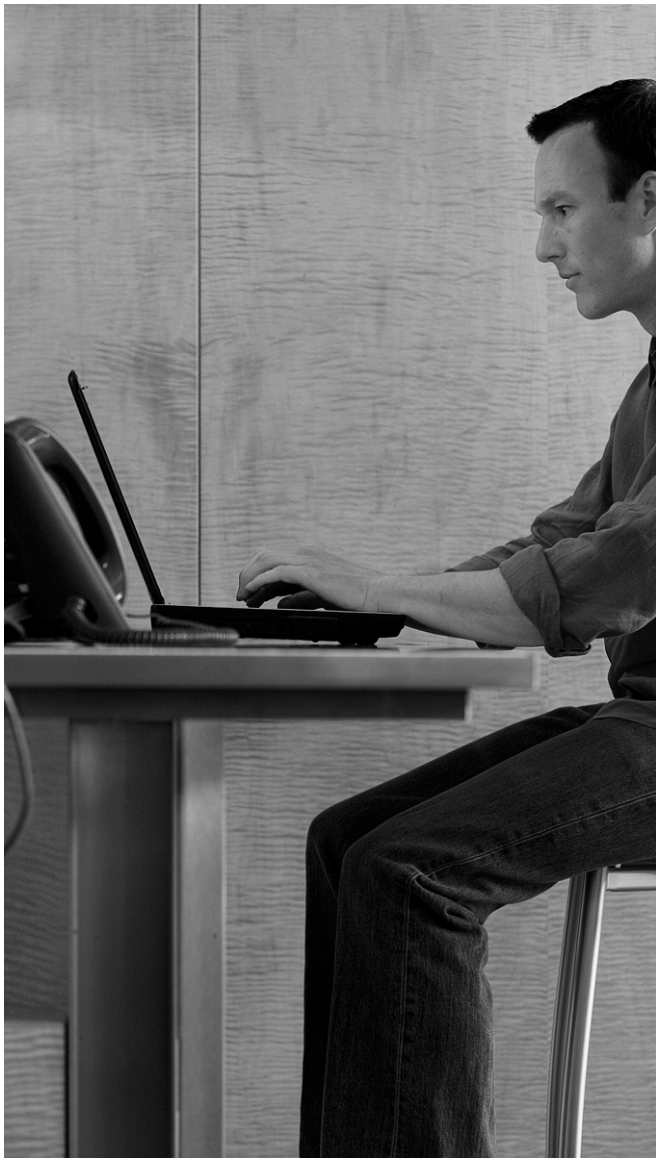


CHAPTER 7

VoIP: An In-Depth Analysis

To create a proper network design, it is important to know all the caveats and inner workings of networking technology. This chapter explains many of the issues facing Voice over IP (VoIP) and ways in which Cisco addresses these issues.

Communications via the Public Switched Telephone Network (PSTN) has its own set of problems, which are covered in Chapter 1, "Overview of the PSTN and Comparisons to Voice over IP," and Chapter 2, "Enterprise Telephony Today." VoIP technology has many similar issues and a whole batch of additional ones. This chapter details these various issues and explains how they can affect packet networks.

The following issues are covered in this chapter:

- Delay/latency
- Jitter
- Pulse Code Modulation (PCM)
- Voice compression
- Echo
- Packet loss
- Voice activity detection
- Digital-to-analog conversion
- Tandem encoding
- Transport protocols
- Dial-plan design

Delay/Latency

VoIP *delay* or *latency* is characterized as the amount of time it takes for speech to exit the speaker's mouth and reach the listener's ear.

Three types of delay are inherent in today's telephony networks: *propagation delay*, *serialization delay*, and *handling delay*. Propagation delay is caused by the length a signal must travel via light in fiber or electrical impulse in copper-based networks. Handling delay—also called processing delay—defines many different causes of delay (actual packetization, compression, and packet switching) and is caused by devices that forward the frame through the network.

Serialization delay is the amount of time it takes to actually place a bit or byte onto an interface. Serialization delay is not covered in depth in this book because its influence on delay is relatively minimal.

Propagation Delay

Light travels through a vacuum at a speed of 186,000 miles per second, and electrons travel through copper or fiber at approximately 125,000 miles per second. A fiber network stretching halfway around the world (13,000 miles) induces a one-way delay of about 70 milliseconds (70 ms). Although this delay is almost imperceptible to the human ear, propagation delays in conjunction with handling delays can cause noticeable speech degradation.

Handling Delay

As mentioned previously, devices that forward the frame through the network cause handling delay. Handling delays can impact traditional phone networks, but these delays are a larger issue in packetized environments. The following paragraphs discuss the different handling delays and how they affect voice quality.

In the Cisco IOS VoIP product, the Digital Signal Processor (DSP) generates a speech sample every 10 ms when using G.729. Two of these speech samples (both with 10 ms of delay) are then placed within one packet. The packet delay is, therefore, 20 ms. An initial look-ahead of 5 ms occurs when using G.729, giving an initial delay of 25 ms for the first speech frame.

Vendors can decide how many speech samples they want to send in one packet. Because G.729 uses 10 ms speech samples, each increase in samples per frame raises the delay by 10 ms. In fact, Cisco IOS enables users to choose how many samples to put into each frame.

Cisco gave DSP much of the responsibility for framing and forming packets to keep router/gateway overhead low. The Real-Time Transport Protocol (RTP) header, for example, is placed on the frame in the DSP instead of giving the router that task.

Queuing Delay

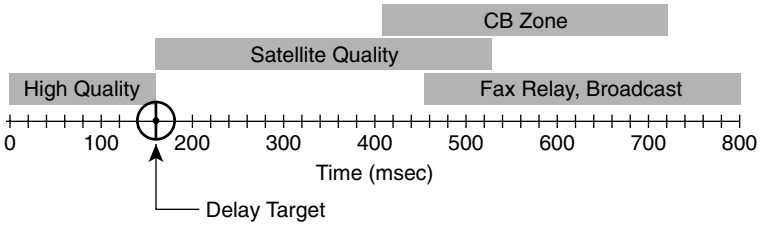
A packet-based network experiences delay for other reasons. Two of these are the time necessary to move the actual packet to the output queue (packet switching) and queuing delay.

When packets are held in a queue because of congestion on an outbound interface, the result is *queuing delay.* Queuing delay occurs when more packets are sent out than the interface can handle at a given interval.

The actual queuing delay of the output queue is another cause of delay. You should keep this factor to less than 10 ms whenever you can by using whatever queuing methods are optimal for your network. This subject is covered in greater detail in Chapter 8, "Quality of Service."

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) G.114 recommendation specifies that for good voice quality, no more than 150 ms of one-way, end-to-end delay should occur, as shown in Figure 7-1. With the Cisco VoIP implementation, *two* routers with minimal network delay (back to back) use only about 60 ms of end-to-end delay. This leaves up to 90 ms of network delay to move the IP packet from source to destination.

Figure 7-1 *End-to-End Delay*



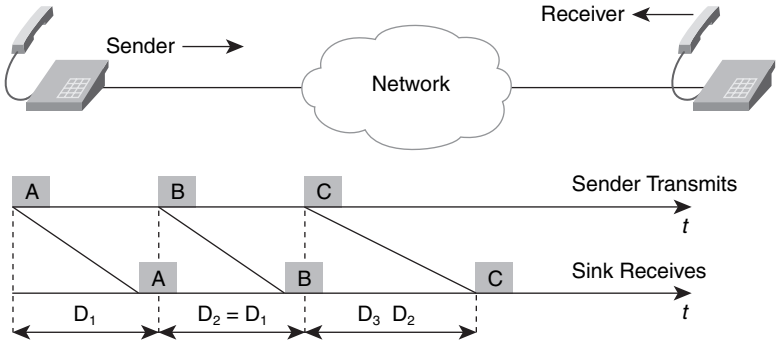
As shown in Figure 7-1, some forms of delay are longer, although accepted, because no other alternatives exist. In satellite transmission, for example, it takes approximately 250 ms for a transmission to reach the satellite, and another 250 ms for it to come back down to Earth. This results in a total delay of 500 ms. Although the ITU-T recommendation notes that this is outside the acceptable range of voice quality, many conversations occur every day over satellite links. As such, voice quality is often defined as what users will accept and use.

In an unmanaged, congested network, queuing delay can add up to two seconds of delay (or result in the packet being dropped). This lengthy period of delay is unacceptable in almost any voice network. Queuing delay is only one component of end-to-end delay. Another way end-to-end delay is affected is through jitter.

Jitter

Simply stated, *jitter* is the variation of packet interarrival time. Jitter is one issue that exists only in packet-based networks. While in a packet voice environment, the sender is expected to reliably transmit voice packets at a regular interval (for example, send one frame every 20 ms). These voice packets can be delayed throughout the packet network and not arrive at that same regular interval at the receiving station (for example, they might not be received every 20 ms; see Figure 7-2). The difference between when the packet is expected and when it is actually received is *jitter.*

Figure 7-2 *Variation of Packet Arrival Time (Jitter)*



In Figure 7-2, you can see that the amount of time it takes for packets A and B to send and receive is equal (D1=D2). Packet C encounters delay in the network, however, and is received *after* it is expected. This is why a *jitter buffer*, which conceals interarrival packet delay variation, is necessary. Voice packets in IP networks have highly variable packet-interarrival intervals. Recommended practice is to count the number of packets that arrive late and create a ratio of these packets to the number of packets that are successfully processed. You can then use this ratio to adjust the jitter buffer to target a predetermined, allowable late-packet ratio. This adaptation of jitter buffer sizing is effective in compensating for delays.

Note that jitter and total delay are *not* the same thing, although having plenty of jitter in a packet network can increase the amount of total delay in the network. This is because the more jitter you have, the larger your jitter buffer needs to be to compensate for the unpredictable nature of the packet network.

Most DSPs do not have infinite jitter buffers to handle excessive network delays. Sometimes it is better to just drop packets or have fixed-length buffers instead of creating unwanted delays in the jitter buffers. If your data network is engineered well and you take the proper precautions, jitter is usually not a major problem and the jitter buffer does not significantly contribute to the total end-to-end delay.

RTP timestamps are used within Cisco IOS Software to determine what level of jitter, if any, exists within the network.

The jitter buffer found within Cisco IOS Software is considered a dynamic queue. This queue can grow or shrink exponentially depending on the interarrival time of the RTP packets.

Although many vendors choose to use static jitter buffers, Cisco found that a well-engineered dynamic jitter buffer is the best mechanism to use for packet-based voice networks. Static jitter buffers force the jitter buffer to be either too large or too small, thereby causing the audio quality to suffer, due to either lost packets or excessive delay. The Cisco jitter buffer dynamically increases or decreases based upon the interarrival delay variation of the last few packets.

Pulse Code Modulation

Although analog communication is ideal for human communication, analog transmission is neither robust nor efficient at recovering from line noise. In the early telephony network, when analog transmission was passed through amplifiers to boost the signal, not only was the voice boosted but the line noise was amplified, as well. This line noise resulted in an often-unusable connection.

It is much easier for digital samples, which are comprised of 1 and 0 bits, to be separated from line noise. Therefore, when analog signals are regenerated as digital samples, a clean sound is maintained. When the benefits of this digital representation became evident, the telephony network migrated to pulse code modulation (PCM).

What Is PCM?

As covered in Chapter 1, PCM converts analog sound into digital form by sampling the analog sound 8000 times per second and converting each sample into a numeric code. The Nyquist theorem states that if you sample an analog signal at twice the rate of the highest frequency of interest, you can accurately reconstruct that signal back into its analog form. Because most speech content is below 4000 Hz (4 kHz), a sampling rate of 8000 times per second (125 microseconds between samples) is required.

A Sampling Example for Satellite Networks

Satellite networks have an inherent delay of around 500 ms. This includes 250 ms for the trip up to the satellite, and another 250 ms for the trip back to Earth. In this type of network, packet loss is highly controlled due to the expense of bandwidth. Also, if some type of voice application is already running through the satellite, the users of this service are accustomed to a quality of voice that has excessive delays.

Cisco IOS, by default, sends two 10-ms G.729 speech frames in every packet. Although this is acceptable for most applications, this might not be the best method for utilizing the expensive bandwidth on a satellite link. The simple explanation for wasting bandwidth is that a header exists for every packet. The more speech frames you put into a packet, the fewer headers you require.

If you take the satellite example and use four 10-ms G.729 speech frames per packet, you can cut by half the number of headers you use. Table 7-1 clearly shows the difference between the various frames per packet. With only a 20-byte increase in packet size (20 extra bytes equals two 10 ms G.729 samples), you carry twice as much speech with the packet.

Table 7-1 *Frames per Packet (G.729)*

G.729 Samples per Frame	IP/RTP/UDP Header	Bandwidth Consumed	Latency*
Default (two samples per frame)	40 bytes	24,000 bps	25 ms
Satellite (four samples per frame)	40 bytes	16,000 bps	45 ms
Low Latency (one sample per frame)	40 bytes	40,000 bps	15 ms

* Compression and packetization delay only

To reduce the overall IP/RTP/UDP overhead introduced by the 54-byte header, multiple voice samples can be packed into a single Ethernet frame to transmit. Although this can increase the voice delay, increasing this count can improve the overall voice quality, especially when the bandwidth is constrained.

How many voice samples to be sent per frame depends on what codec you choose and the balance between bandwidth utilization and impact of packet loss. The bigger this value, the higher the bandwidth utilization because more voice samples are packed into the payload field of a UDP/RTP packet and thus the network header overhead would be lower. The impact of a

packet loss on perceived voice quality will be bigger, however. Table 7-2 lists the values for some of the commonly used codec types.

Table 7-2 *Voice Samples per Frame for VoIP Codecs*

Codec Type	Voice Samples per Frame (Default)	Voice Samples per Frame (Maximum)
PCMU/PCMA	2	10
G.723	1	32
G.726-32	2	20
G.729	2	64
G.728	4	64

Voice Compression

Two basic variations of 64 Kbps PCM are commonly used: μ-law and a-law. The methods are similar in that they both use logarithmic compression to achieve 12 to 13 bits of linear PCM quality in 8 bits, but they are different in relatively minor compression details (μ-law has a slight advantage in low-level, signal-to-noise ratio performance). Usage is historically along country and regional boundaries, with North America using μ-law and Europe and other countries using a-law modulation. It is important to note that when making a long-distance call, any required μ-law to a-law conversion is the responsibility of the μ-law country.

Another compression method used often is *adaptive differential pulse code modulation (ADPCM)*. A commonly used instance of ADPCM is ITU-T G.726, which encodes using 4-bit samples, giving a transmission rate of 32 Kbps. Unlike PCM, the 4 bits do not directly encode the amplitude of speech, but they do encode the differences in amplitude, as well as the rate of change of that amplitude, employing some rudimentary linear prediction.

PCM and ADPCM are examples of *waveform* codecs—compression techniques that exploit redundant characteristics of the waveform itself. New compression techniques were developed over the past 10 to 15 years that further exploit knowledge of the source characteristics of speech generation. These techniques employ signal processing procedures that compress speech by sending only simplified parametric information about the original speech excitation and vocal tract shaping, requiring less bandwidth to transmit that information.

These techniques can be grouped together generally as *source* codecs and include variations such as *linear predictive coding (LPC)*, *code excited linear prediction compression (CELP)*, and *multipulse, multilevel quantization (MP-MLQ)*.

Voice Coding Standards

The ITU-T standardizes CELP, MP-MLQ PCM, and ADPCM coding schemes in its G-series recommendations. The most popular voice coding standards for telephony and packet voice include:

- G.711—Describes the 64 Kbps PCM voice coding technique outlined earlier; G.711-encoded voice is already in the correct format for digital voice delivery in the public phone network or through Private Branch eXchanges (PBXs).
- G.726—Describes ADPCM coding at 40, 32, 24, and 16 Kbps; you also can interchange ADPCM voice between packet voice and public phone or PBX networks, provided that the latter has ADPCM capability.
- G.728—Describes a 16 Kbps low-delay variation of CELP voice compression.
- G.729—Describes CELP compression that enables voice to be coded into 8 Kbps streams; two variations of this standard (G.729 and G.729 Annex A) differ largely in computational complexity, and both generally provide speech quality as good as that of 32 Kbps ADPCM.
- G.723.1—Describes a compression technique that you can use to compress speech or other audio signal components of multimedia service at a low bit rate, as part of the overall H.324 family of standards. Two bit rates are associated with this coder: 5.3 and 6.3 Kbps. The higher bit rate is based on MP-MLQ technology and provides greater quality. The lower bit rate is based on CELP, provides good quality, and affords system designers with additional flexibility.
- iLBC (Internet Low Bitrate Codec)—A free speech codec suitable for robust voice communication over IP. The codec is designed for narrow band speech and results in a payload bit rate of 13.33 kbps with an encoding frame length of 30 ms and 15.20 kbps with an encoding length of 20 ms. The iLBC codec enables graceful speech quality degradation in the case of lost frames, which occurs in connection with lost or delayed IP packets. The basic quality is higher than G.729A, with high robustness to packet loss. The PacketCable consortium and many vendors have adopted iLBC as a preferred codec. It is also being used by many PC-to-Phone applications, such as Skype, Google Talk, Yahoo! Messenger with Voice, and MSN Messenger.

Mean Opinion Score

You can test voice quality in two ways: subjectively and objectively. Humans perform subjective voice testing, whereas computers—which are less likely to be "fooled" by compression schemes that can "trick" the human ear—perform objective voice testing.

Codecs are developed and tuned based on subjective measurements of voice quality. Standard objective quality measurements, such as total harmonic distortion and signal-to-noise ratios, do not correlate well to a human's perception of voice quality, which in the end is usually the goal of most voice compression techniques.

A common subjective benchmark for quantifying the performance of the speech codec is the *mean opinion score (MOS)*. MOS tests are given to a group of listeners. Because voice quality and sound in general are subjective to listeners, it is important to get a wide range of listeners and sample material when conducting a MOS test. The listeners give each sample of speech material a rating of 1 (bad) to 5 (excellent). The scores are then averaged to get the mean opinion score.

MOS testing also is used to compare how well a particular codec works under varying circumstances, including differing background noise levels, multiple encodes and decodes, and so on. You can then use this data to compare against other codecs.

MOS scoring for several ITU-T codecs is listed in Table 7-3. This table shows the relationship between several low-bit rate coders and standard PCM.

Table 7-3 *ITU-T Codec MOS Scoring*

Compression Method	Bit Rate (Kbps)	Sample Size (ms)	MOS Score
G.711 PCM	64	0.125	4.1
G.726 ADPCM	32	0.125	3.85
G.728 Low Delay Code Excited Linear Predictive (LD-CELP)	15	0.625	3.61
G.729 Conjugate Structure Algebraic Code Excited Linear Predictive (CS-ACELP)	8	10	3.92
G.729a CS-ACELP	8	10	3.7
G.723.1 MP-MLQ	6.3	30	3.9
G.723.1 ACELP	5.3	30	3.65
iLBC Freeware	15.2	20	3.9
	13.3	30	

Source: Cisco Labs

For iLBC codec - Research Paper - "COMPARISONS OF FEC AND CODEC ROBUSTNESS ON VOIP QUALITY AND BANDWIDTH EFFICIENCY" - WENYU JIANG AND HENNING SCHULZRINNE. Columbia University, Department of Computer Science, USA.

Perceptual Speech Quality Measurement

Although MOS scoring is a subjective method of determining voice quality, it is not the only method for doing so. The ITU-T put forth recommendation P.861, which covers ways you can objectively determine voice quality using Perceptual Speech Quality Measurement (PSQM).

PSQM has many drawbacks when used with voice codecs (vocoders). One drawback is that what the "machine" or PSQM hears is not what the human ear perceives. In layman's terms, a person can trick the human ear into perceiving a higher-quality voice, but a computer cannot be tricked. Also, PSQM was developed to "hear" impairments caused by compression and decompression and not packet loss or jitter.

Echo

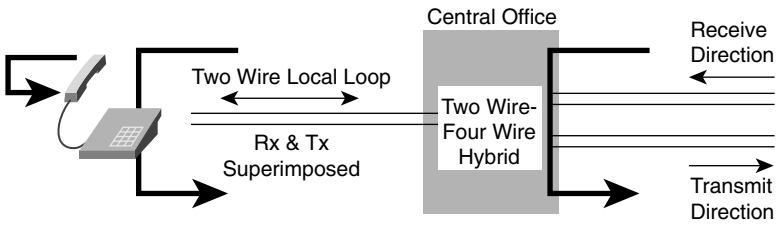
Echo is an amusing phenomenon to experience while visiting the Grand Canyon, but echo on a phone conversation can range from slightly annoying to unbearable, making conversation unintelligible.

Hearing your own voice in the receiver while you are talking is common and reassuring to the speaker. Hearing your own voice in the receiver after a delay of more than about 25 ms, however, can cause interruptions and can break the cadence in a conversation.

In a traditional toll network, echo is normally caused by a mismatch in impedance from the four-wire network switch conversion to the two-wire local loop (as shown in Figure 7-3). Echo, in the standard Public Switched Telephone Network (PSTN), is regulated with echo cancellers and a tight control on impedance mismatches at the common reflection points, as depicted in Figure 7-3.

Echo has two drawbacks: It can be loud, and it can be long. The louder and longer the echo, of course, the more annoying the echo becomes.

Figure 7-3 *Echo Caused by Impedance Mismatch*



Telephony networks in those parts of the world where analog voice is primarily used employ echo suppressors, which remove echo by capping the impedance on a circuit. This is not the best

mechanism to use to remove echo and, in fact, causes other problems. You cannot use Integrated Services Digital Network (ISDN) on a line that has an echo suppressor, for instance, because the echo suppressor cuts off the frequency range that ISDN uses.

In today's packet-based networks, you can build echo cancellers into low-bit-rate codecs and operate them on each DSP. In some manufacturers' implementations, echo cancellation is done in software; this practice drastically reduces the benefits of echo cancellation. Cisco VoIP, however, does all its echo cancellation on its DSP.

To understand how echo cancellers work, it is best to first understand where the echo comes from.

In this example, assume that user A is talking to user B. The speech of user A to user B is called G. When G hits an impedance mismatch or other echo-causing environments, it bounces back to user A. User A can then hear the delay several milliseconds after user A actually speaks.

To remove the echo from the line, the device user A is talking through (router A) keeps an inverse image of user A's speech for a certain amount of time. This is called *inverse speech (–G)*. This echo canceller listens for the sound coming from user B and subtracts the –G to remove any echo.

Echo cancellers are limited by the total amount of time they wait for the reflected speech to be received, a phenomenon known as *echo tail*. Cisco has configurable echo tails of 16, 24, 32, 64, and 128 ms.

It is important to configure the appropriate amount of echo cancellation when initially installing VoIP equipment. If you don't configure enough echo cancellation, callers will hear echo during the phone call. If you configure too much echo cancellation, it will take longer for the echo canceller to converge and eliminate the echo.

Packet Loss

Packet loss in data networks is both common and expected. Many data protocols, in fact, use packet loss so that they know the condition of the network and can reduce the number of packets they are sending.

When putting critical traffic on data networks, it is important to control the amount of packet loss in that network.

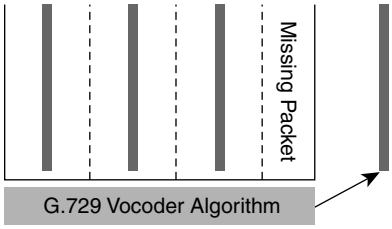
Cisco Systems has been putting business-critical, time-sensitive traffic on data networks for many years, starting with Systems Network Architecture (SNA) traffic in the early 1990s. With protocols such as SNA that do *not* tolerate packet loss well, you need to build a well-engineered network that can prioritize the time-sensitive data ahead of data that can handle delay and packet loss.

When putting voice on data networks, it is important to build a network that can successfully transport voice in a reliable and timely manner. Also, it is helpful when you can use a mechanism to make the voice somewhat resistant to periodic packet loss.

Cisco Systems developed many quality of service (QoS) tools that enable administrators to classify and manage traffic through a data network. If a data network is well engineered, you can keep packet loss to a minimum.

Cisco Systems' VoIP implementation enables the voice router to respond to periodic packet loss. If a voice packet is not received when expected (the expected time is variable), it is assumed to be lost and the last packet received is replayed, as shown in Figure 7-4. Because the packet lost is only 20 ms of speech, the average listener does not notice the difference in voice quality.

Figure 7-4 *Packet Loss with G.729*



Using Cisco's G.729 implementation for VoIP, let's say that each of the lines in Figure 7-4 represents a packet. Packets 1, 2, and 3 reach the destination, but packet 4 is lost somewhere in transmission. The receiving station waits for a period of time (per its jitter buffer) and then runs a *concealment strategy.*

This concealment strategy replays the last packet received (in this case, packet 3), so the listener does not hear gaps of silence. Because the lost speech is only 20 ms, the listener most likely does not hear the difference. You can accomplish this concealment strategy only if one packet is lost. If multiple consecutive packets are lost, the concealment strategy is run only once until another packet is received.

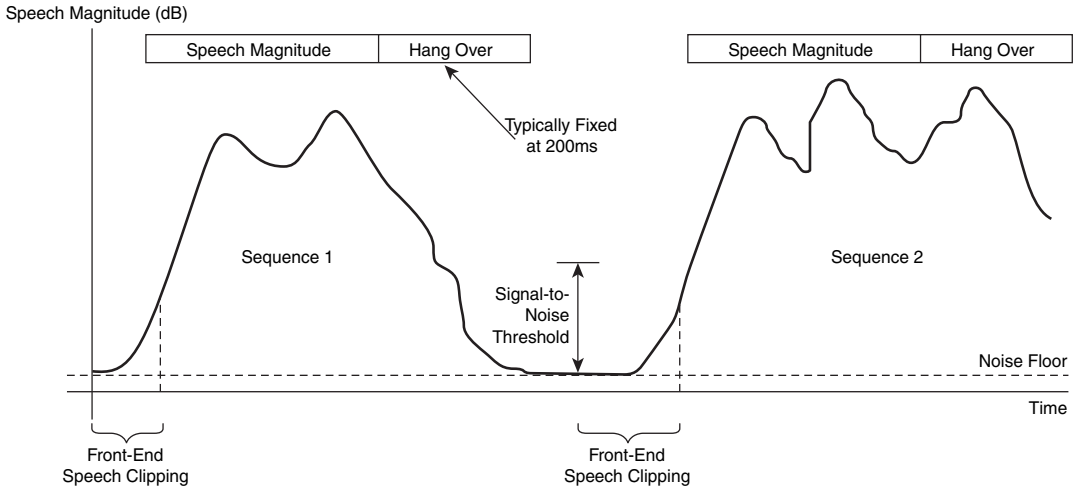
Because of the concealment strategy of G.729, as a rule of thumb G.729 is tolerant to about five percent packet loss averaged across an entire call.

Voice Activity Detection

In normal voice conversations, someone speaks and someone else listens. Today's toll networks contain a bi-directional, 64,000 bit per second (bps) channel, regardless of whether anyone is speaking. This means that in a normal conversation, at least 50 percent of the total bandwidth is wasted. The amount of wasted bandwidth can actually be much higher if you take a statistical sampling of the breaks and pauses in a person's normal speech patterns.

When using VoIP, you can utilize this "wasted" bandwidth for other purposes when voice activity detection (VAD) is enabled. As shown in Figure 7-5, VAD works by detecting the magnitude of speech in decibels (dB) and deciding when to cut off the voice from being framed.

Figure 7-5 *Voice Activity Detection*



Typically, when the VAD detects a drop-off of speech amplitude, it waits a fixed amount of time before it stops putting speech frames in packets. This fixed amount of time is known as *hangover* and is typically 200 ms.

With any technology, tradeoffs are made. VAD experiences certain inherent problems in determining when speech ends and begins, and in distinguishing speech from background noise. This means that if you are in a noisy room, VAD is unable to distinguish between speech and background noise. This also is known as the *signal-to-noise threshold* (refer to Figure 7-5). In these scenarios, VAD disables itself at the beginning of the call.

Another inherent problem with VAD is detecting when speech begins. Typically the beginning of a sentence is cut off or clipped (refer to Figure 7-5). This phenomenon is known as *front-end speech clipping*. Usually, the person listening to the speech does not notice front-end speech clipping.

Digital-to-Analog Conversion

Digital to analog (D/A) conversion issues also currently plague toll networks. Although almost all the telephony backbone networks in first-world countries today are digital, sometimes multiple D/A conversions occur.

Each time a conversion occurs from digital to analog and back, the speech or waveform becomes less "true." Although today's toll networks can handle at least seven D/A conversions before voice quality is affected, compressed speech is less robust in the face of these conversions.

It is important to note that D/A conversion must be tightly managed in a compressed speech environment. When using G.729, just two conversions from D/A cause the MOS score to decrease rapidly. The only way to manage D/A conversion is to have the network designer design VoIP environments with as few D/A conversions as possible.

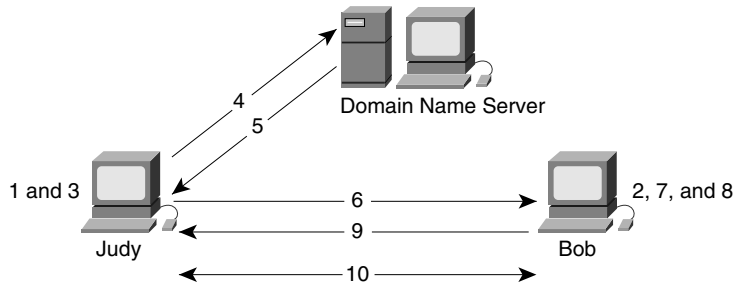
Although D/A conversions affect all voice networks, VoIP networks using a PCM codec (G.711) are just as resilient to problems caused by D/A conversions as today's telephony networks are.

Tandem Encoding

As covered in Chapter 1, all circuit-switched networks today work on the premise of switching calls at the data link layer. The circuit switches are organized in a hierarchical model in which switches higher in the hierarchy are called *tandem switches*.

Tandem switches do not actually terminate any local loops; rather, they act as a *higher-layer* circuit switch. In the hierarchical model, several layers of tandem circuit switches can exist, as shown in Figure 7-6. This enables end-to-end connectivity for anyone with a phone, without the need for a direct connection between every home on the planet.

Figure 7-6 *Tandem Switching Hierarchy*



Typically, a voice call that passes through the two TDM switches and one tandem switch does not incur degradation in voice quality because these circuit switches use 64 Kbps channels.

If the TDM switches compress voice and the tandem switch must decompress and recompress the voice, the voice quality can be drastically affected. Although compression and recompression are not common in the PSTN today, you must plan for it and design around it in packet networks.

Voice degradation occurs when you have more than one compression/decompression cycle for each phone call. Figure 7-7 provides an example of when this scenario might occur.

Figure 7-7 *VoIP Tandem Encoding*

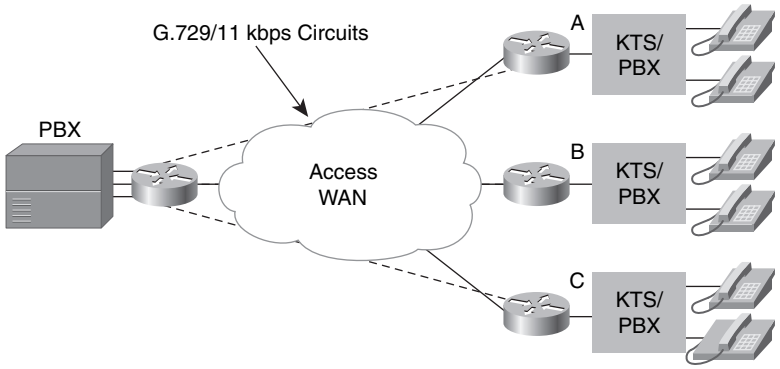


Figure 7-7 depicts three VoIP routers connected and acting as tie-lines between one central-site PBX and three remote-branch PBXs. The network is designed to put all the dial-plan information in the central-site PBX. This is common in many enterprise networks to keep the administration of the dial plan centralized.

A drawback to tandem encoding when used with VoIP is that, if a telephony user at branch B wants to call a user at branch C, two VoIP ports at central site A must be utilized. Also, two compression/decompression cycles exist, which means that voice quality will degrade.

Different codecs react differently to tandem encoding. G.729 can handle two compression/decompression cycles, while G.723.1 is less resilient to multiple compression cycles.

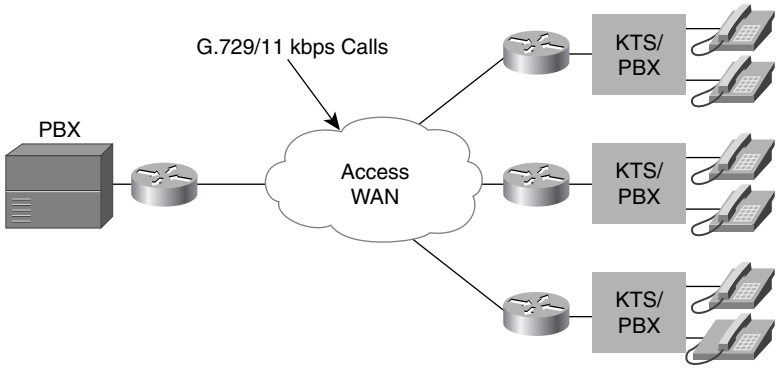
Assume, for example, that a user at remote site B wants to call a user at remote site C. The call goes through PBX B, is compressed and packetized at VoIP router B, and is sent to the central site VoIP router A, which decompresses the call and sends it to PBX A. PBX A circuit-switches the call back to its VoIP router (router A), which compresses and packetizes the call, and sends it to the remote site C, where it is then decompressed and sent to PBX C. This process is known as *tandem-compression*; you should avoid it in all networks where compression exists.

It is easy to avoid tandem compression. This customer simplified the router configuration at the expense of voice quality. Cisco IOS has other mechanisms that can simplify management of dial plans and still keep the highest voice quality possible.

One possible method is to use a Cisco IOS Multimedia Conference Manager (for instance, H.323 Gatekeeper). Another mechanism is to use one of Cisco's management applications, such as Cisco Voice Manager, to assist in configuring and maintaining dial plans on all your routers.

Taking the same example of three PBXs connected through three VoIP routers, but configuring the VoIP routers differently, simplifies the call-flow and avoids tandem encoding, as shown in Figure 7-8.

Figure 7-8 *VoIP Without Tandem Encoding*



You can see one of IP's strengths in Figure 7-8: a tie-line does not have to be leased from the telephone company to complete calls between two PBXs. If a data network connects the sites, VoIP can ride across that network.

The dial plan is moved from the central-site PBX to each of the VoIP routers. This enables each VoIP device to make a call-routing decision and removes the need for tie-lines. The major benefit of this change is the removal of needless compression/decompression cycles.

Transport Protocols

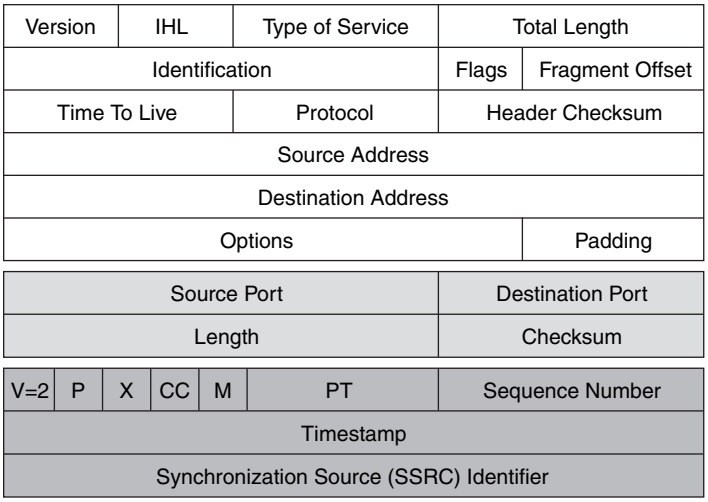
As explained in Chapter 6, "IP Tutorial," two main types of traffic ride upon Internet Protocol (IP): User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). In general, you use TCP when you need a reliable connection and UDP when you need simplicity and reliability is not your chief concern.

Due to the time-sensitive nature of voice traffic, UDP/IP was the logical choice to carry voice. More information was needed on a packet-by-packet basis than UDP offered, however. So, for real-time or delay-sensitive traffic, the Internet Engineering Task Force (IETF) adopted RTP. VoIP rides on top of RTP, which rides on top of UDP. Therefore, VoIP is carried with an RTP/UDP/IP packet header.

RTP

RTP is the standard for transmitting delay-sensitive traffic across packet-based networks. RTP rides on top of UDP and IP. RTP gives receiving stations information that is not in the connectionless UDP/IP streams. As shown in Figure 7-9, two important bits of information are sequence number and timestamp. RTP uses the sequence information to determine whether the packets are arriving in order, and it uses the time-stamping information to determine the interarrival packet time (jitter).

Figure 7-9 *Real-Time Transport Header*



You can use RTP for media on demand, as well as for interactive services such as Internet telephony. RTP (refer to Figure 7-9) consists of a data part and a control part, the latter called RTP Control Protocol (RTCP).

The data part of RTP is a thin protocol that provides support for applications with real-time properties, such as continuous media (for example, audio and video), including timing reconstruction, loss detection, and content identification.

RTCP provides support for real-time conferencing of groups of any size within an Internet. This support includes source identification and support for gateways, such as audio and video bridges as well as multicast-to-unicast translators. It also offers QoS feedback from receivers to the multicast group, as well as support for the synchronization of different media streams.

Another new proposal defined in RFC 3611, RTP Control Protocol Extended Reports (RTCP XR), provides a rich set of data for VoIP management. The data for these extended reports can be provided by technology such as VQmon embedded into VoIP phones or gateways and sent periodically during the call to provide real-time feedback on voice quality. The reports generated present a very useful set of VoIP metrics data on network packet loss, RTP round trip delay, and so on.

Using RTP is important for real-time traffic, but a few drawbacks exist. The IP/RTP/UDP headers are 20, 8, and 12 bytes, respectively. This adds up to a 40-byte header, which is twice as big as the payload when using G.729 with two speech samples (20 ms). You can compress this large header to 2 or 4 bytes by using RTP Header Compression (CRTP). CRTP is covered in depth in Chapter 8.

Reliable User Data Protocol

Reliable User Data Protocol (RUDP) builds in some reliability to the connectionless UDP protocol. RUDP enables reliability without the need for a connection-based protocol such as TCP. The basic method of RUDP is to send multiples of the same packet and enable the receiving station to discard the unnecessary or redundant packets. This mechanism makes it more probable that one of the packets will make the journey from sender to receiver.

This also is known as *forward error correction* (FEC). Few implementations of FEC exist due to bandwidth considerations (a doubling or tripling of the amount of bandwidth used). Customers that have almost unlimited bandwidth, however, consider FEC a worthwhile mechanism to enhance reliability and voice quality.

Cisco currently utilizes RUDP in its PGW2200 product, which enables Signaling System 7 (SS7) to Q.931 over IP conversion. The Q.931 over IP is transmitted over RUDP.

Dial-Plan Design

One of the areas that causes the largest amount of headaches when designing an Enterprise Telephony (ET) network is the *dial plan*. The causes of these head pains might be due to the complex issues of integrating disparate networks. Many of these disparate networks were not designed for integration.

A good data example of joining disparate networks is when two companies merge. In such a scenario, the companies' data networks (IP addressing, ordering applications, and inventory database) must be joined. It is highly improbable that both companies used the same methodologies when implementing their data networks, so problems can arise.

The same problems can occur in telephony networks. If two companies merge, their phone systems (voice mail, billing, supplementary features, and dial-plan addressing) might be incompatible with each other.

These dial-plan issues also can occur when a company decides to institute a corporate dial plan. Consider Company X, for example. Company X grew drastically in the last three years and now operates 30 sites throughout the world, with its headquarters in Dallas. Company X currently dials through the PSTN to all its 29 remote sites. Company X wants to simplify the dialing plan to all its remote sites to enable better employee communication and ease of use.

Company X currently has a large PBX at its headquarters and smaller PBX systems at its remote sites. Several alternatives are available to this company:

- Purchase leased lines between headquarters and all remote sites.
- Purchase a telephony Virtual Private Network (VPN) from the telephone company and dial an access code from anywhere to access the VPN.
- Take advantage of the existing data infrastructure and put voice on the data network.

Regardless of which option Company X chooses, it must face dial-plan design, network management, and cost issues.

Without getting into great detail, most companies must decide on their dial-plan design based on the following issues:

- Plans for growth
- Cost of leased circuits or VPNs
- Cost of additional equipment for packet voice
- Number overlap (when more than one site has the same phone number)
- Call-flows (the call patterns from each site)
- Busy hour (the time of day when the highest number of calls are offered on a circuit)

Depending on the size of the company, the dial plan can stretch from two digits to seven or eight digits. It is important that you not force yourself down a particular path until you address the previous issues.

Company X plans on sustaining 20–30 percent growth and decides on a seven-digit dial plan based on its growth patterns. This choice also cuts down on the number overlap that might be present.

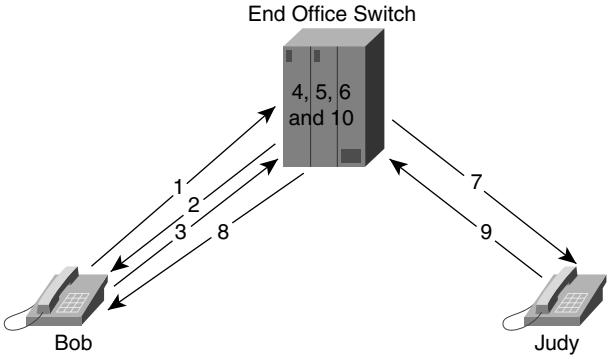
Company X will have a three-digit site code, and four digits for the actual subscriber line. It made this decision because it does not believe it will have more than 999 branch offices.

NOTE For companies that have hundreds of branch offices, it is common to have more site codes and fewer subscriber lines. If a company has several hundred branch offices and needs thousands of subscriber lines, it must use more digits (that is, it must use an eight- or nine-digit dial plan).

End Office Switch Call-Flow Versus IP Phone Call

To simplify a TDM or end office switch call-flow and an IP call-flow, this section looks at ways you can call your next-door neighbor using both the PSTN and the Internet. Figure 7-10 shows a basic call-flow in the PSTN today. Compare this to an IP phone call-flow and notice the similarities of necessary call setup.

Figure 7-10 *Calling My Neighbor with Today's PSTN*



In this example, Bob calls his neighbor Judy. They are both subscribers on the local end office switch, and therefore, no SS7 is needed. The following steps occur:

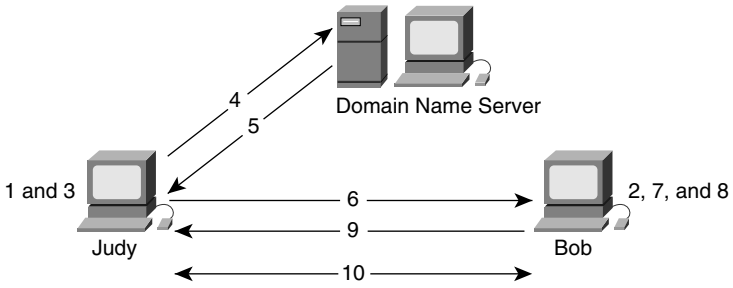
1. Bob picks up his handset (off hook).
2. The local end office switch gives Bob a dial tone.
3. Bob dials Judy's seven-digit phone number.
4. The end office switch collects and analyzes the seven-digit number to determine the destination of the phone call. The end office switch knows that someone from Bob's house is placing the call because of the specific port that it dedicated to Bob.
5. The switch analyzes the seven-digit called number to determine whether the number is a local number that the switch can serve.

NOTE If the same end office switch does not service Judy, Bob's end office switch looks in its routing tables to determine how to connect this call. It can add prefix digits to make the number appear as a fully qualified E.164 number when contacting Judy.

6. The switch determines Judy's specific subscriber line.
7. The end office switch then signals Judy's circuit by ringing Judy's phone.
8. A voice path back to Bob is cut through so that Bob can hear the ring-back tone the end office switch is sending. The ring-back tone is sent to Bob so that he knows Judy's phone is ringing. (The ringing of Judy's phone and the ring-back tone that Bob hears need not be synchronized.)
9. Judy picks up her phone (off hook).
10. The end office switch cuts through the voice path from Bob to Judy. This is a 64 Kbps, full-duplex DS-0 (Digital Service, Level 0) in the end office switching fabric to enable voice transmission.

Figure 7-11 demonstrates the call-flow necessary to complete an Internet phone call using a PC application.

Figure 7-11 *Calling with an Internet-Phone Application*



Both Bob and Judy need to be on the Internet or have some other IP network between their homes to talk to each other. Assuming this IP network exists or that both neighbors have a connection to the Internet, you can then follow this possible call-flow:

1. Judy launches her Internet phone (I-phone) application, which is H.323-compatible.
2. Bob already has his I-phone application launched.
3. Judy knows that Bob's Internet "name," or Domain Name System (DNS) entry, is bob@nextdoorneighbor.com, so she puts that into the "who to call" section in her I-phone application and presses Return.
4. The I-phone application converts Bob.nextdoorneighbor.com to a DNS host name and goes to a DNS server that is statically configured in Judy's machine to resolve the DNS name and get an actual IP address.
5. The DNS machine passes back Bob's IP address.
6. Judy's I-phone application takes Bob's IP address and sends an H.225 message to Bob.
7. The H.225 message signals Bob's PC to begin ringing.
8. Bob clicks on the Accept button, which tells his I-phone application to send back an H.225 connect message.
9. Judy's I-phone application then begins H.245 negotiation with Bob's PC.
10. H.245 negotiation finishes and logical channels are opened. Bob and Judy can now speak to one another through a packet-based network.

The example does not show all the steps and omits some details that a service provider needs to deploy a VoIP network. Because IP is a ubiquitous protocol, as mentioned in Chapter 6, when a call is packetized, it could be destined to your next-door neighbor or to a relative in Norway.

Summary

This chapter brought up many of the issues surrounding VoIP. Many of these issues, such as compression/decompression of the speech frame and propagation delay, are inherent to VoIP, and you can't do much to minimize these effects on VoIP networks.

With careful planning and solid network design, however, you can control and possibly avoid many problematic issues. Some of these issues are jitter, overall latency, handling delay, sampling rates, tandem encodings, and dial-plan design.

References

The following Requests For Comments (RFCs) will help you to continue researching VoIP:

RFC 1889—RTP: A Transport Protocol for Real-Time Applications
RFC 2327—SDP: Session Description Protocol
RFC 2326—RTSP: Real-Time Streaming Protocol
ITU-T Recommendation H.323
ITU-T G. specifications for codecs
ITU-T G.113 Voice Quality Specification
ITU-T P.861 Perceptual Speech Quality Measure(ment), PSQM
iLBC Codec—http://www.ilbcfreeware.org/
RFC 3550—Real Time Transport Control Protocol
RFC 3611—RTCP Extended Reports