CHAPTER 1

Voice-over-IP Overview

Voice-over-IP (VoIP) enables a Cisco 1750 router (hereafter referred to as the router) to carry voice traffic (for example, telephone calls and faxes) over an IP network. Cisco's voice support is implemented using voice packet technology. In VoIP, the digital signal processor (DSP) segments the voice signal into frames and stores them in voice packets. These voice packets are transported using IP in compliance with the International Telecommunications Union-Telecommunications (ITU-T) specification H.323, the specification for transmitting multimedia (voice, video, and data) across a network. Because it is a delay-sensitive application, you need to have a well-engineered, end-to-end network to successfully use VoIP. Fine-tuning your network to adequately support VoIP involves a series of protocols and features to improve quality of service (QoS). Traffic shaping considerations must also be taken into account to ensure the reliability of the voice connection.

VoIP is primarily a software feature; however, you must install the voice interface cards (VICs) in the router. For more information about installing a VIC in the router, refer to the <Emphasis>Cisco WAN Interface Cards Hardware Installation Guide.

Voice Primer

The Voice Primer section provides supplementary information for those users unfamiliar with voice telephony. To understand Cisco's voice implementations, it helps to have some understanding of the analog and digital transmission and signaling. This section provides some very basic, abbreviated voice telephony information as background to help you configure VoIP, Voice over Frame Relay, Voice over ATM, and Voice over HDLC and contains the following topics:

- How VoIP Processes a Typical Telephone Call
- Numbering Scheme
- Analog versus Digital
- CODECs
- Delay
- Echo
- Signaling

How VoIP Processes a Typical Telephone Call

Before configuring VoIP on your router, it helps to understand what happens at an application level when you place a call using VoIP. The general flow of a two-party voice call using VoIP is as follows:

1. The user picks up the handset; this signals an off-hook condition to the signaling application part of VoIP in the router.
2. The session application part of VoIP issues a dial tone and waits for the user to dial a telephone number.
3. The user dials the telephone number; those numbers are accumulated and stored by the session application.
4. After enough digits are accumulated to match a configured destination pattern, the telephone number is mapped to an IP host via the dial plan mapper. The IP host has a direct connection to either the destination telephone number or a PBX that is responsible for completing the call to the configured destination pattern.
5. The session application then runs the H.323 session protocol to establish a transmission and a reception channel for each direction over the IP network. If the call is being handled by a Private Branch Exchange (PBX), the PBX forwards the call to the destination telephone. If Resource Reservation Protocol (RSVP) has been configured, the RSVP reservations are put into effect to achieve the desired QoS over the IP network.
6. The coder-decoder compression schemes (CODECs) are enabled for both ends of the connection and the conversation proceeds using Real-Time Transport Protocol/User Datagram Protocol/Internet Protocol (RTP/UDP/IP) as the protocol stack.
7. Any call-progress indications (or other signals that can be carried inband) are cut through the voice path as soon as end-to-end audio channel is established. Signaling that can be detected by the voice ports (for example, inband dual-tone multifrequency (DTMF) digits after the call setup is complete) is also trapped by the session application at either end of the connection and carried over the IP network encapsulated in Real-Time Transport Control Protocol (RTCP) using the RTCP application-defined (APP) extension mechanism.
8. When either end of the call hangs up, the RSVP reservations are torn down (if RSVP is used) and the session ends. Each end becomes idle, waiting for the next off-hook condition to trigger another call setup.

Numbering Scheme

The standard PSTN is a large, circuit-switched network. It uses a specific numbering scheme, which complies with the ITU-T **international public telecommunications numbering plan** (E.164) recommendations. For example, in North America, the North American Numbering Plan (NANP) is used, which consists of an area code, an office code, and a station code. Area codes are assigned geographically, office codes are assigned to specific switches, and station codes identify a specific port on that switch. The format in North America is 1Nxx-Nxx-xxxx, with N = digits 2 through 9 and x = digits 0 through 9. Internationally, each country is assigned a one- to three-digit country code; the country's dialing plan follows the country code. In Cisco's voice implementations, numbering schemes are configured using the **destination-pattern** command.

Analog versus Digital

Analog transmission is not particularly robust or efficient at recovering from line noise. Because analog signals degrade over distance, they need to be periodically amplified; this amplification boosts both the voice signal and ambient line noise, resulting in degradation of the quality of the transmitted sound.

In response to the limitations of analog transmission, the telephony network migrated to digital transmission using pulse code modulation (PCM) or adaptive differential PCM (ADPCM). In both cases, analog sound is converted into digital form by sampling the analog sound 8000 times per second and converting each sample into a numeric code.

CODECs

Pulse code modulation (PCM) and adaptive differential PCM (ADPCM) are examples of “waveform” CODEC techniques. Waveform CODECs are compression techniques that exploit the redundant characteristics of the waveform itself. In addition to waveform CODECs, there are source CODECs that compress speech by sending only simplified parametric information about voice transmission; these CODECs require less bandwidth. Source CODECs include linear predictive coding (LPC), code-excited linear prediction (CELP) and multipulse-multilevel quantization (MP-MLQ).

Coding techniques for telephony and voice packet are standardized by the ITU-T in its G-series recommendations. The Cisco 1750 router uses the following coding standards:

- G.711—Describes the 64-kbps PCM voice coding technique. In G.711, encoded voice is already in the correct format for digital voice delivery in the PSTN or through PBXs.
- G.729—Describes CELP compression where voice is coded into 8-kbps streams. There are two variations of this standard (G.729 and G.729 Annex A) that differ mainly in computational complexity; both provide speech quality similar to 32-kbps ADPCM.

In Cisco’s voice implementations, compression schemes are configured using the **codec** command.

Mean Opinion Score

Each CODEC provides a certain quality of speech. The quality of transmitted speech is a subjective response of the listener. A common benchmark used to determine the quality of sound produced by specific CODECs is the mean opinion score (MOS). With MOS, a wide range of listeners judge the quality of a voice sample (corresponding to a particular CODEC) on a scale of 1 (bad) to 5 (excellent). The scores are averaged to provide the MOS for that sample. Table 1-1 shows the relationship between CODECs and MOS scores.

Table 1-1 Compression Methods and MOS Scores

Compression Method	Bit Rate (kbps)	Framing Size (ms)	MOS Score
G.711 PCM	64	0.125	4.1
G.729 CS-ACELP[1]	8	10	3.92
G.729 x 2 Encodings	8	10	3.27
G.729 x 3 Encodings	8	10	2.68
G.729a CS-ACELP	8	10	3.7

1. Conjugate structure-algebraic code-excited linear prediction

Although it might seem logical from a financial standpoint to convert all calls to low bit-rate CODECs to save on infrastructure costs, you should exercise additional care when designing voice networks with low bit-rate compression. There are drawbacks to compressing voice. One of the main drawbacks is signal distortion due to multiple encodings (called tandem encodings). For example, when a G.729 voice signal is tandem-encoded three times, the MOS score drops from 3.92 (very good) to 2.68 (unacceptable). Another drawback is CODEC-induced delay with low bit-rate CODECs.

Delay

One of the most important design considerations in implementing voice is minimizing one-way, end-to-end delay. Voice traffic is real-time traffic; if there is too long a delay in voice packet delivery, speech will be unrecognizable. Delay is inherent in voice-networking and is caused by a number of different factors. An acceptable delay is less than 200 milliseconds.

There are basically two kinds of delay inherent in today's telephony networks: propagation delay and handling delay. Propagation delay is caused by the characteristics of the speed of light traveling via a fiber-optic-based or copper-based medium. Handling delay (sometimes called serialization delay) is caused by the devices that handle voice information. Handling delays have a significant impact on voice quality in a packet network.

CODEC-induced delays are considered a handling delay. Table 1-2 shows the delay introduced by different CODECs.

Table 1-2 CODEC-Induced Delays

CODEC	Bit Rate (kbps)	Framing size (ms)	Compression Delay (ms)
G.711 PCM	64	0.125	5
G.729 CS-ACELP	8	10	15
G.729a CS-ACELP	8	10	15

Another handling delay is the time it takes to generate a voice packet. In VoIP, the DSP generates a frame every 10 milliseconds. Two of these frames are then placed within one voice packet; the packet delay is therefore 20 milliseconds.

Another source of handling delay is the time it takes to move the packet to the output queue. Cisco IOS software expedites the process of determining packet destination and getting the packet to the output queue. The actual delay at the output queue is another source of handling delay and should be kept under 10 milliseconds whenever possible by using whatever queuing methods are optimal for your network. Output queue delays are a QoS issue in VoIP and are discussed in the "Configure IP Networks for Real-Time Voice Traffic" section on page 2-2.

In Voice over Frame Relay, you need to make sure that voice traffic is not crowded out by data traffic. Strategies on how to manage Voice-over-Frame-Relay voice traffic are discussed in the "Configure Frame Relay for VoIP" section on page 2-24.

Jitter

Jitter is another factor that affects delay. Jitter occurs when there is a variation between when a voice packet is expected to be received and when it actually is received, causing a discontinuity in the real-time voice stream. Voice devices such as the Cisco 3600 router, Cisco MC3810, and the Cisco 1750 router compensate for jitter by setting up a playout buffer to playback voice in a smooth fashion. Playout control is handled through RTP encapsulation, either by selecting adaptive or non-adaptive playout-delay mode. In either mode, the default value for nominal delay is sufficient.

End-to-End Delay

Figuring out the end-to-end delay is not difficult if you know the end-to-end signal paths/data paths, the CODEC, and the payload size of the packets. Adding the delays from the end points to the CODECs at both ends, the encoder delay (which is 5 milliseconds for the G.711 and G.726 CODECs and 10 milliseconds for the G.729 CODEC), the packet delay, and the fixed portion of the network delay yields the end-to-end delay for the connection.

Echo

Echo is hearing your own voice in the telephone receiver while you are talking. When timed properly, echo is reassuring to the speaker; if the echo exceeds approximately 25 milliseconds, it can be distracting and cause breaks in the conversation. In a traditional telephony network, echo is normally caused by a mismatch in impedance from the four-wire network switch conversion to the two-wire local loop and controlled by echo cancellers. In voice-packet based networks, echo cancellers are built into the low bit-rate CODECs and are operated on each DSP. Echo cancellers are limited by design by the total amount of time they will wait for the reflected speech to be received, which is known as an echo trail. The echo trail is normally 32 milliseconds.

In Cisco’s voice implementations, echo cancellers are enabled using the **echo-cancel enable** command. The echo trails are configured using the **echo-cancel-coverage** command. VoIP has configurable echo trails of 8, 16, 24, and 32 milliseconds.

Signaling

Although there are various types of signaling used in telecommunications today, this document describes only those with direct applicability to Cisco’s voice implementations. The first one involves access signaling, which determines when a line has gone off-hook or on-hook (in other words, dial tone). FXS and FXO are types of access signaling. There are two common methods of providing this basic signal:

- Loop start is the most common technique for access signaling in a standard PSTN end-loop network. When a handset is picked-up (goes off-hook), this action closes the circuit that draws current from the telephone company’s central office (CO), indicating a change in status. This change in status signals the CO to provide a dial tone. An incoming call is signalled from the CO to the handset by sending a signal in a standard on/off pattern, which causes the telephone to ring.
- Ground start is another access signaling method used to indicate on-hook/off-hook status to the CO, but this signaling method is primarily used on trunk lines or tie-lines between PBXs. Ground-start signaling works by using ground and current detectors. This allows the network to indicate off-hook or seizure of an incoming call independent of the ringing signal.

In Cisco’s voice implementations, access signaling is configured using the **signal** command.

Another signaling technique used mainly between PBXs or other network-to-network telephony switches is known as E&M. There are five types of E&M signaling, as well as two different wiring methods. Cisco’s voice implementation supports E&M types I, II, III, and V, using both two-wire and four-wire implementations. In Cisco’s voice implementations, E&M signal types are configured using the **type** command.