (that is, priority settings for QoS pertain only to that router and are not passed to subsequent router hops in the network), identification of the packet is used only within a single router. In some instances, CBWFQ classification is for only a single router. This is contrasted by setting IP precedence bits.

Features such as policy-based routing and committed access rate (CAR) can be used to set precedence based on extended access list classification. This allows considerable flexibility for precedence assignment, including assignment by application or user, by destination and source subnet, and so on. Typically this functionality is deployed as close to the edge of the network (or administrative domain) as possible so that each subsequent network element can provide service based on the determined policy.

Network-based application recognition (NBAR) is used to identify traffic more granularly. For example, URLs in an HTTP packet can be identified. Once the packet has been identified, it can be marked with a precedence setting.

QoS Policy Setting with Policy-Based Routing

Cisco IOS Policy-Based Routing (PBR) enables you to classify traffic based on extended access list criteria, set IP precedence bits, and even route to specific traffic-engineered paths that may be required to allow a specific QoS through the network. By setting precedence levels on incoming traffic and using them in combination with the queuing tools described earlier in this chapter, you can create differentiated service. These tools provide powerful, simple, and flexible options for implementing QoS policies in your network.

Using policy-based routing, route maps are made to match on certain flow criteria and then set precedence bits when ACLs are matched.

The capability to set IP precedence bits should not be confused with PBR's primary capability: routing packets based on configured policies. Some applications or traffic can benefit from QoS-specific routing—transferring stock records to a corporate office (for example, on a higher-bandwidth, higher-cost link for a short time), while transmitting routine application data such as e-mail over a lower-bandwidth, lower-cost link. PBR can be used to direct packets to take different paths than the path derived from the routing protocols. It provides a more flexible mechanism for routing packets, complementing the existing mechanisms provided by routing protocols.

Also available using route maps is the capability to identify packets based on Border Gateway Protocol (BGP) attributes such as community lists and AS paths. This is known as QoS policy propagation via Border Gateway Protocol.

CAR: Setting IP Precedence

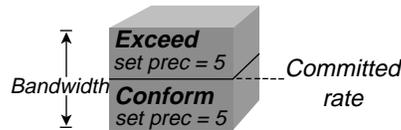
Similar in some ways to PBR, the CAR feature enables you to classify traffic on an incoming interface. It also allows specification of policies for handling traffic that exceeds a certain bandwidth allocation. CAR looks at traffic received on an interface, or a subset of that traffic selected by access list criteria, compares its rate to that of a configured token bucket, and then takes action based on the result (for example, drop or rewrite IP precedence).

There is some confusion with using CAR to set IP precedence bits. An attempt to clear up any confusion follows. As described later in this chapter, CAR (as its name describes) is used to police traffic flows to a *committed access rate*. CAR does this with a token bucket. A token bucket is a bucket with tokens in it that represent bytes (1 token = 1 byte). The bucket is filled with tokens at a user-configured rate. As packets arrive to be delivered, the system checks the bucket for tokens. If there are enough tokens in the bucket to match the size of the packet, those tokens are removed and the packet is passed (this packet *conforms*). If there aren't enough tokens, the packet is dropped (this packet *exceeds*).

When using Cisco IOS's CAR implementation, you have more options than just pass or drop. One option is to set the IP precedence bits. When the conform and exceed actions both say to set precedence bits to the *same* setting, then it is no longer a policing feature, but merely a method of setting IP precedence bits.

Figure 49-3 shows a committed rate that is decided upon. Any packet that is below the rate conforms. Packets above the rate exceed. In this example, the action for both conditions is to set `prec = 5`. In this case, what the rate is does not matter and CAR is simply being used to set precedence bits.

Figure 49-3 Committed Rate That Is Decided Upon



When IP precedence is set in the host or network client, this setting can be used optionally; however, this can be overridden by policy within the network. IP precedence enables service classes to be established using existing network queuing mechanisms (for example, WFQ or WRED), with no changes to existing applications or complicated network requirements. Note that this same approach is easily extended to IPv6 using its Priority field.

Cisco IOS software takes advantage of the end-to-end nature of IP to meet this challenge by overlaying Layer 2 technology-specific QoS signaling solutions with the Layer 3 IP QoS signaling methods of RSVP and IP precedence.

7500 Platform

Cisco IOS software also provides distributed committed access rate (D-CAR) on the 7500 Versatile Interface Processors (VIPs). D-CAR can be used to set IP precedence bits just like CAR. It also can place packets in QoS groups that are used in class-based D-WFQ and for policing in D-CAR.

NBAR: Dynamic Identification of Flows

Cisco's newest method of classification is Network Based Application Recognition (NBAR). For clarity, NBAR is actually only an identification tool, but it will be referred to here as a classification tool. As with any classification tool, the hard part is identifying the traffic. Marking the packet later is relatively easy. NBAR takes the identification portion of classification to another level. Looking deeper into the packet, identification can be performed, for example, to the URL or MIME type of an HTTP packet. This becomes essential as more applications become web-based. You would need to differentiate between an order being placed and casual web browsing. In addition, NBAR can identify various applications that use ephemeral ports. NBAR does this by looking at control packets to determine which ports the application decides to pass data on.

NBAR adds a couple of interesting features that make it extremely valuable. One feature is a protocol discovery capability. This allows NBAR to baseline the protocols on an interface. NBAR lists the protocols that it can identify and provides statistics on each one. Another feature is the Packet Description Language Module (PDLM), which allows additional protocols to be easily added to NBAR's list of identifiable protocols. These modules are created and loaded into Flash memory, which then is uploaded into RAM. Using PDLMs, additional protocols can be added to the list without upgrading the IOS level or rebooting the router.



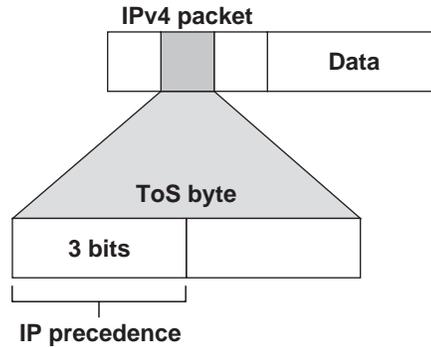
Note

Although NBAR only identifies packets, these packets may also be marked with an IP precedence setting.

IP Precedence: Differentiated QoS

IP precedence utilizes the 3 precedence bits in the IPv4 header's Type of Service (ToS) field to specify class of service for each packet, as shown in Figure 49-4. You can partition traffic in up to six classes of service using IP precedence (two others are reserved for internal network use). The queuing technologies throughout the network can then use this signal to provide the appropriate expedited handling.

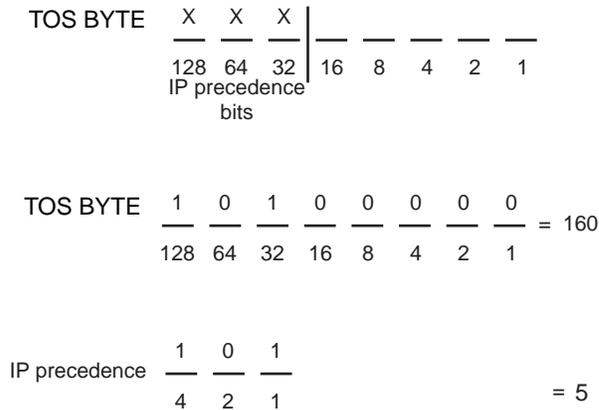
Figure 49-4 This Diagram Shows the IP Precedence ToS Field in an IP Packet Header



The 3 most significant bits (correlating to binary settings 32, 64, and 128) of the Type of Service (ToS) field in the IP header constitute the bits used for IP precedence. These bits are used to provide a priority from 0 to 7 (settings of 6 and 7 are reserved and are *not* to be set by a network administrator) for the IP packet.

Because only 3 bits of the ToS byte are used for IP precedence, you need to differentiate these bits from the rest of the ToS byte. In Figure 49-5, a 1 in the first and third bit positions (viewing from left to right) correlates to an IP precedence setting of 5, but when viewing the ToS byte in a Sniffer trace, it will show 160. You need to be able to translate these settings.

Figure 49-5 IP Precedence



Traffic that is identified can be marked by setting the IP precedence bits. Thus, it needs to be classified only once.

RFC 2475 extends the number of bits used in the ToS byte from 3 to 6. The 6 MSBs will be used for precedence settings (known as DS codepoints), with the 2 least significant bits (the right-most 2 bits) reserved for future use. This specification is commonly referred to as *DiffServ*.

Congestion-Management Tools

One way network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic, and then determine some method of prioritizing it onto an output link. Cisco IOS software includes the following queuing tools:

- First-in, first-out (FIFO) queuing
- Priority queuing (PQ)
- Custom queuing (CQ)
- Flow-based weighted fair queuing (WFQ)
- Class-based weighted fair queuing (CBWFQ)

Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance, as described in the following sections.



Note

Queuing algorithms take effect when congestion is experienced. By definition, if the link is not congested, then there is no need to queue packets. In the absence of congestion, all packets are delivered directly to the interface.

FIFO: Basic Store-and-Forward Capability

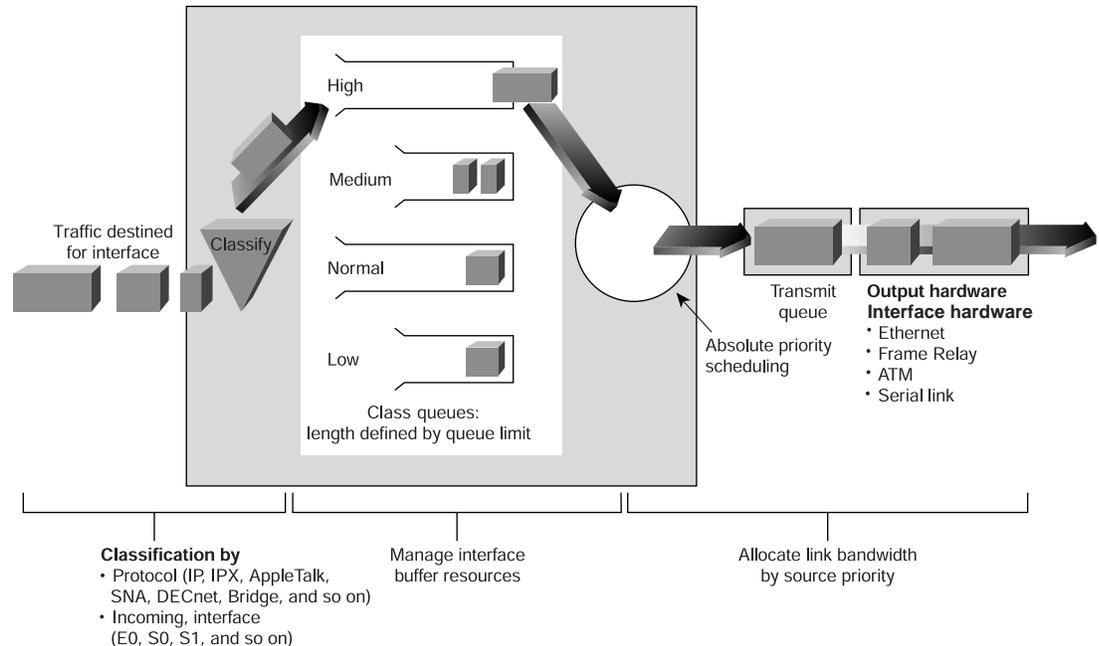
In its simplest form, *FIFO* queuing involves storing packets when the network is congested and forwarding them in order of arrival when the network is no longer congested. FIFO is the default queuing algorithm in some instances, thus requiring no configuration, but it has several shortcomings. Most importantly, FIFO queuing makes no decision about packet priority; the order of arrival determines bandwidth, promptness, and buffer allocation. Nor does it provide protection against ill-behaved applications (sources). Bursty sources can cause long delays in delivering time-sensitive application traffic, and potentially to network control and signaling messages. FIFO queuing was a necessary first step in controlling network traffic, but today's intelligent networks need more sophisticated algorithms. In addition, a full queue causes tail drops. This is undesirable because the packet dropped could have been a high-priority packet. The router couldn't prevent this packet from being dropped because there was no room in the queue for it (in addition to the fact that FIFO cannot tell a high-priority packet from a low-priority packet). Cisco IOS software implements queuing algorithms that avoid the shortcomings of FIFO queuing.

PQ: Prioritizing Traffic

PQ ensures that important traffic gets the fastest handling at each point where it is used. It was designed to give strict priority to important traffic. Priority queuing can flexibly prioritize according to network protocol (for example IP, IPX, or AppleTalk), incoming interface, packet size, source/destination address, and so on. In PQ, each packet is placed in one of four queues—high,

medium, normal, or low—based on an assigned priority. Packets that are not classified by this priority list mechanism fall into the normal queue (see Figure 49-6). During transmission, the algorithm gives higher-priority queues absolute preferential treatment over low-priority queues.

Figure 49-6 Priority Queuing Places Data into Four Levels of Queues: High, Medium, Normal, and Low

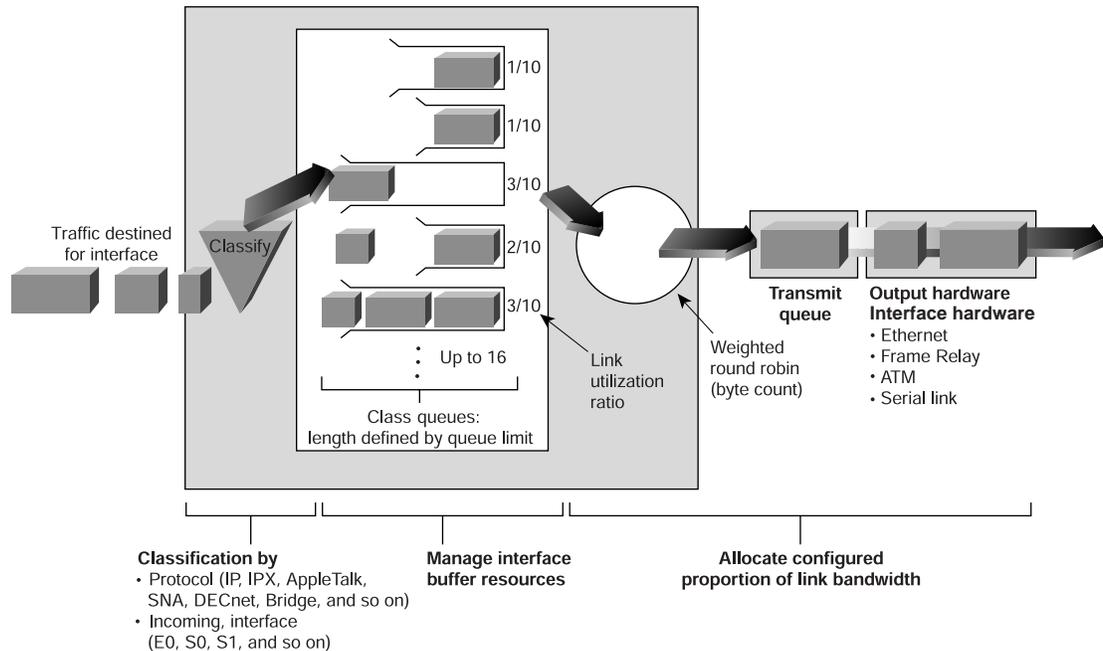


PQ is useful for making sure that mission-critical traffic traversing various WAN links gets priority treatment. For example, Cisco uses PQ to ensure that important Oracle-based sales reporting data gets to its destination ahead of other, less-critical traffic. PQ currently uses static configuration and thus does not automatically adapt to changing network requirements.

CQ: Guaranteeing Bandwidth

CQ was designed to allow various applications or organizations to share the network among applications with specific minimum bandwidth or latency requirements. In these environments, bandwidth must be shared proportionally between applications and users. You can use the Cisco CQ feature to provide guaranteed bandwidth at a potential congestion point, ensuring the specified traffic a fixed portion of available bandwidth and leaving the remaining bandwidth to other traffic. Custom queuing handles traffic by assigning a specified amount of queue space to each class of packets and then servicing the queues in a round-robin fashion (see Figure 49-7).

Figure 49-7 Custom Queuing Handles Traffic by Assigning a Specified Amount of Queue Space to Each Class of Packets and Then Servicing up to 17 Queues in a Round-Robin Fashion



As an example, encapsulated Systems Network Architecture (SNA) requires a guaranteed minimum level of service. You could reserve half of available bandwidth for SNA data and allow the remaining half to be used by other protocols such as IP and Internetwork Packet Exchange (IPX).

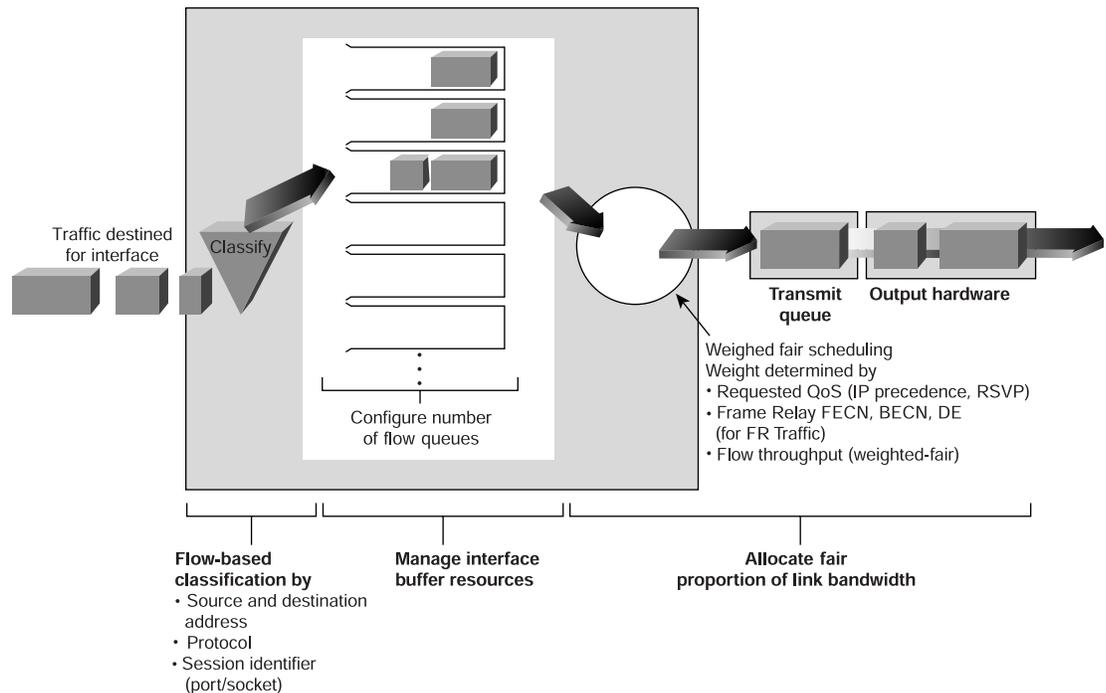
The queuing algorithm places the messages in one of 17 queues (queue 0 holds system messages such as keepalives, signaling, and so on) and is emptied with weighted priority. The router services queues 1 through 16 in round-robin order, dequeuing a configured byte count from each queue in each cycle. This feature ensures that no application (or specified group of applications) achieves more than a predetermined proportion of overall capacity when the line is under stress. Like PQ, CQ is statically configured and does not automatically adapt to changing network conditions.

Flow-Based WFQ: Creating Fairness Among Flows

For situations in which it is desirable to provide consistent response time to heavy and light network users alike without adding excessive bandwidth, the solution is flow-based WFQ (commonly referred to as just WFQ). WFQ is one of Cisco's premier queuing techniques. It is a flow-based queuing algorithm that creates bit-wise fairness by allowing each queue to be serviced fairly in terms of byte count. For example, if queue 1 has 100-byte packets and queue 2 has 50-byte packets, the WFQ algorithm will take two packets from queue 2 for every one packet from queue 1. This makes service fair for each queue: 100 bytes each time the queue is serviced.

WFQ ensures that queues do not starve for bandwidth and that traffic gets predictable service. Low-volume traffic streams—which comprise the majority of traffic—receive increased service, transmitting the same number of bytes as high-volume streams. This behavior results in what appears to be preferential treatment for low-volume traffic, when in actuality it is creating fairness, as shown in Figure 49-8.

Figure 49-8 *With WFQ, If High-Volume Conversations Are Active, Their Transfer Rates and Interarrival Periods Are Made Much More Predictable*



WFQ is designed to minimize configuration effort, and it automatically adapts to changing network traffic conditions. In fact, WFQ does such a good job for most applications that it has been made the default queuing mode on most serial interfaces configured to run at or below E1 speeds (2.048 Mbps).

Flow-based WFQ creates *flows* based on a number of characteristics in a packet. Each flow (also referred to as a *conversation*) is given its own queue for buffering if congestion is experienced. The following descriptions use flow, conversation, and queue interchangeably.



Note

Characteristics defining a flow include source and destination addresses, socket numbers, and session identifiers. These are general characteristics. Review the Cisco Systems technical documents (<http://www.cisco.com>) to see the exact criteria for the definition of a flow. For different protocols, a different criterion is used.

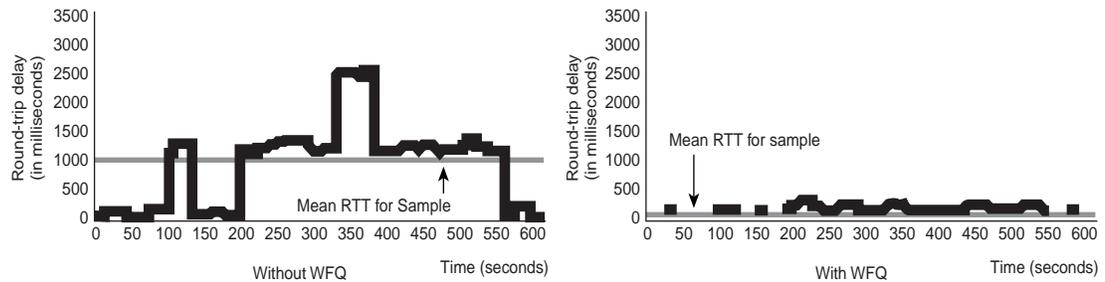
The weighted portion of WFQ comes from the use of IP precedence bits to provide greater service for certain queues. Using settings 0 to 5 (6 and 7 are reserved), WFQ uses its algorithm to determine how much more service to provide to a queue. See the next section “Cooperation Between WFQ and QoS Signaling Technologies,” for more details.

WFQ is efficient in that it uses whatever bandwidth is available to forward traffic from lower-priority flows if no traffic from higher-priority flows is present. This is different from strict time-division multiplexing (TDM), which simply carves up the bandwidth and lets it go unused if no traffic is present for a particular traffic type. WFQ works with both—IP precedence and Resource Reservation Protocol (RSVP), described later in this chapter—to help provide differentiated QoS as well as guaranteed services.

The WFQ algorithm also addresses the problem of round-trip delay variability. If multiple high-volume conversations are active, their transfer rates and interarrival periods are made much more predictable. This is created by the bit-wise fairness. If conversations are serviced in a consistent manner with every round-robin approach, delay variation (or jitter) stabilizes. WFQ greatly enhances algorithms such as

SNA Logical Link Control (LLC) and the Transmission Control Protocol (TCP) congestion control and slow-start features. The result is more predictable throughput and response time for each active flow, as shown in Figure 49-9.

Figure 49-9 This Diagram Shows an Example of Interactive Traffic Delay (128-kbps Frame Relay WAN Link)



Cooperation Between WFQ and QoS Signaling Technologies

As mentioned previously, WFQ is IP precedence-aware; that is, it is capable of detecting higher-priority packets marked with precedence by the IP forwarder and can schedule them faster, providing superior response time for this traffic. This is the weighted portion of WFQ. The IP Precedence field has values between 0 (the default) and 7 (6 and 7 are reserved and normally are not set by network administrators). As the precedence value increases, the algorithm allocates more bandwidth to that conversation to make sure that it is served more quickly when congestion occurs. WFQ assigns a weight to each flow, which determines the transmit order for queued packets. In this scheme, lower weights are provided more service. IP precedence serves as a divisor to this weighting factor. For instance, traffic with an IP Precedence field value of 7 gets a lower weight than traffic with an IP Precedence field value of 3, and thus has priority in the transmit order.



Note

A *weight* is a number calculated from the IP precedence setting for a packet in flow. This weight is used in WFQ's algorithm to determine when the packet will be serviced.

$$\text{Weight} = (4096 / (\text{IP precedence} + 1))$$

$$\text{Weight} = (32384 / (\text{IP precedence} + 1))$$

The numerator of the equation changed from 4096 to 32384 in a v12.0 maintenance release.

Weight settings can be viewed using the **show queue <interface>** command.

Effect of IP Precedence Settings

The effect of IP precedence settings is described here:

If you have one flow at each precedence level on an interface, each flow will get precedence + 1 parts of the link, as follows:

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$$

The flows will get 8/36, 7/36, 6/36, and 5/36 of the link, and so on. However, if you have 18 precedence—1 flow and 1 of each of the others—the formula looks like this:

$$1 + 18 \times 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36 - 2 + 18 \times 2 = 70$$

The flows will get 8/70, 7/70, 6/70, 5/70, 4/70, 3/70, 2/70, and 1/70 of the link, and 18 of the flows will each get approximately 2/70 of the link.

WFQ is also RSVP-aware; RSVP uses WFQ to allocate buffer space and schedule packets, and it guarantees bandwidth for reserved flows. Additionally, in a Frame Relay network, the presence of congestion is flagged by the forward explicit congestion notification (FECN) and backward explicit congestion notification (BECN) bits. WFQ weights are affected by Frame Relay discard eligible (DE), FECN, and BECN bits when the traffic is switched by the Frame Relay switching module. When congestion is flagged, the weights used by the algorithm are altered so that the conversation encountering the congestion transmits less frequently.

7500 Platform

Cisco IOS software also provides distributed weighted fair queuing (D-WFQ), a high-speed version of WFQ that runs on VIP-distributed processors. The D-WFQ algorithm provides two types of WFQ: flow-based fair queuing and class-based fair queuing. The flow-based implementation of D-WFQ differs from WFQ by not recognizing IP precedence bits—thus, there is no weighting to flows.

Class-Based WFQ: Ensuring Network Bandwidth

Class-based WFQ (CBWFQ) is one of Cisco's newest congestion-management tools for providing greater flexibility. When you want to provide a minimum amount of bandwidth, use CBWFQ. This is in comparison to a desire to provide a maximum amount of bandwidth. CAR and traffic shaping are used in that case.

CBWFQ allows a network administrator to create minimum guaranteed bandwidth classes. Instead of providing a queue for each individual flow, a class is defined that consists of one or more flows. Each class can be guaranteed a minimum amount of bandwidth.

One example in which CBWFQ can be used is in preventing multiple low-priority flows from swamping out a single high-priority flow. For example, a video stream that needs half the bandwidth of T1 will be provided that by WFQ if there are two flows. As more flows are added, the video stream gets less of the bandwidth because WFQ's mechanism creates fairness. If there are 10 flows, the video stream will get only 1/10th of the bandwidth, which is not enough. Even setting the IP precedence bit = 5 does not solve this problem.

$$1 \times 9 + 6 = 15$$

Video gets 6/15 of the bandwidth, which is less than the bandwidth video needs. A mechanism must be invoked to provide the half of the bandwidth that video needs. CBWFQ provides this. The network administrator defines a class, places the video stream in the class, and tells the router to provide 768 kbps (half of a T1) service for the class. Video is now given the bandwidth that it needs. A default class is used for the rest of flows. This class is serviced using flow-based WFQ schemes allocating the remainder of the bandwidth (half of the T1, in this example)



Note

This is *not* to discount the use of WFQ. For most implementations, WFQ is an excellent congestion-management tool (that's why it's default on interfaces E1 and below). The previous example was meant to show a situation in which CBWFQ is very effective.

In addition, a low-latency queue (LLQ) may be designated, which essentially is a priority queue. Note that this feature is also referred to as priority queue class-based weighted fair queuing (PQCBWFQ).

Low-latency queuing allows a class to be serviced as a strict-priority queue. Traffic in this class will be serviced before any of the other classes. A reservation for an amount of bandwidth is made. Any traffic above this reservation is discarded. Outside of CBWFQ, you can use IP RTP priority (also known as PQWFQ) or IP RTP reserve to provide similar service for RTP traffic only.

**Note**

With CBWFQ, a minimum amount of bandwidth can be reserved for a certain class. If more bandwidth is available, that class is welcome to use it. The key is that it is guaranteed a minimum amount of bandwidth. Also, if a class is not using its guaranteed bandwidth, other applications may use the bandwidth.

7500 Platform

Cisco IOS software also provides distributed class-based weighted fair queuing (still referred to as D-WFQ), a high-speed version of WFQ that runs on VIP-distributed processors. Class-based WFQ in D-WFQ differs from CBWFQ by using different syntax, but it essentially provides the same service. In addition to providing the capability to guarantee bandwidth, class-based WFQ in D-WFQ has an option to recognize IP precedence bits not recognized in flow-based (this is called ToS-based).

Queue Management (Congestion-Avoidance Tools)

Congestion avoidance is a form of queue management. *Congestion-avoidance techniques* monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks, as opposed to congestion-management techniques that operate to control congestion *after* it occurs. The primary Cisco IOS congestion avoidance tool is weighted random early detection (WRED).

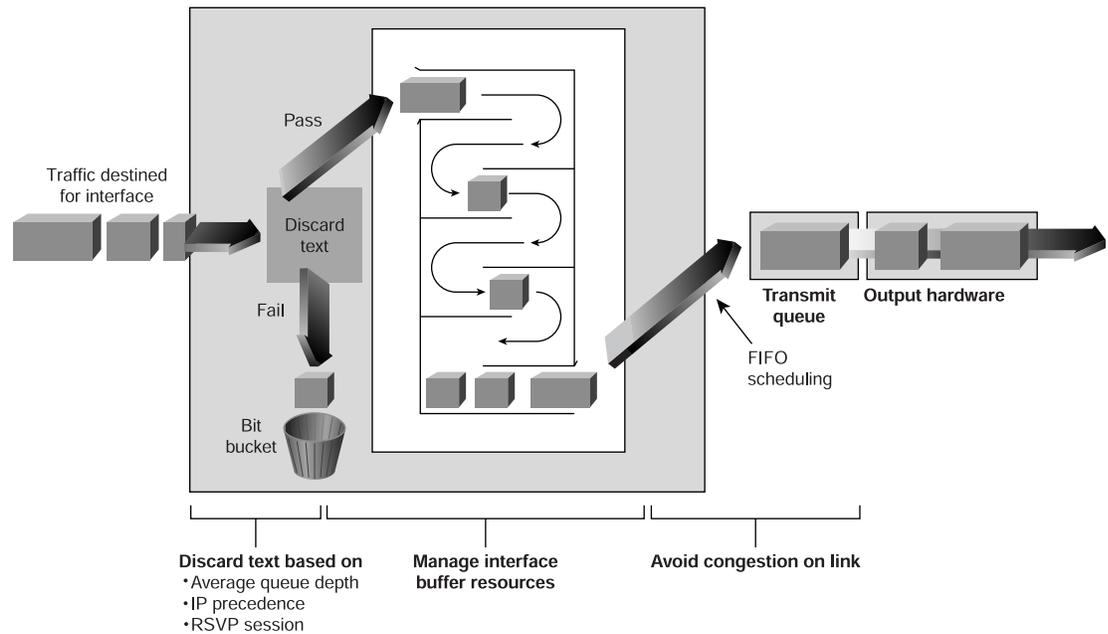
WRED: Avoiding Congestion

The *random early detection (RED)* algorithms are designed to avoid congestion in internetworks before it becomes a problem. RED works by monitoring traffic load at points in the network and stochastically discarding packets if the congestion begins to increase. The result of the drop is that the source detects the dropped traffic and slows its transmission. RED is primarily designed to work with TCP in IP internetwork environments.

WRED Cooperation with QoS Signaling Technologies

WRED combines the capabilities of the RED algorithm with IP precedence. This combination provides for preferential traffic handling for higher-priority packets. It can selectively discard lower-priority traffic when the interface starts to get congested and can provide differentiated performance characteristics for different classes of service (see Figure 49-10). WRED is also RSVP-aware and can provide an integrated services controlled-load QoS.

Figure 49-10 WRED Provides a Method That Stochastically Discards Packets if the Congestion Begins to Increase



Within each queue, a finite number of packets can be housed. A full queue causes tail drops. *Tail drops* are dropped packets that could not fit into the queue because the queue was full. This is undesirable because the packet discarded may have been a high-priority packet and the router did not have a chance to queue it. If the queue is not full, the router can look at the priority of all arriving packets and drop the lower-priority packets, allowing high-priority packets into the queue. Through managing the depth of the queue (the number of packets in the queue) by dropping various packets, the router can do its best to make sure that the queue does not fill and that tail drops are not experienced. This allows the router to make the decision on which packets get dropped when the queue depth increases. WRED also helps prevent overall congestion in an internetwork. WRED uses a minimum threshold for each IP precedence level to determine when a packet can be dropped. (The minimum threshold must be exceeded for WRED to consider a packet as a candidate for being dropped.)

Take a look at this WRED example:

Depth of the queue: 21 packets

Minimum drop threshold for IP precedence = 0: 20

Minimum drop threshold for IP precedence = 1: 22

Because the minimum drop threshold for IP precedence = 0 has been exceeded, packets with an IP precedence = 0 can be dropped. However, the minimum drop threshold for IP precedence = 1 has not been exceeded, so those packets will not be dropped. If the queue depth deepens and exceeds 22, then packets with IP precedence = 1 can be dropped as well. WRED uses an algorithm that raises the probability that a packet can be dropped as the queue depth rises from the minimum drop threshold to the maximum drop threshold. Above the maximum drop threshold, all packets are dropped.

Flow RED: RED for Non-TCP-Compliant Flows

WRED is primarily used for TCP flows that will scale back transmission if a packet is dropped. There are non-TCP-compliant flows that do not scale back when packets are dropped. Flow RED is used to deal with such flows. The approach is to increase the probability of dropping a flow if it exceeds a threshold.

Flow-based WRED relies on these two main approaches to remedy the problem of linear packet dumping:

- It classifies incoming traffic into flows based on parameters such as destination and source addresses and ports.
- It maintains state about active flows, which are flows that have packets in the output queues.

Flow-based WRED uses this classification and state information to ensure that each flow does not consume more than its permitted share of the output buffer resources. Flow-based WRED determines which flows monopolize resources, and it more heavily penalizes these flows.

This is how flow-based WRED ensures fairness among flows: It maintains a count of the number of active flows that exist through an output interface. Given the number of active flows and the output queue size, flow-based WRED determines the number of buffers available per flow.

To allow for some burstiness, flow-based WRED scales the number of buffers available per flow by a configured factor and allows each active flow to have a certain number of packets in the output queue. This scaling factor is common to all flows. The outcome of the scaled number of buffers becomes the per-flow limit. When a flow exceeds the per-flow limit, the probability that a packet from that flow will be dropped increases.

7500 Platform

Cisco IOS software also provides distributed weighted random early detection (D-WRED), a high-speed version of WRED that runs on VIP-distributed processors. The D-WRED algorithm provides the same functionality as what WRED provides, such as minimum and maximum queue depth thresholds and drop capabilities for each class of service.



Warning

Although IOS allows the configuration of the minimum and maximum queue depth thresholds and drop capabilities, it is recommended that you use the defaults. Consult Cisco Technical Support before changing any of these defaults.

Traffic-Shaping and Policing Tools

Cisco's QoS software solutions include two traffic-shaping tools—generic traffic shaping (GTS) and Frame Relay traffic shaping (FRTS)—to manage traffic and congestion on the network. Cisco's IOS policing tool is committed access rate (CAR). This was briefly described in the “Classification” section, earlier in this chapter, as it pertains to classification. Here it will be described for its policing function.

CAR: Managing Access Bandwidth Policy and Performing Policing

As described earlier, fundamentally, QoS provides priority either by raising the priority of one flow or by limiting the priority of another. CAR is used to limit the bandwidth of a flow in order to favor another flow.

In the “Classification” section, earlier in this chapter, a generic token bucket was described. In that description, packets that conform are passed, and packets that exceed are dropped.

With Cisco's IOS implementation of CAR, a number of actions can be performed. These actions consist of transmitting, dropping, setting IP precedence bits, and continuing (this refers to cascading CAR statements). This flexibility allows for a number of ways to act upon traffic. Here are some scenarios:

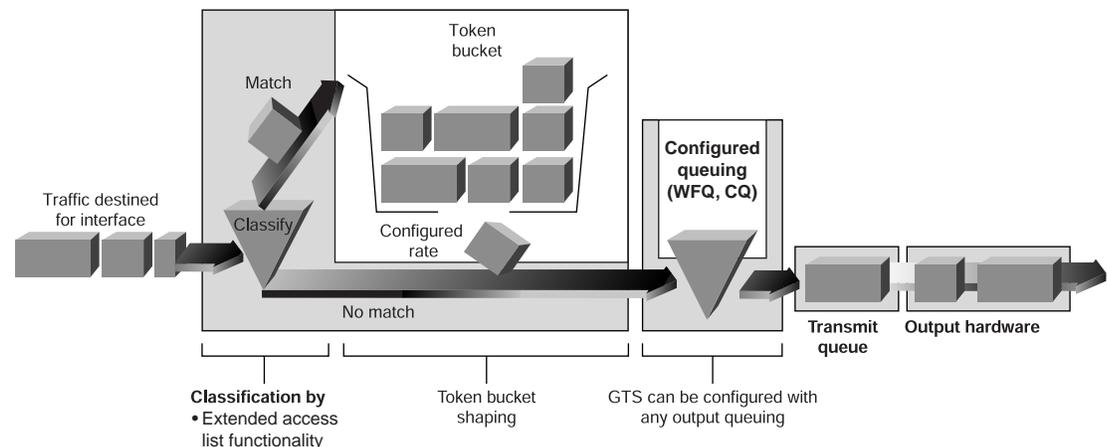
- Conforming traffic can be classified with an IP precedence of 5, and exceeding traffic can be dropped.
- Conforming traffic can be transmitted with an IP precedence setting of 5, while exceeding traffic can also be transmitted, but with an IP precedence setting of 1.
- Conforming traffic can be transmitted, and exceeding traffic can be reclassified to a lower IP precedence setting and then sent to the next CAR statement for additional conditions.

Cisco IOS's CAR implementation also provides an excess burst bucket not found in a generic token bucket. In this bucket are additional tokens above the original (or normal) burst bucket. When these tokens are used, the packet has the possibility of being dropped (even if the action is to transmit). A RED-like algorithm is used that says, "The more tokens you use from this bucket, the higher probability that the next packet will be dropped." This allows the flow to be scaled back slowly as in WRED, while still getting the opportunity to send above the normal bucket.

GTS: Controlling Outbound Traffic Flow

GTS provides a mechanism to control the traffic flow on a particular interface. It reduces outbound traffic flow to avoid congestion by constraining specified traffic to a particular bit rate (it also uses a token bucket approach) while queuing bursts of the specified traffic. So, any traffic above the configured rate is queued. This differs from CAR, in which packets are not queued. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, eliminating bottlenecks in topologies with data-rate mismatches. Figure 49-11 illustrates GTS.

Figure 49-11 Generic Traffic Shaping Is Applied on a Per-Interface Basis



GTS applies on a per-interface basis, can use access lists to select the traffic to shape, and works with a variety of Layer 2 technologies, including Frame Relay, ATM, Switched Multimegabit Data Service (SMDS), and Ethernet.

On a Frame Relay subinterface, GTS can be set up to adapt dynamically to available bandwidth by integrating BECN signals, or it can be set up simply to shape to a prespecified rate. GTS can also be configured on an ATM Interface Processor (ATM/AIP) interface card to respond to RSVP signaled over statically configured ATM permanent virtual circuits (PVCs).

FRTS: Managing Frame Relay Traffic

FRTS provides parameters that are useful for managing network traffic congestion. These include committed information rate (CIR), FECN and BECN, and the DE bit. For some time, Cisco has provided support for FECN for DECnet, BECN for SNA traffic using direct LLC2 encapsulation via RFC 1490, and DE bit support. The FRTS feature builds on this Frame Relay support with additional capabilities that improve the scalability and performance of a Frame Relay network, increasing the density of virtual circuits and improving response time.

For example, you can configure rate enforcement—a peak rate configured to limit outbound traffic—to either the CIR or some other defined value, such as the excess information rate (EIR), on a per-virtual-circuit (VC) basis.

You can also define priority and custom queuing at the VC or subinterface level. This allows for finer granularity in the prioritization and queuing of traffic, and provides more control over the traffic flow on an individual VC. If you combine CQ with the per-VC queuing and rate enforcement capabilities, you enable Frame Relay VCs to carry multiple traffic types such as IP, SNA, and IPX, with bandwidth guaranteed for each traffic type.

FRTS can eliminate bottlenecks in Frame Relay networks with high-speed connections at the central site and low-speed connections at the branch sites. You can configure rate enforcement to limit the rate at which data is sent on the VC at the central site. You can also use rate enforcement with the existing data-link connection identifier (DLCI) prioritization feature to further improve performance in this situation. FRTS applies only to Frame Relay PVCs and switched virtual circuits (SVCs).

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

FRTS also provides a mechanism for sharing media by multiple VCs. Rate enforcement allows the transmission speed used by the router to be controlled by criteria other than line speed, such as the CIR or EIR. The rate enforcement feature can also be used to preallocate bandwidth to each VC, creating a virtual TDM network. Finally, with Cisco's FRTS feature, you can integrate StrataCom ATM Foresight closed-loop congestion control to actively adapt to downstream congestion conditions.

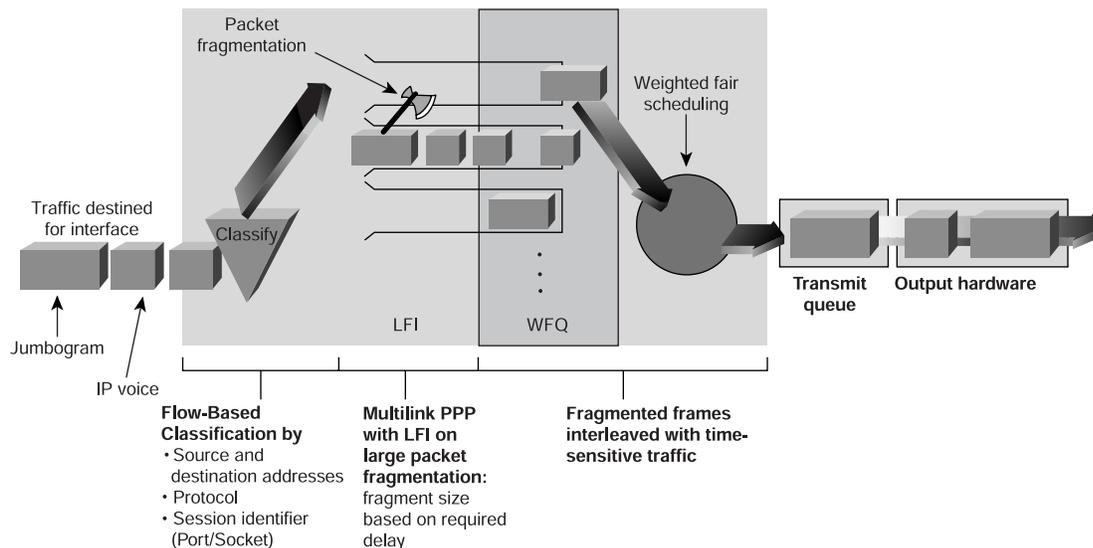
Link Efficiency Mechanisms

Currently, Cisco IOS software offers two link efficiency mechanisms—link fragmentation and interleaving (LFI) and real-time protocol header compression (RTP-HC)—which work with queuing and traffic shaping to improve the efficiency and predictability of the application service levels.

LFI: Fragmenting and Interleaving IP Traffic

Interactive traffic (Telnet, Voice over IP, and the like) is susceptible to increased latency and jitter when the network processes large packets (for example, LAN-to-LAN FTP transfers traversing a WAN link), especially as they are queued on slower links. The Cisco IOS LFI feature reduces delay and jitter on slower-speed links by breaking up large datagrams and interleaving low-delay traffic packets with the resulting smaller packets (see Figure 49-12).

Figure 49-12 By Dividing Large Datagrams with the LFI Feature, Delay Is Reduced on Slower-Speed Links



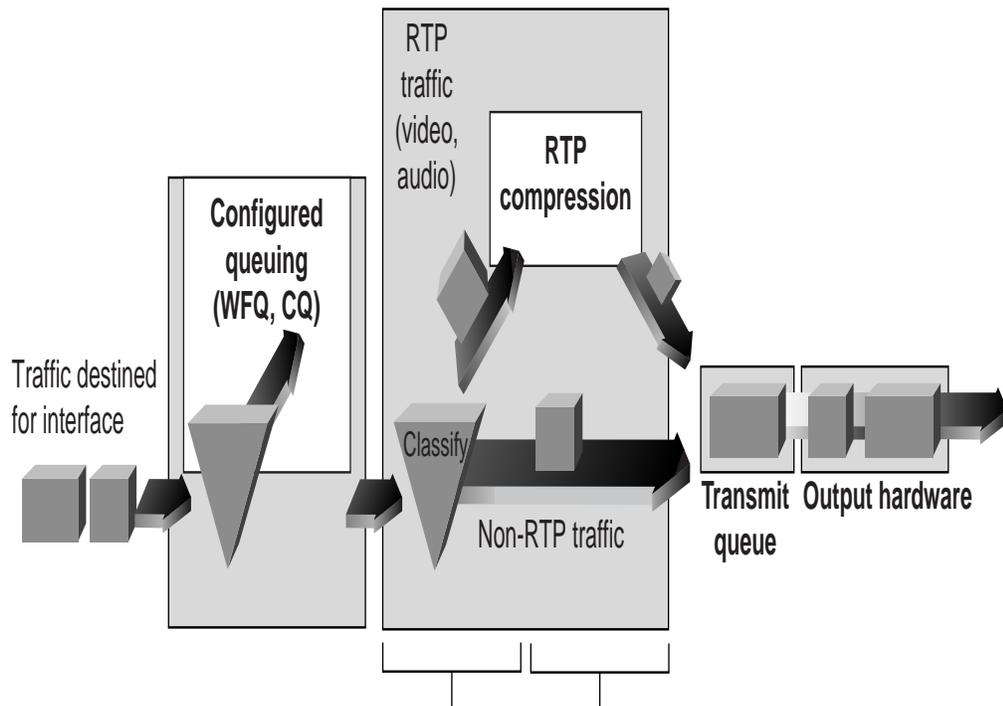
LFI was designed especially for lower-speed links in which serialization delay is significant. LFI requires that multilink Point-to-Point Protocol (PPP) be configured on the interface with interleaving turned on. A related IETF draft, called “Multiclass Extensions to Multilink PPP (MCML),” implements almost the same function as LFI.

Note that for implementation of fragmentation over Frame Relay, you should use the FRF.12 feature, which provides the same results.

RTP Header Compression: Increasing Efficiency of Real-Time Traffic

Real-Time Transport Protocol is a host-to-host protocol used for carrying newer multimedia application traffic, including packetized audio and video, over an IP network. Real-Time Transport Protocol provides end-to-end network transport functions intended for applications transmitting real-time requirements, such as audio, video, or simulation data over multicast or unicast network services. Real-Time Transport Protocol header compression increases efficiency for many of the newer voice over IP or multimedia applications that take advantage of Real-Time Transport Protocol, especially on slow links. Figure 49-13 illustrates Real-Time Transport Protocol header compression.

Figure 49-13 This Diagram Illustrates Real-Time Transport Protocol Header Compression



For compressed-payload audio applications, the RTP packet has a 40-byte header and typically a 20- to 150-byte payload. Given the size of the IP/UDP/Real-Time Transport Protocol header combination, it is inefficient to transmit an uncompressed header. Real-Time Transport Protocol header compression helps Real-Time Transport Protocol run more efficiently—especially over lower-speed links—by compressing the Real-Time Transport Protocol/UDP/IP header from 40 bytes to 2 to 5 bytes. This is especially beneficial for smaller packets (such as IP voice traffic) on slower links (385 kbps and below), where RTP header compression can reduce overhead and transmission delay significantly. Real-Time Transport Protocol header compression reduces line overhead for multimedia Real-Time Transport Protocol traffic with a corresponding reduction in delay, especially for traffic that uses short packets relative to header length.

RTP header compression is supported on serial lines using Frame Relay, High-Level Data Link Control (HDLC), or PPP encapsulation. It is also supported over ISDN interfaces.

A related IETF draft, called “Compressed RTP (CRTP),” defines essentially the same functionality.

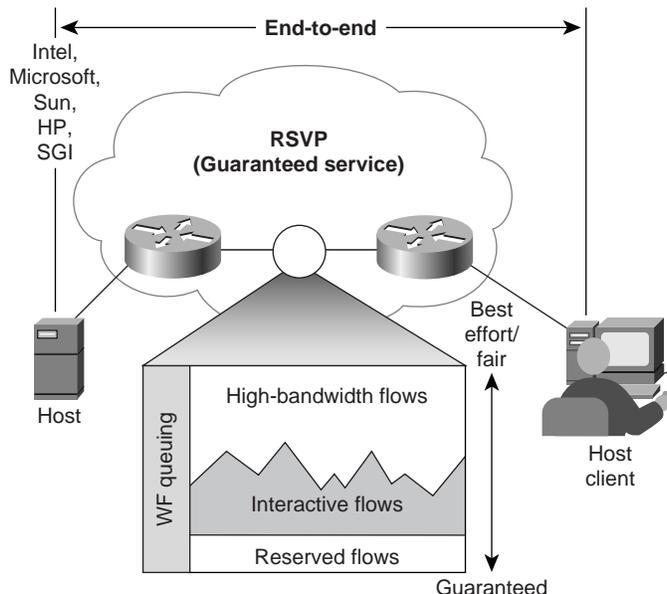
RSVP: Guaranteeing QoS

RSVP is an IETF Internet standard (RFC 2205) protocol for allowing an application to dynamically reserve network bandwidth. RSVP enables applications to request a specific QoS for a data flow, as shown in Figure 49-14. Cisco’s implementation also allows RSVP to be initiated within the network, using configured proxy RSVP. Network managers can thereby take advantage of the benefits of RSVP in the network, even for non-RSVP-enabled applications and hosts.

Hosts and routers use RSVP to deliver QoS requests to the routers along the paths of the data stream and to maintain router and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate, the largest amount of data that the router will keep in queue, and minimum QoS to determine bandwidth reservation.

WFQ or WRED acts as the workhorse for RSVP, setting up the packet classification and scheduling required for the reserved flows. Using WFQ, RSVP can deliver an integrated services guaranteed service. Using WRED, it can deliver a controlled load service. WFQ continues to provide its advantageous handling of nonreserved traffic by expediting interactive traffic and fairly sharing the remaining bandwidth between high-bandwidth flows; WRED provides its commensurate advantages for non-RSVP flow traffic. RSVP can be deployed in existing networks with a software upgrade.

Figure 49-14 This Figure Shows RSVP Implemented in a Cisco-Based Router Network



QoS Management

The introduction discussed a common method (and by no means the only method) for QoS management.

For baselining a network, you can use RMON probes and an application (such as Traffic Director) to develop a good understanding of traffic characteristics. The discovery feature in NBAR (discussed earlier in this chapter) provides a brief look at utilization on an interface basis, but RMON probes provide more complete information. In addition, targeted applications should be baselined (this is commonly measured by response time). This information helps to validate any QoS deployment. From this data, QoS policy is set and deployed.

Once deployed, it is important to evaluate the QoS policies and deployment and to decide whether additional services are needed. Internetwork Performance Monitor (IPM) can assist in determining if QoS policies continue to be effective by measuring response times within the internetwork. Comparing new baseline data for specific applications with the original baseline data will validate the QoS policies deployed. In addition, RMON probes should still continue to monitor the network because the traffic characteristics likely will change. A constant look at network traffic will help with changing trends and allow a network administrator to address new network requirements more expeditiously.

For the network-wide configuration of QoS in a Cisco network, Cisco's QoS Policy Manager (QPM) provides a graphical user interface for managing QoS in a network. Rules or policies are created and then downloaded to the devices. This simplifies QoS configuration of devices. QPM is compatible with Common Open Policy Server (COPS), a standard protocol for downloading policy to any COPS-compatible devices. The proposed standard (RFC 2748) is a simple client/server model for supporting policy control over QoS signaling protocols.

For device management of QoS, there is Cisco's Quality of Service Device Manager (QDM). QDM is a web-based Java application that is stored in the Flash file system of the router. The client browser makes a connection to the embedded web server of the router where the QDM application is stored and can configure that device for this web-based Java interface.

QoS on Ethernet

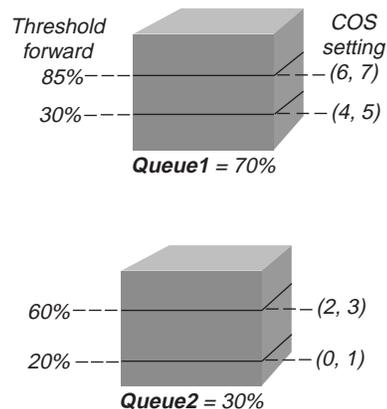
In the Catalyst line of multilayer switches is the capability to provide QoS at Layer 2. At Layer 2, the frame uses class of service (CoS) in 802.1p and Interlink Switch Link (ISL). CoS uses 3 bits, just like IP precedence, and maps well from Layer 2 to layer 3, and vice versa.

The switches have the capability to differentiate frames based on CoS settings. If multiple queues are present, frames can be placed in different queues and serviced via weighted round robin (WRR). This allows each queue to have different service levels. Within the queue, WRED thresholds are set. These thresholds are similar to the minimum thresholds set in WRED at Layer 3. They act as the starting point for the probability that a packet will be dropped.

Figure 49-15 explains the use of WRR with WRED using two queues with two thresholds each. This is referred to as *2Q2T*. In this instance, settings 4 to 7 are put in queue 1. Settings 0 to 3 are put in queue 2. Queue 1 is set to get service 70 percent of the time, and queue 2 gets service 30 percent of the time. In queue 1, when the queue is 30-percent full, settings 4 and 5 can be dropped. Not until the queue is 85-percent full are settings 6 and 7 dropped. In queue 2, when the queue is 20-percent full, settings 0 and 1 can be dropped. Not until the queue is 60-percent full can settings 2 and 3 be dropped.

Many implementations provided mapping of ToS (or IP precedence) to CoS. In this instance, an Ethernet frame CoS setting can be mapped to the ToS byte of the IP packet, and vice versa. This provides end-to-end priority for the traffic flow.

Figure 49-15 WRR with WRED Using Two Queues with Two Thresholds Each



Multiprotocol Label Switching: Allowing Flexible Traffic Engineering

Cisco's MPLS (also known as *tag switching*) feature contains the mechanisms to interoperate with and take advantage of both RSVP and IP precedence signaling. The tag switching header contains a 3-bit field that can be used as a traffic prioritization signal. It can also be used to map particular flows and classes of traffic along engineered tag-switching paths to obtain the required QoS through the tag-switching portion of a network.

QoS Policy Control

The QoS policy control architecture is being developed as a key initial piece of the CiscoAssure policy networking initiative. This initiative leverages standards-based QoS policy control protocols and mechanisms to implement QoS policy from a single console interface.

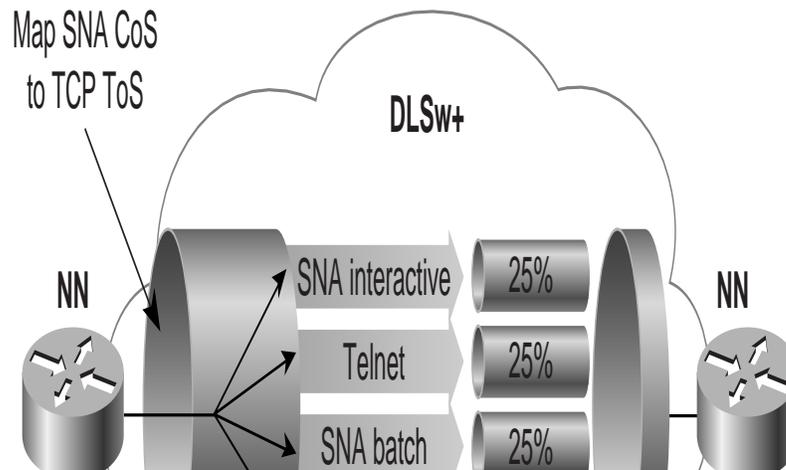
At the infrastructure level, packet classification is a key capability for each policy technique that allows the appropriate packets traversing a network element or particular interface to be selected for QoS. These packets can then be marked for the appropriate IP precedence in some cases, or can be identified as an RSVP. Policy control also requires integration with underlying data link layer network technologies or non-IP protocols.

SNA ToS

SNA ToS, in conjunction with data-link switching plus (DLSw+), allows mapping of traditional SNA class of service (CoS) into IP differentiated service. This feature takes advantage of both QoS signaling and pieces of the architecture. DLSW+ opens four TCP sessions and maps each SNA ToS traffic into a different session. Each session is marked by IP precedence. Cisco's congestion control technologies (CQ, PQ, and WFQ) act on these sessions to provide a bandwidth guarantee or other improved handling across an intranet, as shown in Figure 49-16. This provides a migration path for traditional SNA customers onto an IP-based intranet, while preserving the performance characteristics expected of SNA.

Thus, traditional mainframe-based, mission-critical applications can take advantage of evolving IP intranets and extranets without sacrificing the QoS capabilities historically provided by SNA networking.

Figure 49-16 SNA ToS, in Conjunction with DLSw, Allows Mapping of SNA CoS into IP Differentiated Services



QoS for Packetized Voice

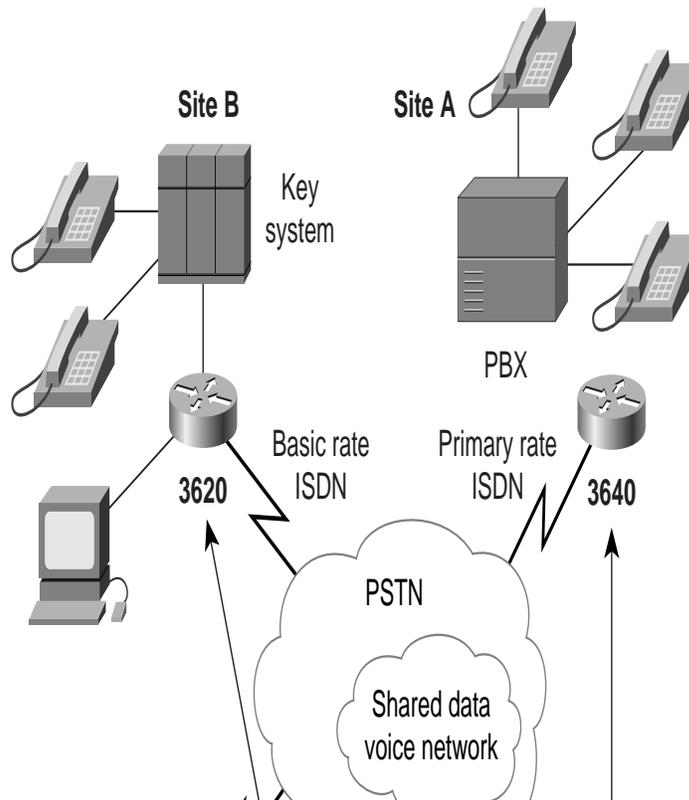
One of the most promising uses for IP networks is to allow sharing of voice traffic with the traditional data and LAN-to-LAN traffic. Typically, this can help reduce transmission costs by reducing the number of network connections sharing existing connections and infrastructure, and so on.

Cisco has a wide range of voice networking products and technologies, including a number of Voice over IP (VoIP) solutions. To provide the required voice quality, however, QoS capability must be added to the traditional data-only network. Cisco IOS software QoS features give VoIP traffic the service that it needs, while providing the traditional data traffic with the service that it needs as well.

Figure 49-17 shows a business that has chosen to reduce some of its voice costs by combining voice traffic onto its existing IP network. Voice traffic at each office is digitized on voice modules on 3600 processors. This traffic is then routed via H.323 Gatekeeper, which also requests specific QoS for the voice traffic. In this case, IP precedence is set to high for the voice traffic. WFQ is enabled on all the router interfaces for this network. WFQ automatically expedites the forwarding of high-precedence voice traffic out each interface, reducing delay and jitter for this traffic.

Because the IP network was originally handling LAN-to-LAN traffic, many datagrams traversing the network are large 1500-byte packets. On slow links (below T1/E1 speeds), voice packets may be forced to wait behind one of these large packets, adding tens or even hundreds of milliseconds to the delay. LFI is used in conjunction with WFQ to break up these jumbograms and interleave the voice traffic to reduce this delay as well as jitter.

Figure 49-17 This Diagram Provides an Overview of a QoS VoIP Solution

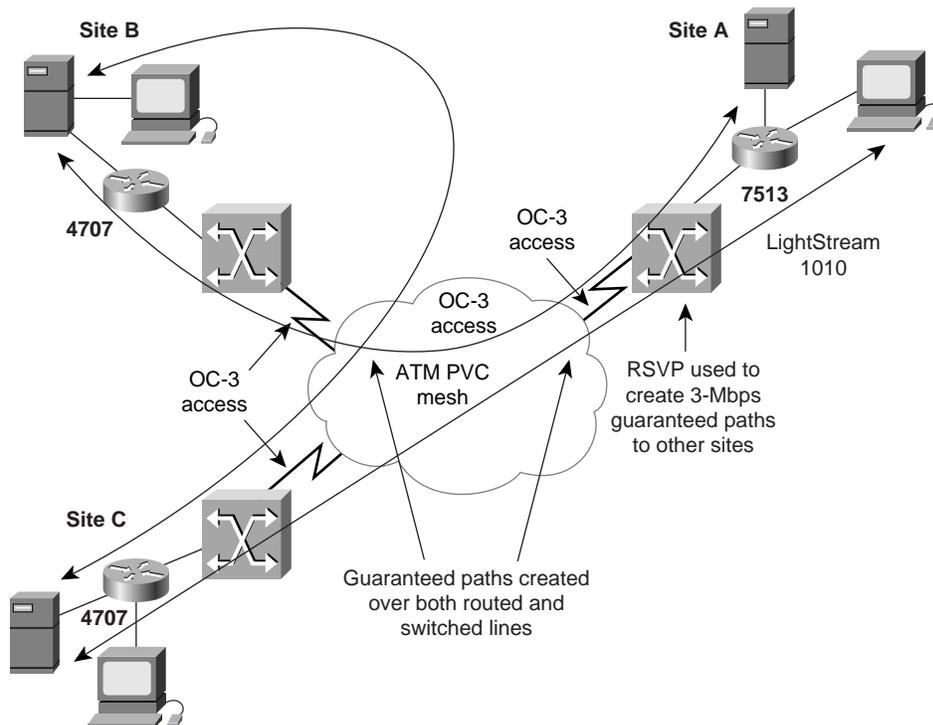


QoS for Streaming Video

One of the most significant challenges for IP-based networks, which have traditionally provided only best-effort service, has been to provide some type of service guarantees for different types of traffic. This has been a particular challenge for streaming video applications, which often require a significant amount of reserved bandwidth to be useful.

In the network shown in Figure 49-18, RSVP is used in conjunction with ATM PVCs to provide guaranteed bandwidth to a mesh of locations. RSVP is configured from within Cisco IOS to provide paths from the router networks, at the edges, and through the ATM core. Simulation traffic then uses these guaranteed paths to meet the constraints of geographically distributed real-time simulation. Video-enabled machines at the various sites also use this network to do live videoconferencing.

Figure 49-18 The Network Diagram Shows the Use of RSVP in a Meshed ATM Environment



In this instance, OC-3 ATM links are configured with multiple 3-Mbps PVCs connecting to various remote sites. RSVP ensures that QoS from this PVC is extended to the appropriate application across the local routed network. In the future, Cisco IOS will extend this RSVP capability to dynamically set up ATM SVCs. This will reduce configuration complexity and add a great degree of automatic configuration.

Summary

Cisco IOS QoS provides a set of tools to provide a flow(s) with the necessary network services to work successfully.

QoS provides differentiated services, which provide higher-priority to flows, or guaranteed services that provide an assured service level. Both of these are contrasted by best-effort services, which is provided by what is generally considered a lack of QoS. FIFO provides best-effort service. Here, flows are not differentiated and are serviced on a first-come, first-served basis.

Using classification tools (PBR, CAR, and NBAR), flows are identified and optionally are marked for use by other QoS tools throughout the internetwork. Congestion-management tools (PQ, CQ, WFQ, and CBWFQ) all manage the delivery of packets when there is more bandwidth than the link can handle. Queue management (WRED) is used for congestion avoidance within individual queues, as well as to prevent congestion in the internetwork. Using the behavior of TCP, WRED can throttle the speed of flows by dropping certain flows. It can also provide priority by dropping low-priority flows before high-priority flows. Link efficiency tools (LFI and RTP header compression) provide relief for time-sensitive low-bandwidth flows. LFI does this by fragmenting large packets. RTP header compression does this by reducing the overhead for RTP packets.

Guaranteed services is generally provided by RSVP, although CBWFQ could be considered a form of guaranteed services. RSVP is a signaling protocol that signals the network to provide guaranteed services for the entire path of the packet. CBWFQ differs in that it guarantees service onto an interface.

QoS Looking Forward

In a continued evolution toward end-to-end services, Cisco is expanding QoS interworking to operate more seamlessly across heterogeneous link layer technologies, and working closely with host platform partners to ensure interoperability between networks and end systems.

QoS is on the forefront of networking technology. The future brings us the notion of user-based QoS in which QoS policies are based on a user as well as application. Capabilities such as NBAR and its ability to read deeper into the packet provides a robust implementation for identifying flows. Cisco's end-to-end QoS solutions (from desktop to desktop) make a Cisco network the premier provider of end-to-end quality of service.

Review Questions

Q—*What is the main goal of QoS?*

A—QoS provides preferential treatment to an identified flow(s). You must also provide enough service for other flows to successfully pass traffic. Providing priority to a certain flow(s) by breaking other applications is not desirable.

Q—*What are the types of QoS tools?*

A—

- **Classification**—These tools identify and (if desired) mark flows.
- **Congestion management**—These tools queue and service flows in different ways to provide preferential treatment to a certain flow(s).
- **Congestion avoidance**—This tool prevents a queue from filling, to allow high-priority traffic to enter the queue. This tool also provides for overall congestion avoidance in an Internet/intranet.
- **Shaping/policing**—These tools limit the bandwidth that a flow(s) uses.
- **Link efficiency**—These tools provide a method of mitigating delay experienced on lower-speed links.

Q—*What is signaling?*

A—Signaling notifies the network in regard to the priority of a flow(s). Most commonly, this is accomplished through the setting of IP precedence bits in the ToS byte, setting Class of Service bits (for Ethernet), and RSVP for end-to-end reservation.

Q—*What is IP precedence?*

A—IP precedence consists of the 3 most significant bits of the ToS byte in the IP header. It is used to mark a packet to notify the network in regard to the importance of the packet. The 3 bits allow settings from 0 to 7 (6 and 7 are reserved and should not be set by a network administrator).

Q—*How does flow-based WFQ (WFQ) differ from class-based WFQ (CBWFQ)?*

A—

- WFQ provides a queue for each flow. CBWFQ provides classes that can consist of more than one flow.

- WFQ creates fairness among all flows (given equal IP precedence settings). CBWFQ has classes of flows that are provided a user-determined minimum amount of bandwidth.
- CBWFQ supports WRED.

Q—*What is used for queue management to provide congestion avoidance? How does it avoid congestion?*

A—Weighted early random detection avoids congestion by the following actions:

- Trying to make sure that the queue does not fill up, so there is room for high priority packets
- Providing an algorithm that drops packets that are of lower priority before dropping higher-priority packets

Q—*What are the two primary uses for CAR?*

A—

- Classifying packets using IP precedence bits or QoS groups (for D-WFQ)
- Limiting the amount of traffic (or policing) that a flow(s) can pass

Q—*What QoS tool would you use to guarantee a minimum amount of bandwidth?*

A—CBWFQ.

Q—*What QoS tool would you use to limit a flow to a maximum amount of bandwidth?*

A—CAR or GTS/FRTS.

Q—*What does NBAR do? What are two unique features of it?*

A—NBAR provides for greater granularity of identification of a flow. By looking deeper into the packet, NBAR can identify flows such as URL (instead of merely by HTTP port 80).

Two unique features are:

- Protocol discovery, in which the router can identify protocols and provide statistical data on each protocol
- PDLMs, which provide easy upgrade of the protocols that NBAR can identify

Q—*What is a common use for traffic shaping?*

A—One common use is in a hub-and-spoke topology, where a single high-speed link at the central site terminates a number of lower-speed remote links. With such a topology, many will traffic shape at the central site, so the slower remote site links are not overrun, causing packets to drop.

Q—*What tool is used for Integrated QoS?*

A—RSVP.

For More Information

Cisco Systems. *Cisco IOS 12.0 Quality of Service*. Indianapolis: Cisco Press, 1999.

Ferguson, Paul, and Huston, Geoff. *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*. New York: John Wiley & Sons, 1998.

Lee, Donn. *Enhanced IP Services*. Indianapolis: Cisco Press, 1999.

Vegesna, Srinivas. *IP Quality of Service for the Internet and the Intranets*. Indianapolis: Cisco Press, 2000.

Cisco IOS QoS (<http://www.cisco.com/warp/public/732/Tech/quality.shtml>)

RFC 2386, "A Framework for QoS-Based Routing in the Internet."

■ For More Information