



CCNP ONT

Quick Reference Sheets

Exam 642-845

Brent Stewart
Denise Donohue

Network Architecture

Cisco VoIP

QoS Overview

QoS Details

AutoQoS

Wireless Scalability



About the Authors

Brent Stewart, CCNP, CCDP, MCSE, Certified Cisco Systems Instructor, is a network administrator for CommScope. He participated in the development of BSCI, and has separately developed training material for ICND, BSCI, BCMSN, BCRAN, and CIT. Brent lives in Hickory, NC, with his wife, Karen and children, Benjamin, Kaitlyn, Madelyn, and William.

Denise Donohue, CCIE No. 9566, is a Design Engineer with AT&T. She is responsible for designing and implementing data and VoIP networks for SBC and AT&T customers. Prior to that, she was a Cisco instructor and course director for Global Knowledge. Her CCIE is in Routing and Switching.

Icons Used in This Book



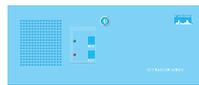
Router

7507
RouterMultilayer Switch
with TextMultilayer
SwitchCommunication
Server

Switch



Internal Firewall



IDS

Web
Browser

Database



App Server

CHAPTER 1

Network Architecture

Modern converged networks include different traffic types, each with unique requirements for security, Quality of Service (QoS), transmission capacity, and delay. Some examples include:

- Voice signaling and bearer
- Core application traffic, such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM)
- Database transactions
- Multicast multimedia
- Network management
- “Other” traffic, such as web pages, e-mail, and file transfer

Cisco routers are able to implement filtering, compression, prioritization, and policing (dedicating network capacity). Except for filtering, these capabilities are referred to collectively as QoS.

Although QoS is wonderful, it is not the only way to address bandwidth shortage. Cisco espouses an idea called the Intelligent Information Network (IIN). IIN builds on standard network design models to enable these new services to be reliable and layered on top of traditional data delivery.

SONA and IIN

IIN describes an evolutionary vision of a network that integrates network and application functionality cooperatively and allows the network to be smart about how it handles traffic to minimize the footprint of applications. IIN is built on top of the Enterprise Composite Model and describes structures overlaid on to the Composite design as needed in three phases.

Phase 1, “Integrated Transport,” describes a converged network, which is built along the lines of the Composite model and based on open standards. This is the phase that the industry has been transitioning. The Cisco Integrated Services Routers (ISR) are an example of this trend.

Phase 2, “Integrated Services,” attempts to virtualize resources, such as servers, storage, and network access. It is a move to an “on-demand” model.

By “virtualize,” Cisco means that the services are not associated with a particular device or location. Instead, many services can reside in one device to ease management, or many devices can provide one service that is more reliable.

An ISR brings together routing, switching, voice, security, and wireless. It is an example of many services existing on one device. A load balancer, which makes many servers look like one, is an example of one service residing on many devices.

VRFs are an example of taking one resource and making it look like many. Some versions of IOS are capable of having a router present itself as many virtual router (VRF) instances, allowing your company to deliver different logical topologies on the same physical infrastructure.

CHAPTER 1

NETWORK ARCHITECTURE

Server virtualization is another example. The classic example of taking one resource and making it appear to be many resources is the use of a virtual LAN (VLAN) and a virtual storage area network (VSAN).

Virtualization provides flexibility in configuration and management.

Phase 3, “Integrated Applications,” uses application-oriented networking (AON) to make the network application-aware and to allow the network to actively participate in service delivery.

An example of this Phase 3 IIN systems approach to service delivery is Network Admission Control (NAC). Before NAC, authentication, VLAN assignment, and anti-virus updates were separately managed. With NAC in place, the network is able to check the policy stance of a client and admit, deny, or remediate based on policies.

IIN allows the network to deconstruct packets, parse fields, and take actions based on the values it finds. An ISR equipped with an AON blade might be set up to route traffic from a business partner. The AON blade can examine traffic, recognize the application, and rebuild XML files in memory. Corrupted XML fields might represent an attack (called *schema poisoning*), so the AON blade can react by blocking that source from further communication. In this example, routing, an awareness of the application data flow, and security are combined to allow the network to contribute to the success of the application.

Services-Oriented Network Architecture (SONA) applies the IIN ideal to Enterprise networks. SONA breaks down the IIN functions into three layers:

- Network Infrastructure—Hierarchical converged network and attached end systems.
- Interactive Services—Resources allocated to applications.
- Applications—Includes business policy and logic

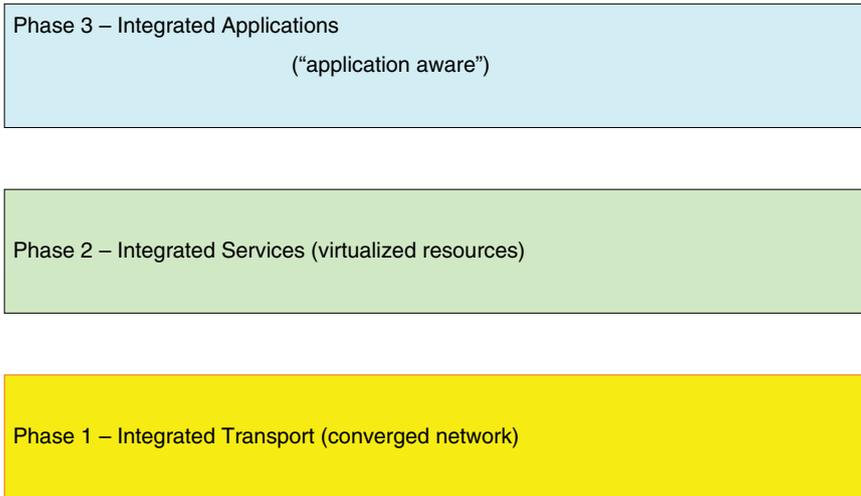
IOS features, such as Survivable Remote Site Telephony (SRST) and AutoQoS, cooperate with centralized services to increase the resiliency of the network by easily distributing network application logic to the edges of the enterprise, so that the entire network participates in operations instead of just the core.

Figure 1-1 shows how IIN and SONA more specifically compare.

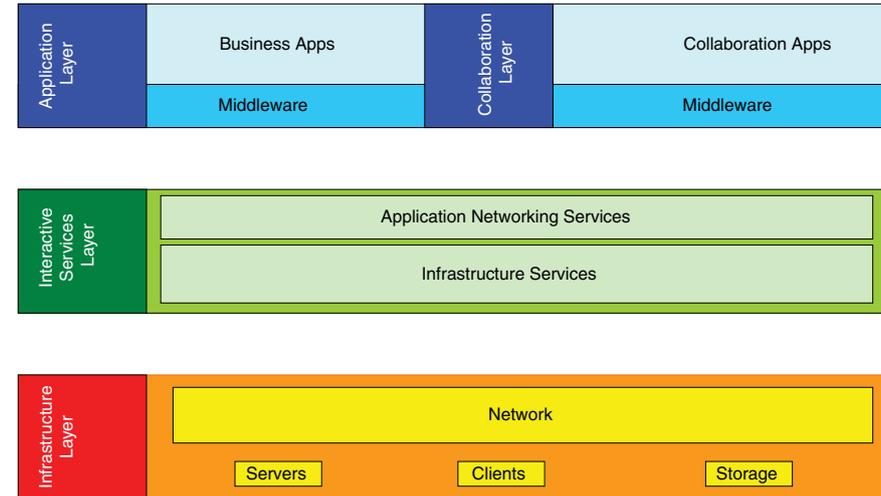
NETWORK ARCHITECTURE

FIGURE 1-1 IIN and SONA

IIN Phases



SONA Framework Layers



Network Models

Cisco has developed specific architecture recommendations for Campus, Data Center, WAN, branches, and telecommuting. These recommendations add specific ideas about how current technologies and capabilities match the network roles within an enterprise.

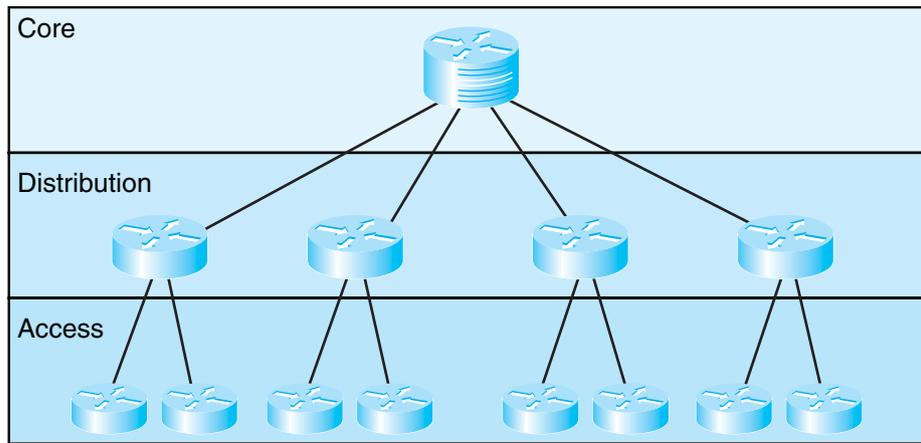
Each of these designs builds on a traditional hierarchical design and adds features such as security, QoS, caching, and convergence.

Hierarchical Design Model

The traditional model provided a high-level idea of how a reliable network could be conceived, but it was short on specific guidance.

Figure 1-2 is a simple drawing of how the three-layer model might have been built. A distribution layer-3 switch is used for each building on campus, tying together the access switches on the floors. The core switches link the various buildings together.

NETWORK ARCHITECTURE

FIGURE 1-2 Three-Layer Hierarchical Design

The layers break a network in the following way:

- Access layer—End stations attach to the network using low-cost devices.
- Distribution layer—Intermediate devices apply policies.
 - Route summarization
 - Policies applied, such as:
 - Route selection
 - Access lists
 - Quality of Service (QoS)

- Core layer—The backbone that provides a high-speed path between distribution elements.
 - Distribution devices are interconnected.
 - High speed (there is a lot of traffic).
 - No policies (it is tough enough to keep up).

Enterprise Composite Network Model

The newer Cisco model—the Enterprise Composite Model—is significantly more complex and attempts to address the shortcomings of the Hierarchical Design Model by expanding the older version and making specific recommendations about how and where certain network functions should be implemented. This model is based on the principles described in the Cisco Architecture for Voice, Video, and Integrated Data (AVVID).

The Enterprise Composite Model is broken into three large sections:

- Enterprise Campus
- Enterprise Edge
- Service Provider Edge

CHAPTER 1

NETWORK ARCHITECTURE

The first section, the Enterprise Campus, looks like the old Hierarchical Design Model with added details. It features six sections:

- Campus Backbone
- Building Distribution
- Building Access
- Management
- Edge Distribution—A distribution layer out to the WAN
- Server Farm—For Enterprise services

The Enterprise Edge details the connections from the campus to the Wide Area Network and includes:

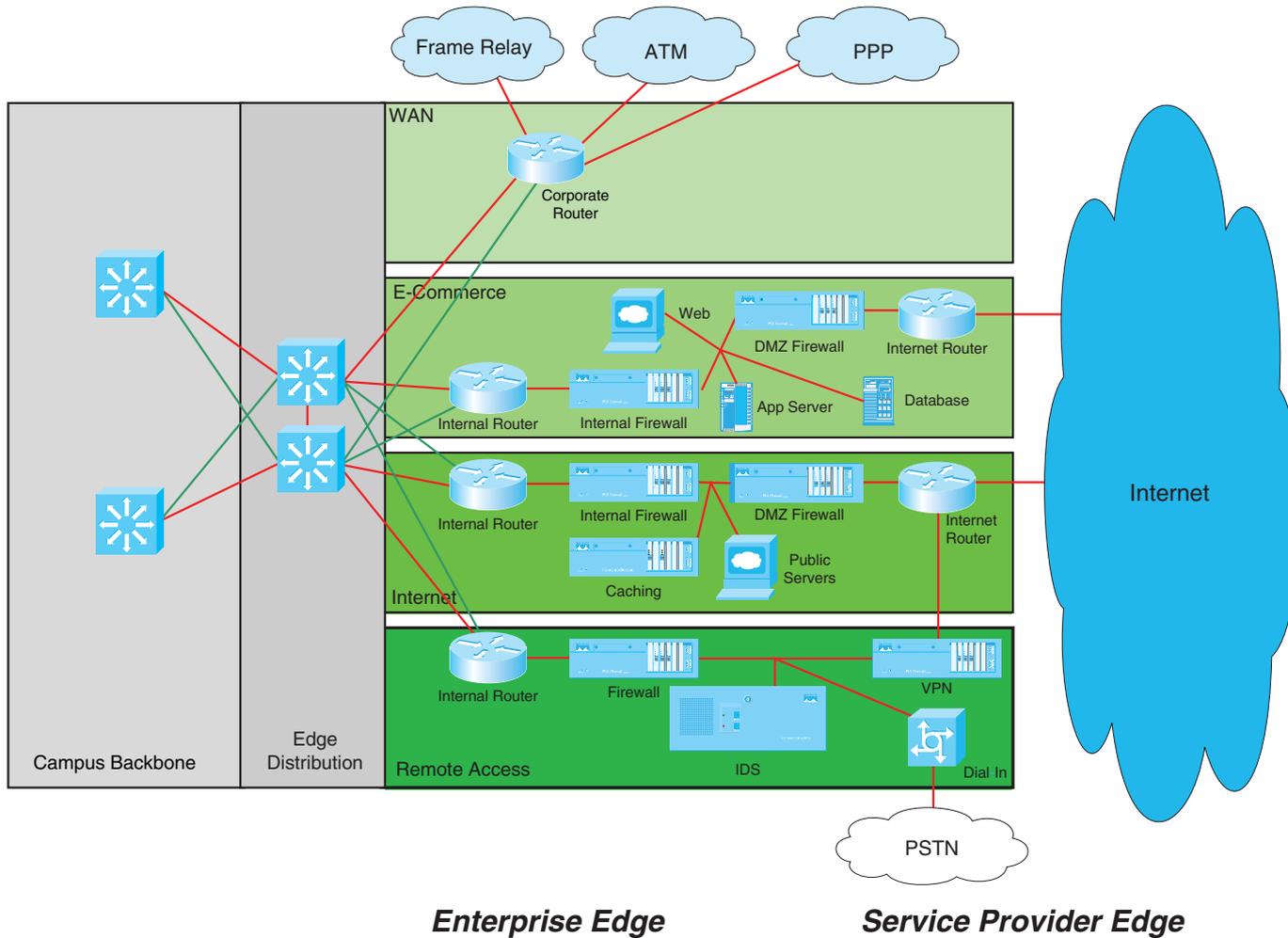
- E-commerce
- Internet connectivity
- Remote access
- WAN

The Service Provider Edge is just a list of the public networks that facilitate wide-area connectivity. These include:

- Internet service providers (ISP)
- Analog phone dial up
- Frame Relay, ATM, and PPP, which have private connectivity

Figure 1-3 shows the Campus, Enterprise Edge, and Service Provider Edge modules assembled. Security implemented on this model is described in the Cisco SAFE (Security Architecture for Enterprise) blueprint.

FIGURE 1-3 Enterprise Design



Cisco VoIP

Introduction

Voice over IP (VoIP) is a set of technologies that seeks to replace traditional analog voice services. There are three main compelling benefits to VoIP:

- VoIP makes better use of network capacity. Traditional voice uses a 64-Kbps circuit, even when it is not active, but VoIP can use much less and no capacity when the line is not in use.
- VoIP allows new and revolutionary features, such as the following:
 - Integration of voice and data systems (so that operators get customer information popped on to the screen when a phone call arrives).
 - Voice CODECs can improve sound quality (at the expense of bandwidth).
 - Integration with new clients. Instead of an analog phone, VoIP clients can include television boxes, Personal Digital Assistants (PDAs), cell phones, laptops, and so on.
- VoIP can save money by avoiding toll calls.

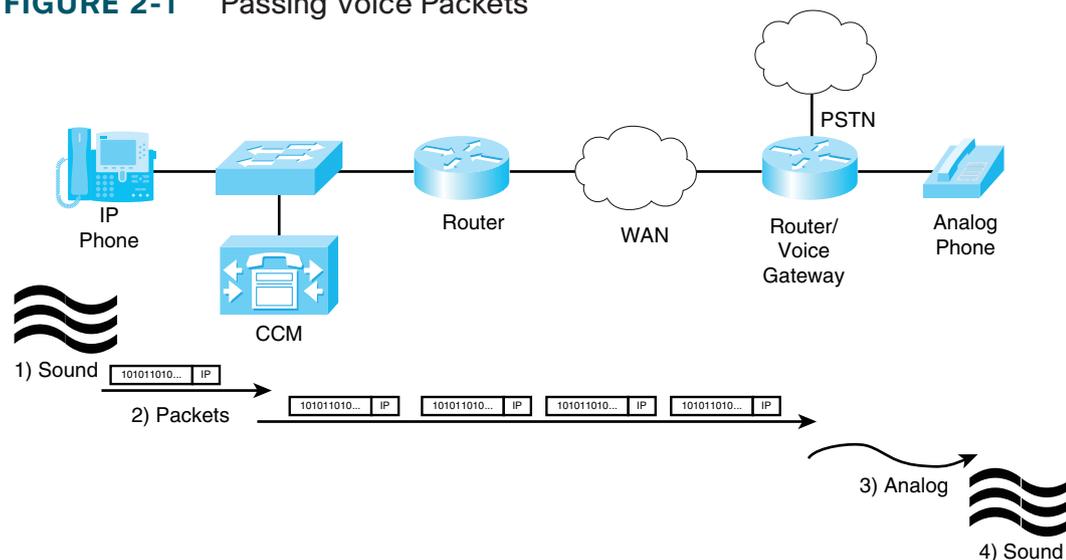
IP telephony solutions include many pieces:

- Internet Protocol (IP) phones
- Analog phones connected to IP by a Gateway
- Control and number resolution by a Gatekeeper
- Conferencing capabilities provided by a multipoint control unit (MCU)
- Applications, such as directories and product information that interface with smart IP phones

Transmission

Figure 2-1 shows a VoIP transmission scenario.

FIGURE 2-1 Passing Voice Packets



CHAPTER 2

CISCO VoIP

Voice is passed over an IP network by packetization. Example 2-1 shows an IP phone communicating with an older analog phone, but any combination of the two is supported. The numbered list below matches the steps involved in taking sound and converting it packets and then back to sound:

1. Incoming sounds are grouped into slices of sound (typically 20 ms), sampled, and digitized.
2. Each slice of sound is fitted with headers (data link, IP, User Datagram Protocol [UDP], and Reliable Transport Protocol [RTP]) and transmitted across the IP network.
3. Because the analog phone doesn't understand packets, a gateway (in this case, it is housed in a router) translates the stream of packets into an analog electrical signal.
4. The analog phone receives an analog electrical signal and sends it to a speaker, where the recording is restored to audio.

Cisco routers are commonly deployed as gateways. Three types of analog connections are supported:

- Foreign Exchange Station (FXS)—FXS ports connect analog phones. FXS ports supply line voltage.
- Foreign Exchange Office (FXO)—FXO ports connect to a Private Branch Exchange (PBX) or to the Public Switched Telephone Network (PSTN). FXO ports receive line voltage.

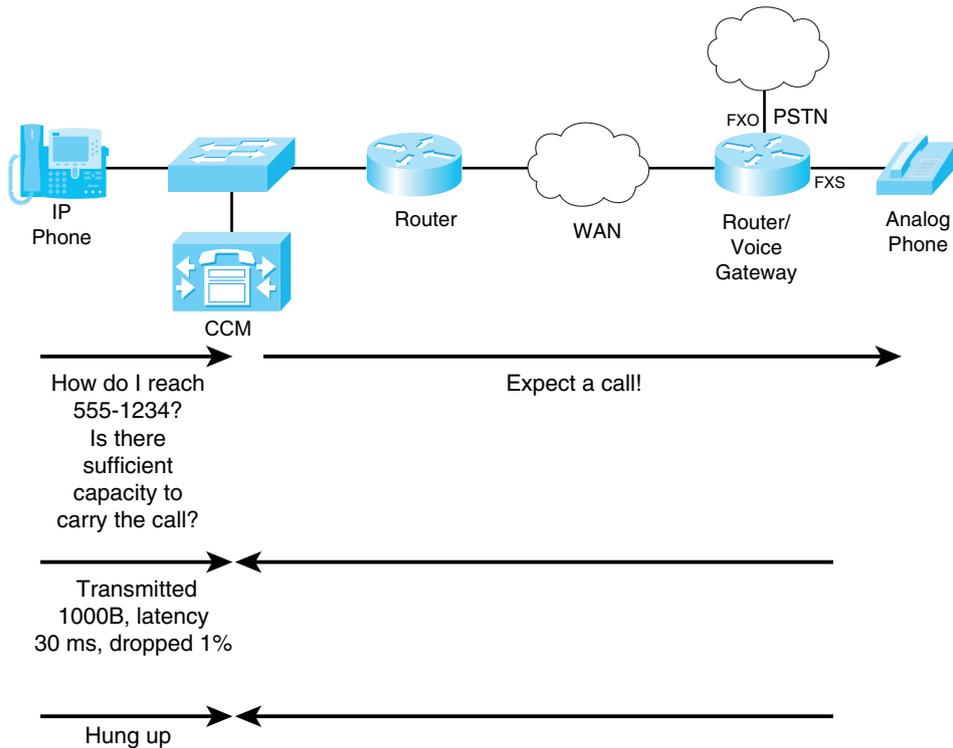
- E&M—E&M (which is alternately said to stand for Ear and Mouth or Earth and Magneto) interfaces supply advanced signaling to a PBX using a separate set of wires.

Three digital phone ports are supported:

1. ISDN—ISDN interfaces support advanced Q.931 signaling.
2. T1/E1 CCS (Common Channel Signaling)—T1/E1 CCS uses a channel for signaling. ISDN PRI uses CCS.
3. T1/E1 CAS (Channel Associated Signaling)—Robs bits from the analog waveform for signaling and is not as full-featured.

Although Figure 2-1 focused on the flow of voice records, signaling is equally important to understand. Signaling is what tells the system which phone to ring and when the line is hung up. Phone companies, in particular, are interested in this (and might write it Signaling) because signaling is used in billing. Figure 2-2 shows the types of signaling that are expected.

FIGURE 2-2 Signaling



Signaling plays several important roles:

- Information about the receiver is obtained.
- Capacity is checked before starting; otherwise, call quality suffers.
- Call quality is monitored so that adjustments may be made to maintain call quality.

- Connect and disconnect times are kept for billing.

In Figure 2-2, a Call Manager is shown receiving the signaling. A Call Manager allows centralized call control, which provides oversight of the call and records of connections and quality. Voice trunking may be accomplished without such supervision (called distributed call control), but care must be taken to not overburden links and quality must be manually maintained.

Packetization

Before voice may be transmitted over a network, sound has to be captured from a microphone and digitized. The digital recording is then chopped into sections (each is typically 20 ms), which are sent sequentially and replayed in order out a speaker.

Sound is captured at a microphone by sampling (periodically taking a power reading). The Nyquist theorem says that to reproduce a signal, sampling must occur at twice the maximum frequency. The phone system is designed to capture frequencies less than 4 kHz, which are samples of 8,000 times per second.

Pulse Amplitude Modulation (PAM) is used in the PSTN. Samples are quantized to 8-bit numbers 8,000 times per second (yielding a 64-kbps DS0).

Two forms of quantization are used. A linear scale is used in the U.S., while abroad, a logarithmic scale is used. The U.S. system (called μ -law) was developed earlier, and it suffered from lower-powered

sampling systems. A-law (logarithmic sampling) was developed later to be different and give domestic opportunities to European companies that were still recovering from World War II. A-law benefits from greater computing resources, and the logarithmic scale does a better job of reproducing sound.

After captured, Pulse Amplitude Modulation (PAM) samples are encoded using a coder/decoder (CODEC). Coders work using two main techniques: PCM, which encodes the signal straight to bits, and CELP, which matches the waveform to a predefined library and sends a code.

G.711 and G.726 use PCM. G.711 uses 8 bits per sample, whereas G.726 uses 7, 6, or 5, depending on the desired quality. G.728 and 729 use CELP. Resulting voice quality is shown in Table 2-1. Remember that the figures for bandwidth do not include headers.

TABLE 2-1 Details of Five CODECs

CoDec	Technique	Bandwidth	20 ms Sample Size	Quality
G.711	PCM	64	160	4.10
G.726	ADPCM	32, 24, 16	80, 40, 20	3.85
G.728	LDCELP	16	40	3.61
G.729	CS-ACELP	8	20	3.92
G.729A	CS-ACELP	8	20	3.90

Voice quality is measured on a scale called Mean Opinion Score (MOS). MOS has been scored by averaging judges' scores: a MOS of 5 is perfect, whereas 4 is toll quality, and anything less gets less and less acceptable. Perceptual Speech Quality Measurement (PSQM) is a

newer technique that compares wave forms pre- and post-transmission and grades on a scale of 0 to 6.5. PSQM is repeatable and less arbitrary, but the non-traditional scale made it hard to compare to MOS, so Perceptual Evaluation of Speech Quality (PESQ) is a version of PSQM that uses an MOS scale.

All the ideas discussed in this section—sampling, quantization, encoding, and compression—depend on specialized processors called Digital Signal Processors (DSP). DSPs are also used for translating CODECs (transcoding) and for conferencing.

Transmitting

VoIP depends on three pillars:

- Signaling is used for call setup and teardown. Common protocols include H.323, SIP, and MGCP.
- Packetization sends voice samples inside IP packets.
- QoS prioritizes VoIP traffic.

There are three reasons users will throw new VoIP phones at you and beg for old analog headsets: packet loss, delay, and echo. The biggest reason for packet loss is tail-drop in queues, which is solved through QoS. The biggest issue with delay is variation in delay (called jitter), which causes large de-jitter buffers to be used and causes more delay. The solution to jitter is QoS. Echo is solved through a technique called echo-cancellation (G.168), which is on by default and compensates for delay.

Voice samples are encapsulated in Real Time Protocol (RTP) packets. Voice does not need the reliability provided by TCP; by the time a retransmission happened, the moment to play the sound would have passed. Voice does need a way to order samples and recognize the time between samples, which UDP by itself doesn't allow. RTP is a protocol within UDP that adds the necessary features.

A complete VoIP packet needs to include a data link header (Ethernet has a 14 Byte header and 4 Bytes CRC), an IP header (20 Bytes), an 8 Byte UDP header, and 12 Bytes for RTP. Each 20ms sample therefore includes 58 Bytes of overhead. G.711 sends 8000 Bytes per second (20ms would therefore need 160 Bytes), so about a quarter of the transmission is headers!

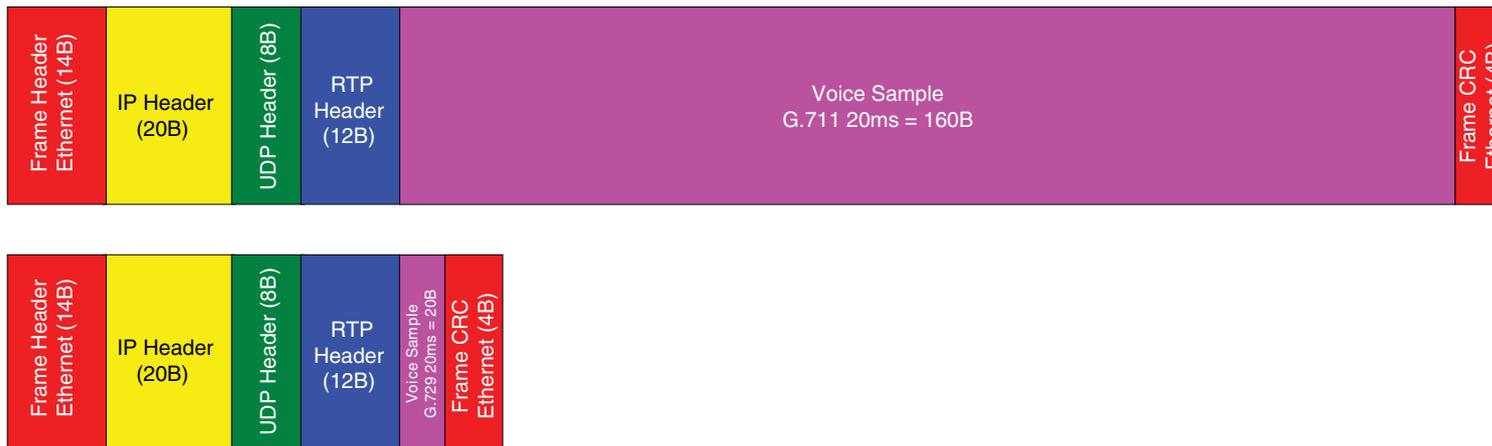
Figure 2-3 shows the header overhead graphically and Table 2-1 shows the bandwidth consumed by the various CODECs, including headers. If the phone uses 20 ms samples (50 samples per second), then there will be 50 headers. G.711, instead of being 64 Kbps, turns out to be:

$$(\text{Headers} + \text{Sample}) * 50/\text{s} =$$

$$(14\text{B} + 20\text{B} + 8\text{B} + 12\text{B} + 160\text{B} + 4\text{B}) * 50/\text{s} =$$

$$218 \text{ B} * 50/\text{s} = 10900 \text{ B/s} = 87,200 \text{ b/s} = 87.2 \text{ Kbps}$$

FIGURE 2-3 Protocol Layout for Voice Transmission over IP



CHAPTER 2

CISCO VoIP

Note that G.729 uses 20-byte samples, and so it needs only 31.2kbps.

At this point, you may have sticker shock. If G.729 is billed as 8 Kbps per conversation, 31.2 Kbps seems extreme. There are ways to mitigate the difference, although the techniques do not completely erase the need for headers.

One way is to use RTP header compression. Header compression is configured per link and remembers previous IP, UDP, and RTP headers, substituting 2B- or 4B-labels subsequently. By taking the header set from 40B to 4B, cRTP delivers G.729 using 22-B headers and a consumption of 16.8 Kbps!

Voice Activity Detection (VAD) is a technology that recognizes when you are not talking and ceases transmission, thus saving bandwidth. In normal speech, one person or the other is talking less than 65 percent of the time (there are those long, uncomfortable silences right after you say, “*You did what?*”). VAD can therefore dramatically reduce demands for bandwidth.

The bad news with VAD is that it doesn’t help with music (such as hold music) and that it creates “dead air,” which can be mistaken for disconnection. Some phones, in fact, will play soft static to reinforce that the line is still live (this is called comfort noise).

Bandwidth Requirements

Various tradeoffs go into selecting the parameters for a VoIP implementation, each of which affect voice quality and bandwidth. These include:

- Sample period—Each packet represents a period of time. Longer periods mean that fewer headers have to be sent, but add delay while accumulating samples. Shorter periods mean more header overhead, but less delay.
- Packets per second—One second divided by the sample period.
- CODEC—Each coding protocol uses more or less bandwidth and offers more or less quality. See Table 2-1 for details.
- IP/UDP/RTP overhead—40 B, or 4 B if using cRTP with checksum, or 2B if using cRTP without checksum.
- Data Link overhead—Ethernet uses 18 B. This varies by protocol.

A Worksheet for Calculating VoIP Bandwidth

Sample period = _____ Packets per second = _____

CODEC = _____ Sample size = _____

Header size (40 B without cRTP, 4 B with) = _____

Data Link overhead = _____

Total packet = sample + header + data link = _____

*Packets per second = × _____

Multiply by 8 to get b/s × 8 _____

Divide by 1000 to get kb/s / 1000 _____

An Example for G.711, No Compression over Ethernet, 20 ms Samples

Sample period = 20 ms Packets per second = 50/s

CODEC = G.711 Sample size = 160B

Header size (40 B w/o cRTP, 4 B with) = + 40 B

Data Link overhead = + 18B

Total packet = sample + header + data link = 218B

* Packets per second = × 50/s

10900 B/s

Multiply by 8 to get b/s × 8 b/B

87200 b/s

Divide by 1000 to get kb/s / 1000 Kb/b

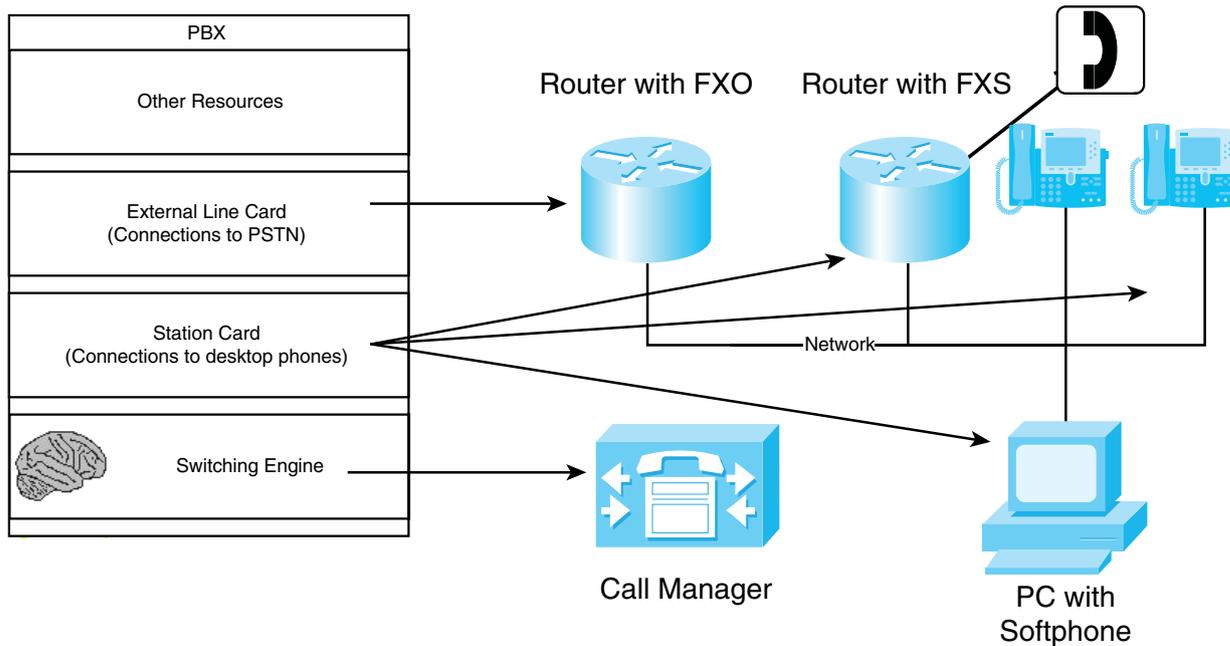
87.2 Kb/s

Additionally, allot 30–73 bytes for IPSec headers if used.

Implementing IP Telephony

In the enterprise, IP telephony is deployed to replace a PBX. A typical PBX contains a switching function (the “brains”) and cards that attach extensions (station cards) and connect to the outside world (line cards). Figure 2-4 shows the evolution from an old PBX to a modern distributed IP telephony solution.

FIGURE 2-4 Evolution from PBX to IP Telephony



A Cisco Call Manager takes the place of the “brains” and helps end stations understand how to reach each other. CCM also oversees the dial plan, produces utilization reports, and determines functionality. CCM is typically deployed in a cluster, so that the system does not rely on one machine.

NOTE

Cisco Call Manager Express runs on a router and can be used for small offices. Routers are also deployed as backup call managers (this is called Survivable Remote Site Telephony or SRST), so being disconnected from a remote CCM does not disable a branch phone system.

IP phones and soft phones connect directly to the network, whereas legacy phones connect to the network through FXS ports on routers. Routers operating this way are called gateways. Think of the network and gateways as being equivalent to the station cards in an old PBX.

Routers with external connections, such as FXO ports, are also called gateways. In this scenario, however, the router takes the place of an external line card.

Telephony deployments follow one of four models:

- **Single Site**—One office uses a CCM cluster to handle local phones.
- **Multisite with centralized call processing**—One CCM cluster at headquarters handles local and remote phones. Branch offices typically are set up with SRST.
- **Multisite with distributed call processing**—Each site has a CCM cluster.
- **Clustering over WAN**—The CCM cluster is distributed between locations.

One other piece, not shown or discussed so far, is Call Admission Control (CAC). Usually data is described as “better to degrade service than to deny service,” which is to say that when more users need

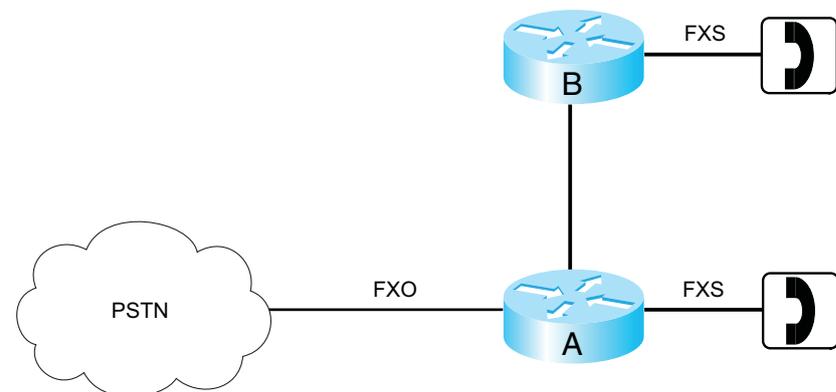
service, everyone goes slower. But the voice world has never said that one more user would cause quality to go down. In fact, voice engineers would say “It’s better to deny service than to degrade service.”

The problem is, how do you limit the number of calls going across a VoIP network? Intuitively, there is nothing to prevent one more person from calling. This is where CAC comes in. CAC is a tool that tracks the number of calls and—when it reaches a threshold value—prevents another call. CAC is an important part of an IP telephony solution.

Configuring Cisco Routers to Support VoIP

Consider Figure 2-5 as a precursor to reading about the configuration of a router with FXO, FXS, and VoIP neighbors.

FIGURE 2-5 Voice over IP Network Topology



CHAPTER 2

CISCO VoIP

The configuration is shown in Example 2-1.

EXAMPLE 2-1 Configuration of Router A

```
hostname router_A
interface s0/0
 ip address 172.21.77.1 255.255.255.0

dial-peer voice 1 voip
 destination-pattern 720
 session-target ipv4:172.21.77.2

dial-peer voice 2 pots
 destination-pattern 721
 port 1/0/0

dial-peer voice 3 pots
 destination-pattern 9
 port 2/0/0
```

In Example 2-1, the dial plan consists of three patterns: Dialing 9 gets an outside line, dialing 720 rings the phone on the other router, and 721 rings the pots line on the local router.

All patterns need a destination-pattern statement to configure the dial plan. Phones reached over IP also need a session target, whereas directly attached analog phones are referenced by port.

CHAPTER 3

QoS Overview

Quality of service (QoS) configurations give special treatment to certain traffic at the expense of others. This helps make your network performance more deterministic and predictable for this traffic. Using QoS in the network addresses the following problems:

- Lack of bandwidth for important applications
- Delay of sensitive data, such as voice and video
- Jitter (variable delay)
- Packet loss due to data being dropped at a congested interface

Bandwidth

In a network with several hops, the available bandwidth is only as much as the smallest link. When multiple applications and multiple flows use the same links, the available bandwidth per application is even smaller—it equals the smallest link bandwidth divided by the number of flows. Insufficient bandwidth especially affects time-sensitive and interactive traffic, and traffic with large flows.

You can increase link speeds to get more bandwidth—that can be expensive, time-consuming, and introduce technological difficulties. Alternatively, QoS mechanisms can guarantee bandwidth to specific applications.

Compressing the traffic on slower links creates more useable bandwidth; because each frame is smaller, there are fewer bits to transmit. However, compressing data uses processor and memory resources and introduces some latency while the compression is being done. Because of this, use compression only on T1 links or less. You can compress the whole payload or just compress the protocol headers with TCP or Real-time Protocol (RTP) header compression (cRTP). Cisco supports three payload compression algorithms:

- Stacker
- Predictor
- Microsoft Point-to-Point Compression (MPPC)

For voice, use Low Latency Queuing (LLQ) and cRTP compression, and for data, use Class-Based Weighted Fair Queuing (CBWFQ) and TCP compression. LLQ and CBWFQ are discussed later in this chapter.

Delay and Jitter

Network traffic experiences four types of delay:

- Processing Delay—The time it takes a packet to move from the input interface of a router or Layer 3 switch, to the output interface. Processing delay depends on switching mode, CPU speed and utilization, the router's architecture, and interface configuration. This is a variable delay.

CHAPTER 3

QoS OVERVIEW

- **Queuing Delay**—The length of time a packet waits in the interface queue before being sent to the transmit ring. Queuing delay depends on the number and size of packets in the queue, and the queuing methods in place. This is a variable delay.
- **Serialization Delay**—The length of time it takes to place the bits from the interface transmit ring onto the wire. Serialization delay depends on the bandwidth of the interface—higher bandwidth equals smaller serialization delay. This is a fixed delay.
- **Propagation Delay**—The length of time it takes the packet to move from one end of the link to the other. Propagation delay depends on the type of media, such as fiber or satellite links. This is a fixed delay.

The total delay is the sum of all four delays on every link along the path. Because processing and queuing delay times can vary, end-to-end delay can vary from packet to packet. This variation is called *jitter*.

To decrease delay, you can increase the link bandwidth, prioritize important packets (note that this increases the delay for non-priority traffic), or compress the packet headers or the payloads. On links under T1 speed, you can fragment large packets and interleave smaller, interactive, packets between them—this is called Link Fragmentation and Interleave (LFI).

When your traffic traverses an ISP network, you might need to reprioritize it to match the provider's standards.

Packet Loss Issues

Packet loss can cause jerky transmission of voice or video, slow application performance, or corrupt data. By default, when a software queue is full (congested), the switch or router drops all other traffic bound for that queue. This is called *tail drop*. It can cause some problems:

- TCP global synchronization.
- TCP buffer starvation.
- Delay and jitter.
- High-priority traffic is dropped, whereas low-priority traffic is sent.

Congestion avoidance attempts to prevent tail drop. To accomplish this, increase link bandwidth, use queuing to guarantee a certain amount of traffic to each application, or use Weighted Random Early Detection (WRED). WRED drops lower-priority traffic (based on Differentiated Services Code Point [DSCP] or IP Precedence values) as a queue starts to fill and drops high-priority traffic only when the queue is almost full. If the queue fills completely, however, tail drop is used. The drop thresholds and the drop ratios are configurable. WRED works best with TCP traffic, because TCP dynamically adjusts its sending rate when packets are dropped. Do not use WRED for voice traffic. The “Congestion Avoidance” section describes this more completely.

Four other causes of packet drop are: frame errors, lack of buffer space (called an *ignore*), a CPU that is unable to assign a free buffer to it (called an *overrun*), or a CPU that is too busy to process inbound packets so the inbound queue fills.

CHAPTER 3

QoS OVERVIEW

Defining QoS Requirements for Network Traffic

To implement QoS, you need to identify the types of network traffic, determine the requirements for each, divide the traffic into classes, and then set policies for those classes.

A network audit helps identify the types of traffic on the network.

The relative importance of each application is a business decision, accomplished by a business audit. Applications should be grouped into classes that have about the same QoS requirements. Some common classes include: Voice, Interactive, Mission-critical, Transactional, Best-effort, and Scavenger.

A QoS policy then can be created for each class of traffic. You need to decide such things as allocated bandwidth (minimum and/or maximum), prioritization, and congestion avoidance.

QoS Models

There are three QoS models:

- Best effort—Traffic is sent with no guarantees of bandwidth or priority.
- Integrated Services (IntServ)—The QoS parameters are signaled throughout the path and guaranteed for the length of the session.
- Differentiated Services (DiffServ)—QoS parameters are applied to traffic classes at each hop along the path.

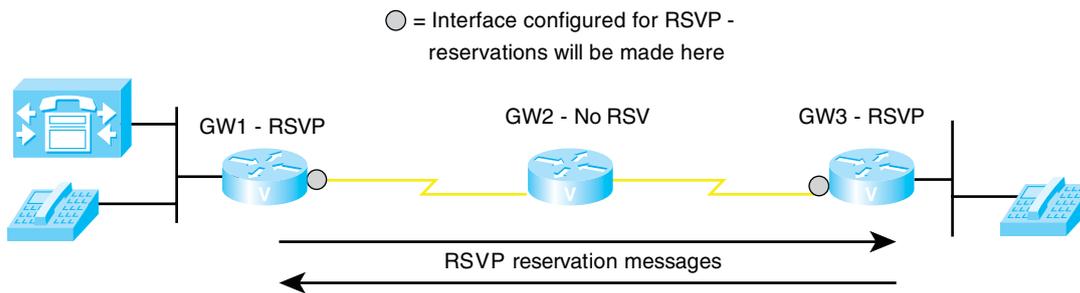
Best Effort

Best-effort delivery is the default method—traffic is sent out in the order it arrives with no differentiation between types of traffic and no guarantee of delivery. Benefits of best effort include its scalability (the Internet is based on best-effort delivery), and its ease of deployment. Drawbacks include the fact that all traffic is given the same service level.

IntServ

IntServ is a QoS model that guarantees a specific level of service to each flow of identified traffic, throughout the entire network, for the length of the session. This is done using Resource Reservation Protocol (RSVP). An RSVP-aware application, or a router or CallManager acting in proxy for a nonRSVP-aware device, requests a specific level of service from its next-hop router. A check is made along the path between the two endpoints, and each RSVP-enabled router along the way reserves bandwidth for that flow. If the network cannot provide the required bandwidth, the session is not allowed or its service level is downgraded.

RSVP works for any type of traffic, but it is usually used for real-time applications that are either rate-sensitive or delay-sensitive, such as voice and video. Figure 3-1 shows a call between two IP phones.

FIGURE 3-1 Using RSVP for Voice Calls

Two of the routers in the path—GW1 and GW3—are configured with RSVP; however, GW2 is not. When GW1 and GW3 receive the RSVP messages requesting a service level, they reserve that amount of bandwidth on their WAN interface. There must be some sort of QoS configured on the routers to implement the reservation. When GW2 receives the RSVP messages, it merely passes them on to the next hop router unchanged. Note that reservations are made in both directions because this is a voice call.

All routers in the path are not required to be configured with RSVP, but reservations are made only on those routers and those interfaces with it enabled. To ensure end-to-end service, configure RSVP on all router interfaces in the data path.

The path between endpoints is determined by the routing protocol, not by RSVP. If there is a network change, and the routing protocol changes the path, then RSVP reconverges also.

Current applications use DiffServ to enact IntServ QoS policies, such as guaranteed rate, and controlled load. One of the biggest benefits of IntServ is that it provides

per-flow admission control. This can help with VoIP calls. RSVP supports applications that use dynamic port numbers and static ones. Some drawbacks include its overhead—signaling is exchanged at the beginning of a flow, so there can be some delay. It must continue to cross the network for the length of the flow to adjust for changes in path due to network changes, thus causing extra overhead. Additionally, because you need to track each flow, it is not scalable in a large enterprise.

For more information on using RSVP with VoIP, see the Cisco Press book *Cisco Voice Gateways and Gatekeepers* by David Mallory, Ken Salhoff, and Denise Donohue.

DiffServ

DiffServ groups network traffic into *classes* comprised of traffic needing the same type of QoS treatment. For instance, voice traffic is separated from email traffic. However, e-mail might be placed in the same class as web traffic. The exact classes, traffic, and QoS policies used are a business decision.

These classes are distinguished from each other based on the value of certain bits in the IP or ISL header or the 802.1Q tag. Each hop along the way must be configured to treat the marked traffic the way you want—this is called per-hop behavior (PHB).

CHAPTER 3

QoS OVERVIEW

- In the Layer 3 IP header, you use the 8-bit Type of Service (ToS) field. You can set either IP Precedence, using the top 3 bits, or DSCP using the top 6 bits of the field. The bottom 2 bits are not used for setting priority. The default DSCP value is zero, which corresponds to best-effort delivery.
- At Layer 2, with ISL, you can set 3 of the 4 bits in the ISL priority field to reflect the class of service (CoS). With 802.1Q, you set the 3 802.1p bits to the CoS. The values of these 3 bits correspond to the IP Precedence values.

Benefits of DiffServ include the many classes of service possible, and its scalability. As a drawback, it can be complex to configure. It also does not absolutely guarantee a level of service.

QoS Implementation Methods

The legacy method of configuring QoS was at each interface, on each router, using the Command Line Interface (CLI). The current recommended method is to use the Modular QoS CLI (MQC), which allows you to create one configuration that can then be applied to many interfaces. Common QoS settings have been automated with AutoQoS. For those who prefer a GUI interface, there is the Cisco Router and Security Device Manager (SDM).

Legacy CLI

The traditional QoS configuration using legacy CLI involves accessing the router via Telnet or console port. Traffic classification and policy enforcement are combined in the configuration at each interface, which is time-consuming and can lead to errors.

The types of QoS possible are limited, also. For example, you can do simple priority queuing, custom queuing, and compression. Legacy CLI QoS might be used to tweak AutoQoS settings.

MQC

Modular QoS CLI (MQC) is a method of classifying traffic, marking the traffic, and setting policies for that traffic that can be used on most devices with most kinds of policies. It's most important contribution is the separation of traffic classification from policy implementation. Here are general steps for implementing MQC:

- Step 1.** Create the necessary access control lists, if classifying traffic by ACL, or configure network-based application recognition (NBAR). (Click [here](#) for an explanation of NBAR.)
- Step 2.** Create class maps that specify matching such items as ACLs, protocol, DSCP, or IP Precedence values.
- Step 3.** Create a policy map that links to each class map and defines the policy for each.
- Step 4.** Apply the policy map to the appropriate interfaces.

QoS OVERVIEW

When access control lists (ACL) are used to classify traffic, the way a router or switch reacts to specific access control entries (ACE) is different in a QoS context than with security-based ACLs. In a QoS access list:

- If the traffic matches a *permit* statement, the designated QoS action is taken.
- If the traffic matches a *deny* statement, the rest of the ACEs in that ACL are skipped and the switch goes to the next ACL.
- If there are multiple ACLs in a policy applied to an interface, the switch stops reading them as soon as a permit statement match is found for the traffic.
- If the traffic does not match any ACL entry, the switch just gives best-effort delivery to the traffic.

MQC Configuration

First, configure the ACLs if using them to identify traffic.

Second, configure a class map for each classification of traffic. Class map names are case-sensitive.

```
(config)#class-map [match-any | match-all] name
(config-cmap)#match {match options, such as ACL}
```

Third, configure a policy map that calls the class maps and sets policies or types of treatment for each class. Policy map names are also case sensitive.

```
(config)#policy-map name
(config-pmap)#class class-map-name
(config-pmap-c)#policy options, such as set DSCP or bandwidth
```

Finally, apply the MQC policy to the desired interface(s), either inbound or outbound:

```
(config-if)#service-policy {output | input} name
```

Verifying QoS Configuration

Use the following commands to verify your QoS configurations and actions:

- **show class-map [name]**—Displays the configured class maps or just the one named.
- **show policy-map [name]**—Displays the configured policy maps or just the one named.
- **show policy-map [interface [interface-number [input | output]] | [class class-name]**—Displays the policy maps and statistics by interface or class.
- **show queuing [interface interface-number]**—Shows the queuing strategy and statistics for any queues configured on the interface.
- **show policy interface interface-number**—Displays the policies for all classes applied to the interface, along with statistics.
- **debug ip rsvp**—If using RSVP for voice, shows information about packets received and sent.

CHAPTER 3

QoS OVERVIEW

AutoQoS

AutoQoS is a utility that automates and simplifies QoS configuration, giving a consistent configuration across the network. It discovers the applications traversing the router or switch and configures standard best practice QoS policies for them. It can be used with both LAN and WAN interfaces.

Automatic configurations can be tuned if necessary by using the MQC or with legacy CLI. AutoQoS was originally only for VoIP applications, but recent versions can be used with data applications also.

When configured on a WAN interface, AutoQoS:

- Detects and classifies VoIP and data traffic (typically using NBAR).
- Builds appropriate services policies, including placing Real-Time Protocol (RTP) traffic into a low-latency queue (LLQ) and guaranteeing bandwidth to VoIP control traffic.
- Sets up traffic shaping, fragmentation, or compression where needed.
- Enables SNMP traps and syslog alerting for VoIP events.

When configured on a LAN interface, AutoQoS:

- Sets up priority/expedited queuing on the switch interface.
- Configures the COS mapping to queues, and adjusts queue size and weights.
- Sets up trust boundaries on user access ports and links between switches. Trusts the incoming CoS only when an IP phone is present.

To use AutoQoS, CEF must be enabled, and the correct bandwidth configured on each interface, then AutoQoS is enabled as follows. This example enables AutoQoS for VoIP only. Notice that after the commands are given, the router has created a policy map (not shown) and applied it to the interface:

```
Router(config)#int s1/0/0:1
Router(config-if)#bandwidth 1544
Router(config-if)#auto qos voip
!
Router#show auto qos int s1/0/0:1

Serial1/0/0:1 -
!
interface Serial1/0/0:1
  service-policy output AutoQoS-Policy-UnTrust
```

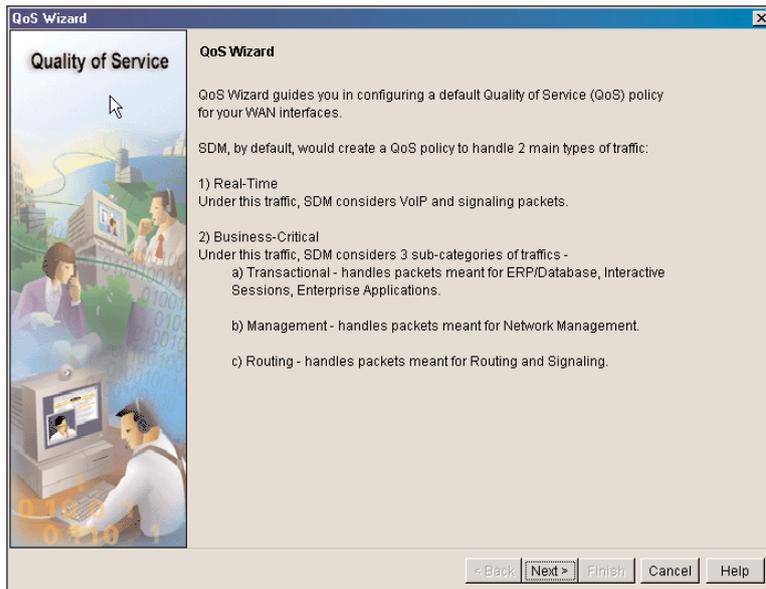
SDM QoS Wizard

SDM allows GUI configuration of router interfaces, firewall, ACL features, VPNs, routing, Network Address Translation (NAT), Intrusion Prevention, Network Access Control (NAC), and QoS. It helps nonexpert users to configure these router functions. SDM comes preinstalled on the ISR routers, but to use the SDM Wizard, the router's HTTP server function must be enabled.

With the SDM's QoS Wizard, you can configure, monitor, and troubleshoot QoS configurations. Browse to <http://10.10.10.1>—the default IP address for SDM. From the “Configure” menu, choose to configure QoS. This launches the QoS Wizard, shown in Figure 3-2.

QoS OVERVIEW

FIGURE 3-2 SDM QoS Wizard

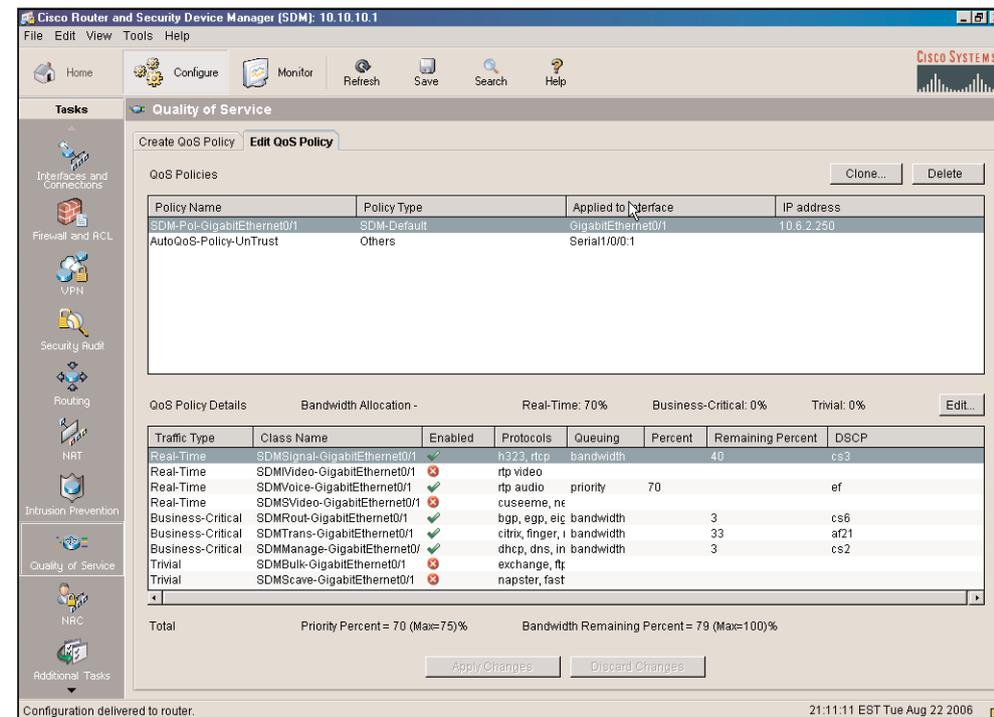


Notice that the wizard creates policies for two types of traffic:

- Real-Time—VoIP and signaling packets
- Business-Critical—This has three subcategories:
 - Transactional—Database, interactive sessions, and enterprise applications
 - Management—Network management applications
 - Routing—Routing protocols

You can specify bandwidth allocation for these classes or use the wizard's recommendations. After the wizard is done, it shows you the policies and loads them into the router's configuration. You can edit them later, as shown in Figure 3-3.

FIGURE 3-3 Editing the QoS Wizard Policies



If you select “Monitor” from the menu and then “QoS Status”, you can monitor the amounts and types of traffic through each interface configured for QoS. The interval, traffic direction, and type of statistics shown can be adjusted.

CHAPTER 3

QoS OVERVIEW

QoS Methods Comparison

Thus, Cisco provides four ways for you to configure QoS in your network. They each have their strengths and weaknesses.

- Legacy CLI—Hardest to use, little capability to fine-tune, takes the longest to implement, and is not modular.
- MQC—Easier to use and takes less time to implement on multiple interfaces than does legacy CLI. Has excellent capability to fine-tune configurations, and it is modular.
- AutoQoS—Easy to use, but it has limited inherent fine-tuning, takes the least time to implement, and has excellent modularity.
- SDM QoS Wizard—Simple to use, can do some limited fine-tuning, is fast to implement, and has good modularity.

CHAPTER 4

QoS Details

This chapter explores, in detail, ways of choosing and configuring quality of service. The way you classify and mark traffic, and the type of QoS policies you implement, will depend on the policy location and types of network traffic present.

Classification and Marking

Classification is the most basic Quality of Service (QoS) step—until traffic is identified, it cannot be provided a unique level of service. Traffic is often classified by application, source or destination IP address, or inbound interface.

After traffic is classified, an appropriate marking can be applied to it. The location where traffic is marked defines a *trust boundary*. If the device that marked the traffic is trusted, then that marking is passed through the network and honored by each device. If that device is untrusted, then some trusted network entity must re-mark the traffic.

Classification and marking should be done as close to the traffic source as possible because they can be resource intensive. Marking at the end device, such as an IP phone, is ideal. Otherwise, mark (or re-mark) traffic at the access switch or distribution switch if necessary.

Layer 2 markings include 802.1Q Class of Service (CoS) and Multiprotocol Label Switching (MPLS) experimental bits. Frame relay markings are different—they include setting the Backward Explicit

Congestion Notification (BECN) bit, the Forward Explicit Congestion Notification (FECN) bit, or the Discard Eligible (DE) bit in the frame relay header. Layer 3 markings include Differentiated Services Code Point (DSCP) and IP precedence. After traffic is classified and marked, other routers and switches in the network can be configured to provide QoS to it.

Using NBAR for Classifying Traffic

There are several ways to identify traffic so that it can be classified. Access lists are commonly used to identify application data, but Cisco has an IOS-based tool that provides more granularity and goes beyond static port numbers. Network-Based Application Recognition (NBAR) is an IOS protocol discovery and classification mechanism. It monitors the traffic going in and out of an interface, identifies it by protocol, port number, or payload contents (up to 400 bytes), and provides traffic statistics. NBAR recognizes common applications, even those that use dynamic ports. For instance, Real-Time Protocol (RTP) carries voice and video traffic and uses dynamic port numbers within a large range. An access list can match traffic within that range of port numbers, but NBAR can match on the following RTP characteristics:

- Audio traffic (using payload types 0–23)
- Video traffic (using payload types 24–33)
- Payload type for a specific payload type value

Note

NBAR does not identify RTP control traffic, just RTP bearer traffic.

You can additionally configure NBAR to recognize custom applications. Cisco provides downloadable Packet Description Language Modules (PDLM) that also add additional applications.

CEF must be enabled on each interface where NBAR is used. To enable NBAR at an interface, and then view the traffic that it discovers, use the commands:

```
Router(config-if)#ip nbar protocol-discovery
Router#show ip nbar protocol-discovery
```

You can download new PDLMs from the Cisco web site: <http://www.cisco.com/cgi-bin/tablebuild.pl/pdlm>. You must be a registered user. After the file is downloaded, you should either save it in the router's flash or place it on a TFTP server reachable by the router. Instruct the router to load the PDLM with the following command:

```
Router(config)#ip nbar pdlm pdlm_name
```

The name is in URL format, and points the router either to the file in flash or to the TFTP server. For example, you might use **ip nbar pdlm flash://bittorent.pdlm** to load the PDLM for Bit Torrent from flash memory.

Sometimes users map protocols to different ports than NBAR expects. To tell NBAR to look for a protocol on additional ports and to then verify your configuration, use the commands:

```
Router(config)#ip nbar port-map protocol [tcp | udp] port
Router#show ip nbar port-map
```

To use NBAR for classifying traffic with the MQC, follow these steps:

- Step 1.** Enable NBAR on all appropriate interfaces.
- Step 2.** Create a class map that matches against one or more of the NBAR protocols, using the **match protocol** option. Repeat this step for each class desired.
- Step 3.** Create a policy that links to those class maps, and assigns desired service to it.
- Step 4.** Apply the policy to an interface.

Example 4-1 shows NBAR enabled on a GigEthernet interface, and class maps created to match three types of traffic discovered by NBAR: RTP, any web traffic that has the word “ccnp” in its URL, and eDonkey. A policy map is created that marks this traffic, and it is applied inbound to the LAN interface.

EXAMPLE 4-1 Using NBAR with the MQC

```
Router(config)#int gi 0/0
Router(config-if)#ip nbar protocol-discovery
!
Router(config)#class-map VOIP
Router(config-cmap)#match protocol rtp audio
Router(config-cmap)#!
Router(config-cmap)#class-map Exams
Router(config-cmap)#match protocol http url ccnp*
Router(config-cmap)#!
Router(config-cmap)#class-map eDonkey
```

QoS DETAILS

```

Router(config-cmap)#match protocol edonkey
!
Router(config)#policy-map NBAR
Router(config-pmap)#class VOIP
Router(config-pmap-c)#set ip dscp ef
Router(config-pmap-c)#class Exams
Router(config-pmap-c)#set ip dscp 31
Router(config-pmap-c)#class eDonkey
Router(config-pmap-c)#set ip dscp 13
!
Router(config-pmap-c)#int gi 0/0
Router(config-if)#service-policy input NBAR

```

This classifies and marks the traffic and uses NBAR to identify it. Classification and marking needs to happen only once—all other devices in the network can just look for the DSCP markings and set policies based on those. Thus, the next part must be to configure some way to treat this classified and marked traffic. An example of this configuration is the section on LLQ and CBWFQ. Click here to review that configuration: [Configure](#)

For more detailed information on NBAR, including a list of applications it currently is able to recognize, see this link: http://www.cisco.com/en/US/products/ps6616/products_qanda_item09186a00800a3ded.shtml

Marking at Layer 2

CoS uses the three 802.1p bits in the 802.1Q trunking tag to mark traffic. These three bits have eight possible values, ranging between zero and seven. IP Precedence uses three bits in the IP header, so it has the same range of values as does CoS. Table 4-1 lists the values and their standard meanings.

TABLE 4-1 IP Precedence and CoS Values

IP Precedence/CoS	Name
7	Network
6	Internet
5	Critical
4	Flash-override
3	Flash
2	Immediate
1	Priority
0	Routine

When frames enter a switch, the Layer 2 header is stripped off. The switch maps the CoS value to an internal DSCP value as the packet moves through it. This DSCP value is then translated back to a CoS value if the packet is sent over another trunk link. There are default values for the mappings between and CoS and DSCP, but they can also be configured.

QoS DETAILS

MPLS labels have a three-bit field, called the MPLS experimental (MPLS EXP) field, which has the same eight possible values as CoS and IP Precedence. By default, any IP Precedence value is copied into this field and becomes the Layer 2 marking for MPLS traffic. Service providers alternatively can set these bits independently, thus marking the traffic within their network without changing their customer's Layer 3 marking. The value of MPLS EXP bits is preserved through the MPLS network.

Table 4-2 lists the eight Layer 2 markings, and some suggested applications for them.

TABLE 4-2 Layer 2 Markings and Applications

CoS Value	Application
7	Reserved
6	Reserved
5	Voice Bearer
4	Video
3	Call Signaling
2	High-Priority Data
1	Medium-Priority Data
0	Best-Effort Data

Marking at Layer 3

The concept behind DiffServ (DS) is to group traffic into classes and mark it once at the edge of the network. DiffServ was created to be highly scalable by separating classification from policy creation and by servicing aggregate classes of traffic rather than individual flows.

DiffServ uses Layer 3 markings, setting the eight-bit ToS field in the IP header. Unlike the Layer 2 header, this marking remains with the packet as it traverses the network, and changes only if some device overwrites the value of these bits. You can set either IP Precedence, using the top three bits, or Differentiated Services Code Points (DSCP), using the top six bits of the field. The bottom two bits can be used for congestion notification. The default DSCP value is zero, which corresponds to best-effort delivery. When properly configured, DSCP is backward compatible with IP Precedence.

Each hop in the network is provisioned to treat traffic differently based on its markings; this is called “per-hop behavior” (PHB). RFC 2475 defines PHB as “the externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate.” A *behavior aggregate* is a logical grouping of traffic that needs similar service levels. It is also referred to as a *service class*. Four PHBs are defined:

- Default
- Assured forwarding
- Class selector
- Expedited forwarding

Default PHB

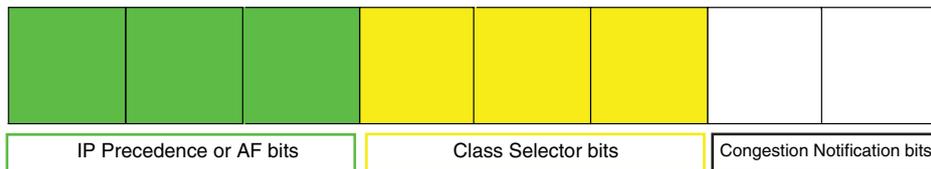
All the bits in the TOS byte are set to “0,” which gives best-effort delivery. Any unmarked traffic is mapped to this PHB.

Assured Forwarding and Class Selector PHB

Figure 4-1 shows the TOS byte in the IP header.

FIGURE 4-1 The TOS Byte in the IP Header

TOS Byte



The six DSCP bits can be broken down into two sections: The highest three bits define the DiffServ Assured Forwarding (AF) class (the area in green), and the next three bits are called the “Class Selector (CS)” (the area in yellow). When the three CS bits are all zero, you have a value that is equal to IP Precedence. The lowest two bits (the area in white) are not used in DiffServ marking—they allow the sending of congestion notification information.

Each AF class becomes its own queue at the interface. AF uses the first two CS bits to define the drop probability within that queue. The last bit is always zero and is not used in calculating drop probability values.

AF classes 1–4 are defined and within each class, 1 is low drop probability, 2 is medium, and 3 is high (meaning that traffic is more likely to get dropped if there is congestion).

AF guarantees a specified amount of bandwidth to a class. By default, it allows the traffic to burst above that amount if there is extra bandwidth available, although this can be policed.

Table 4-3 lists the classes and their associated AF values.

TABLE 4-3 Assured Forwarding Values

	Low Drop	Medium Drop	High Drop
Class 1	AF11	AF12	AF13
Class 2	AF21	AF22	AF23
Class 3	AF31	AF32	AF33
Class 4	AF41	AF42	AF43

DiffServ Expedited Forwarding PHB

Another predefined DiffServ classification is Expedited Forwarding (EF), which is DSCP 46. This is equivalent to IP precedence 5. EF traffic becomes a separate queue at the QoS-enabled router interface. You must configure each hop in the network for the type of service you want EF traffic to receive. EF is usually used to put traffic in a low-latency queue, which results in low delay, guarantees a specified amount of bandwidth, and also polices the traffic to prevent it from exceeding that bandwidth.

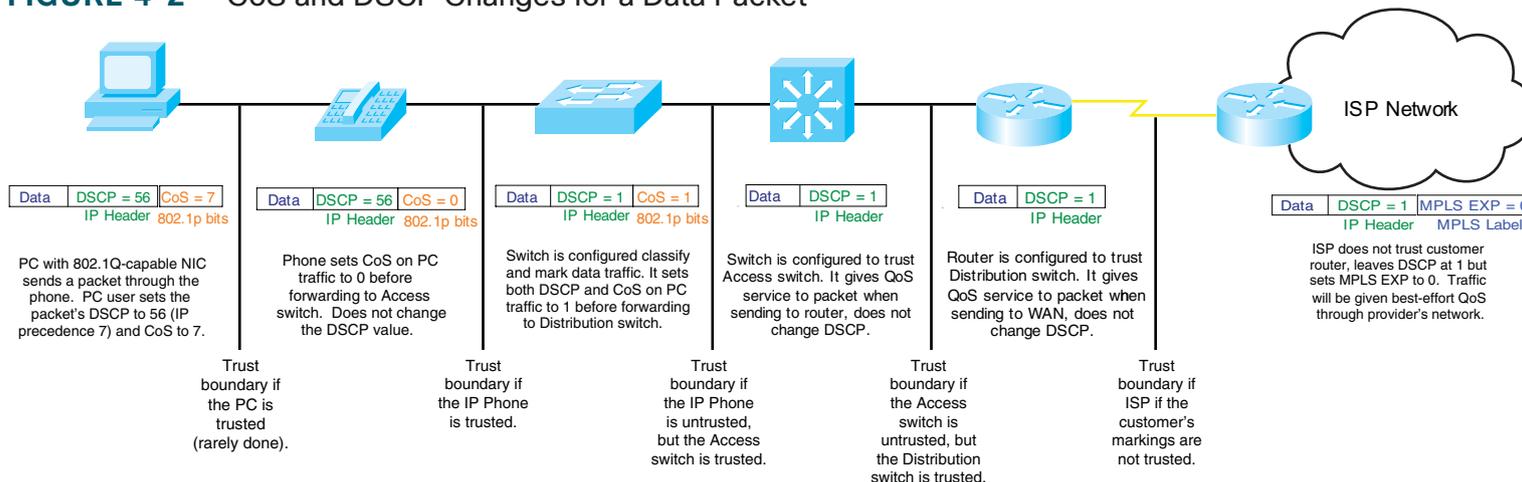
Classifying and Marking in a VoIP Network

Figure 4-2 shows what happens to the CoS and DSCP settings of a data packet as it moves through a QoS-enabled LAN.

- In the figure, users with an 802.1Q-enabled Network Interface Card (NIC) on their PC attempts to give their data higher priority within the network. They send a frame with an 802.1Q tag in which they have set the 802.1p bits to CoS of 7. They have also set the DSCP on the packet to 56. This animation shows just the relevant parts of the headers used.
- The IP phone by default creates an 802.1Q trunk between itself and the Access switch. It sets the 802.1p CoS on data traffic to zero, but it does not change any Layer 3 markings.

- The Access switch gets the frame from the phone and strips the Layer 2 header. By default it translates into an internal DSCP of zero as it moves through the switch fabric; however, this switch is configured to classify and mark data traffic. This particular application falls into a class that gets a Layer 3 marking of AF11, or DSCP 10 (binary value 001010). The switch remarks the DSCP value, and then sets the CoS to 1 in the 802.1Q tag when it sends the packet to the Distribution switch.
- The Distribution switch is configured to trust the Access switch's markings. It strips off the Layer 2 header, looks at the DSCP value, and provides the type of QoS service it is configured to provide to AF11 traffic. The switch's interface to the router is a Layer 3 interface, so no trunk tag is used. Instead, it puts on a normal Ethernet frame header and forwards the packet to the router.

FIGURE 4-2 CoS and DSCP Changes for a Data Packet



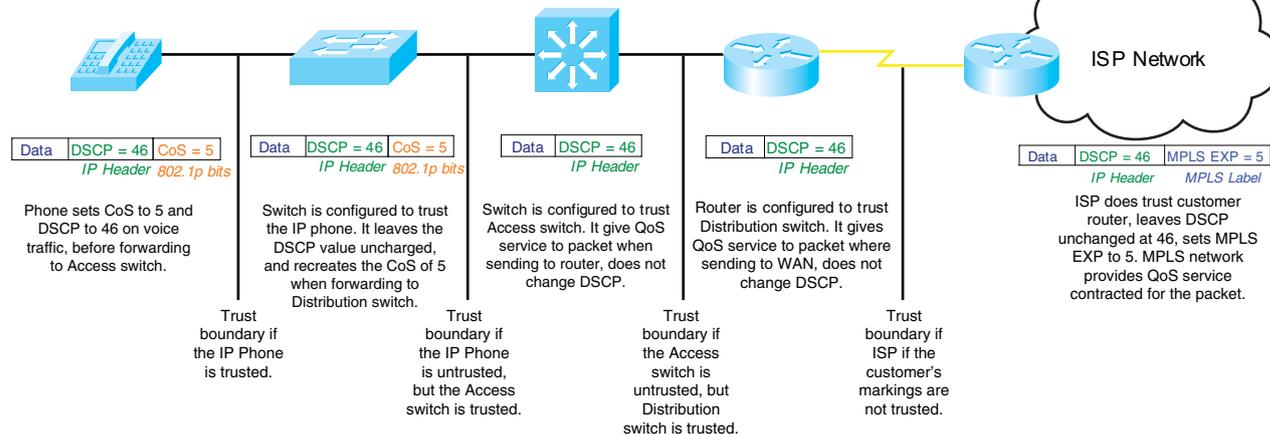
QoS DETAILS

- The router is configured to trust the packet's markings. It strips off the Layer 2 header, looks at the DSCP value, and provides the type of QoS service it is configured to provide to AF11 traffic. This might include allocating a certain amount of bandwidth and using Weighted Random Early Detection (WRED) in the queue. The router then forwards the packet to its ISP edge router.
- The ISP is not configured to trust the customer's markings. It could overwrite all DSCP values with zero, but in this case it just sets the MPLS Experimental Bits in the MPLS label to zero. The DSCP stays unchanged. The packet receives only best-effort service as it moves through the ISP's network, but devices in the destination network can use the unchanged DSCP values to provide QoS service to the packet.

Figure 4-3 shows what happens to the CoS and DSCP settings of a voice packet as it moves through a QoS-enabled LAN. In this example, the ISP trusts the customer's markings.

- The Cisco IP phone creates an 802.1Q trunk between itself and the Access switch. It sets the 802.1p CoS on its own voice media traffic to 5, and it sets the Layer 3 marking to EF or DSCP 46 (binary value 101110). If this were voice signaling traffic, the phone would set a CoS of 3 and a Layer 3 marking of CS3, or DSCP 24 (binary value 011000).

FIGURE 4-3 CoS and DSCP Changes for a Voice Packet



CHAPTER 4

QoS DETAILS

- The Access switch gets the frame from the phone and strips the Layer 2 header. It is configured to trust the IP phone, so it translates the CoS value into an internal DSCP as the packet moves through the switch fabric. The switch then translates that internal DSCP value back to a CoS of 5 in the 802.1Q tag when it sends the packet to the Distribution switch. The Access switch applies any outbound policies configured for this traffic, such as putting it into a priority interface queue, as it sends it to the outbound interface.
- The Distribution switch is configured to trust the Access switch's markings. It strips off the Layer 2 header, looks at the DSCP value, and provides the type of QoS service it is configured to provide to EF traffic. This typically includes placing the packet in a priority queue at the interface. The switch's interface to the router is a Layer 3 interface, so no trunk tag is used. Instead it puts on a normal Ethernet frame header and forwards the packet to the router.
- The router is configured to trust the packet's markings. It strips off the Layer 2 header, looks at the DSCP value, and provides the type of QoS service it is configured to provide to EF traffic. This typically includes placing the packet in an LLQ and allocating a certain amount of bandwidth to that queue. The router then forwards the packet to its ISP edge router.
- The ISP is configured to trust the customer's markings. It translates the IP precedence value of 5 into an MPLS Experimental Bits value of 5. The DSCP stays unchanged. The packet receives prioritized, real-time service as it moves through the ISP's network, and devices in the destination network can use the unchanged DSCP values to also provide QoS service to the packet.

Queuing Overview

Queuing configuration usually acts on outbound traffic. Each interface has a *hardware queue*, or transmit ring (TxQ), that holds traffic ready to be serialized onto the interface media. This queue is always First In/First Out (FIFO). When outbound traffic arrives at an interface, the interface scheduler sends it to the transmit ring to be placed on the media. If the transmit ring is full, other traffic must wait in some buffer memory space assigned to that interface called a *software queue*. When traffic must be placed into queues, the interface is said to be *congested*.

Causes of congestion include:

- Speed mismatches —Typically LAN traffic needing to go across a much slower WAN link (persistent congestion). It can also be GigEthernet traffic bound out a FastEthernet interface (transient congestion).
- Link aggregation—For WAN links, this occurs when multiple remote sites access a hub site across an oversubscribed link. For LAN links, this typically occurs on a distribution switch with multiple access switches feeding in to it on the uplink port(s) to the core switch.

Hardware Queue

Logical interfaces, such as subinterfaces and tunnel interfaces, use the hardware queue of the physical interface; they do not have their own transmit queues.

CHAPTER 4

QoS DETAILS

The number of packets a TxQ can hold depends on the speed of the interface. It is automatically determined, and that length is typically fine but can be tuned if desired. Most devices use the **tx ring-limit** command for tuning TxQ size.

Lowering the Tx ring size lessens the length of time a packet waits in the FIFO queue, and it increases the chance of a packet hitting the software queue. However, too short of a TxQ results in router resource use, as the CPU must be interrupted each time the Tx ring requests a packet from a software queue. A decrease in ring size means an increase in interrupts.

Increasing the TxQ size is usually not recommended when using QoS, because it decreases the use of the software queue.

Use the **show controllers interface** command to find information about the transmit ring size.

Software Queue

The software queuing strategy is configurable, and the next sections deal with various techniques to do this. This is where you can influence the order in which packets are scheduled into the TxQ and the number of packets sent. When using queuing mechanisms, several different logical queues are created, and traffic is placed into the appropriate software queue when it arrives at the interface. Each software queue has size limits, and packets above that limit are dropped.

Remember that fast-switched or CEF-switched traffic enters the software queue only if the hardware queue is congested. Process-switched traffic always goes into the software queue.

Legacy Queuing Techniques

Queuing mechanisms allow you to control the way congested traffic is buffered and sent out the interface by placing each type of traffic in its own queue. The router or switch then services each queue, scheduling transmittal of its traffic, according to a configured policy

FIFO Queuing

By default, most interfaces use FIFO queuing—there is just one software queue, and traffic is buffered and then scheduled onto the interface in the order it is received.

Note

Serial interfaces of E1 speed and below use weighted fair queuing by default, rather than FIFO queuing.

Priority Queuing

With Priority queuing, queues are assigned different priority values and placed in one of four queues. The high-priority queue is a strict priority

CHAPTER 4

QoS DETAILS

queue, which means that it gets serviced before anything else until it is empty. After that, each queue is serviced in turn, as long as the priority queue remains empty. The lower-priority queues may never be serviced if there is sufficient traffic in higher-priority queues (a condition called “starvation”).

Round Robin Queuing

Round Robin queuing takes one packet from each queue and then starts over. Each queue is serviced, none starve, but there is no way to prioritize any of the traffic or apply any sort of differential treatment to it.

During times of interface congestion, Weighted Round Robin (WRR) queuing weights queues, and more packets are sent from higher weighted queues, thus giving them more bandwidth. However, the bandwidth allocations are done in a way that might lead to more bytes being sent from some queues than desired, which causes other packets to be delayed.

Weighted Fair Queuing

Weighted Fair Queuing (WFQ) attempts to address some of the failing of FIFO and Priority queuing by allowing all traffic some access to the interface. Some characteristics of WFQ include:

- Queues traffic by flow or conversation.
- Flows are identified by header information, such as source and destination IP address, protocol, source and destination ports, and

type of service field value. These are used by a hash algorithm to create a queue index number.

- Each interface has a limited number of WFQ queues. Default for most interfaces is 256; it can be configured from 16–4096.
- If the number of flows exceeds the number of queues, multiple flows are placed in the same queue, resulting in less bandwidth per flow.
- Provides queues for system traffic and RSVP traffic separate from the WFQ queues.
- Traffic is weighted by flow, based on IP precedence.
- WFQ schedules small interactive flows before high-bandwidth flows.
- Allows lower-weighted flows relatively more bandwidth than higher-weighted conversations.
- Drops packets from high-volume flows more aggressively than those of low-volume flows.
- The *hold-queue limit* determines how many packets can be held in the WFQ system before all new packets are dropped (tail drop). The default is 1000.
- The *congestive discard threshold (CDT)* determines how many packets can be held by the WFQ before it begins dropping packets from high-volume flows. Packets are dropped from the queues with the most packets first. Packets are not dropped from low-volume conversations. The default CDT is 64.

QoS DETAILS

Advantages of WFQ include simplicity of configuration, support on most platforms and IOS versions, allowing some bandwidth to all traffic, and dropping more packets from higher-bandwidth flows.

Disadvantages of WFQ include lack of support for tunneling or encryption, lack of manual control over traffic classification, lack of minimum or fixed bandwidth guarantees, and chance of multiple flows placed in the same queue.

Configuring WFQ

WFQ is enabled by default on physical interfaces with a bandwidth less than 2.048 Mbps. If it has been disabled, or to enable it on another interface, use the interface command **fair-queue** [*congestive-discard-threshold* [*dynamic-queues* [*reservable-queues*]]]. For example, the following commands enable WFQ on the serial interface, set a CDT of 100, increase the number of flow queues to 512, and reserve 10 queues for RSVP to use.

```
Router(config)#int s1/0/0:0
Router(config-if)#fair-queue 100 512 10
```

To change the size of the hold queue, use the **hold-queue** *number* {**in** | **out**} interface command, as shown:

```
Router(config-if)#hold-queue 2000 out
```

To monitor the interface queues, use either **show interface** *interface* or **show queue** *interface*:

```
Router#show interface s1/0/0:0
[output omitted]
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output
  drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/2000/100/0 (size/max total/threshold/drops)
    Conversations 0/0/512 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 2250 kilobits/sec
[output omitted]
```

```
Router#show queue s1/0/0:0
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output
  drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/100/0 (size/max total/threshold/drops)
    Conversations 0/0/512 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 2250 kilobits/sec
```

CBWFQ and LLQ

CBWFQ

Class-Based Weighted Fair Queuing (CBWFQ) addresses some of the problems with WFQ. It allows manual configuration of traffic classification and minimum bandwidth guarantees. It uses the MQC to configure classes and allocate bandwidth. Therefore, you can group traffic

CHAPTER 4

QoS DETAILS

into classes based on any of the criteria available through the MQC. Each class becomes its own FIFO queue at the interface. This is still considered Weighted Fair Queuing because each queue is assigned a weight based on the class bandwidth guarantee, and the scheduler takes packets from each queue based on those weights.

There are three ways to designate bandwidth within an MQC policy map. All the CBWFQ classes within a single policy map must use the same method.

- The **bandwidth** *bandwidth-in-kbps* command
- The **bandwidth percent** command
- The **bandwidth remaining percent** command

Each of these methods designates a minimum bandwidth to allocate to the class. Traffic is allowed to burst above that amount if extra bandwidth is available. You can allocate up to 75 percent of the interface bandwidth by default; the rest is reserved for system traffic, such as routing updates. This can be changed using the **max-reserved-bandwidth** interface command.

You can optionally specify the maximum number of packets allowed in each class queue with the policy map command **queue-limit** *number-of-packets*. The default is 64.

Each policy map has a default class that is created automatically. All traffic not identified by one of the other classes is placed in this queue. You can allocate bandwidth to this class or enable WFQ for its traffic,

but not both. To enable WFQ, use the **fair-queue** command. If WFQ is enabled, you can also configure the number of dynamic queues with the **fair-queue** *number-of-queues* command.

Benefits of CBWFQ include:

- Control over traffic classification
- Minimum bandwidth guarantees
- Granular control and scalability

Drawbacks of CBWFQ include:

- Voice traffic can be delayed

[This example](#) shows CBWFQ configured in a policy map along with LLQ.

LLQ

LLQ addresses the needs of real-time traffic for low delay and guaranteed bandwidth. It creates one priority queue (in addition to the CBWFQs) with bandwidth that is both guaranteed and policed. This is a strict priority queue—traffic is sent from it before any other queues. However, when the interface is congested, the priority queue is not allowed to exceed its configured bandwidth to avoid starving the other queues. Voice traffic is typically enqueued into the LLQ. You can place more than one class of traffic in the LLQ. If so, the router still creates just one priority queue but allocates bandwidth to each class, and meters the traffic so that it does not exceed the bandwidth assigned to that class.

QoS DETAILS

Configure LLQ under the class statement in the policy map:

```
(config-pmap-c)#priority {bandwidth [burst] ; percent percentage
 [burst]}
```

Bandwidth is configured in kilobits per second, and burst is configured in bytes. These bandwidth amounts include the layer 2 headers.

Example 4-2 shows RTP voice traffic put into an LLQ, then guaranteed and policed to 256K of bandwidth. Traffic bound to URLs that include the string “ccnp” are placed in another queue, guaranteed 128K of bandwidth, and congestion avoidance is applied via WRED. Traffic bound for eDonkey applications is dropped. All other traffic falls into the default class, is placed in its own queue, and WRED is applied. The policy is applied outbound on the serial interface.

This example is based on an earlier classification and marking example. To refer back to it, click here: [Classify](#)

EXAMPLE 4-2 Configuring LLQ and CBQFQ

```
Router(config)#class-map VOIP-Out
Router(config-cmap)#match ip dscp ef
Router(config-cmap)#!
Router(config-cmap)#class-map Exams-Out
Router(config-cmap)#match ip dscp 31
Router(config-cmap)#!
Router(config-cmap)#class-map eDonkey-Out
Router(config-cmap)#match ip dscp 13
Router(config-cmap)#!
Router(config-cmap)#policy-map QOS
Router(config-pmap)#class VOIP-Out
Router(config-pmap-c)#priority 256
```

```
Router(config-pmap-c)#class Exams
Router(config-pmap-c)#bandwidth 128
Router(config-pmap-c)#random-detect dscp-based
Router(config-pmap-c)#class eDonkey
Router(config-pmap-c)#drop
Router(config-pmap-c)#class class-default
Router(config-pmap-c)#fair-queue
Router(config-pmap-c)#random-detect dscp-based
!
Router(config)#int s1/0/0:1
Router(config-if)#service-policy output QOS
```

Use the **show policy-map interface *interface*** command to see the service policy configuration and also the effect it has had as shown in Example 4-3.

EXAMPLE 4-3 Verifying a Policy Map

```
Router#show policy-map output interface s1/0/0:0
```

```
Serial1/0/0:0
```

```
Service-policy output: TestPolicy
```

```
Class-map: VoIP (match-all)
 0 packets, 0 bytes
 5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group 100
Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 264
  Bandwidth 128 (kbps) Burst 3200 (Bytes)
  (pkts matched/bytes matched) 0/0
  (total drops/bytes drops) 0/0
```

CHAPTER 4

QoS DETAILS

```

Class-map: class-default (match-any)
  19 packets, 1990 bytes
  30 seconds offered rate 0 bps, drop rate 0 bps
  Match: any

```

Congestion Avoidance

WRED solves two major problems with TCP and tail drop:

- **TCP Synchronization** occurs when all TCP packets exiting an interface are repeatedly dropped. At each tail drop, each session goes into slow start, and then ramps up its sending rate. When the interface queue fills, all packets are dropped again, and all sessions reduce their sending again. Eventually this results in waves of increased and decreased transmission, causing underutilization of the interface.
- **TCP Starvation** results when large flows, with increased window sizes, fill the interface queue. Packets from smaller or less aggressive flows are then dropped. This can cause jitter for those smaller flows due to a lack of a differentiated dropping strategy.

Random Early Detection (RED) attempts to avoid tail drops by preventing the interface from becoming totally congested. Once the queue fills above a threshold level, it drops random packets from the interface queue, dropping a higher percentage of packets as the queue fills. TCP sessions experience a packet loss at different times, and so they go into slow start at different times, thus preventing TCP synchronization.

You need to understand three RED concepts:

- **Minimum threshold**—The queue depth at which RED begins dropping packets.
- **Maximum threshold**—The queue depth at which the maximum number of packets are being dropped.
- **Mark probability denominator**—Controls how many packets are dropped when the maximum threshold is reached. The probability of a packet being dropped equals $1/\text{max-prob-denom}$.

RED has three packet-dropping modes:

- **No drop**—When the queue depth is between zero and the minimum threshold.
- **Random drop**—When the queue depth is between the minimum and the maximum thresholds.
- **Tail drop**—When the queue depth is above the maximum threshold.

Basic RED does not distinguish between flows or types of traffic. Weighted RED, on the other hand, drops traffic differently depending on its IP precedence or DSCP value. WRED is combined with CBWFQ to implement DiffServ's Assured Forwarding PHB. Each PHB has a unique WRED profile, identifying a minimum threshold, maximum threshold, and mark probability denominator for that profile. There are default profiles for each PHB, but they can also be manually configured.

CHAPTER 4

QoS DETAILS

Configure WRED under a physical interface, a VC, or a class within a policy map with the command **random-detect [dscp-based]**. Packet drop decisions are based on IP precedence value unless the **dscp-based** option is configured. To change the default profile for a specific DSCP value, use the **random-detect dscp dscp-value min-threshold max-threshold mark-probability-denominator** command.

Traffic Policing and Shaping

Traffic policing and traffic shaping are both ways to control the amount of traffic through an interface. Policing drops traffic, whereas shaping buffers it for sending later. Policing can be used inbound or outbound, but shaping is used only on outbound traffic.

Both mechanisms use a token bucket concept to measure the amount of bandwidth allowed. Enough tokens enter the bucket at regular intervals to allow the interface to send the number of bytes configured for that interval. If a packet enqueued has fewer bytes than are represented by tokens in the bucket, it is considered a *conforming* packet. The packet is sent and an equivalent number of tokens are subtracted from the bucket. If it has more bytes than there are tokens to send, it is considered an *exceeding* packet. The router then takes action based on whether policing or shaping is configured. Some implementations use two token buckets. If a packet exceeds the first bucket, it is checked against the second one. If there are not enough tokens in the second bucket to send the packet, it is considered a *violation*.

If all the tokens are not used within the interval, they can accrue and remain available to future packets if bursting is enabled.

Traffic Policing

By using the QoS policing function, bandwidth use can be controlled on physical interfaces. Policing specifies an amount of bandwidth allowed for a particular type of traffic, and generally drops traffic over that amount. It can also be configured to allow the excess traffic, but it marks it with a different (usually lower) DSCP value.

Some uses for policing include:

- Limiting the bandwidth of a high-speed interface to a lower rate
- Limiting the rate of an application or a traffic class
- Remarking the DSCP of traffic exceeding a specified rate

Class-based policing is configured in a policy map. Its possible conditions are: Conform, Exceed, and Violate. Its possible actions are: Drop, Set (DSCP, for example), and Transmit.

Traffic Shaping

Traffic shaping also controls the amount of traffic sent out an interface, but shaping buffers excess traffic instead of dropping it. Because data is usually bursty, the buffered traffic can be sent out between bursts. Thus, shaping smoothes out the flow of traffic. This also results in fewer packet drops, and thus fewer TCP retransmits. It does, however, introduce some delay for traffic that must be buffered. It does not support remarking traffic.

CHAPTER 4

QoS DETAILS

Some uses for shaping include:

- Making the outgoing traffic rate match the contracted committed information rate (CIR).
- To avoid overrunning remote links in networks, such as ATM, Frame Relay, and Metro Ethernet, that might have different bandwidths on hub and spoke devices.
- Interacting with Frame Relay congestion notifications, causing the router to throttle-back its sending rate.

Class-based traffic shaping is configured under the policy map. It works with any type of interface, not just Frame Relay interfaces.

Link Efficiency Mechanisms

Although QoS mechanisms cannot actually create bandwidth, they can help your network use the available bandwidth more efficiently. Two ways of doing this are compression and fragmentation. These mechanisms are typically applied at the WAN edge, where links are slower than within the LAN.

Compression

Compressing the traffic on a line creates more useable bandwidth; because each frame is smaller, there are fewer bits to transmit. Thus, the serialization delay is reduced, and more frames can be sent. Cisco IOS supports two types of compression: payload and header.

Payload compression can be done at either Layer 2 or Layer 3.

- **Layer 2 Payload Compression**—The Layer 2 payload compression compresses the Layer 3 and 4 headers and the packet data. Layer 2 payload compression is typically a hop-by-hop mechanism, because the Layer 2 header is removed at each hop. Layer 2 compression done in software is CPU-intensive and might actually add extra delay to the traffic flow. Hardware compression, however, adds little delay. Cisco supports three Layer 2 payload compression algorithms:
 - Stacker
 - Predictor
 - Microsoft Point-to-Point Compression (MPPC)
- **Layer 3 Payload Compression**—Layer 3 payload compression compresses the Layer 4 header and the packet data. It is generally done session-by-session.

Header compression leaves the payload intact but compresses the headers. TCP header compression compresses the IP and TCP headers. RTP header compression compresses the IP, UDP, and RTP headers. It is most effective when the headers are much larger than the payload, such as with Telnet or VoIP. Headers do not change much over the life of a flow and contain many redundant fields (such as source and destination IP address, protocol, and port). Compression removes the redundant information and sends only the new information and an index pointing to the unchanged information.

QoS DETAILS

Compression configured on a physical interface applies to all flows. For more granular control over which traffic is compressed, configure it in the MQC policy map under the desired classes. Header compression uses fairly few CPU resources.

Link Fragmentation and Interleave (LFI)

A typical network has a range of packet sizes. Small packets can be delayed waiting for a large packet to be sent out the interface. This can happen even if LLQ is configured—a small voice packet might be sent immediately to the hardware queue. However, the hardware queue is FIFO. If a large packet arrived there just before the voice packet, it is serialized out the interface first. The voice packet has to wait. This causes delay and jitter.

LFI breaks large packets into smaller segments and intersperses the smaller packets between the pieces of the big ones. Thus, delay and jitter are reduced for the small packets.

The target serialization delay for voice is 10–15 ms. At 2 Mbps link speed, a 1500 byte packet can be serialized in 10 ms. Thus, there is typically no need for LFI on links over E1 speed.

QoS with VPNs

A Virtual Private Network (VPN) is a way of creating a virtual point-to-point link over a shared network (often over the Internet). It can be used either for user remote access or for intrasite links. Two types of remote access VPNs are:

- **Client-initiated**—The user has a VPN client application, such as Cisco’s VPN Client, on their computer. After they are connected to the Internet, they use the application to connect them to their network.
- **Network Access Server (NAS) initiated**—Users connect into an access server at their ISP. The NAS then sets up a VPN to the private network.

Two types of intrasite VPNs are:

- **Intranet VPN**—Links sites within the same company to each other.
- **Extranet VPN**—Links an external group (such as a customer or supplier) to the company’s private network.

VPNs have several advantages, including:

- The ability to encrypt traffic across the public network and keep it confidential.
- The ability to verify that the data was not changed between the source and destination.
- The ability to authenticate the packet sender.

Router-to-router VPN tunnels use a logical tunnel interface that is created on the router. This interface is where you put configuration pertaining to the tunnel itself. Tunnel traffic uses one of the router’s physical interfaces, determined by the routing table. Configuration on this interface applies to all traffic, even if several tunnels use that interface.

VPNs create an extra challenge for QoS. A VPN tunnels traffic from one device to another by adding an IP header on top of the original one. Thus, the original header, with its QoS markings, is hidden from routers in the packet's path. If the packet needs any special QoS treatment, the markings must be copied from the original IP header into the tunnel IP header.

GRE Tunnels

Generic Routing Encapsulation (GRE) tunnels add a GRE header and a tunnel IP header to the packet. By default, TOS markings on the original packet are copied into the tunnel IP header. When the packet arrives at the physical interface, classification and queuing are based on the markings in the tunnel IP header.

IPSec Tunnels

IP Security (IPSec) can operate in either tunnel mode or transport mode. In tunnel mode, it creates a tunnel through the underlying network. In transport mode, it provides security over normal physical links or over a tunnel created with a different protocol. IPSec can also provide either authentication alone using Authentication Headers (AH) or encryption and authentication using Encapsulation Security Protocol (ESP). Table 4-4 describes the differences between AH and ESP.

TABLE 4-4 IPSec AH and ESP

	AH	ESP
Protocol	51	50
Fields Added	Authentication Header	ESP Header, ESP Trailer, ESP Authentication Trailer
IP Header— Tunnel Mode	Creates new tunnel IP header	Creates new tunnel IP header
IP Header— Transport Mode	Uses original IP header	Uses original IP header
TOS Byte— Transport Mode	Copies original TOS markings to new IP header	Copies original TOS markings to new IP header
TOS Byte— Transport Mode	Original TOS byte is available	Original TOS byte is available
Payload Change	None	Encrypts payload
Authentication Protocols Supported	MD5, SHA	MD5, SHA
Encryption Protocols Supported	None	DES, 3DES, AES

MD5 = Message Digest 5

SHA = Secure Hash Algorithm

DES = Data Encryption Standard

AES = Advanced Encryption Standard

CHAPTER 4

QoS DETAILS

Although both GRE and IPSec allow traffic to be classified based on its original TOS markings, there are times when you might want to classify based on other fields, such as port number or original IP address. In that case, packets must be classified before the original IP header is hidden or encrypted. To do this, use the **qos pre-classify** command. This command causes the router to make a copy of the original IP header, and classify the packet based on that information.

Qos pre-classify can be given on a tunnel interface, in a crypto map, or on a virtual template interface, and it works only on IP packets. Use it on the tunnel interface for a GRE tunnel, on the virtual interface for a L2TP tunnel, and under both the crypto map and the tunnel interface for an IPSec tunnel—IF classification must be done on non-TOS fields.

Enterprise-Wide QoS Deployment

SLA

A company might use a Service Level Agreement (SLA) to contract with their ISP for certain levels of service. This typically provides levels of throughput, delay, jitter, packet loss, and link availability, along with penalties for missing the SLA. With Layer 2 links (such as frame relay), the service provider is not involved in providing QoS through its network. With Layer 3 links (such as MPLS), the service provider can contract for QoS SLAs through its network.

Service providers use a set number of classes, and your marking must conform to their guidelines to use QoS SLAs. When calculating the amount of delay (or latency), jitter, and packet loss for your SLA, remember to take into account your internal network performance. For example, voice is best with an end-to-end delay of 150 ms or less. If the latency in the LAN at each site is 25 ms, then your latency SLA with the ISP should be no more than 100 ms.

Enterprise QoS

Each block within an enterprise network has its own QoS needs and considerations. In general, you should:

- Classify and mark traffic as close to the access edge as possible. Switches that can accomplish this in hardware are more efficient than routers that must do it in software.
- Establish the correct trust boundaries.
- Prioritize real-time traffic (such as voice and video).
- Configure the appropriate queues on outbound interfaces.

At the Access switch level:

- Set the trust boundary appropriately.
- Classify and mark non-VoIP traffic.
- Place VoIP traffic in interface priority queue.
- Set speed and duplex.
- Can use multiple queues, especially on uplink ports.

CHAPTER 4

QoS DETAILS

At the Distribution switch level:

- Set the trust boundary appropriately.
- Place VoIP traffic in interface priority queue.
- Set speed and duplex.
- Can use multiple queues, especially on uplink ports.
- Might use Layer 3 policing and marking.
- Use WRED in data queues.

At the WAN edge:

- Determine SLA.
- Might need to reclassify and remark traffic.
- Use LLQ for real-time traffic.
- Use WRED in data queues.
- Might need to use shaping, compression, or LFI.

Within the Service Providers's network:

- Have a DiffServ-compliant backbone.
- Use LLQ or modified deficit round robin (MDRR).
- Plan for adequate capacity.
- Use WRED in data queues.

The actual configuration done on WAN edge routers depends on whether or not the router is managed by the provider. If it is managed, then the provider configures output policies on the customer router and does not need any input policies on the provider edge router. For traffic bound from the provider network to the customer network, the provider edge router has the configuration to enforce SLAs.

If the customer edge router is not managed, then customers must configure their own QoS policies. The service provider likely also configures their edge router to enforce contracted SLAs on traffic from the customer. For traffic bound from the provider network to the customer network, the provider edge router has the configuration to enforce SLAs. The customer might have other types of configuration, such as reclassifying and remarking to fit their internal QoS policies.

CoPP

Control Plane Policing (CoPP) allows QoS to be applied to the router's control plane to avoid overrunning the router's CPU. The control plane consists of high-level processes that run on the route processor, and handles management tasks, such as traffic bound to or from the router or switch itself.

CoPP uses the MQC to control the traffic bound to and from the router or switch's control plane. Two policy options are available—police or drop. To configure CoPP, take the following steps:

CHAPTER 4

QoS DETAILS

- Step 1.** Configure a class map that identifies traffic to be policed.
- Step 2.** Configure a policy map that either polices the traffic permitted by the class map or drops it.
- Step 3.** Enter control plane configuration mode using the global **control-plane** command.
- Step 4.** Apply the service policy to the control plane.

During a DOS attack, or times of heavy processor use, CoPP can ensure that the network device remains available and critical network traffic can be processed.

CHAPTER 5

AutoQoS

Chapter 3 contained an introduction to AutoQoS. This chapter expands on that and offers some restrictions, caveats, and ways to tune it.

Some benefits of AutoQoS include:

- Classification of applications
- Automatic policy generation
- QoS configuration
- Monitoring and recording via SNMP and Cisco QPM
- Consistent QoS policies

AutoQoS originally supported only VoIP applications. AutoQoS for VoIP is available on all Cisco routers and switches. Implementing it is basically a one-step process, as shown in the example in Chapter 3 (click [here](#) to review that example for router configuration.)

AutoQoS for Switches

To configure AutoQoS on a switch, use the interface command **auto qos voip {cisco-phone | cisco-softphone | trust}**. Use the **cisco-phone** keyword when the interface connects to a phone; QoS markings are trusted when a Cisco IP phone is detected. Use the **cisco-softphone** keyword when the interface connects to a computer using the Cisco' SoftPhone. Use the **trust** keyword when the interface links to a trusted switch or a router. Giving this command automatically enables global QoS support (**mls qos**). Use **show auto qos** or **show mls qos interface interface-id** to view the AutoQoS configuration and the QoS actions.

AutoQoS for Routers

Routers can also use AutoQoS; recent IOS versions support AutoQoS for Enterprise applications. AutoQoS for Enterprise is currently supported only on routers, and is a two-step process. The configuration can be manually tuned after it is automatically generated.

Step 1. Application discovery and policy generation—The first step is to enable application discovery on interfaces where QoS is configured. NBAR is typically used for this. The router then collects application data for the desired number of days, analyzes the data, and creates QoS templates. You can review these configurations before applying them to the interface.

Use the interface command **auto discovery qos [trust]** to enable application discovery. Without the optional **trust** keyword, the router uses NBAR. With it, the router classifies traffic by DSCP markings. Use **show auto discovery qos** to view the traffic discovered and the configuration that is implemented.

Step 2. Implement the AutoQoS policies—Apply the policies generated by AutoQoS to the interface(s). Use the interface command **auto qos [voip [trust] fr-atm]**. The optional **voip** keyword enables only AutoQoS for VoIP. If you use this, you can then optionally choose to trust the existing DSCP markings with the keyword **trust**, or enable AutoQoS for frame-relay to ATM with the optional keyword **fr-atm**. Use **show auto qos** to view the AutoQoS configuration.

CHAPTER 5

AUTOQoS

AutoQoS Restrictions and Caveats

CEF must be enabled for AutoQoS to work. The interface must not have an existing QoS policy configured. Bandwidth should be configured on each interface. If you change the bandwidth after enabling AutoQoS, the router does not change its QoS policies to reflect the new bandwidth. It classifies links as slow or fast, with slow being 768 kbps or less. An IP address must be configured on slow speed links prior to enabling AutoQoS because the router uses Multilink PPP and transfers the IP address to the multilink interface by default. On access switches, CDP must be enabled for the switch to detect a Cisco IP phone.

SNMP support and an SNMP server address must be configured on the router for SNMP traps to work. The SNMP string “AutoQoS” needs to have write permissions.

AutoQoS supports the following WAN interfaces:

- Frame-relay point-to-point subinterfaces. The PVC must not have a map class or virtual template already assigned to it. If LFI is needed, AutoQoS configures it for G.729 codec use. Manual tuning is needed for G.711 use.
- ATM point-to-point PVCs. The PVC must not have a virtual template already assigned to it. Configure it as VBR-NRT.
- Serial interfaces using PPP or HDLC. AutoQoS must be configured on both sides of the link, and both sides must use the same bandwidth.

Tuning AutoQoS

AutoQoS might need tuning for three common reasons. First, it can configure too many classes for your network needs. Second, it does not adapt to changing network conditions. Third, it just might not include the types of policies you want.

Some questions to ask as you evaluate the policies generated by AutoQoS include:

- How many classes were created using class maps?
- What classification criterion was used to place traffic into each class?
- What DSCP and COS markings were configured for each traffic class?
- What types of queuing or other QoS mechanisms were implemented?
- Was the policy applied to the interface, PVC, or subinterface?

AutoQoS Classes

AutoQoS uses up to ten different traffic classes, as shown in Table 5-1. The table also shows the type of traffic included in each class, along with its DSCP and COS markings.

TABLE 5-1 AutoQoS Traffic Classes

Traffic Class	Traffic Type	DSCP	COS
IP Routing	Network control traffic (for example, routing protocols)	CS6	6
Interactive Voice	Voice bearer traffic	EF	5
Interactive Video	Interactive video data traffic (for example, video conferencing)	AF41	4
Streaming Video	Streaming media traffic	CS4	4
Telephony Signaling	Voice signaling and control traffic	CS3	3
Transactional & Interactive Data	Transactional database applications such as SQL	AF21	2
Network Management	Network management traffic such as telnet	CS2	2
Bulk Data	Email traffic, general data traffic, bulk data transfers	AF11	1
Scavenger	Traffic needing less-than-best-effort treatment	CS1	1
Best Effort	Default class, includes all other traffic	0	0

Too many classes might be generated for your needs. Most companies use between three and six classes. You might want to manually consolidate some classes with similar QoS needs after AutoQoS has finished its configuration.

AutoQoS and Changing Network Conditions

AutoQoS creates its configuration based on the traffic it discovered during the initial discovery phase with NBAR. If conditions change, you might need to disable it, run autodiscovery again, and then re-enable AutoQoS.

Manually Tuning AutoQoS Configurations

The **show auto qos** command shows the class maps that AutoQoS created, along with their match criteria and any access lists it created. It shows the policy maps, and the policy configured for each class. It shows where the policy was applied and any monitoring that was implemented. This can help you determine what changes are needed to the configuration.

You can modify AutoQoS configuration in two ways:

- Using Cisco QPM
- Manually with the MQC

CHAPTER 5

AUTOQoS

To modify using the MQC, allow the router/switch to apply its AutoQoS configuration. Copy the relevant portions to a text editor and make the desired changes. This might include changing the classification criteria, combining classes, or altering the policy for a particular class, for instance. Then replace the old configuration with the new one.

CHAPTER 6

Wireless Scalability

Wireless LANs (WLAN) are an extension to wired networks using wireless standards, such as 802.11A/B/G. The 802.11 standards take the place of the Ethernet standard, but both data-links support the same types of services. The benefit of WLANs is that it allows users to relocate within the workspace, closer to machinery or conference rooms, for instance.

WLAN QoS

802.11 wireless uses carrier sense multiple access/collision avoidance (CSMA/CA), meaning transmissions are pre-announced, because systems may not be able to hear each other or recognize collisions later. CA uses a Distributed Coordination Function (DCF) to implement timers and delays to ensure cooperation.

Unfortunately, DCF timers interfere with low-latency applications, such as voice and video. Wi-Fi Multimedia (WMM or 802.11e) is an attempt to shorten timers—proportional to Differentiated Services Code Point (DSCP) priority—and prioritize important traffic. WMM replaces DCF with enhanced DCF (EDCF) that creates four categories (platinum, gold, silver, and bronze) and forces longer interframe waits on lower-priority traffic.

LWAP

Cisco introduced Lightweight Access Points (LWAP) that use the concept of “split MAC,” which separates the real-time communication and management functions. An LWAP controls beaconing, buffering, and encryption and uses a controller for 802.1x, **Extensible Authentication Protocol** (EAP), key management, and bridging functions.

In the LWAP scenario, QoS is handled at the controller. QoS is marked at Layer 2 using 802.11e. 802.11e, like 802.1p, will not pass through a router, so it has to be converted to DSCP if used end-to-end in a large network. Similarly, .1p and DSCP fields must be mapped back to WMM when traffic goes to the client.

Controllers host profiles that describe traffic handling. At the controller, an administrator can specify:

- Average and burst “best effort” data rate
- Average and burst “real-time” data rate
- Maximum RF usage (set to 100%)
- Queue depth, which is the number of packets that will be in the queue if the line is busy
- WMM-to-802.1p mapping

Furthermore, the controller may be set up to ignore, allow, or require 802.11e.

802.1x and WLAN Security

WLAN security is important because wireless systems are designed to allow easy access and may extend beyond the physical perimeter of your building. Many WLAN implementations do not have encryption or authentication. Small wonder then that “war driving,” or the act of randomly wondering in search of an open AP, is so easy to perform.

The number-one problem is that most APs are insecure by default and few have any security added to them. When present, security for WLANs is accomplished through authenticating users and encrypting traffic. Old forms of authentication and encryption have been found vulnerable, so APs must be kept current. Types of wireless security include:

- Service Set Identifier (SSID)
- Authentication by MAC
- Static Wired Equivalent Privacy (WEP) keys
- One-way authentication

Network administrators must not only ensure their APs are secure, they must always look for rogue APs (access points put up by users to accomplish a narrow goal without regard to corporate security).

Note

LWAPs and their controllers help with AP security and rogue AP detection. LWAPs, because they are controlled from a central point, are more scalable because administration is much easier. Cisco LWAP/Controller model also has rogue detection baked in.

Figure 6-1 shows a timeline of WLAN security.

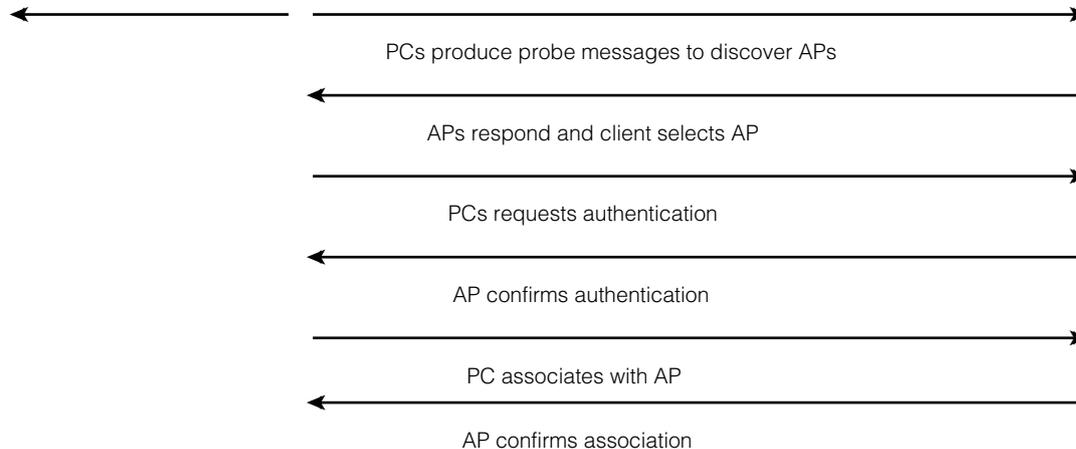
FIGURE 6-1 WLAN Security over Time



802.11 WEP supports open and shared key authentication. Open authentication means that no authentication is used and any user is allowed to associate with an AP. Shared key authentication expects a cryptographic key to be known before accessing the AP; this key is subsequently used to encrypt the payload. To authenticate using a shared key, an AP sends a plain-text challenge, which the PC encrypts and sends back. If it is encrypted correctly, the PC is authenticated. More detail is provided in Figure 6-2, which shows the entire authentication process.

WIRELESS SCALABILITY

FIGURE 6-2 WLAN Authentication



Enhanced WEP was a Cisco proprietary fix to WEP that added two improvements:

- 802.1x for authentication
- Cisco Key Integrity Protocol (CKIP) to protect the key

WPA (Wi-Fi Protected Access), the pre-standard version of 802.11i, mirrored the Cisco Enhanced WEP by enhancing encryption and authentication in much the same way. Encryption is improved by incorporating Temporal Key Integrity Protocol (TKIP). WPA2 (standard 802.11i) added Advanced Encryption Standard (AES) encryption. Authentication was improved to support 802.1x and the Extensible Authentication Protocol (EAP).

Key improvements in WPA/WPA2 include the following:

- Per-session keys allow users a different key each time the user accesses the AP.
- TKIP changes the way the key is applied to consecutive packets.
- Encryption uses a starting number called an Initialization Vector (IV). WPA uses an IV that is harder to guess.
- The cryptographic function is changed to 128-bit AES. AES is a standard that is common in security functions, such as virtual private networks (VPN).
- 802.1x for encrypted RADIUS authentication. RADIUS can be linked back to Active Directory, so users sign in with familiar usernames and passwords.

802.1x requires that the client and AP support EAP and that a RADIUS server is present. There are several methods based on EAP to accomplish authentication:

- Lightweight EAP (LEAP)
- EAP Flexible Authentication via Secure Tunnel (EAP-FAST)
- EAP-Transport Layer Security (EAP-TLS)
- Protected EAP (PEAP)

CHAPTER 6

WIRELESS SCALABILITY

Configuring WLAN Security on Controller

Open (no authentication) is typically set up on public APs. On a Cisco WLAN controller, go to **WLANs>Edit** and then set Layer 2 Security to None.

Setting Layer 2 Security to Static WEP, WPA, or WPA2 allows control of parameters for a static key. If no WPA static key is entered, then the controller will use EAP 802.1x to RADIUS.

Setting Layer 2 Security to 802.1X supports dynamic WEP keys. Key size may be selected, but remember that Windows XP supports only 40-bit and 104-bit keys.

If Layer 3 Security is selected, then users will enter credentials on a customizable web page, which then checks an internal database or a remote RADIUS server.

WLAN Management

Cisco supports two WLAN models.

- Autonomous APs:
 - Users connect to APs.
 - APs are aggregated by Wireless Domain Services (WDS).
 - WDS is controlled by a Wireless Solution Engine (WLSE), which centralizes control similar to an LW Controller.

- Lightweight APs connected to a controller:
 - Users attach to LWAPs.
 - LWAPs are controlled by controllers.
 - Controllers are managed by Wireless Control System (WCS).

The benefit of LWAPs is centralized control. The problem is that loss of the controller brings the whole campus down, so redundancy is recommended. The lightweight model provides displays of RF coverage, dynamic management of the radio environment, detection of rogue APs, and easier roaming.

WLSE brings many of the benefits of a controller to an existing autonomous deployment. WLSE is offered in two versions, both of which also handle AAA:

- Ciscoworks WLSE for large deployments
- Ciscoworks WLSE Express for less than 100 APs

WCS allows management of the entire network as a unit. It runs as a service on Linux or Windows. Three feature sets are supported:

- Base, which detects rogue APs and tracks a device to the closest AP.
- WCS with Location, which adds support for RF fingerprinting and tracks a device to within 10 meters.
- WCS with Location+, which adds the ability to track 1500 clients at the same time and collects historical information.

Location is important to support VoIP calls to 911.

CCNP ONT Quick Reference Sheets

Brent Stewart
Denise Donohue

Copyright© 2007 Cisco Systems, Inc.

Published by: Cisco Press

800 East 96th Street

Indianapolis, IN 46240 USA

All rights reserved. No part of this digital short cut may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing October 2006

ISBN: 1-58705-315-2

Warning and Disclaimer

This digital short cut is designed to provide information about networking. Every effort has been made to make this digital short cut as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this digital short cut that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this digital short cut should not be regarded as affecting the validity of any trademark or service mark.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers’ feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this digital short cut or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please make sure to include the digital short cut title and ISBN in your message.

We greatly appreciate your assistance.

Corporate and Government Sales

Cisco Press offers excellent discounts on this digital shortcut when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the U.S., please contact: International Sales international@pearsoned.com



Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

©2006 Cisco Systems, Inc. All rights reserved. CCVP, the Cisco logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking Academy, Network Registrar, Packet, PIX, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0609R)