Kevin Wallace, CCIE No. 7945, CCSI, CCNP, CCDP, MCSE 4, CNE 4/5, is a full-time instructor for KnowledgeNet, a pioneer of next-generation e-learning. With 15 years of Cisco internetworking experience, Kevin has been a network design specialist for The Walt Disney World Resort and a network manager for Eastern Kentucky University. Kevin holds a bachelor of science degree in electrical engineering from the University of Kentucky. Among Kevin’s other publication credits are the CCDB/CCDP Flash Cards and Exam Practice Pack and the CCIE Routing and Switching Flash Cards and Exam Practice Pack, both coauthored with Anthony Sequeria and available from Cisco Press. In addition, Kevin authored the Cisco Enterprise Voice over Data Design (EVoDD) 3.3 course and has written for the Cisco Packet magazine. Kevin also holds the IP Telephony Design Specialist and IP Telephony Support Specialist CQS certifications.

Jeremy D. Cioara, CCIE, MCSE 4/2000, CNE 4/5, is a trainer at KnowledgeNet, a Cisco Learning Partner providing career Cisco certification training. In addition to training in all major certification programs (Cisco, Microsoft, and Novell) for more than seven years, he has authored many books, including Cisco IP Telephony (CIPT) 3.3 Curriculum and Cisco IP Telephony Troubleshooting (IPTT) 3.3 Curriculum, and he attained the three Cisco IP Telephony Specialist certifications. Outside of training, Jeremy has consulted for companies such as Qwest, MicroAge, Terminal Processing Systems, and IKON.

Paul Giralt, CCIE R&S, CCIE Voice No. 4793, is an escalation engineer at the Cisco Systems Technical Assistance Center in Research Triangle Park, NC. He has been troubleshooting complex IP telephony networks since the release of CallManager 3.0. Paul has troubleshoot problems in some of the largest Cisco IP Telephony deployments and has provided training for TAC teams around the globe. He holds a bachelor of science degree in computer engineering from the University of Miami and is the author of Troubleshooting Cisco IP Telephony.

Jose Martinez, CCIE No. 1690, is an escalation engineer at the Cisco Systems Technical Assistance Center in Research Triangle Park, NC. Since 1995, he has worked in the TAC, supporting multiple technologies, including IBM protocols, L2 and L3 switches, and most recently, IP telephony. As an escalation engineer, Jose has been involved in numerous network deployments. He has troubleshoot problems involving critical situations in customer’s networks and has provided training to other Cisco engineers, partners, and customers.
Overview of Legacy and IP Telephony Networks

Introduction
Understanding emerging Voice over IP (VoIP) technologies requires a solid understanding of legacy telephony networks. Therefore, the Quick Reference Sheets in this section review many of the terms and concepts that surround legacy and packet telephony networks, and those technologies are then contrasted with IP Telephony technologies.

Legacy Telephony Networks
The Public Switched Telephone Network (PSTN) is at the heart of the legacy telephony network. PSTN components include the following items:

- Edge devices (for example, phones) are used by customers to interface with the PSTN.
- Local loops connect customer locations to a local central office (CO) over a pair of wires called tip and ring.
- Phone switches allow one phone to connect to another phone by dialing a phone number. The switch interprets the dialed digits and interconnects the dialing phone's local loop with the destination phone's local loop. The “phone company” has switches that are located in COs. However, companies can have their own phone switches [for example, Private Branch Exchanges (PBXs) or key systems] located locally.
- Trunks interconnect phone switches.

Companies that have their own phone switches can select between PBXs or key systems. PBXs are typically more scalable than key systems, supporting 20 to 20,000 phones. PBX users in the United States typically dial a 9 to access an outside line. However, key systems traditionally have buttons on a key phone that the user presses to access a specific outside line. For example, one might have been in a store and heard an intercom announcement such as, “Kevin, pick up line 2.” In that example, Kevin would go to a “key phone” and press the line 2 button to access the call. Because of their scalability limitations, key systems typically support a maximum of 30 to 40 users.

Call signaling makes it possible to place an end-to-end voice call. Consider the following steps that are used to establish an end-to-end voice call:

1. A phone goes off-hook and sends digits to the local phone switch.
2. The local phone switch examines the dialed digits, makes a forwarding decision, and sends signaling information to the destination phone switch.
3. The destination phone switch signals the destination phone by sending ringing voltage to the phone.

© 2007 Cisco Systems Inc. All rights reserved. This publication is protected by copyright. Please see page 49 for more details.
When a user dials digits on an analog phone, those digits can be communicated to the local phone switch using either dual-tone multifrequency (DTMF) or pulse dialing. DTMF sends tones that are composed of two frequencies, whereas pulse dialing rapidly opens and closes the local loop to indicate dialed digits.

With digital circuits, such as T1s or E1s, multiple conversations can be carried in different channels on the same circuit. Each of these digital channels needs bits for signaling information. Common approaches include the following:

- **Common channel signaling (CCS)**—Has a channel that is dedicated to signaling. For example, in an Integrated Services Digital Network (ISDN) circuit, the D channel is dedicated to signaling.
- **Channel associated signaling (CAS)**—Can use framing bits from a few of the channels to serve as signaling bits. Sometimes this is called *robbed-bit signaling*.

Analog circuits have their own signaling mechanisms, such as the following:

- **Loop-start**—Causes a phone switch to seize a line when loop current is flowing
- **Ground-start**—Causes a phone switch to seize a line after the phone temporarily grounds the “ring” side of the circuit

Digital circuits can use multiplexing techniques to place multiple conversations on a single link. For example, time-division multiplexing can give a “time slice” to a specific channel, and by “taking turns,” you can send 24 conversations across a single link.

Frequency-division multiplexing (FDM) allows multiple conversations to be sent at the same time using different frequencies. For example, dense-wavelength-division multiplexing (DWDM) simultaneously sends multiple light frequencies over a fiber-optic cable.

### Packet Telephony Networks

Many companies that have PBXs at more than one site and interconnect those PBXs through the PSTN are migrating to a packet telephony network. A **packet telephony network** allows companies to preserve their existing investment in PBX technologies, while eliminating the recurring expense for the trunks that interconnect their PBXs. Specifically, companies can connect their PBXs to routers that are already interconnected through a wide-area network (WAN). The PBXs can then send their signaling information and voice calls over the WAN.

Call-forwarding intelligence can reside in the routers. For increased scalability, however, you can configure routers to point to external call
agents. Such a topology lays the foundation for other packet telephony technologies, including the following:

- IP phones have an Ethernet connection that sends and receives voice calls.
- Call agents replace much of the functionality that was provided previously by the PBX. For example, a call agent can be configured with route plans that dictate how voice calls are forwarded. The Cisco CallManager (CCM) is an example of a call agent.
- Gateways can forward calls between different types of networks. For example, a call from an IP phone could be forwarded through a gateway to the PSTN.
- Gatekeepers keep track of WAN resources and, based on available resources, either permit or deny a request to place a call across the WAN. In addition, the gatekeeper can provide E.164 number resolution.
- Multipoint Control Units (MCUs) contain digital signal processor (DSP) resources and can support the mixing of audio streams in a conference call.

Simply placing a voice call across a WAN does not guarantee the quality of the voice call. Data applications, for example, tend to be more forgiving of dropped or delayed packets than applications such as voice or video. Therefore, the quality of service (QoS) technology is an integral part of Cisco VoIP designs, and an entire section in these Quick Reference Sheets focuses on QoS technologies.

### IP Telephony Networks

Although packet telephony is more of a generic term, covering Voice over IP (VoIP), Voice over Frame Relay (VoFR), and Voice over ATM (VoATM), the primary focus of these Quick Reference Sheets is creating IP-based telephony networks using VoIP technologies. Therefore, with the foundational understanding of legacy and packet telephony networks, you delve into some of the components of IP Telephony.

First, you consider the analog interfaces that are available on voice-enabled routers. A Foreign Exchange Station (FXS) port allows you to connect plain old telephone service (POTS) devices to a router. For example, you could attach a traditional analog phone, speakerphone, or fax machine to an FXS port on a Cisco router, and that FXS port can act like a PBX or CO switch. For example, an FXS port can provide a dial tone when the phone goes off-hook, interpret dialed digits, and send ringing voltage to the attached phone.

A Foreign Exchange Office (FXO) port connects to a phone switch (for example, a PBX or the PSTN). The FXO port can connect into the traditional tip-and-ring connection that comes from a CO or a PBX. Because it is acting as a phone, an FXO port can go off-hook, dial digits, and answer incoming calls.

E&M is the third type of analog port, and this port interconnects PBXs. The “E” and “M” originally referred to “earth” and “magneto,” although you can think of “ear” and “mouth” to better visualize the receive and transmit functions of E&M.
OVERVIEW OF LEGACY AND IP TELEPHONY NETWORKS

Two primary digital ports (that is, interfaces) are the T1 and E1 interfaces. A T1 interface can send 24 voice channels using channel associated signaling (CAS). Alternatively, with a common channel signaling (CCS) approach, where one channel is dedicated to signaling information, a T1 interface can carry 23 voice channels.

E1 interfaces have 32 channels. However, regardless of the CAS or CCS approach, only 30 of the channels are typically available for voice paths.

Integrated Services Digital Network (ISDN) interfaces are great examples of digital CCS, where one channel is dedicated to signaling. The two flavors of ISDN are Basic Rate Interface (BRI) and Primary Rate Interface (PRI). A BRI has two 64-kbps “B” channels (that is, bearer channels) that carry the voice, video, or data. For signaling, a BRI has a single “D” channel. PRIs, however, are based on T1 or E1 interfaces, where either 23 or 30 voice channels are available.

These digital and analog interfaces just described provide connectivity from Cisco routers to legacy telephony networks. However, we now consider how IP phones connect into this topology.

Some of the Cisco IP Phones are actually three-port switches. One port connects to the IP phone itself and a second port connects to a Catalyst switch in the wiring closet; the third port can connect to a PC. By having a port that connects to a PC, the IP phone allows the PC to daisy-chain through the phone, back to the switch, thus eliminating the need for extra wiring for the IP phone. As the IP phone forwards its voice packets and the PC’s data packets back to the wiring closet switch, the IP phone can place the different packets in different virtual LANs (VLANs) and give the packets different priority markings. As discussed in the section “Ensuring Voice Quality,” later in these Quick Reference Sheets, those priority markings that are assigned by the IP phone can be referenced by switches or routers, which can make forwarding or dropping decisions based on those markings. Cisco IP Phones register with a Cisco CallManager (CCM), which acts as a call agent.

When you have multiple sites (for example, a main headquarters site and remote office sites) that contain IP phones, those CCMs can be located centrally at the headquarters location. In such an example, IP phones at the remote sites register with CCMs over the WAN link. If the WAN were to go down, these phones would lose connectivity with the headquarters site. To preserve basic service for those IP phones at remote sites, you can configure Survivable Remote Site Telephony (SRST), which allows a Cisco router to stand in for a CCM and perform basic functions in the event of a WAN outage.
Although a centralized deployment can minimize the number of CCMs that must be purchased, the IP WAN is a potential point of failure. Therefore, you can choose a distributed deployment model, in which you have CCMs at all remote offices. In this example, if a WAN link were to fail, the remote offices' IP phones maintain connectivity to local CCMs, and they still can route calls out to the PSTN by leveraging, for example, FXO ports on a Cisco router acting as a gateway.
The Mechanics of Analog and Digital Voice Circuits

Introduction
In the previous section, you were introduced to analog and digital voice connections. Now, the material gets more specific and examines the electrical characteristics of these connections.

Analog Voice
First, consider the following types of signaling that are present in the analog telephony world:

- **Supervisory signaling**—Indicates the on-hook or off-hook condition of a phone, based on whether loop current is flowing. In addition, “ringing” is considered to be supervisory signaling. Ringing voltage is sent from the phone switch to alert the destination phone that it is receiving an incoming call. In the United States, the pattern of ringing (that is, *ring cadence*) is 2 seconds on and 4 seconds off.

- **Address signaling**—Allows a phone to dial (that is, specify the address of) a destination phone. The older method of dialing digits was with a rotary phone, which used “pulse” dialing. Pulse dialing rapidly opens and closes the tip-and-ring circuit. This series of open and closed circuit conditions within specific timing parameters indicates a dialed digit.

- **Information signaling**—Like DTMF, information signaling uses combinations of frequencies to, in this case, indicate the status of a call (that is, to provide information to the caller). For example, a busy signal is a combination of a 480-Hz tone and a 620-Hz tone, with on/off times of 0.5/0.5 seconds.

In the previous section, you were introduced to the concept of a **trunk**, which interconnected phone switches. Also, you saw how loop-start signaling seized a line when loop current began to flow, and how ground-start signaling seized a line by giving its tip lead a ground potential. However, consider E&M signaling more closely. Five types of E&M signaling exist (that is, Type I through Type V), and these types define such things as the number of wires used for an E&M circuit and the polarity of those wires. Note that the voice path does not use the E&M leads. The E&M leads are intended only for signaling.

With E&M, three types of signaling can occur over the E&M leads: wink start, immediate start, and delay start. However, the most common type of E&M signaling is wink start. With wink-start signaling, the calling equipment (for example, the router) seize a line by applying...
voltage to its M lead. The called equipment (for example, the PBX) “winks” by toggling its M lead on and off. When the calling equipment sees this wink, it sends its dialed digits across the voice path.

Voice ports on voice-enabled Cisco routers also can help you with the problem of echo. An impedance mismatch in a 2-wire–to–4-wire hybrid circuit (such as those found in analog phones) is the typical cause of echo. Fortunately, Cisco routers can listen to the analog voice waves that are being sent out of, for example, an FXS port. If that same waveform comes back into the router (within 8 ms by default on pre-IOS 12.3 platforms), the router interprets the waveform as echo and cancels the echo by internally playing an inverse waveform (that is, a waveform that is 180 degrees out of phase with the echo waveform).

**Digitizing the Spoken Voice**

To transmit the spoken voice across a digital network or an IP network, you need to digitize the analog speech patterns. In this section, you see how this conversion happens. Also, you might want to conserve WAN bandwidth by compressing those now-digitized voice packets.

To digitize an analog waveform, you periodically take samples of the analog waveform’s amplitude. However, the question is this: How many samples should you take? The Nyquist Theorem, developed by Harry Nyquist in 1933, says that you need to sample at a rate that is at least twice as high as the highest frequency that is being sampled. For voice, in theory, the highest sampled frequency is 4 kHz. Therefore, the Nyquist Theorem indicates that you need to take 8000 samples per second, which means that you need to take a sample every 125 microseconds.

According to the Nyquist Theorem, you should sample a waveform at least twice as many times as the highest frequency. For example, if the highest frequency you wanted to sample were 4000 Hz, sample at twice that rate. Specifically, you should sample $4000 \times 2 = 8000$ samples per second.

The Cisco router can, by default, store 8 ms of transmitted waveforms. If a received waveform matches the stored waveform, the router can internally play an inverse waveform (that is, 180 degrees out of phase) to cancel the echo waveform.
These samples, consisting of a single frequency, have amplitudes equaling the amplitudes of the sampled signaling at the instant of the sampling. This is called Pulse Amplitude Modulation (PAM). The next step is to take these PAM amplitudes and assign them a number, which can be sent in binary form. The process of assigning a number to an amplitude is called **quantization**.

After the analog waveforms have been digitized, you might want to save WAN bandwidth by compressing those digitized waveforms. The processes of encoding and decoding these waveforms are defined by **codecs**. The various forms of waveform compression are as follows:

- **Pulse Code Modulation (PCM)**—Does not actually compress the analog waveform. Rather, PCM samples and performs quantization (as previously described) with no compression. The G.711 codec uses PCM.

- **Adaptive Differentiated PCM (ADPCM)**—Uses a “difference signal.” Instead of encoding an entire sample, ADPCM can send the difference in the current sample versus the previous sample. G.726 is an example of an ADPCM codec.

- **Conjugate Structure Algebraic Code Excited Linear Predication (CS-ACELP)**—Dynamically builds a codebook based on the speech patterns. It then uses a “look-ahead buffer” to see whether the next sample matches a pattern that is already in the codebook. If it does, the codebook location can be sent, instead of the actual sample. G.729 is an example of a CS-ACELP codec.

- **Low-Delay Conjugate Excited Linear Predication (LDCELP)**—Is similar to CS-ACELP. However, LDCELP uses a smaller codebook, resulting in less delay but requiring more bandwidth. G.728 is an example of an LDCELP codec.

To assign a number to these samples, you establish logarithmic thresholds, and you assign numbers to samples whose amplitudes fall between specific thresholds. Because this process is really “rounding off” to a threshold value, you are introducing **quantization error**, which adds noise (that is, hissing) to the signal. This hissing is reduced, because a logarithmic scale is being used, and small signals are more likely to occur than large signals. Also, large signals tend to mask the noise.

After Pulse Amplitude Modulation (PAM) samples, you need to quantize these samples (that is, assign numbers to represent their amplitudes). However, if you use a linear scale (as shown), the quantization error (as indicated by the deltas) causes distortion in the voice. This distortion is especially noticeable at lower volumes. Therefore, instead of a linear scale, use a logarithmic scale, which has more measurement intervals at lower volumes.

To assign a number to these samples, you establish logarithmic thresholds, and you assign numbers to samples whose amplitudes fall between specific thresholds. Because this process is really “rounding off” to a threshold value, you are introducing quantization error, which adds noise (that is, hissing) to the signal. This hissing is reduced, because a logarithmic scale is being used, and small signals are more likely to occur than large signals. Also, large signals tend to mask the noise.

After Pulse Amplitude Modulation (PAM) samples, you need to quantize these samples (that is, assign numbers to represent their amplitudes). However, if you use a linear scale (as shown), the quantization error (as indicated by the deltas) causes distortion in the voice. This distortion is especially noticeable at lower volumes. Therefore, instead of a linear scale, use a logarithmic scale, which has more measurement intervals at lower volumes.

To assign a number to these samples, you establish logarithmic thresholds, and you assign numbers to samples whose amplitudes fall between specific thresholds. Because this process is really “rounding off” to a threshold value, you are introducing quantization error, which adds noise (that is, hissing) to the signal. This hissing is reduced, because a logarithmic scale is being used, and small signals are more likely to occur than large signals. Also, large signals tend to mask the noise.

After Pulse Amplitude Modulation (PAM) samples, you need to quantize these samples (that is, assign numbers to represent their amplitudes). However, if you use a linear scale (as shown), the quantization error (as indicated by the deltas) causes distortion in the voice. This distortion is especially noticeable at lower volumes. Therefore, instead of a linear scale, use a logarithmic scale, which has more measurement intervals at lower volumes.

To assign a number to these samples, you establish logarithmic thresholds, and you assign numbers to samples whose amplitudes fall between specific thresholds. Because this process is really “rounding off” to a threshold value, you are introducing quantization error, which adds noise (that is, hissing) to the signal. This hissing is reduced, because a logarithmic scale is being used, and small signals are more likely to occur than large signals. Also, large signals tend to mask the noise.

After Pulse Amplitude Modulation (PAM) samples, you need to quantize these samples (that is, assign numbers to represent their amplitudes). However, if you use a linear scale (as shown), the quantization error (as indicated by the deltas) causes distortion in the voice. This distortion is especially noticeable at lower volumes. Therefore, instead of a linear scale, use a logarithmic scale, which has more measurement intervals at lower volumes.

To assign a number to these samples, you establish logarithmic thresholds, and you assign numbers to samples whose amplitudes fall between specific thresholds. Because this process is really “rounding off” to a threshold value, you are introducing quantization error, which adds noise (that is, hissing) to the signal. This hissing is reduced, because a logarithmic scale is being used, and small signals are more likely to occur than large signals. Also, large signals tend to mask the noise.
Working with Cisco products, you normally use G.711 (which requires 64 kbps of bandwidth for voice payload) in the LAN environment and G.729 (which requires 8 kbps of bandwidth for voice payload) over the WAN. G.729 has a couple of variants. Although all forms of G.729 require 8 kbps of bandwidth, G.729a uses a less-complex algorithm, which saves processor resources with slight quality degradation. G.729b enables voice activity detection (VAD), which suppresses the sending of silence if a party in the conversation does not speak for, by default, 250 ms.

Codecs vary in their bandwidth requirements, and in their quality. To measure quality, you can use a mean opinion score (MOS), which uses a “trained ear” to judge the quality of voice after passing through the codec that is being tested. MOS values range from 1, for unsatisfactory quality, to 5, for no noticeable quality degradation. For toll-quality voice, however, an MOS value in the range of 4 is appropriate. The G.711 codec has an MOS value of 4.1. Accompanied by a significant bandwidth savings, G.729 has an MOS of 3.92, while the less-processor-intensive G.729a has an MOS of 3.9.

The challenge with MOS is that at its essence, it is based on opinion. Another approach to quality measurement is Perceptual Speech Quality Measurement (PSQM), which digitally measures the difference in the original signal and the signal after it passes through a codec.

Digital Signaling

Previously, you reviewed how analog signaling (for example, loop-start) functions. Next, consider digital signaling. On a T1 circuit, each frame (including the framing bit) is 193 bits. Typically, you use a framing approach called extended super frame (ESF), which groups 24 of those standard 193-bit frames together. Because the frames are grouped, you do not need all 24 framing bits. Therefore, you can use robbed-bit signaling (that is, channel associated signaling [CAS]) to send signaling information in the framing bit of every sixth frame. Specifically, you can use the framing bit from frames 6, 12, 18, and 24 in an ESF for signaling purposes.

Alternatively, T1s can use common channel signaling (CCS), which supports 23 voice channels and one channel, the 24th channel, that is dedicated to carrying only signaling information.

An E1 circuit has 32 channels, and you can use up to 30 of them for voice. The first channel (that is, time slot 0) is used for framing information; the 17th channel (that is, time slot 16 or TS 16) carries signaling information.

In a CAS implementation, one frame might use channel 17 to carry signaling information for channels 2 and 18, while the next frame uses channel 17 to carry signaling information for channels 3 and 19.

With a CCS E1 implementation, the 17th channel (that is, TS 16) is dedicated to carrying signaling information for all other channels that use a protocol called Q.931. ISDN is an example of a CCS technology, which reserves its D channel to carry signaling information.

Whereas the Q.931 signaling protocol is often used from a customer site to a CO, the Q-Signaling (QSIG) protocol is a standards-based approach to signaling between different PBX vendors.
You can also encounter the Digital Private Network Signaling System (DPNSS) protocol, which also can interconnect PBXs. DPNSS was developed by European PBX vendors in the early 1980s, which was before ISDN standards were established. Numerous Cisco IOS gateways can function in a DPNSS network, because DPNSS can run over a standard ISDN interface.

COs typically use Signaling System 7 (SS7) as the signaling protocol between CO switches. For VoIP networks, you can use Signaling Transport (SIGTRAN) to send SS7 messages over an IP network. Specifically, SIGTRAN transports these SS7 messages using a Layer 4 protocol called Stream Control Transport Protocol (SCTP).

The Challenge of Compressing Nonvoice Streams

Although codecs such as G.729 do a great job of compressing voice, they are not designed to compress nonvoice signals such as fax or modem tones. Fax and modem information can be transmitted using G.711 without a problem, but the G.729 codec corrupts these signals to a point where they cannot be interpreted.

Cisco has a proprietary solution for this situation, called Cisco Fax Relay. With Cisco Fax Relay, the router’s DSPs hear the fax tones and do not compress those tones using G.729. A similar industry-standard approach is T.38 Fax Relay. As an additional benefit, Cisco routers can send faxes to PCs and servers that are configured with T.38 fax software.

Another approach to sending faxes across the WAN is T.37 Fax Store and Forward. With the T.37 approach, a Cisco router (called an on-ramp) can convert fax data into a TIFF attachment in an e-mail message and transmit that attachment to a store-and-forward e-mail server. This server can then deliver the fax e-mail messages to an off-ramp Cisco router, which initiates a session with the destination fax machine.

To transmit modem tones across a WAN when you have specified the G.729 codec for voice traffic, you can use modem relay, which sends modem information through the Simple Packet Relay Transport (SPRT). The last-hop router then remodulates the data and sends it to the destination router.
Configuring Router Voice Ports

Introduction
Now that you understand, in theory, the operation of various voice interfaces, in this section, you review the configuration of these interfaces on Cisco voice-enabled routers. First, consider the following categories of voice calls:

- **Local calls**—Occur when both the calling and called phones are attached to the same router.
- **On-net calls**—Span more than one router. Specifically, the calling phone is attached to one router, and the called phone attaches to a different router. In this case, routers are part of the same network.
- **Off-net calls**—Origin on a router but terminate on the PSTN.
- **Private Line Automatic Ringdown (PLAR) calls**—Occur when a caller picks up a phone and the phone automatically dials a preconfigured number.
- **PBX-to-PBX calls**—Are on-net calls, where the source and destination are PBXs.
- **CallManager-to-CallManager calls**—Occur when IP phones register with Cisco CallManagers (CCMs) and one CCM forwards a call to another CCM.
- **On-net to off-net calls**—Originally intend to be on-net calls. However, because of conditions such as WAN oversubscription or a WAN outage, the call is diverted off-net (for example, to the PSTN).

### Analog Voice Ports

A phone connects to an FXS port, just as a phone would connect to a PBX or the PSTN. Therefore, you can configure parameters such as signal type (that is, loop-start or ground-start), ring pattern, impedance (to match the impedance of the connecting device), and call progress tones (for example, what a busy signal sounds like). Consider the following FXS configuration example.

**FXS Port Configuration**

```
Router(config)#voice-port 1/1/1
Router(config-voiceport)#signal loopstart
Router(config-voiceport)#impedance 600r
Router(config-voiceport)#ring cadence pattern02
Router(config-voiceport)#output attenuation -2
Router(config-voiceport)#input gain 3
Router(config-voiceport)#echo-cancel coverage 32
```

In this example, voice port 1/1/1 is an FXS port, and you are specifying that it should use loop-start signaling, as opposed to ground-start. Also, the impedance is set to 600 ohms resistive (that is, no capacitive component). The ringing pattern is set to the predefined pattern02.
which specifies a cadence of 1 second on and 4 seconds off. Because
you want the volume of VoIP calls to be approximately the same as the
volume of the calls in a PBX environment (to make the VoIP network
as transparent to the users as possible), you can make gain and attenua-
tion adjustments. In this example, you are attenuating the volume of
calls that are being sent out of the port to the attached phone by 2 deci-
bels (dB), with the output attenuation command. The input gain
command, in this example, is increasing the volume of the waveforms
that are coming from the phone into the router by 3 dB.

To combat echo, instead of the default 8-ms period of time during
which a router can recognize an incoming waveform as echo, you are
increasing the coverage to 32 ms, with the echo-cancel coverage
command. Finally, to maximize echo cancellation, you enabled the
nonlinear feature, which suppresses all incoming waveforms from the
phone until the volume is loud enough to be interpreted as speech. Note
that this nonlinear feature can lead to “clipping” when the other party
begins to speak. You can enter the nonlinear feature with the non-
linear voice-port configuration-mode command.

An FXO port connects to a phone switch (for example, a CO switch or
PBX). Therefore, an FXO port acts like a phone. Typical parameters
that you can configure on an FXO port are signaling (which must
match the signaling type of the phone switch to which you are connect-
ing), dial type (that is, DTMF or pulse), and the number of incoming
rings before the FXO port answers (that is, goes off-hook). Consider
the following FXO configuration example.

Router(config)#voice-port 1/2/1
Router(config-voiceport)#signal loopstart
Router(config-voiceport)#ring number 3
Router(config-voiceport)#dial-type pulse

In this example, voice port 1/2/1 is an FXO port, and you are specify-
ing that it should use loop-start signaling. Also, because the FXO port
acts like a phone, you can specify how many rings it receives before it
answers. In this case, the FXO port will answer after 3 rings. Also,
when the FXO port dials, it is configured to use pulse dialing.

An E&M port typically connects to an existing E&M port on a PBX.
Typical parameters that you can configure on an E&M port include the
signaling type (for example, wink start), the E&M type (that is, 1, 2, 3,
or 5), and the number of wires that are used for the voice path (that is,
the “operation”). Consider the following E&M configuration:

Router(config)#voice-port 2/1/1
Router(config-voiceport)#type 1
Router(config-voiceport)#operation 4-wire
Router(config-voiceport)#signal wink-start

© 2007 Cisco Systems Inc. All rights reserved. This publication is protected by copyright. Please see page 49 for more details.
In this example, an E&M port is configured as E&M Type I. In addition, the voice path, which does not use the E&M leads, uses four wires, meaning that both the tip and ring leads have their own return path. The default signaling method of wink start is specified also.

Numerous timing options can be configured for Cisco voice ports. For example, consider the following configuration:

```
Router(config)#voice-port 1/1/1
Router(config-voiceport)#timeouts interdigit 20
Router(config-voiceport)#timeouts initial 20
Router(config-voiceport)#timeouts call-disconnect 20
```

In this example, voice port 1/1/1 is an FXS port. The `interdigit` parameter determines the maximum number of seconds allowed between dialed digits. The `initial` parameter specifies how long the caller can receive a dial tone before he dials the first digit. Finally, the `call-disconnect` parameter indicates how long this port remains in an off-hook condition if the other party disconnects first. All units of measure in this example are seconds.

Some PBXs interconnect using trunks that use E&M ports. You can replace a trunk connection with an IP WAN connection by connecting the PBX E&M ports into E&M ports on Cisco voice-enabled routers.

Digital Voice Ports

Digital voice interfaces, such as T1, E1, and ISDN interfaces, have unique interface-specific configuration parameters. These parameters can include the type of line coding (for example, B8ZS) and framing (for example, ESF) that is used on the digital circuit. Consider the following T1 configuration example, noting that a T1 is configured from controller-configuration mode, as opposed to interface- or voice-port configuration mode.

```
Router(config)#controller 2/0
Router(config-controller)#clock source line
Router(config-controller)#framing esf
Router(config-controller)#linecode b8zs
```

In this example, T1 controller 2/0 gets its clocking from the network, as indicated by the `clock source line` command. In addition, the T1 is configured for extended superframing (ESF) and B8ZS line coding.

Sometimes, PBXs use proprietary signaling approaches that a Cisco router cannot interpret. In such a situation, you can configure...
transparent common channel signaling (T-CCS). With T-CCS, the router allows PBX signaling to flow through the router with no codec manipulation. In controller-configuration mode, you can specify that a T1 is carrying signaling, in the 24th channel, from an external source with the following command:

```
Router(config-controller)#ds0-group 1 timeslots 24 type ext-sig
```

In the next section, you learn about the concept of a dial peer, which tells a Cisco router how to send voice packets that are destined for a specific phone number. However, for the sake of completion, a dial-peer configuration-mode command is introduced that is required for the T-CCS configuration. In dial-peer configuration mode, you need to tell the router to transmit the specified signaling channel with no manipulation (for example, from a codec) with the following command:

```
Router(config-dial-peer)#codec clear-channel
```

For verification and troubleshooting purposes, consider the following commands:

- `show voice port`—Displays detailed settings for the voice ports
- `show voice port summary`—Displays a concise view of the installed voice ports
- `show voice dsp`—Displays the codecs that are currently being supported by the router’s digital signal processors (DSPs)
- `show controller {T1 | E1}`—Displays the operating parameters for T1 and E1 controllers
- `show isdn status`—Displays Layer 1, 2, and 3 information for an ISDN interface

Various test commands are also available for troubleshooting purposes. For example, you can force a voice port to send ringing voltage with the following command:

```
Router#test voice port 1/1 relay ring on
```

You also can play a tone out to an attached phone with the following command:

```
Router#test voice port 1/1 inject-tone local 500hz
```

Note that other frequencies can be specified. You can even cause a gateway to dial a number with the `csim start number` command. For example, you could enter the `csim start 5551212` command to cause the router to place a call to a destination phone number of 555-1212. This command is “hidden,” meaning that it does not appear in the IOS’s context-sensitive help. Cisco also warns that this command occasionally can cause a router to crash. Therefore, use this command with caution.
Configuring Router Dial Peers

Introduction
At this point in the Quick Reference Sheets, you have seen how to configure voice ports on Cisco voice-enabled routers. However, you have not yet trained the routers to reach specific destinations. That is the focus of this section. Specifically, you are going to create dial peers that inform the routers how to reach specific phone numbers. Consider the following topology.

Routers R1 and R2 each have a plain old telephone service (POTS) dial peer that points to their locally attached phone and a VoIP dial peer that points to the IP address of the remote router.

Therefore, when extension 1111 dials extension 2222, router R1 searches for a dial peer that matches a destination pattern of 2222. In this case, R1 has a VoIP dial peer that points to R2’s IP address of 10.1.1.2. R1 then forwards the call to R2. Then, R2 receives the incoming call that is destined for extension 2222, and it finds a POTS dial peer that specifies FXS port 1/1/1. The FXS port then sends ringing voltage out port 1/1/1. Extension 2222 goes off-hook, and the end-to-end call is complete.

Notice that you have a total of four dial peers that allow a call in the opposite direction. Also, notice that four stages of the call (that is, “call legs”) are defined, two call legs from the perspective of each router, as follows:

- **Call Leg #1**—The call comes in to R1 on FXS port 1/1/1.
- **Call Leg #2**—The call is sent from R1 to IP address 10.1.1.2.
- **Call Leg #3**—R2 receives an incoming call that is destined for extension 2222.
- **Call Leg #4**—R2 forwards the call out FXS port 1/1/1.
POTS Dial Peers
When configuring a POTS dial peer, specify the following two parameters:
- The destination-pattern (that is, the phone number)
- The physical port address

Consider the following POTS dial-peer configuration:

```
R1(config)#dial-peer voice 1111 pots
R1(config-dial-peer)#destination-pattern 1111
R1(config-dial-peer)#port 1/1/1
```

In this example, an analog phone is attached to FXS port 1/1/1. You entered dial-peer configuration mode for a POTS dial peer with the `dial-peer voice 1111 pots` command. Notice that the 1111 in the `dial-peer` command does not need to match the phone number. The number is merely a locally significant tag. However, to make the configuration more intuitive to interpret, you might want to adopt a practice of using the extension number as the dial-peer tag. The phone’s extension number of 1111 is specified with the `destination-pattern 1111` command. The phone’s physical location is specified with the `port 1/1/1` command.

VoIP Dial Peers
When configuring a VoIP dial peer, you specify a remote phone number with the same `destination-pattern` command that was used in a POTS dial peer. However, instead of identifying a local port, a VoIP dial peer specifies the voice packets’ destination IP address. Consider the following VoIP dial-peer configuration:

```
R1(config)#dial-peer voice 2222 voip
R1(config-dial-peer)#destination-pattern 2222
R1(config-dial-peer)#session target ipv4:10.1.1.2
```

A VoIP dial peer associates a phone number with an IP address. In this instance, a phone number of 2222 is associated with an IP address of 10.1.1.2.

```
R1(config)#dial-peer voice 2222 voip
R1(config-dial-peer)#destination-pattern 2222
R1(config-dial-peer)#session target ipv4:10.1.1.2
```
In this example, router R1 is sending voice packets that are destined for extension 2222 across the IP WAN to IP address 10.1.1.2, as specified with the `session target` command. Note that you must prepend `ipv4:` to the IP address.

At this point, you understand how to establish a dial plan for two phones that are separated by an IP WAN. However, what if you had 1000 extensions on the other side of the WAN? You certainly would not want to create 1000 dial peers. Fortunately, the Cisco IOS allows you to use wildcards to specify a range of addresses. A few of the more frequently used wildcards are as follows:

- **T**—A T represents a dial string of any length. For example, you could specify a destination pattern of 9T to match any number that begins with a 9, and then you could point any calls matching that pattern out to the PSTN. Because the T can be any number of digits, the router forwards the call, by default, after no digits have been dialed for 10 seconds (by default) or after the caller presses the # key.
- **.**—A period indicates any single digit. For example, you could specify extension numbers in the range 7000–7999 with the 7... destination pattern.
- **[]**—Brackets can specify a range of numbers. For example, a 123[4-6] destination pattern identifies a four-digit number of 1234, 1235, or 1236.

At this point, you have learned about using the `destination-pattern` and `port` commands to match dial peers.

However, a couple of additional options exist for matching inbound dial peers, `incoming called-number` and `answer-address`. To illustrate, consider that all calls destined for a call center are sent out the same trunk, but when the calls reach the call center, they might need to go to different locations (for example, to different customer service agents). You can configure this type of behavior with the `incoming called-number` command.

You can use the `answer-address` command instead of the `destination-pattern` command. For example, if you operate a call center that supports international customers, you can use the `answer-address` command.
number command to match a calling number, using caller ID information, to direct the call to an appropriate customer service agent. You can have a configuration in which multiple destination patterns match a dialed number. The logical question is “Which destination pattern do you use?” The “most specific” destination pattern is used when multiple destination patterns match a called number. Consider the following configuration:

Router(config)#dial-peer voice 1 voip
Router(config-dial-peer)#destination-pattern 2468
Router(config-dial-peer)#session target ipv4:192.168.1.1
Router(config-dial-peer)#dial-peer voice 2 voip
Router(config-dial-peer)#destination-pattern 2...
Router(config-dial-peer)#session target ipv4:192.168.2.2
Router(config)#dial-peer voice 3 voip
Router(config-dial-peer)#destination-pattern 2T
Router(config-dial-peer)#session target ipv4:192.168.3.3

In this configuration, a dialed number of 2468 is matched by all three dial peers. However, dial peer 1 is the most specific dial peer, matching the pattern exactly without the use of wildcards used to forward the call.

You might want to have multiple dial peers that match specific numbers, for redundancy reasons. For example, you might want to send calls that are destined for extension 4444 across the IP WAN. However, if the WAN is not available, you want to place the call through the PSTN. To accomplish this, you can have two dial peers: a VoIP dial peer that points across the IP WAN and a POTS dial peer that points to a physical port attached to a PBX or the PSTN. You then can indicate that you prefer to use the VoIP dial peer by assigning “preference” values to the dial peers with the preference number command.

For example, in dial-peer configuration mode for the VoIP dial peer, you could enter the preference 0 command. Similarly, you could use the preference 1 command for the POTS dial peer. Because lower preference values are preferred, the VoIP dial peer is used, if it is available. If the VoIP dial peer is not available, the call would fail over to the POTS dial peer. Note that the valid range of preference values is 0–10.

Digit Forwarding

When the router is connected to a PBX or the PSTN, you need to understand how the router forwards digits. By default, digits that are matched explicitly by the destination-pattern command in a POTS dial peer are not forwarded. For example, consider a destination pattern of 123..., which matches the dialed digits 1235555. By default, the router forwards the digits 5555 only to the attached equipment, because explicitly matched digits are stripped. To prevent this behavior, you can use the no digit-strip command in dial-peer configuration mode.

In addition to forwarding dialed digits, you also might want to forward additional digits. For example, you might want to prepend a dialed number with a 9 before forwarding the number to a PBX. You can prepend digits to a dialed number with the prefix number command.
As opposed to using the `no digit-strip` command, you can specify exactly how many digits to forward by using the `forward-digits` number command. For example, you could use the dial-peer configuration-mode command `forward-digits 5` to forward the five right-most digits in the dialed number.

You can even instruct the router to replace one dialed number with another. For example, you might have a telecommuter whose home phone number is 555-1234. However, all employees at the headquarters location have four-digit extension numbers. To make the telecommuter’s phone appear as a local extension, you can use the `num-exp` command. In this example, if you wanted to represent the telecommuter’s home phone with a four-digit extension number of 2020, you could enter the `num-exp 2020 5551234` command in global-configuration mode.

**Connection Types**

Recall the operation of a PLAR connection, as described in a previous section, “Introduction.” A PLAR connection automatically dials a predetermined number when a phone goes off-hook. In the following example, when extension 1111 goes off-hook, router R1 automatically dials extension 2222.

With a Private Line Automatic Ringdown (PLAR) call, when a phone goes off-hook, the attached router automatically places a call to a preconfigured destination. In this instance, when 1111 goes off-hook, router R1 automatically places a call to 2222.

```
R1(config)#dial-peer voice 2222 voip
R1(config-dial-peer)#destination-pattern 2222
R1(config-dial-peer)#session target ipv4:10.1.1.2
R1(config-dial-peer)#voice-port 1/1/1
R1(config-voiceport)#connection plar 2222
```

PLAR Off-Premise Extension (PLAR-OPX) is similar to PLAR. However, with PLAR-OPX, instead of a router answering an incoming call from a PBX—in which case the PBX would consider the call complete—the call is not answered until the destination phone goes off-hook. As a result, an unanswered call can be rerouted to voice mail by the PBX.

In the following PLAR-OPX example, when extension 1111 dials extension 2222, the PBX forwards the call to router R1. Router R1 then places a call to extension 2222. However, because of the PLAR-OPX configuration, R1 (and therefore the PBX) does not consider the call complete until extension 2222 goes off-hook.

```
PLAR

V

FXS
1/1/1

10.1.1.1

R1

R2

10.1.1.2

FXS
1/1/1

With a Private Line Automatic Ringdown (PLAR) call, when a phone goes off-hook, the attached router automatically places a call to a preconfigured destination. In this instance, when x1111 goes off-hook, router R1 automatically places a call to x2222.

R1(config)#dial-peer voice 2222 voip
R1(config-dial-peer)#destination-pattern 2222
R1(config-dial-peer)#session target ipv4:10.1.1.2
R1(config-dial-peer)#voice-port 1/1/1
R1(config-voiceport)#connection plar 2222
```

```
PLAR-OPX

V

FXS
1/1/1

10.1.1.1

R1

R2

10.1.1.2

FXS
1/1/1

With a Private Line Automatic Ringdown (PLAR-OPX) call, when a phone goes off-hook, the attached router automatically places a call to a preconfigured destination. In this instance, when x1111 goes off-hook, router R1 automatically places a call to x2222.

R1(config)#dial-peer voice 2222 voip
R1(config-dial-peer)#destination-pattern 2222
R1(config-dial-peer)#session target ipv4:10.1.1.2
R1(config-dial-peer)#voice-port 1/1/1
R1(config-voiceport)#connection plar 2222
```

© 2007 Cisco Systems Inc. All rights reserved. This publication is protected by copyright. Please see page 49 for more details.
The first three digits (that is, 859) indicate an area code, which is typically associated with a geographical location within North America. The following three digits (that is, 555) are the central office code (that is, the NXX code), which identifies a central office location within the area that is specified by the area code. The final four digits (that is, 1212) point the local central office to a specific local loop that goes out to a subscriber’s physical location.

When designing a dial plan for a customer, consider the following items:

- Where is the dial plan logic located (for example, in a gateway [GW] or in a CCM)?
- Will digit translation be required?
- Can the numbers in the dial plan be reached by multiple paths, for fault-tolerant purposes?
- Should you use digit translations to give dial plans a consistent feel?
- Does an attached voice-mail system require a different number of digits than an extension?
- When do area codes need to be dialed?
- How can the dial plan support different countries, which can have country codes of varying lengths?

With a Private Line Automatic Ringdown Off-Premise Extension (PLAR-OPX) call, a router connected to a PBX does not indicate to the PBX that the call is complete until the destination phone goes off-hook. As a result, the PBX might be able to reroute the call to another location, such as to a voice mail system. In this instance, when x1111 dials x2222, router R1 does not indicate to the PBX that the call is complete until x2222 goes off-hook.

Router(config)#dial-peer voice 2222 voip
Router(config-dial-peer)#destination-pattern 2222
Router(config-dial-peer)#session target ipv4:10.1.1.2
Router(config-dial-peer)#voice-port 1/1/1
Router(config-voiceport)#connection plar-opx 2222

Dial Plans
Dial plans organize a group of phone numbers in a hierarchical fashion. Consider the North American dialing plan, which consists of ten digits, as follows:

859-555-1212
Voice over IP Design Considerations

Introduction
In the previous section, you learned about the configuration of a basic VoIP network. However, additional considerations are required. In this section, you examine various VoIP design considerations and are introduced to two additional voice-over-data technologies: Voice over Frame Relay (VoFR) and Voice over ATM (VoATM).

RTP
Consider the characteristics of IP networks. They are considered to be “connectionless,” meaning that the IP protocol by itself does not provide guaranteed packet delivery. Also, thanks to various routing protocols, IP networks can support multiple paths between a source and destination; this provides load balancing and fault tolerance.

Voice packets use a Layer 4 protocol, called the Real-Time Transport Protocol (RTP). RTP is encapsulated within the User Datagram Protocol (UDP), another Layer 4 protocol that is connectionless. Because these RTP packets travel over an IP network, with multiple paths, the packets can arrive at different times. Fortunately, RTP provides sequence number information so that the far side of a VoIP call can reorder voice packets.

However, other quality issues must be addressed on a VoIP network. The three main quality challenges are as follows:

- **Delay**—Delay is the time that is required for a packet to travel from its source to its destination. You might have witnessed delay on the evening news, when the news anchor is talking through a satellite to a foreign news correspondent. Because of the satellite delay, the conversation might have seemed unnatural.

- **Jitter**—Jitter is the uneven arrival of packets. For example, consider that in a Voice over IP (VoIP) conversation, packet 1 arrives. Then, 20 ms later, packet 2 arrives. After another 70 ms, packet 3 arrives, and then packet 4 arrives 20 ms behind packet 3. This variation in arrival times (that is, variable delay) is not dropping packets, but this jitter can be interpreted by the listener as dropped packets.

- **Drops**—Packet drops occur when a link is congested and a buffer overflows. Some types of traffic, such as UDP traffic (for example, voice), are not retransmitted if packets are dropped.

In the section “Ensuring Voice Quality,” later in these Quick Reference Sheets, a variety of the Cisco quality of service (QoS) solutions address these quality issues.
VoFR and VoATM

In addition to Voice over IP, you can encounter a Voice over Frame Relay (VoFR) or Voice over ATM (VoATM) network. Whereas VoIP networks forward voice traffic based on Layer 3 information (that is, an IP address), VoFR and VoATM networks forward voice traffic at Layer 2. Specifically, instead of having the session target command specify a remote IP address, VoFR uses the session target command to point to a local Data Link Connection Identifier (DLCI). In a similar fashion, VoATM uses the session target command to forward voice traffic to a local virtual path identifier/virtual channel identifier (VPI/VCI).

Although forwarding voice traffic at Layer 2 offers the advantage of eliminating Layer 3 overhead (that is, IP header, UDP header, and RTP header), VoFR and VoATM networks do not offer the ability to load-share across multiple paths, as VoIP networks do. Also, you can lose some redundancy in your design, because VoFR and VoATM do not use multiple paths.

Although bandwidth requirements are due in large part to the codec that you select, bandwidth demands also vary based on the transport (for example, FR or ATM) that you use. You can use the following formula to calculate per-call bandwidth requirements for a VoFR circuit:

\[ \text{Per\_Call\_Bandwidth} = \frac{\text{Codec\_Bandwidth} \times (\text{Payload} + \text{Overhead})}{\text{Payload}} \]

The units of measure for Codec_Bandwidth are bps, whereas the Payload and Overhead parameters are measured in bytes. Note that the default payload size that is used by G.729 on a VoFR network is 30 bytes.

Each asynchronous Transfer Mode (ATM) cell is 53 bytes in size: 48 bytes of payload and 5 bytes of header. If your voice payload does not consume all 48 payload bytes in a cell, the remainder of those 48 bytes is filled with “padding,” which can lead to inefficient bandwidth usage. The following formula is used to calculate the per-call bandwidth requirements for a VoATM circuit:

\[
\text{Per\_Call\_Bandwidth} = \text{Codec\_Bandwidth} \times (\text{Number\_of\_Cells} \times 53) / \text{Payload}
\]

Note that whereas the G.711 codec requires six ATM cells to send a single voice sample, the G.729 codec requires only one cell per voice sample.

**Gateway Considerations**

When selecting a gateway (GW) for a VoIP network, consider the following items:

- Do you need an analog or digital gateway?
- What specific ports are required on the gateway?
- What signaling protocol does the gateway need to support?
- Does the gateway need to support Survivable Remote Site Telephony (SRST)?

**Voice Protocols**

Gateway protocols that you should be familiar with include H.323, the Media Gateway Control Protocol (MGCP), and the Session Initiation Protocol (SIP). Session protocols map to Layer 5 (that is, the session layer) of the OSI model. The session layer is responsible for the setup and teardown of sessions. In the voice environment, you can use these gateway control protocols to set up and tear down voice sessions. In the next section, you learn about each of these protocols.

Voice packets are transmitted using the Real-time Transport Protocol (RTP). A companion protocol, called the Real-Time Transport Control Protocol (RTCP), can monitor the quality of an RTP stream. Both RTP and RTCP operate at Layer 4 and are encapsulated in UDP. UDP ports 16,384 to 32,767 are used by RTP and RTCP. However, RTP uses the even port numbers in that range, whereas RTCP uses the odd port numbers.

One of the challenges with RTP is its overhead. Specifically, the combined IP, UDP, and RTP headers are approximately 40 bytes in size, and the typical voice payload size on a VoIP network is only 20 bytes, which includes 20 ms of voice by default. In this case, the header is twice the size of the payload. Fortunately, Cisco supports RTP Header Compression (cRTP), which can reduce the 40-byte header to 2 or 4 bytes in size (depending on whether UDP checksums are in use).
Because cRTP requires router processor resources, you should not enable it in every instance. Specifically, Cisco recommends that you enable cRTP on slow link speeds (that is, speeds of less than 2 Mbps) or when you need to optimize your bandwidth usage on a WAN interface.

### Bandwidth Considerations

In the section “Digitizing the Spoken Voice,” earlier in these Quick Reference Sheets, you reviewed the codec bandwidth requirements for the G.711 and G.729 codecs, as follows:

- **G.711** (64 kbps of bandwidth required for voice payload)
- **G.729** (8 kbps of bandwidth required for voice payload)

However, other factors impact the overall bandwidth requirements for voice traffic. For example, the decision of whether to use cRTP can impact the overall bandwidth requirements dramatically. Layer 2 header sizes can vary also. For example, whereas Ethernet has an 18-byte Layer 2 header, both Multilink PPP (MLP) and Frame Relay require only 6 bytes for their Layer 2 header.

In a tunneling environment, you might need to consider the overhead of such protocols as IP Security (IPSec) (50- to 57-byte overhead) and Layer 2 Tunneling Protocol/generic Routing encapsulation (L2TP/GRE) (24-byte overhead). Also, in a service provider environment, you can use Multiprotocol Label Switching (MPLS), which has 4 bytes of overhead.

When you read about the G.729b codec, the concept of voice activity detection (VAD) was introduced. When you have multiple simultaneous conversations (that is, approximately 24 or more), you can benefit from the economies of scale that are offered by VAD to reduce the overall bandwidth consumption. Although you should not use this number for design purposes because VAD’s benefit varies with speech patterns, you can enjoy bandwidth savings on the order of a 35 percent reduction when using VAD.

### Security Considerations

As another design consideration, you must ensure that your VoIP solution can function in the presence of an enterprise’s existing security infrastructure. The Cisco model for security design is called SAFE (Secure Architecture For Enterprise). The main goal of the SAFE blueprint is to provide best-practice information for designing and implementing secure networks. Specifically, the SAFE blueprint seeks to detect and defend against intrusions into the network, to control which traffic can enter and leave the network, and to protect data as it travels through the network. For more information on the Cisco SAFE blueprint, consult [http://www.cisco.com/go/safe](http://www.cisco.com/go/safe).

© 2007 Cisco Systems Inc. All rights reserved. This publication is protected by copyright. Please see page 49 for more details.
An enterprise network might already have a firewall and a Virtual Private Network (VPN) concentrator in place when you decide to overlay voice traffic on that network. A firewall maintains a set of rules that specify which protocols you can use to communicate from specific outside devices to specific inside devices, and vice versa. The challenge with establishing a voice call through a firewall is that the UDP port that is negotiated for the voice stream varies from session to session. Specifically, voice streams can use any even-numbered UDP port in the range 16,384 to 32,767. Therefore, a “stateful” firewall is required to monitor this negotiation and to open the appropriate port. The Cisco PIX firewall product is an example of a stateful firewall.

Various VPN mechanisms exist. For example, you can use Layer 2 Tunneling Protocol (L2TP) at Layer 2 or IPSec at Layer 3. Because various VPN technologies have different header sizes, your overall voice bandwidth requirements vary with the VPN technology that you select. However, as a rule, you typically add 30 to 60 bytes of overhead per VoIP packet when you transmit the packet over a VPN.

One way to protect voice packets from eavesdropping is to send them through a VPN connection. A VPN is a tunnel over which you can send encrypted packets. As a result, if the voice packets were intercepted, they would be worthless to anyone who intercepted the packets, because the packets are encrypted.

© 2007 Cisco Systems Inc. All rights reserved. This publication is protected by copyright. Please see page 49 for more details.
Gateway Protocols

Introduction
In this section, you learn about the theory and configuration of the following gateway control protocols: H.323, SIP, and MGCP. The discussion begins with a generic overview of gateway protocols.

Gateway Protocol Theory
The ultimate goal of call control is to allow RTP streams to flow directly between endpoints. Therefore, during the call setup process, each endpoint (for example, IP phone) needs to learn the IP address and UDP port to use to get a phone call to the other end.

In addition to setting up a call, you might want to perform call administration and accounting features. These features can keep track of bandwidth usage on the WAN and maintain call records, which you can use for billing or planning purposes. You also might need a way to obtain status information about a current call.

Although your local gateway might know how to reach local phones, the gateway might need to contact an external database of addresses to resolve the location of a remote phone. This external database can learn about remote phones by having the gateway that is used by those remote phones register those phone numbers with the database. By having this central repository of phone number-to-IP address mappings, less configuration needs to be performed on each of the local gateways.

In the Cisco IP telephony environment, the Cisco CallManager (CCM) acts as a database that can direct a gateway (for example, a Cisco voice-enabled router) to a remote gateway that is connected to the destination phone. If all the CCMs are in a single location, that deployment model is called centralized call control.

Centralized Deployment Model

With a centralized deployment model, a CCM cluster is in a central location, and remote IP Phones register with the centralized CCM cluster over the IP WAN. SRST allows IP Phones at remote sites to function in the event of a WAN failure.
However, if CCMs are scattered across all your remote office locations, that deployment model is referred to as distributed call control. You can incur additional costs with a distributed call control model, because of the extra equipment and administration required. However, you enjoy greater scalability with the distributed call control model.

Consider the following pros and cons of each approach:

- **Centralized call control**
  - Pros:
    - Less expensive
    - Easier to maintain
  - Cons:
    - Single point of failure
    - Less scalable
    - Uses WAN bandwidth for call setup

- **Distributed call control**
  - Pros:
    - More fault tolerant
    - Does not consume WAN bandwidth for call setup
    - More scalable
  - Cons:
    - Requires additional equipment
    - Requires more work to perform updates
**H.323 Theory and Configuration**

H.323 is not a single protocol. Rather, it is a suite of protocols. This suite specifies such things as audio/video codecs and call signaling.

You should be aware of the following two major protocols:

- **H.225**—Performs call setup and Registration, Admission, and Status (RAS) functions
- **H.245**—Performs call control, including a capabilities exchange

In addition to various protocols, H.323 identifies the physical components of an IP telephony network, as follows:

- **Terminals**—An H.323 terminal is an end-user device that communicates with another end-user device. By definition, an H.323 terminal must support the G.711 codec.
- **Gateways**—A gateway (GW) converts between different audio formats/signaling. For example, on one side of the GW, you might use H.225 and H.245 signaling, and on the other side of the GW, you might use E&M wink-start signaling.
- **IP-to-IP gateways**—These gateways allow different VoIP networks to interconnect. For example, if two companies had their own internal VoIP network with their own service provider, an IP-to-IP gateway could allow the two companies to join their VoIP networks. This solution requires no extension-number overlapping between the two companies.

- **Gatekeepers**—To prevent voice calls from oversubscribing the WAN bandwidth, groups of routers, called *zones*, can be identified. Then, a gatekeeper (GK) can keep track of the number of calls made from one zone to another, or calls made within a zone. Based on the available bandwidth within or between zones, a GK can permit or deny a call attempt. The gatekeeper also can perform centralized E.164 number resolution.
- **Multipoint Control Units (MCUs)**—An MCU supports conference calling by adding and removing participants from a call and by mixing voice streams together.

Gatekeepers are optional components, because you can have an H.323 GW directly communicate with another H.323 GW. However, this approach has scalability limitations. If you introduce GKs into the network, GWs can communicate with GKs using the RAS channel. In larger topologies, you can have multiple GKs, and those GKs communicate with each other using the RAS channel.

Consider how a call is completed in the following H.323 networks:

- **GW-to-GW calls**—This topology does not require a GK. Specifically, both GWs communicate directly with each other. First, H.225 performs the call setup, followed by H.245 performing a capabilities exchange. However, this negotiation requires numerous packet exchanges between the GWs. Another option is to use H.323 Fast Connect, which performs call setup and does a capabilities exchange in a single exchange of messages between the two GWs.
GW-to-GK-to-GW calls—With a GK in your topology, the originating GW requests permission from the GK to place a call using an admission request (ARQ) message, after which the GK can send an admission confirm (ACF) or admission reject (ARJ) message. If permission is granted, the call setup proceeds. The destination GW also sends an ARQ to the GK. If permission is granted, the call setup proceeds as usual, using H.225 and H.245, after which RTP is used to stream audio directly between the GWs.

For even larger environments, you can have multiple GKs involved in the call setup. The main difference with such a configuration is that when the first GK gets an admission request, it sends a location request (LRQ) and must receive a location confirm (LCF) from the remote GK before sending an admission confirm (ACF) to the originating GW.

To increase the availability of H.323 networks, you can configure multiple GKs and GWs to service the same phone numbers. You also can use high-availability technologies such as Hot Standby Router Protocol (HSRP) to maintain uptime in an H.323 network.
Numerous options are available to you when configuring H.323 GWs and GKs. For a thorough example, visit http://www.cisco.com/warp/public/788/voip/zone_gw_gk.pdf. However, use the following three basic steps to configure an H.323 GW:

1. Enable the GW using the `gateway` command in global-configuration mode.
2. Configure the relationship between the GW and the GK with the `h323-gateway voip id` command in interface-configuration mode.
3. Configure dial peers to point to the GK, with the `session target ras` command.

You can verify and troubleshoot an H.323 configuration with various `show` commands, as follows:

- `show gatekeeper calls` — Displays current phone calls that the GK participated in
- `show call active voice [brief]` — Displays details for current voice calls
- `show call history voice [last n | record | brief]` — Displays call record logs

**SIP Theory and Configuration**

Session Initiation Protocol (SIP) uses the concept of inviting participants into sessions, and those sessions can be advertised by Session Announcement Protocol (SAP). Like H.323, SIP is a peer-to-peer protocol. These peers are called User Agents (UAs). The two types of UAs are as follows:

- **User Agent Clients (UACs)** — Initiate the connection by sending an INVITE message
- **User Agent Servers (UASs)** — Reply to INVITE messages

SIP also leverages the following types of SIP servers:

- **Proxy server** — Performs forwarding for a UAC
- **Registrar server** — Registers the location of current clients
- **Redirect server** — Informs the UA of the next server to contact
- **Location server** — Performs address resolution for SIP proxy and redirect servers

SIP uses clear text for sending messages; the two types of SIP messages are as follows:

- **Request** — A message from a client to a server
- **Response** — A message from a server to a client

A request includes messages such as an INVITE (which requests a participant to join the session) or a BYE (which disconnects the current call). Conversely, a response message uses HTML status messages. For example, you probably have attempted to connect to a website and...
received a “404 error” or a “500 error.” Those same types of responses are used in the SIP environment.

For a SIP client to get the IP address of a SIP server, it has to resolve a SIP address. These addresses are actually URLs that begin with “sip:” as opposed to “http:” which is commonly used in web browsers. SIP addresses can include a variety of information, such as username, password, host name, IP address, and phone number. Following is an example of a SIP address:

    sip:18595551212@ciscopress.com;user=phone

In this example, the user=phone argument specifies that the user portion of the URL (that is, 18595551212) is a phone number and not a user ID.

SIP devices can make their addresses known dynamically by registering with a SIP registrar server. When SIP devices have their addresses registered, a SIP client can resolve a SIP address by itself, perhaps through DNS or through a local host table. However, a SIP client can use a SIP proxy server to query a SIP location database to resolve the SIP IP address. Consider a basic SIP call, where one SIP GW communicates directly with another SIP GW, without the use of proxy or redirect servers.

The basic call setup begins when a SIP client sends an INVITE to a SIP server (noting that a SIP IP phone can act as either a client or server, depending on whether it is originating or terminating the call). The destination server (that is, a UAS) responds if it is willing to join the session to which it has been invited. The originating client (that is, a UAC) sends an acknowledgment (that is, an ACK message) to the destination server, and at this point, the RTP streams can flow directly between the SIP GWs.

If you introduce a SIP proxy server into your topology, the call setup procedure is similar to that just discussed. However, the INVITE is sent to the proxy server rather than the destination UAS. The proxy server can consult a location server to learn the IP address of the final endpoint. The destination exchanges call parameters with the proxy server, which responds to the originating UAC. The UAC then forwards an ACK through the proxy server to the destination UAS, after which the RTP stream is established.
When you use a redirect server, the originating UAC sends an INVITE message to a redirect server, which can consult a location server to determine the path to the destination. The registrar server responds to the UAC with a moved message, telling the UAC the IP address of the destination UAS. This operation is much like when you connect to a website and you receive a message saying that the page you are looking for has moved to a new URL; then you are redirected automatically to the new URL. When the UAC learns the location of the destination UAS, a direct connection can be set up between the UAC and UAS. Therefore, the main purpose of a redirect server is essentially to off-load the IP resolution procedure from the UAC.

If one of your SIP servers goes down, the voice network could be rendered unavailable. One way to provide redundancy is to have multiple instances of proxy and redirect servers. Therefore, the UAs can have multiple entries, and if the first server fails, the second server takes over.

Use the following two basic steps to configure SIP on a Cisco router:

1. Enable the UA.
2. Configure dial peers.

Consider the following example:

```
Router(config)#sip-ua
Router(config-sip-ua)#sip-server dns:SERVER1
Router(config-sip-ua)#dial-peer voice 1 voip
```

In this example, you enable the SIP UA with the `sip-ua` global-configuration-mode command. Then, you specify the DNS name of your SIP server with the `sip-server` command. When the SIP server has been defined, in VoIP dial-peer configuration mode, you specify that you want the dial peer to use SIP version 2, with the `session protocol sipv2` command. Finally, instead of specifying an IP address as the session target, you use the `session target sip-server` command, which points your dial peer to the SIP server that is defined under sip-ua configuration mode.

In addition to the `show call` commands that you used for H.323 verification, you can also use the `show sip-ua statistics` command to display three different sets of statistics (that is, SIP response statistics, SIP total traffic statistics, and retry statistics).

**MGCP Theory and Configuration**

Whereas H.323 and SIP were peer-to-peer protocols, MGCP is more of a client-server approach to call control. Specifically, MGCP allows GWs to point to a centralized call agent for processing. In a Cisco environment, this centralized call agent is the Cisco CallManager (CCM).
The physical pieces that make up an MGCP network, such as call agents, gateways, and endpoints, are called **components**. However, the logical pieces of an MGCP network, such as calls and connections, are called **concepts**. First, consider the following MGCP components:

- **Endpoints**—An endpoint is where you interface between the VoIP network and the traditional telephony network. For example, an FXS port that connects to a telephone would be considered an endpoint. Endpoint names look much like an e-mail address (for example, circuitID@mgcpgw.ciscopress.com). These names are composed of two parts: the locally significant name of the endpoint (before the @ sign) and the DNS name of the MGCP GW (after the @ sign).

- **Gateways**—Gateways are in charge of converting audio between a VoIP network and a circuit-switched network. For example, a residential gateway supports devices that you typically find in residential environments (for example, POTS telephones).

- **Call agents**—A call agent is the intelligence of an MGCP network and controls the gateways and their endpoints. An MGCP gateway can report events to the call agent, and the call agent can, for example, tell the endpoint what type of signaling to send to the phone.

Recall that an MGCP concept is a logical piece of an MGCP network. Consider the following MGCP concepts:

- **Call**—A call is formed when two or more endpoints are interconnected.

- **Events**—An event is what an endpoint has been instructed (by the call agent) to watch for. As an example, an FXS port might notice the event of an attached POTS device going off-hook.

- **Signal**—A call agent instructs an endpoint to send a specific signal when a certain event occurs. For example, after the event of a POTS phone going off-hook, the call agent might instruct the FXS endpoint to send the signal of “dial tone” to the phone.

MGCP groups relevant events and signals into **packages**. For example, a line package contains events and signals that are used on subscriber lines, such as an off-hook event and a dial-tone signal. MGCP GW
Gateway Protocols

types are defined by the types of packages that they support. For example, a trunk gateway supports the generic media, DTMF, trunk, and RTP packages.

You do not have to configure an MGCP GW with dial peers for every destination phone number. Instead, the GW can send each dialed digit to the call agent, until a match is made. However, that approach can put a heavy burden on the call agent. An alternate approach is for the gateway to download a digit map, which is essentially a copy of the dial plan that is contained in the call agent. When a GW has a digit map, it then has the responsibility of collecting all the dialed digits and finding a pattern match.

To enhance the availability of an MGCP network, survivability strategies such as “MGCP switchover and switchback” and “MGCP gateway fallback” are supported. MGCP switchover and switchback uses two or more CCMs. When a GW sees no MGCP messages from a call agent for a period of time, the GW attempts to establish a connection with a backup CCM. If the primary CCM comes back up, the GW can be configured to switch back to the primary.

MGCP gateway fallback works with SRST to maintain a remote office that connects to a centralized CCM. If the WAN link, which the GW uses to reach the CCM, fails, the GW relies on a fallback H.323 configuration that provides basic functionality for attached IP phones.

MGCP is enabled in global-configuration mode with the mgcp command. The IP address of the call agent then is specified with the mgcp call-agent ip_address command. When MGCP is enabled, you specify which packages the GW expects the call agent to use when communicating with the GW. Consider the following configuration example, where you are specifying that a router is acting as an MGCP GW and that its call agent is located at IP address 10.1.1.1:

Router(config)#mgcp
Router(config)#mgcp call-agent 10.1.1.1

A POTS dial peer that MGCP uses does not have a destination-pattern command. Rather, it uses the application MGCPAPP command to indicate that the port specified by the dial peer is controlled by an MGCP call agent. To view information about current MGCP connections on a GW, you can use the show mgcp connection command.
In a Cisco environment, because an MGCP gateway works in conjunction with a Cisco CallManager (CCM), the following items need to be configured on the CCM:

- From the CCM main screen, choose the Device menu and select Gateway.
- Click the Add a New Gateway link.
- Select the Gateway Type from the drop-down menu, and click Next. (Note that by selecting the appropriate hardware, the CCM automatically determines that you are using MGCP.)
- In the Domain Name field, enter the DNS name of the gateway router.
- Select the appropriate CCM group from the Cisco CallManager Group menu.
- From the Module in Slot 1 drop-down menu, select the appropriate installed module, and click the Insert button.
- After the screen refreshes, you see one or more “subunits” listed below the module that you specified. Select the appropriate subunit(s) from the Subunit drop-down menu(s), and click the Update button.
- After the screen refreshes, you see one or more “endpoint identifiers” that correspond to physical voice ports on the router. Click an Endpoint Identifier link.
- Select the appropriate Device Pool from the drop-down menu, and click the Insert button.

For an FXS port, after the screen refreshes, click the Add DN link to add a directory number. Enter the directory number (that is, phone number) for the voice port in the Directory Number field, and click the Insert button.

At this point, the Cisco MGCP gateway and the CCM (acting as the MGCP call agent) can communicate between themselves using MGCP signaling.

Selecting a Gateway Protocol

When selecting from the three call control protocols (that is, H.323, SIP, and MGCP), consider the following characteristics:

- **H.323**
  - Mature
  - Uses Abstract Syntax Notation 1 (ASN.1) for call control messages
  - Uses a peer-to-peer model
  - Scales well in an enterprise

- **SIP**
  - Not as mature as H.323
  - Uses clear text for call control
  - Uses a peer-to-peer model
  - Allows interoperability between diverse vendor equipment
Ensuring Voice Quality

Introduction

In this section, you examine several quality of service (QoS) tools to help you preserve voice quality as voice packets traverse voice-enabled networks. Specifically, you make the following distinctions that affect voice quality:

- Selecting codecs
- Minimizing delay
- Provisioning sufficient bandwidth
- Overcoming jitter

However, you first learn to measure voice quality.

Challenges to Voice Quality

As mentioned earlier in these Quick Reference Sheets, voice quality can be impacted when voice packets are dropped or delayed. Also, variation in the delay (that is, jitter) affects the listener’s perception of the voice quality. The fidelity of a signal is a measure of the frequency range that is represented by a signal. However, in the voice environment, your main concern is transporting the majority of the frequencies that make up speech patterns, rather than reproducing high-fidelity music, as you...
would on a home stereo system. Therefore, the G.711 codec samples frequencies up to 4000 Hz, which contains more than 90 percent of the frequencies that are used in human speech patterns.

Earlier, you learned how to measure voice quality using mean opinion score (MOS) or Perceptual Speech Quality Measurement (PSQM). The choice of codecs impacts the score that is given to the voice quality. For example, the G.711 codec has an MOS of 4.1, whereas G.729 has an MOS of 3.92.

Another approach to measuring voice quality is Perceptual Evaluation of Speech Quality (PESQ). Like PSQM, PESQ uses test equipment to measure voice quality, as opposed to a “trained ear.” However, PESQ attempts to match MOS rankings. For example, if a codec had an MOS of 3.92, it should have a PESQ score of 3.92. However, the maximum value on the PESQ scale is 4.5, whereas the MOS scale reaches 5.0.

**Overcoming Delay**

Voice is sent over IP networks in RTP packets, which use UDP for transport. Because UDP is connectionless, the network does not attempt to retransmit a lost packet. Also, because IP networks can load-balance traffic across multiple links, packets can arrive with various interpacket intervals. These characteristics can lead to variable delay, which is called **jitter**. Another example of variable delay is queuing delay, which your queuing strategy can introduce. A queuing strategy determines the order that packets are transmitted out of a queue in the presence of congestion.

Fortunately, the Cisco voice-enabled routers have a “jitter buffer,” which stores packets as they come into a router and then plays the packets out in a smooth fashion. When packets are in the jitter buffer, you can reorder them based on RTP sequence numbers. Although the IOS determines the size of the jitter buffer is automatically, if jitter problems are not eliminated, you can manually configure the jitter buffer with the `playout-delay mode` command. Specifically, the dial-peer configuration-mode command `playout-delay mode adaptive` along with the `playout-delay maximum size` command can specify, in milliseconds, the maximum size of the jitter buffer, while allowing the jitter buffer to dynamically adjust its size based on network conditions. Alternatively, you can use the `playout-delay mode fixed` and the `playout-delay nominal size` dial-peer configuration-mode commands to specify statically the size of the jitter buffer. You can view current jitter conditions with the `show call active voice` command.
ENSURING VOICE QUALITY

Some delay components, however, are fixed, meaning they do not change during a call. The following are examples of fixed delay:

- **Coder delay**—The amount of time required for a codec to process a voice packet. (For example, a G.729 codec can require 2.5 to 10 ms to do the coding, depending on how many voice calls a DSP is coding. Similarly, the G.723 codec requires 5 to 20 ms for coding.)

- **Serialization delay**—The amount of time required for a packet to exit a serial interface. (For example, a 1500-byte frame requires 214 ms to exit a serial interface running at 56 kbps.)

- **Propagation delay**—The amount of time required for a packet to travel across the network.

Although serialization delay is considered a fixed delay, for the duration of a call, you can minimize the effect of serialization delay by performing Link Fragmentation and Interleaving (LFI). LFI fragments large data payloads and interleaves smaller voice packets among the fragments. On a Voice over IP over PPP circuit, the LFI mechanism that you use is multilink PPP (MLP). However, on a Voice over IP over Frame Relay (VoIPovFR) network, your LFI tool of choice is FRF.12.

The combined fixed and variable delay components create the delay budget for a design. However, too much delay adversely affects voice quality. As a design guideline, the ITU G.114 recommendation states that the one-way delay for a voice packet should not exceed 150 ms.

LAN Quality of Service

Normally, quality of service (QoS) technologies are related to the WAN. However, the LAN also needs QoS. For example, you might have multiple Fast Ethernet devices simultaneously communicating to a server that is also connected through Fast Ethernet. Such a situation could lead to “oversubscription” of the bandwidth. Therefore, frames that cannot be sent at the moment can be buffered by a Catalyst switch in a queue. If the queue overflows, all incoming packets are discarded, even high-priority voice packets.

However, if you place voice frames in a different queue from lower-priority data frames, even if the data queue overflows, the voice queue still accepts incoming voice frames.

After frames are queued, the switch’s queuing strategy determines in what order frames are emptied from the queues and how bandwidth is made available to the queues. Specifically, many Cisco Catalyst switches support Weighted Round Robin (WRR) queuing, which specifies the “weight” that is given to each queue.

As an example, consider that queue 1 has a weight of 1 and that queue 4 has a weight of 4. In this example, queue 4 receives four times the bandwidth of queue 1 because of its weight.

Catalyst switches such as the 2950 and the 3550 offer four queues where you can place frames. Other switch models can have a different number of queues. However, Cisco recommends a 2Q1T approach, which specifies that you are using two queues, where each of the...
queues has one threshold. Having a single threshold causes a queue to delay discarding frames until the queue fills to capacity.

As the name suggests, Differentiated Services differentiates among different types of traffic and provides different levels of service based on those distinctions. Instead of forcing every network device to classify traffic, DiffServ can mark packets with a particular priority marking that other network devices can reference.

A common type of packet marking is Differentiated Services Code Point (DSCP). DSCP uses the 6 left-most bits in an IPv4 header’s type of service (ToS) byte. With 6 bits at its disposal, up to 64 DSCP values (0 to 63) can be assigned to various classes of traffic.

Another type of marking that uses the IPv4 ToS byte is IP Precedence, which uses the 3 left-most bits in the ToS byte. Because it uses 3 bits, IP Precedence has values in the range 0 to 7. However, Cisco recommends that IP Precedence values of 6 or 7 never be used, because they are reserved for network and Internet use. Just marking a packet does not change its operation, unless QoS tools are enabled that can reference that marking. Fortunately, multiple QoS tools exist that can make decisions based on these markings.

WAN Quality of Service

Two broad categories of QoS tools exist: Integrated Services (IntServ) and Differentiated Services (DiffServ). Integrated Services provides QoS by guaranteeing treatment to a particular traffic flow. A commonly used IntServ tool is the Resource Reservation Protocol (RSVP).
You can place most QoS tools in one of the following areas:

- Classification and marking
- Congestion avoidance
- Congestion management
- Traffic conditioning
- Signaling
- Link efficiency mechanisms

**Classification and Marking**

When configuring QoS, decide which devices in the network you “trust” to make markings. These devices should be as close to the source as possible. For example, you can select a Cisco IP phone as the trust boundary.

When you decide on a trust boundary, configure your edge devices to classify traffic into classes. For example, you could have a Voice over IP (VoIP) class, a database class, an FTP class, a video class, and a default class.

When the traffic is categorized into traffic classes, mark the various traffic classes with DSCP markings. This prevents other devices in the network from having to reclassify the traffic. Instead, these other devices can reference the DSCP markings. However, as mentioned earlier, marking by itself does not alter the behavior of traffic.

**Congestion Avoidance**

The purpose of congestion avoidance is to prevent an interface’s output queue from filling to capacity, because if a queue is full, all newly arriving packets are discarded. Some of those packets can be high priority, and some can be low priority. However, if the queue is full, no room exists for a packet.

With a congestion-avoidance tool, drop thresholds are defined for various markings (for example, DSCP markings). Therefore, as a queue begins to fill, lower-priority packets are dropped more aggressively than higher-priority packets, thus preventing the queue from ever filling to capacity. The Cisco congestion-avoidance tool of choice is Weighted Random Early Detection (WRED).

Inside an IPv4 header is a Type of Service (ToS) byte. The three left bits in that byte can be used to mark the packet with an IP Precedence value (0 to 7). Alternatively, the six left bits in the ToS byte can be used to mark the packet with a DSCP value (0 to 63).
Ensuring Voice Quality

Congestion Management

Congestion-management tools are queuing tools. These queuing tools decide how packets are placed in an interface’s output queues and how the packets are forwarded out of the queues. Several queuing tools are available in the IOS. However, modern queuing approaches include the following:

- **Low Latency Queuing (LLQ)** — The preferred queuing method for voice and video traffic, where traffic can be classified in up to 64 different classes, with different amounts of bandwidth given to each class. This method includes the ability to give priority treatment to one or more classes.

- **Class-Based Weighted Fair Queuing (CB-WFQ)** — Similar to LLQ, with the exception of having no priority-queuing mechanism.

Many of the Cisco QoS mechanisms are configured using a three-step configuration process called Modular QoS CLI (MQC).

© 2007 Cisco Systems Inc. All rights reserved. This publication is protected by copyright. Please see page 49 for more details.
ENSURING VOICE QUALITY

Following is an MQC example where you are classifying various types of e-mail traffic (for example, SMTP, IMAP, and POP3) into one class-map. The KaZaa protocol, which is frequently used for music downloads, is placed in another class-map. Voice over IP (VoIP) traffic is placed in yet another class-map. Then, the policy-map assigns bandwidth allocations or limitations to these traffic types. The code is as follows:

Router(config)#class-map match-any EMAIL
Router(config-cmap)#match protocol pop3
Router(config-cmap)#match protocol imap
Router(config-cmap)#match protocol smtp
Router(config-cmap)#exit

Router(config)#class-map MUSIC
Router(config-cmap)#match protocol kazaa2
Router(config-cmap)#exit

Router(config)#class-map VOICE
Router(config-cmap)#match protocol rtp

Notice that the CVOICE policy-map makes 128 kbps of bandwidth available to e-mail traffic. However, KaZaa version 2 traffic has its bandwidth limited to 32 kbps. Voice packets not only have access to 256 kbps of bandwidth, but they also receive “priority” treatment, meaning that they are sent first (that is, ahead of other traffic), up to the 256-kbps limit.

The next logical question is this: What happens to all the traffic that you did not classify? Interestingly, Cisco created a class-map named class-default, which categorizes any traffic that is not matched by one of the defined class-maps. Finally, in the previous example, the policy-map is applied in the outbound direction on the Serial 0/1 interface.
Traffic Conditioning

Although some of the congestion-management techniques can guarantee bandwidth amounts, in some situations, you might want to limit bandwidth usage. For example, you might want to prevent oversubscription of a link. The two categories of traffic conditioning are as follows:

- **Policing**—Limits traffic rates, with excess traffic being dropped
- **Shaping**—Limits traffic rates, with excess traffic being delayed

Shaping buffers excess traffic, while policing drops excess traffic. These characteristics suggest that policing is more appropriate on high-speed interfaces, whereas shaping is more appropriate on low-speed interfaces.

Signaling

The IntServ model uses signaling to allow an application to reserve bandwidth for the duration of the application. RSVP is the primary QoS signaling protocol. One of the main characteristics of signaling is that signaling is performed *end to end*, meaning that all routers along the path from the source to the destination must be configured to support the IntServ mechanism (for example, RSVP). Also, note that when the Cisco voice-enabled routers are configured to reserve bandwidth using RSVP, reservations are made in both directions.
Link Efficiency Mechanisms

Link efficiency mechanisms help make the most of the limited bandwidth that is available on WAN links. Two link efficiency mechanisms are as follows:

- **Link Fragmentation and Interleaving (LFI)**—Takes large payloads, fragments them, and interleaves smaller packets among the fragments to reduce serialization delay for latency-sensitive traffic (for use on link speeds of less than 768 kbps)

- **RTP Header Compression (cRTP)**—Takes a 40-byte VoIP header and compresses it to 2 or 4 bytes (for use on link speeds of less than 2 Mbps)

**AutoQoS**

Optimizing a QoS configuration for VoIP can be a daunting task. Fortunately, Cisco added a feature called AutoQoS to many of its router and switch platforms. AutoQos automatically generates router-based or switch-based VoIP QoS configurations.

On a router platform, the following command enables AutoQoS from either interface-configuration mode or from DLCI-configuration mode, if you have a Frame Relay circuit:

```
Router(config-if)#auto qos voip
```

After you enter this command, the IOS automatically executes a series of QoS configuration commands that depend on the configured interface bandwidth. For example, on a PPP interface running at less than 768 kbps, the AutoQoS feature automatically configures LLQ, cRTP, MLP, and even an SNMP configuration, where quality issues can be reported to an SNMP network management server.

**Preventing WAN Oversubscription**

Consider a scenario in which you have 50 kbps of bandwidth on an IP WAN, and perhaps a single voice call requires 25 kbps of bandwidth. After two voice calls are placed across the IP WAN, no more bandwidth is available. Therefore, if a third phone call is set up across the IP WAN, you have oversubscribed the bandwidth; this leads to poor quality for all three calls. To prevent such an occurrence, you can...
configure one or more Call Admission Control (CAC) mechanisms to prevent this oversubscription. CAC mechanisms fall in one of the following three categories:

- **Local CAC mechanisms**—Make admission-control decisions based on information known to the local router (for example, the number of connections placed using a particular dial peer).
- **Measurement-based CAC mechanisms**—Send Service Assurance Agent (SAA) probes out into the network to determine network conditions. A request to place a phone call across the IP WAN is then permitted or denied based on the probes’ results.
- **Resource-based CAC mechanisms**—Determine how much bandwidth has been used between different network locations. For example, a gatekeeper can keep track of the amount of bandwidth that is used between two defined zones.

Cisco also supports CAC features for the following call control protocols, which were discussed earlier:

- **H.323**—CAC can permit or deny a call attempt based on a router’s available resources (for example, CPU and memory).
- **SIP**—CAC can send SAA probes into the network to determine whether a call attempt should be permitted.
- **MGCP**—CAC can make admission-control decisions based on available router resources, SAA probes, and available network bandwidth.

**Bandwidth Provisioning**

The essence of bandwidth provisioning for voice networks is to determine how many calls that you might need to support during peak periods. You can obtain current call volumes from various sources, such as your telephony service provider or your PBX’s Station Message Detail Recorder (SMDR) records.

As a designer, you must determine an acceptable grade of service (GoS), which is the probability that a call will be blocked because of bandwidth oversubscription. Typically, you use a GoS of $P(0.01)$, which specifies a 1 percent probability of a blocked call during the busiest hour.

With the GoS determined, you next need to determine the number of Erlangs that you need to support during your busiest hour. An Erlang is defined as 1 hour of conversation. For example, two phone calls that lasted 30 minutes each would equal one Erlang.

As a design best practice, you can calculate the number of Erlangs that you need to support with the following formula:

\[ \text{Erlangs} = \frac{\text{Hours of phone use in a month}}{22} \times 0.15 \times 1.10 \]

The 22 represents the number of business days per month. The 0.15 indicates that approximately 15 percent of your phone system usage occurs during the busiest hour. Finally, the 1.10 adds 10 percent for call-processing overhead.
ENSURING VOICE QUALITY

With the number of required Erlangs and the GoS value, you can determine the number of required trunks using an Erlang table or calculator, such as the one found at http://www.erlang.com/calculator/erlb.

When you have the number of trunks that you need, you can translate that number into the amount of bandwidth that you require by determining how much bandwidth is needed for a single voice call and then multiplying that amount by the number of trunks that you need. As a reminder, you can use the Cisco voice bandwidth calculator for this purpose. You can find this calculator at http://tools.cisco.com/Support/VBC/do/CodecCalc1.do.

Note that a Cisco.com login is required for the bandwidth calculator.