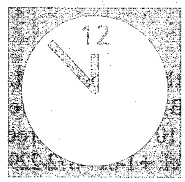
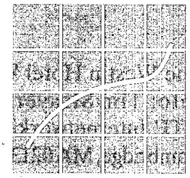




Real-Time Voice Over Packet-Switched Networks

Thomas J. Kostas, Michael S. Borella, Ikhlaq Sidhu, Guido M. Schuster, Jacek Grabiec, and Jerry Mahler
3COM

Abstract

We discuss the architecture and technical viability of transporting real-time voice over packet-switched networks such as the Internet. The value of integrating voice and data networks onto a common platform is well known. The telephony industry has proposed the ATM standard as a means of upgrading the Internet to provide both real-time and data services. In contrast, voice services may be added to traditional IP networks that were originally designed for data transmission alone. Here, we consider the feasibility and expected quality of service of audio applications over IP networks such as the Internet. In particular, we examine possible architectures for voice over IP and discuss measured Internet delay and loss characteristics.

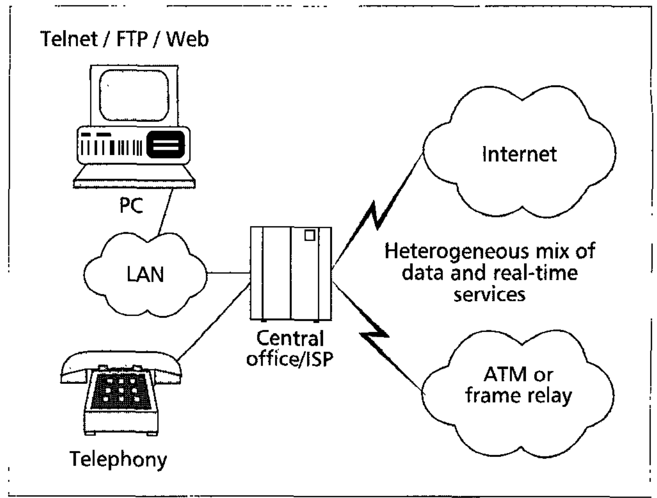
The concept of an integrated services network with both real-time and data services is not new. In fact, two alternate schemes are currently contending to provide all the services seen in Fig. 1. From one point of view, a new backbone from the telephony world — interexchange carrier (IXC)/asynchronous transfer mode (ATM) or frame relay (FR) — would provide all the required quality of service (QoS) levels for integrated services. From the other point of view, the existing Internet and corporate intranets would carry real-time voice traffic in addition to data.

Traditionally, the networking world has been divided along such lines. There has long been a telecommunications network that is circuit-switched and designed for point-to-point communication of real-time audio. Subsequently it has been adapted for the growing needs of data communication via modem technologies, ISDN, digital carriers, and, most recently, integrated services ATM and FR backbones. In contrast, there also exists a data networking world of store-and-forward packet technologies created primarily for data transport over local and wide areas. These networks are the vast collection of small and large IP networks that are intertwined in the form of the Internet and many partitioned intranets. Data networks comprise links, routers, bridges, and switches in the form of local and wide area networks.

Although these two networking worlds are coming together in a model of shared data, voice, and video, proponents of each view are looking at the future as an extension of their own technology. The telecommunications world has envisioned an integrated network via a large-scale ATM backbone that supports many levels of QoS, including traditional n x 64 kb/s voice. Since the telecommunications world has always been very QoS-focused, ATM is provisioned with mechanisms to provide different QoS levels. From the IP community, the long-term view is that real-time voice and video services can multiplex with existing data traffic. However, QoS has not been considered with the same intensity — the current Internet service model is flat, offering a classless, best-effort delivery service. As such, QoS is an ad hoc extension to the IP infrastructure. The next generation of IP, version 6, includes support for "flows" of packets between one or more hosts [1]. In conjunction with a hop-by-hop resource reservation protocol such as RSVP [2], end-to-end capacity can be set aside for real-time traffic. Much can be said of the trade-offs between IP and ATM solutions for providing integrated services. However, in this article we focus on the technical issues involved in supporting audio in the current Internet.

The dominant standard for transmitting multimedia in packet-switched networks is International Telecommunication Union (ITU) Recommendation H.323 [3, 4], which uses IP/UDP/RTP encapsulation for audio. RTP, the Real-Time Protocol [5], is a generic mechanism for supporting the integration of voice, video, and data. RTP headers provide the sequence number and timestamp information needed to reassemble a real-time stream from packets. H.323 does not provide any QoS guarantees, but does specify that a reliable transport protocol, such as TCP, be used for transmitting control information. The Voice Over IP (VoIP) standards committee is proposing a subset of H.323 for audio over IP [6]. Currently, a number of vendors have developed or are developing Internet telephony products that conform to this standard.

This article is organized as follows. The first section provides an overview of Internet telephony architecture and a discussion of the issues that must be addressed in order for an implementation to succeed. The second section discusses the factors affecting QoS, including the trade-offs between various end-user technologies and between different coding and error recovery schemes. It also examines an early implementation of Internet telephony. The fourth section provides an overview of our extensive Internet delay and loss measurements and examines their implications for VoIP deployment.



■ Figure 1. *Integrated services architecture.*

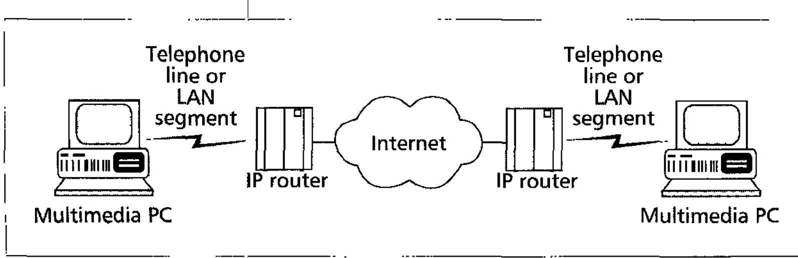
Internet Telephony Architecture

Description

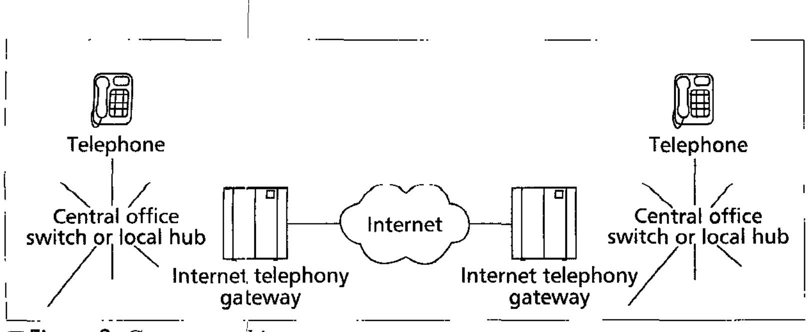
The current model of Internet telephony, shown in Fig. 2, is based on the assumption that two (or more) users have access to multimedia computers that are connected to the Internet. These computers can be on a LAN, as in the case of many corporate computers, or connected via telephone lines to Internet service providers (ISPs), as is the case for most home computers. All sampling, compression and packetization of the voice signal occurs in codec hardware and software on the sender's PC, while playout of the received signal occurs through a sound card on the receiver's PC. Alternatively, the codec could be implemented in hardware, possibly as part of a modem, network interface card, or sound board. A user places a call by specifying the IP address of the recipient, or by looking up the recipient's name in a public directory.[1]

In contrast to the PC-to-PC architecture, we can use a standard telephone to place and receive a phone call over the Internet. A home or office user calls an Internet telephony gateway that is located near a central office switch or local hub. Based on caller ID, the user is recognized (for authentication and billing purposes) and asked to enter the phone number of the intended recipient. The sender's gateway then looks up and initiates an H.323 session to the IP address of a gateway that is as close as possible to the recipient. The recipient's gateway places a call to the recipient's phone, and then the end-to-end communication can proceed, with voice sent in IP packets between the two gateways. Encoding and packetization occurs in the sender's gateway, while decoding and reassembly occurs in the recipient's gateway. The central office or hub may digitize the voice before passing it to the gateway, or a central office bypass may be in place that passes the analog signal to the gateway for digitization. In each user's local loop, the signal is analog. The gateways are implemented in hardware — digital signal processors (DSPs) and ASICs and are designed for low latency and to be able to handle many simultaneous calls.

Naturally, hybrid schemes will arise in which a gateway user places a call to or receives a call from a PC user. In such a situation, there must be a mapping or translation service between IP addresses and phone numbers. There are four different types of unidirectional paths: PC to PC (PC-PC), gateway to gateway (GW-GW), PC to gateway (PC-GW), and gateway to PC (GW-PC). In all of these architectures, both endpoints must employ the same voice codec.



■ Figure 2. *PC-to-PC architecture.*



■ Figure 3. *Gateway architecture.*

Implementation Issues

In this section we compare the architectures above by discussing a number of issues which must be addressed in order for a VoIP implementation to be feasible. In doing so, we also implicitly emphasize the advantages and disadvantages of each architecture.

Endpoint Requirements — In order for a PC to support real-time interactive voice, it must have considerable processing power. The computational requirements of voice codecs increases with the voice compression ratio (see the section on codecs). This is bad news for home users, because even fast modems are limited to about 33.6 Kb/s transmission rates. Corporate users connected to the Internet at T1 or greater speeds, as well as home users with ISDN or ADSL connectivity, may be able to employ

[1] *This will be the case if the recipient is using DHCP [7], with which an ISP server dynamically assigns IP addresses to dial-up users.*

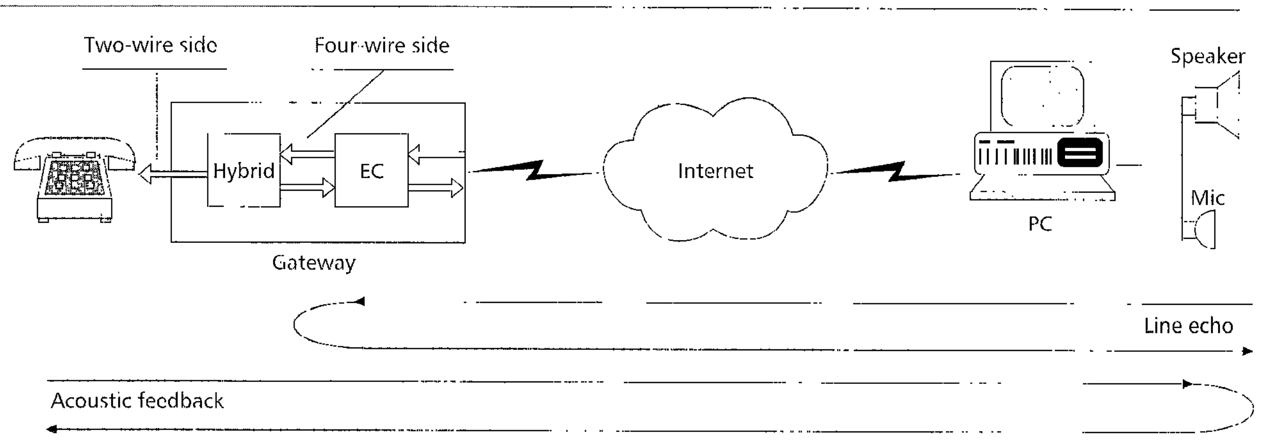


Figure 4. *Line echo and acoustic feedback.*

codecs with lower compression ratios, and thus lower processor utilization.

Echo Cancellation — There are two types of echo that can impact VoIP. The first is the usual far-end echo caused by the four-wire to two-wire hybrid conversion. End users will hear their own voice signal bouncing off the remote central office's line-card hybrid. Adaptive cancellation is programmed into outgoing long distance trunks to subtract the echo from the line on the four-wire side. If the hybrid conversion is done in a gateway, the gateway must also implement the echo cancellation. This echo will not exist in the PC-PC architecture. The second form of echo occurs when a free-air microphone and speakers are used, as is the case for most PC endpoints. The remote user's voice signal produced by the speakers is picked up by the microphone and echoed back to the remote user. The microphone may receive the PC speaker signal[2] from multiple paths (i.e., bouncing off the walls and ceiling of the room). This multipath echo is the most difficult type of echo to alleviate. Modern high-end speaker phones do a reasonably good job of echo cancellation with built-in DSPs. In the case of PCs, the echo cancellation can be designed into the codec. These two forms of echo are shown in Fig. 4.

Dual Tone Multifrequency (DTMF) Transmission — Touch-tone phones transmit phone numbers using a simple combination of two sinusoidal tones for each digit. Either these tones, or the phone number itself, must be reliably passed through the network in the GW-GW or PC-GW architectures. Most modern telephony hardware is guaranteed to reject tones with a duration of less than 20 ms and to accept tones with a duration of greater than 40 ms. Tone durations between these thresholds may be either accepted or rejected, depending on implementation. Some autodialers generate tones with a duration close to the 40 ms threshold. Linear predictive codecs, which are the most promising for VoIP, do not handle DTMF very well — they distort the on/off transitions at the beginnings and ends of the tones. This distortion may effectively shorten the duration of the tones so that a tone of appropriate length is rejected. Alternatives to tone transmission include using in-band H.245 [8], a control message protocol for multimedia, or initiating a separate RTP control stream for sending an ASCII representation of the phone number. When the number reaches the recipient's gateway, the gateway will regenerate the DTMF signal on the recipient's local loop.

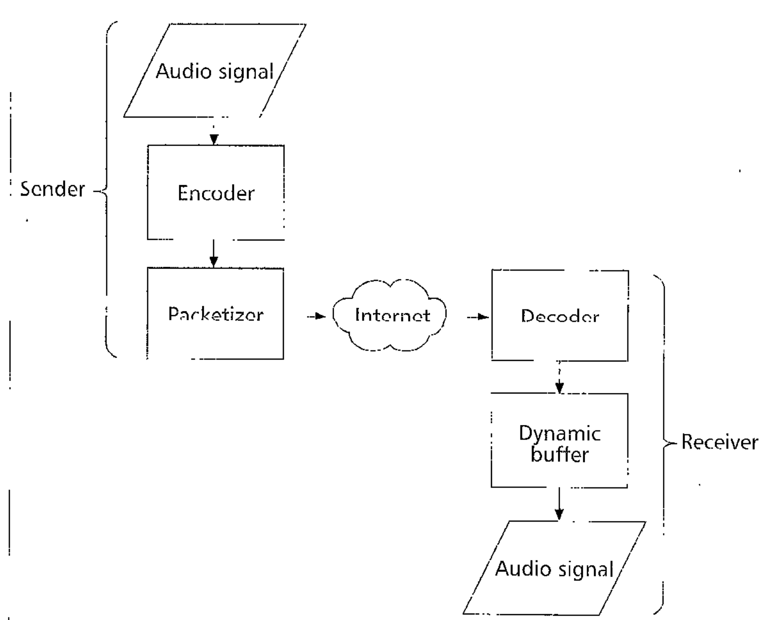


Figure 5. *VoIP data flow.*

Clock Synchronization — Whether the communication endpoints are gateways or PCs, low-frequency clock drift between the two can cause the receiver buffer (see the section on packet delays and losses) overflow or underflow. At the receiver, a clock synchronization mechanism must be in place that corrects for clock drift by comparing the timestamps of the received RTP packets with a local clock [5].

Billing — Gateways support the billing of local users. Thus, the GW-GW architecture supports billing of both the sender and recipient, while the GW-PC and PC-GW architectures support billing of the sender and recipient, respectively. The PC-PC architecture does not explicitly support billing in current IP networks. However, an overall billing paradigm for real-time services has yet to emerge. It is possible that the caller will be billed for the

[2] *In some rooms, the local user's voice may also be subject to multi-path echo.*

call while the recipient will be charged for airtime, not unlike the current cellular telephony model.

IP Address/Phone Number Mapping — In order for a call to be connected in any of the architectures, the sender must supply either an IP address (for a PC-based recipient) or a phone number (for a gateway-based recipient). For PC-PC communication, the caller must specify the IP address of the recipient — no mapping is necessary unless DHCP is employed. For all other architectures, the caller must specify the phone number of the recipient. The network maps the phone number to the IP address of the gateway closest to the recipient. These mappings must be stored in a dynamic distributed database, not unlike the Internet Domain Name Service (DNS). No standard mapping technique has yet emerged, although a promising candidate is the H.323 gatekeeper service [3].

Value-Added Services and Human Factors

The PC-PC architecture offers support for a number of value-added services, such as multiparty calls via IP multicast as well as voice mail, document sharing, and "distributed whiteboard" applications. The latter is particularly attractive to corporate users. Home users with modems are at a disadvantage due to their limited access bandwidth — coding and decoding of voice signals may not leave any spare processor or link capacity for other applications. PC telephony can also be integrated with video telephony when the latter technology matures. However, the gateway architecture has several important advantages over the PC architecture. First, the telephone is a familiar tool to a huge customer base, and does not require the user to buy a PC or learn new skills. Second, unless headphones are used, PC speakers decrease the privacy of calls, especially in cubicle-based corporate environments. Third, in order to receive a call on a PC, the PC must be turned on and connected to the Internet. And finally, PC telephony does not allow the user the mobility common to cordless telephony customers — the user may walk around the room, but cannot leave it.

Factors Affecting Quality of Service

To transport audio over a nonguaranteed packet-switched network, audio samples must be coded (usually with some form of compression), inserted into packets that have sequence numbers and creation timestamps, transported by the network, received in a playout buffer, decoded in sequential order, and played back as seen in Fig. 5. A symmetric scheme is used in the other direction for interactive conversation. All real-time transport schemes use this mechanism. In this section, we describe the barriers to the operation of these schemes, including requirements for codecs, bandwidth, delays, and losses.

Codecs

Internet telephony services must operate in a bandwidth-, delay-, loss-, and cost-constrained environment. This environment has been passed down to the codec development efforts of the ITU. Recently three ITU codecs, G.723.1 [9], G.729 [10], and G.729A [11], have been designed to work well in the presence of these constraints. Although they were designed with different applications in mind, they all are candidates for enabling VoIP.

Codec	G.723.1	G.729	G.729A
Bit rate	5.3 / 6.4 kb/s	8 kb/s	8 kb/s
Frame size	30 ms	10 ms	10 ms
Processing delay	30 ms	10 ms	10 ms
Lookahead delay	7.5 ms	5 ms	5 ms
Frame length	20/24 bytes	10 bytes	10 bytes
DSP MIPS	16	20	10.5
RAM	2200	3000	2000

■ Table 1. *Codec bit rates, delays, and complexity [12].*

Table 1 lists the characteristics of the codecs [12]. Bit rate refers to the output bit rate of the encoder when the input is standard 64 kb/s pulse code modulated (PCM) voice. Frame size is the length of the voice signal compressed into each packet. Processing delay is the delay required to run the encoding algorithm on a single frame. The lookahead delay is the amount of the next frame the coder uses to encode the current frame in order to take advantage of correlation. The effective one-way latency of the encoder is the sum of the frame size, lookahead, and processing delay. Typical decode delays are on the order of half the encode delays. Frame length is the number of bytes in an encoded frame (excluding headers). The DSP MIPS rating specifies the minimum processor speed, in millions of instructions per second, required for a DSP implementation of the encoder. Note that the DSP MIPS rating is not equivalent to MIPS ratings of general-purpose CPU microprocessors, such as those used in PCs. The latter, not specifically designed for the task, require greater speeds to encode or decode at the required rate. The required RAM for each encoder is given in 16-bit words.

From Table 1, we find that while G.723.1 provides the lowest bit rate, it also suffers from the largest delays. G.729 trades off a slightly higher bit rate and more complexity for a significant decrease in delay. G.729A provides the same performance as G.729, but with about half the complexity.

Bandwidth

A prerequisite condition for audio transport is, of course, that enough network bandwidth is available. Many users connect to the Internet at 33.6 kb/s rates or less. The small frame sizes of G.729 and G.729A allow for low-latency encoding, but also add a significant overhead if only one frame is encapsulated in an RTP packet. This implies that G.723.1 would be favorable to home PC users, who must share what little bandwidth they have with data traffic. Corporate users with direct access to Ethernet or T1 media may prefer to G.729A for its favorable delay characteristics.

Packet Delays and Losses

Network bandwidth is not the only requirement for quality audio. Each piece in the data flow pipeline, from coding to transport to reception to decoding, adds delay to the overall transmission. Some delays are relatively fixed, such as coding and decoding, while others depend on network conditions. Delay due to the transport network is nondeterministic in nature. If network conditions are poor, average packet delay and packet delay variance (jitter) will be high (on the order of 75–300 ms). Receive buffers can hide jitter at the cost of additional delay; however, packets delayed past the point at which they are supposed to be played out (the playout point) are effectively lost.

IP networks do not guarantee delivery of packets. Due to the stringent delay requirements of real-time interactive applications, reliable transport protocols such as TCP cannot be used. Packet loss is unavoidable; but can be compensated for by codec loss-concealment schemes. For example, G.723.1 interpolates a lost frame by simulating the vocal characteristics of the previous frame and slowly damping the signal [9].

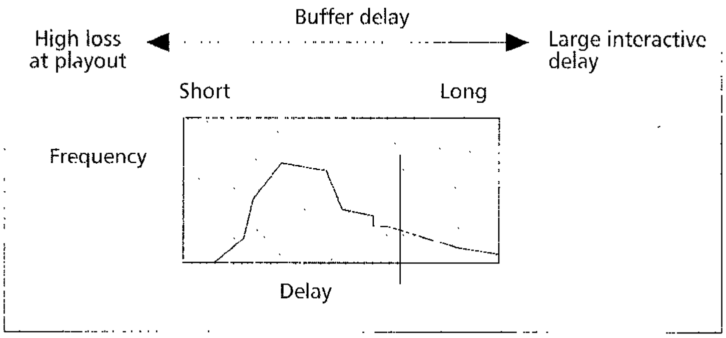


Figure 6. *Delay vs. loss trade-off.*

Informal tests in our lab have found that random, independent packet loss rates of up to 10 percent have little noticeable impact on G.723.1 speech transmission.

While single packet losses are of little consequence due to these schemes, loss bursts, like those produced by the Internet [13], can cause noticeable dropouts in the received signal. Forward error correction (FEC) schemes have been proposed to alleviate loss bursts of a small number of packets [14]. The effectiveness of FEC in the presence of loss concealment has not been rigorously studied. The drawback to FEC-based loss recovery is that in order to recover packet n (where n is the packet's sequence number), we need, at the very least, to successfully receive packet $n + 1$. Thus, we will be subjected to at least one extra frame delay in addition to the processing delay of the FEC encoding and decoding. These additional delays may cause the recovered packet to arrive too late (beyond its playout point) so that it is lost anyway.

An alternative loss recovery scheme involves adding copies of the previous k frames in the packet containing frame n. For example, when $k = 2$, packet n will contain frames n, $n - 1$, and $n - 2$. Then, if packet $n - 1$ is lost, we can still reconstruct frame $n - 1$ from either packet n or packet $n + 1$. Like other FEC schemes, this one will be most effective in scenarios in which we have a receiver buffer depth of several frames.

In the Internet, packet delay is highly variable. It may be to our advantage to dynamically modify the receiver buffer depth. Figure 6 shows the interaction between delay and loss for a representative delay distribution. The vertical line represents the playout point. As we move the line to the left, delays decrease but loss increases. As we move the line to the right, losses are reduced at the expense of higher delays.

If the network is not congested, it is possible to satisfy both delay and loss constraints. When network congestion is high enough, one of the two constraints must be broken. User studies [15] indicate that telephony users find round-trip delays of greater than about 300 ms more like a half-duplex connection than a conversation. However, user tolerance of delays varies significantly from user to user and from application to application. The most critical users required delays of 200 ms or less, while more tolerant users were satisfied with delays of 300–800 ms.

[3] *Modem processing delays vary from vendor to vendor and configuration to configuration. Informal tests in our laboratory have found that 33.6 kb/s V.34 modems with error correction and data compression turned off exhibit delays slightly less than 20 ms.*

Access Delays

In all proposed end-to-end architectures, users are vulnerable to significant hardware, operating system, and processing delays at one or more PCs (in PC-GW and GW-PC architectures, the gateway user will also be affected by the PC user's latencies). Modern PC sound cards typically add 20–180 ms of delay. V.34 modems will add a further 20–40 ms of DSP, equalization, and processing delay.[3] Modems will also incur transmission delays which are based on the ratio of packet size (including all headers) to bit rate. Processor and operating system delays are highly variable. In particular, computation- or communication-intensive applications will interfere with real-time applications. Gateway delays are also nonnegligible. A realistic design goal for maximum unidirectional gateway latency would be in the range of 20-40 ms, not including codec delays.

Trade-offs

For packet-switched audio transport, any of the various solutions that exist can be categorized by a trade-off space, as illustrated in Fig. 7. The three axes shown are bandwidth, delay, and computational complexity. Any given solution for packetized audio can be characterized by its required bandwidth, end-to-end delay, and computational complexity. Therefore, any particular solution can also be mapped into the space shown by these axes (note that zero-complexity, zero-bandwidth, and zero-delay solutions are not practically feasible). Each point on this surface results in decoded speech of the same quality. Note that the surface shown is a simple monotonic function. In reality, we expect it to be more complex.

The high point on the bandwidth axis represents traditional telephony, which requires a large amount of bandwidth (64 kb/s) but low computational effort, and exhibits low delay. The point near the delay axis represents streamed compressed audio over the Internet, which may suffer seconds of delay. For Internet and intranet audio applications, we suggest solutions that lie in the region of low to moderate bandwidth, intermediate delay, and high computational complexity. These solutions are generally based on dedicated DSP hardware, such as that found in a gateway, for lower-bit-rate coding.

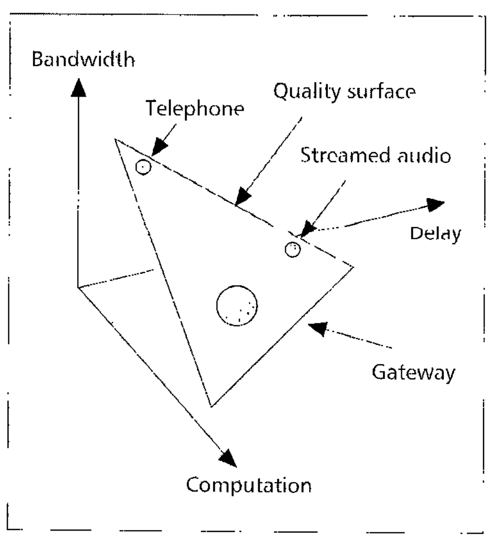
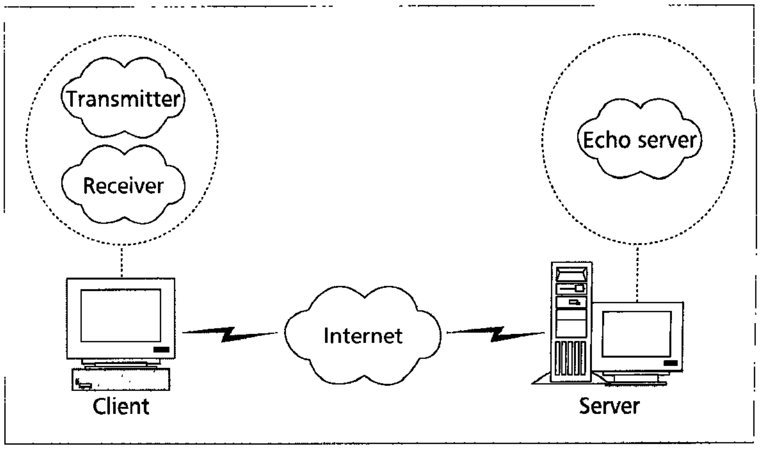


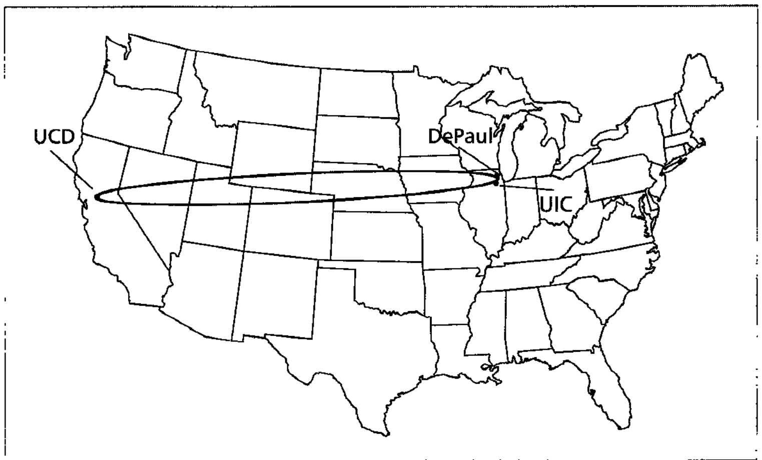
Figure 7. *Bandwidth, delay, and computation trade-offs.*

Implementation Examples

One of the earliest implementations of Internet telephony is the INRIA Free Phone [16]. Free Phone has been implemented entirely in software. The current version, 3.5, utilizes RTP as well as a separate signaling protocol. A number of codecs are supported, including 64 kb/s PCM [17], 32 kb/s and 24 kb/s adaptive differential PCM (ADPCM) [18], 13 kb/s Global System for Mobile Communication (GSM), and 4.8 kb/s linear predictive coding (LPC). Free Phone will attempt to keep the loss rate between user-defined watermarks by using adaptive redundancy and loss concealment techniques. In particular, lost frames can be reconstructed or interpolated through redundant copies in later packets or copying adjacent packets that were successfully received, respectively. As a last resort, silence is inserted into the stream when a lost packet cannot be rebuilt. Other freely available software implementations include Berkeley's VAT [19] and GMD Fokus's NeVoT [20]. A number of commercial VoIP products have been released as well, most software-based. Hardware-based commercial implementations are likely to become widely available in 1998.



■ Figure 8. *MID architecture.*



■ Figure 9. *Measurement paths.*

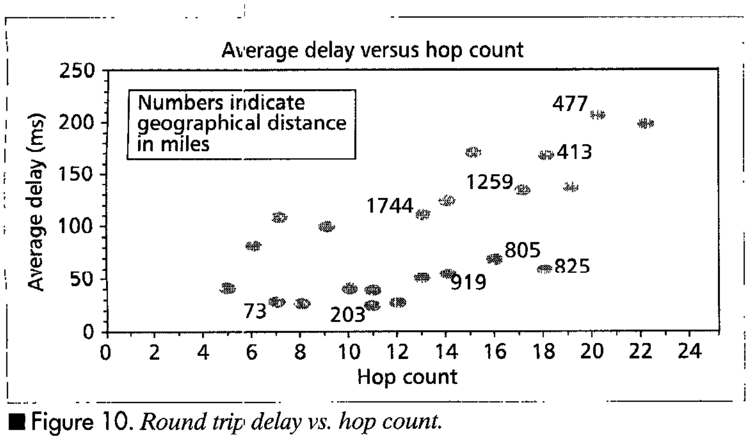
Internet Delay and Loss

In this section, we describe a set of sample delay and loss measurements of the Internet. Previous research [21–23] in this area has shown that round-trip Internet delays are often in the hundreds of milliseconds, and are usually correlated with packet loss. However, these studies were limited in scope, usually encompassing just a few hours or days of measurement. Our measurements were made on three Internet paths over a six-month period. They are powerful because they show long-term trends, as well as short-term and daily characteristics of the Internet. They also provide real-world numbers that can be used as a reference by VoIP implementers.

Measurement Barriers

In order to evaluate the impact of Internet delay on real-time applications, knowledge of unidirectional delay, delay jitter, and loss rates are necessary. These metrics must be available in order to determine the most effective codecs, transmission redundancy rates, and receiver buffer size. Accurately determining one-way packet delay from a client host to a server host in the Internet is difficult due to the need for synchronizing the client and server clocks. The clock resolution on the most popular computer architecture (the Intel Pentium family) is poor at 10 ms. This, as well as the variability of Internet delay, limits the accuracy of clock synchronization techniques to a few tens of milliseconds at best [24]. In theory, unidirectional delays can be measured accurately by equipping each endpoint with a global positioning system (GPS) satellite transceiver, but this is an expensive solution that does not easily scale.

Measuring the jitter of one-way Internet delay has been done [25] by comparing the relative difference between client-server and server-client delay. It was shown that the second moments of client-server and server-client delays are usually not symmetrical. This phenomenon is likely due to the asymmetric end-to-end routes common in the Internet [26]. Thus, we know that measuring round-trip delays and simply dividing the results by two does not necessarily give us realistic one-way delays. Given that there is no reasonably accurate technique for measuring unidirectional delays, we must use round-trip delays for our network latency metric. While our measurements cannot be used directly to infer one-way delays, they do provide us with general short-term and long-term trends of Internet packet delay.

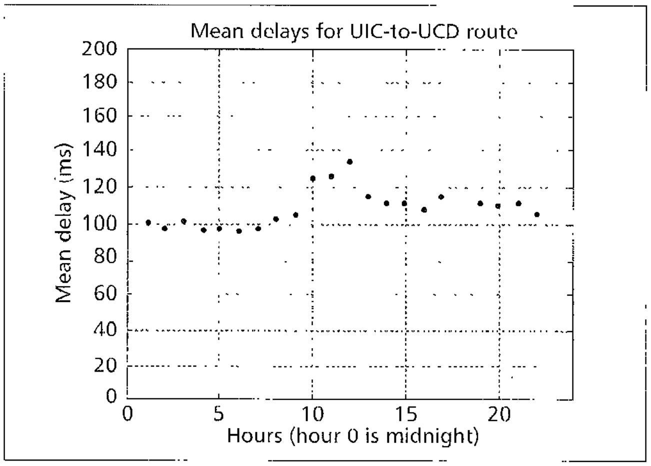


■ Figure 10. *Round trip delay vs. hop count.*

Measurement Architecture

The Internet `ping` service is commonly used to measure round-trip delay. ICMP echo packets with timestamps are used. We chose not to measure round-trip delays with `ping` because some routers will drop ICMP packets before UDP or TCP packets when they are congested. As an alternative, we have developed a UDP-based measurement package called MID.

MID consists of a client and a server to be run on two different hosts. The client program consists of two processes running in parallel: a transmitter and a receiver. The client's transmitter process uses the host's operating system timer to schedule transmission of a stream of UDP packets to the server with regular interdeparture times. Each packet contains a sequence number and a client identifier (CID), which is a random 10-bit string chosen at the beginning of a session. The server saves the last CID it has received. When the server receives a packet with a CID different than the one it has stored, it sets its packet count (PC) register to 0. When the server receives a packet with a CID identical to the one it has stored, it increments its PC by 1. Each packet received by the server is echoed back to the client, and, if not lost, is received by the client's receiver process. The receiver process logs the sequence number of all packets it receives. When the sender process has completed sending a stream, it transmits an end-of-transmission (EOT) token to the server, which responds with the current contents of its PC register. If either the EOT or PC packets are lost, the transmitter will timeout and retransmit the EOT until it successfully receives the server's response. Using this software, we can measure client-to-server loss rate and server-to-client loss rate, as well as round-trip loss rate.



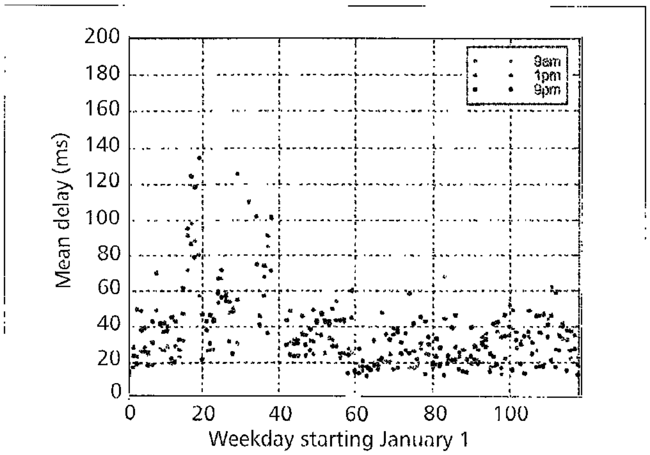
■ Figure 11. *Mean delays over a 24-hour period. Data measured on Friday, January 10 on UIC to UCD route.*

Sample Measurements

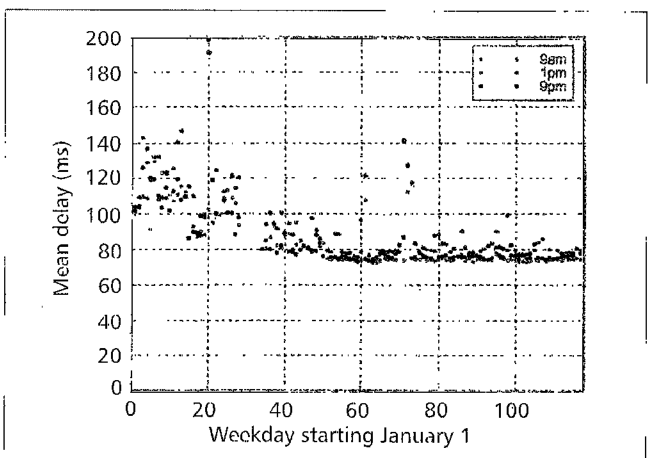
Delay and loss characteristics were measured from DePaul University to University of Illinois, Chicago (UIC), University of California, Davis (UCD) to DePaul University, and from UIC to UCD (Fig. 9). These sites were chosen based on availability and geographical dispersion. Each experiment was conducted as follows: once per hour, the client would transmit to a server for three minutes. UDP packet size was 80 bytes, and interdeparture times were 30 ms apart (except for the larger packet size, our experiments conformed with G.723.1). We refer to the data collected from such a run as a *trace*. The measurements produced 24 traces/path per day, from a collection period starting January 1, 1997, and continuing until mid-June 1997.

Delay and Hop Count Correlation — Observed delays are much more correlated to the "hop distance" between hosts than to geographical distance. As a quick but effective test, we used the ping program to measure delay versus hop distance. The results of this test in Fig. 10 illustrate this phenomenon.

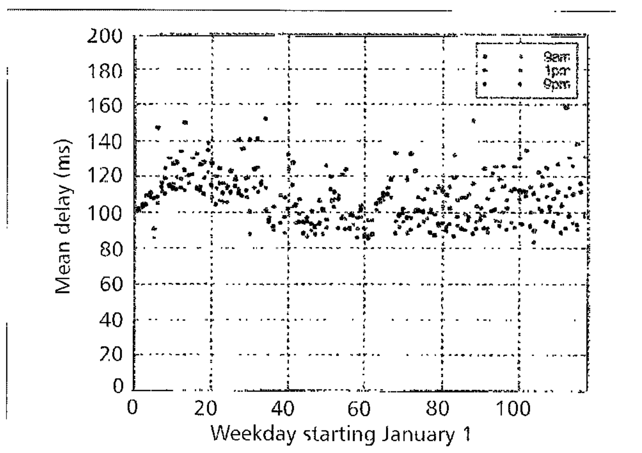
The results, obtained by running the `ping` program 100 times to each site during one hour and taking average delay, clearly shows the trend. Notice that although sites may be spaced the same hop distance apart, average delays experienced by packets sent to these sites reveals dramatically different values. For example, delay between Duke University and DePaul University was measured at 50 ms, while delay between University of California, Davis and DePaul was 110 ms, although both destinations are 13 hops away. The two sites that were 413 and 477 miles from the sender exhibited larger delays than some sites several times more distant. This is likely to be due to higher levels of congestion on the paths to the closer sites.



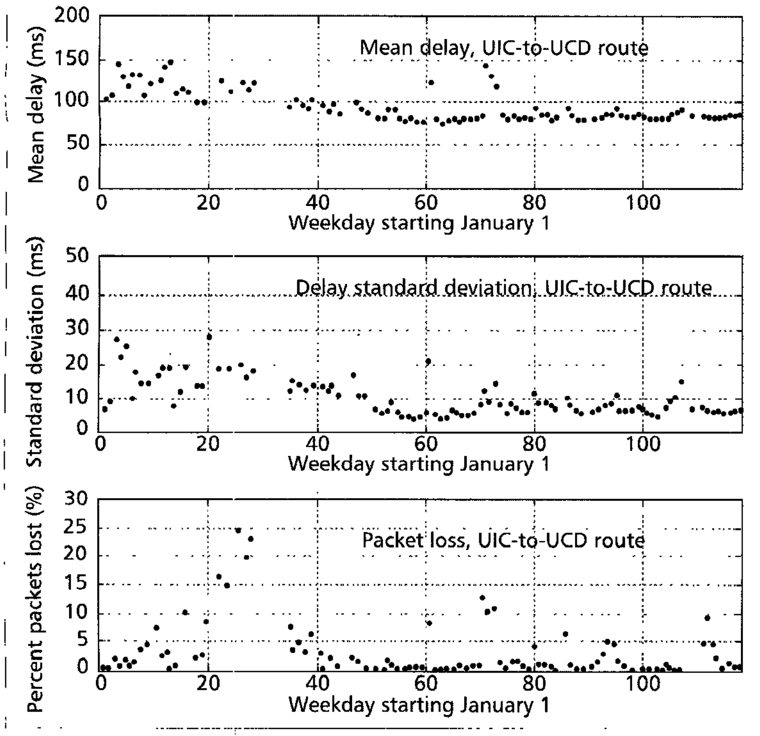
■ Figure 12. *Weekday mean delay on DePaul to UIC route, January–June 1997.*



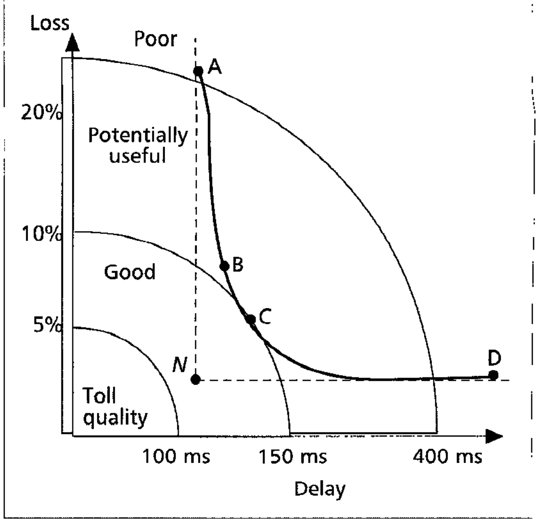
■ Figure 13. *Weekday mean delay on UIC-UCD route, January–June 1997.*



■ Figure 14. *Weekday mean delay on UCD - DePaul route, January–June 1997.*



■ Figure 15. *Round-trip delay, standard deviation of delay, and loss rate from UIC to UCD at 1:00 pm.*



■ Figure 16. *QoS mapping for unidirectional Internet delay and loss.*

Daily Mean Delay Samples — It has been established that Internet delay follows a diurnal cycle [23]. During the "working hours" of about 8 a.m. to 6 p.m., delays are usually greater than during "overnight" hours of midnight to 8 a.m. Evening hours of 6 p.m. to midnight exhibit moderate delays. In Fig. 11, note the increase in mean round-trip delay by approximately 20 ms during the middle of the business day. We have also observed a weekly cycle in which delays during a given hour are generally larger during weekdays than on weekends. Thus, directly comparing daytime versus nighttime or weekday versus weekend delays is misleading.

Figures 12–14 show the daily round-trip mean delays for all three paths. The average round-trip packet delays at 9 a.m., 1 p.m., and 9 p.m. are plotted as a function of the number of weekdays since January 1. For example, the mean delay at 4 o'clock on Monday, January 6 would be shown above the fourth position on the x axis since January 6 is the fourth weekday of the year. Mean delays for weekend hours are not shown on the same graphs as those for weekdays for the reasons described above.

In Figs. 13 and 14, the majority of the mean delays range from 70 ms and 160 ms. This is in sharp contrast to the mean delay points plotted in Fig. 12, the majority of which range between 20 ms and 40 ms. The previous figures represent routes between California and Illinois, which have more hops and a much greater propagation distance than those between UIC and DePaul within Chicago. We also notice that the delay at 1 p.m. in each graph is often larger than delays at 9 p.m. and 9 a.m. The increased delay is due to congestion during heavy-use hours. Poor quality of service is subjectively described in terms of higher delay means, standard deviations, and loss rates. These figures also show a trend (especially seen in Fig. 13) toward smaller mean delay from January to June.

We gain more insight by further characterizing the Internet by measuring standard deviation and loss rates. A sample of these measurements is seen in Fig. 15, which represents the UIC-to-UCD route at 1 p.m. Note the different scale in the standard deviation measurements from the mean delay. A positive correlation between mean delay, standard deviation, and loss is seen. During hours of higher delay toward the beginning of the year, greater standard deviation and loss rates are evident. Throughout the next few months, periods when delays are higher than their surrounding days are also marked by higher loss rates and standard deviations.

In many of our traces, packet loss is highly correlated. In other words, given that packet n is lost, the probability that packet $n + 1$ also is lost increases. In [13] packet loss for the month of April is studied in more detail, and we find that packet loss over all three paths is well modeled by a nonlinear Pareto distribution. The latter indicates that the Internet's packet loss process is highly bursty, with a disproportionate amount of individual packet losses occurring in a relatively small number of bursts.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Encode/decode	82 ms	30 ms	82 ms	30 ms
Access (source and destination)	40–80 ms	40–80 ms	100–340 ms	100–340 ms
Transmission (source and destination)	< 1 ms	< 1 ms	20–40 ms	10–30 ms
Internet (propagation and queuing)	30–100 ms	30–100 ms	30–100 ms	30–100 ms
Totals	152–262 ms	100–210 ms	232–562 ms	170–500 ms

NOTE: In this table we do not consider operating system delays which, in the PC-based architectures, may cause significant additional delays as well as occasional packet loss due to kernel buffer overflows.

■ Table 2. *Expected unidirectional delays using G.723.1 and G.729A.*

Discussion

Given the factors that affect QoS from the second section and measured Internet delays from this section, we would like to provide a mapping from delay and loss rate to QoS. As discussed earlier, such a mapping is very complex, and includes a great deal of site- and implementation-dependent variability, such as codec, access, network, operating system, and sound card delays. These factors cannot be easily quantified into a parsimonious analytical model. As an alternative, we present an approximate and intuitive representation of the QoS trade-offs involved in implementing VoIP.

Table 2 shows *approximate* unidirectional latencies that we expect to experience with Internet telephony applications using G.723.1 and G.729A in the GW-GW and PC-PC architectures. The encode/decode delays are derived from Table 1, assuming that decode delays are about half of encode delays. Access delays are estimated in the GW-GW case, based on current high-end hardware performance. Access delays in the PC-PC case include sound card delay and four modem processing delays (user and ISP modems at each endpoint). Transmission delays are assumed to be nearly negligible in the GW-GW case, due to the likelihood that gateways will be attached to Ethernet or T1 lines. In the PC-PC architecture, transmission delays are quite severe and occur at both ends. Our calculations in the latter case are based on the frame lengths from Table 1, assuming compressed IP/UDP/RTP headers in the best case and uncompressed headers in the worst case. Internet delays are based on our measurements from the 2000-mile Chicago-to-California link. Most of the round-trip delays that we measured were between 70 and 160 ms. We chose an expected unidirectional best case of 30 ms, but an expected worst case of 100 ms to cover a reasonable amount delay asymmetry. Note that although Table 2 may imply that G.729A is more promising than G.723.1, G.729A requires additional bandwidth and processing due to a higher header-to-payload ratio.

Figure 16 shows a hypothetical mapping from delay and loss to QoS. This mapping is meant to provide the reader with an understanding of the effect that various delay and loss rates, as well as buffer dynamics, have on QoS, and should not be taken to be based on user studies. We show four ranges of QoS based on codec properties and the discussion in [15]: *toll quality*, for delays of less than 100 ms and low loss rates; *good*, for delays 100–150 ms and slightly higher loss; *potentially useful*, for delays from 150–400 ms and higher loss rates; and *poor*, for delays greater than 400 ms and very high loss rates. We assume that a codec with a reasonably good loss concealment algorithm is being used; otherwise loss rates of 5–10 percent would result in a poor-quality connection.

Point N in Fig. 16 represents the total delay and loss that is experienced due to the "network" factors listed in Table 2, including all delays and losses up to the receive buffer. The system will never actually operate at this point. Point B represents a receiver operating under a certain amount of delay and loss. Part of this delay is due to the size of the receiver's buffer. Increasing the buffer size will bring us to point C, which increases the delay and decreases the loss rate. This decrease in loss is due to the fact that by pushing back our playout point we are not dropping as many packets. If we continue to increase this buffer toward point D and on to infinity, we reach the "network" loss rate since we no longer drop additional packets. On the other hand, if we decrease the buffer size toward point A and onward to a zero buffer length, we approach the "network" delay. However, the loss rate approaches 100 percent because the packets must arrive exactly at the playout point.

Given our assumptions, we expect that under the best circumstances VoIP implementations can reach a QoS acceptable by many users. Lower QoS may be acceptable by a large user base if appropriate pricing incentives are used (e.g., users may accept lower than toll-quality calls if they pay less per minute than comparable long distance rates).

Conclusions

Internet telephony promises to combine our separate data and voice networks into a single transport mechanism. The potential benefits for corporate and home users include the reduced cost of only needing to buy a single line to the outside world as well as lower per-minute telephone rates. However, there are significant barriers to acceptable QoS that must be overcome. Many of these barriers are in the form of trade-offs; finding the best combination of codec, access technology, and end-to-end architecture is challenging. The delay constraints for QoS are the most limiting, in the sense that we may be able to build faster hardware in the future, but we cannot increase the speed of light. These constraints can be somewhat mitigated by using Internet telephony on a local or metropolitan basis. In this article we have summarized the implementation and network issues that form these trade-offs and have presented six months of Internet delay and loss measurements. The latter can be used as a guide by VoIP implementers for evaluating the effectiveness of their schemes.

Acknowledgments

We are grateful to Ying Wu, Tom Gentles, Stan Naudus, and Vijay Nadkarni of 3Com for their insight into the implementation issues of Internet telephony and to Biswanath Mukherjee of the University of California, Davis for allowing us to use his equipment for measurements. We would also like to thank the anonymous reviewers for many useful recommendations.

References

[1] C. Huitema, *IPv6: The New Internet Protocol*, Prentice Hall, 1996.
[2] R. Braden *et al.*, "Resource Reservation Protocol (RSVP) — Version 1, Functional Specification," RFC 2205, Sept. 1997.
[3] G. A. Thom, "H.323: The Multimedia Communications Standard for Local Area Networks," *IEEE Commun. Mag.*, Dec. 1996.
[4] ITU Rec. H.323, "Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service," Nov. 1996.
[5] H. Schulzrinne *et al.*, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, Jan. 1996.
[6] Hyperlink at http://www.imtc.org/i/activity/i_voip.htm; http://www.imtc.org/i/activity/i_voip.htm.
[7] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, Mar. 1997.
[8] ITU Rec. H.245, "Control Protocol for Multimedia Communication," Mar. 1996.
[9] ITU Rec. G.723.1, "Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s," Mar. 1996.
[10] ITU Rec. G.729, "Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)," Mar. 1996.
[11] ITU Rec. G.729 Annex A, "Reduced Complexity 8 kbit/s CS-ACELP Speech Codec," Nov. 1996.
[12] R. V. Cox, "Three New Speech Coders from the ITU Cover a Range of Applications," *IEEE Commun. Mag.*, Sept. 1997.
[13] M. S. Borella *et al.*, "Analysis of End-to-End Internet Packet Loss: Dependence and Asymmetry," Preprint, 1997.
[14] J.-C. Bolot and A. Vega-Garcia, "The Case for FEC-Based Error Control for Packet Audio in the Internet," *ACM Multimedia Sys.*, 1997.
[15] ITU Rec. G.114, "One-Way Transmission Time," Feb. 1996.
[16] Hyperlink at http://zenon.inria.fr/rodeo/fphone/.
[17] ITU Rec. G.711, "Pulse Code Modulation (PCM) of Voice Frequencies," 1988.
[18] ITU Rec. G.726, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," June 1990.
[19] Hyperlink at http://www-nrg.ee.lbl.gov/vat/.
[20] Hyperlink at http://www.fokus.gmd.de/step/hgs/nevot/.
[21] D. Sanghi *et al.*, "Experimental Assessment of End-to-End Behavior on Internet," *Proc. IEEE INFOCOM '93*, Mar. 1993, pp. 867–74.
[22] J.-C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet," *J. High Speed Networks*, vol. 2, 1993, pp. 305–23.
[23] A. Mukherjee, "On the Dynamics and Significance of Low Frequency Components of Internet Load," *Internetworking: Research and Experience*, vol. 5,

1994, pp. 163–205.
[24] D. L. Mills, "The Network Time Protocol," RFC 1129, Oct. 1989.
[25] K. Claffy, G. Polyzos, and H.-W. Braun, "Measurement Considerations for Assessing Unidirectional Latencies," *Internetworking: Research and Experience*, vol. 4, no. 3, Sept. 1993, pp. 121–32.
[26] V. Paxson, "End-to-End Routing Behavior in the Internet," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, Oct. 1997, pp. 601–15.

Additional Reading

[1] D. Oran, "Voice Quality: An End-to-End Problem," presented at Nat'l. Commun. Forum, Nov. 1997.
[2] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," Ph.D. dissertation, UC-Berkeley, Apr. 1997.

Biographies

THOMAS J. KOSTAS received a B.S. degree in computer engineering and a B.A. degree in economics from Lehigh University, Bethlehem, Pennsylvania, in 1990, and an M.S. degree in electrical engineering from Northwestern University, Evanston, Illinois, in 1993. He is currently working at 3Com Advanced Technologies Research Center in Mount Prospect, Illinois. He is also pursuing a Ph.D. degree in electrical engineering at Northwestern University.

MICHAEL S. BORELLA (mike_borella@mw.ecom.com) received a B.S. in computer science and technical communications from Clarkson University in 1991, and M.S. and Ph.D. degrees in computer science from UC Davis, in 1994 and 1995, respectively. He is currently a senior engineer at the 3Com Advanced Technologies Research Center in Mount Prospect, IL. He has authored over 35 papers and patents in various areas of networking. His research interests include internetworking, multimedia, performance evaluation, traffic analysis, and operating systems.

IKHLAQ SIDHU received his B.S.E.E. degree from the University of Illinois, Urbana-Champaign, and M.S.E.E. and Ph.D. degrees from Northwestern University in Evanston, Illinois. He heads the Advanced Technologies Research Center, which is the applied research program for the Carrier Systems Business Unit of 3Com Corporation. The center's research agenda includes architectures, technologies, and new services for next-generation network access.

GUIDO M. SCHUSTER received an Ing HTL degree in elektronik, mess- und regeltechnik in 1990 from the Neu Technikum Buchs (NTB), Buchs, St.Gallen, Switzerland. He received M.S. and Ph.D. degrees, both in electrical engineering, from Northwestern University, Evanston, Illinois, in 1992 and 1996, respectively. In 1996 he joined the Network Systems Division of U.S. Robotics in Mount Prospect (now the Carrier Systems Business Unit of 3Com), Illinois, where he cofounded the Advanced Technologies Research Center. He has filed and holds several patents in fields ranging from adaptive control over video compression to Internet telephony. He is also the author of the book *Rate-Distortion Based Video Compression* (Kluwer, 1997). His current research interests are operational rate-distortion theory, source and channel coding, and networked multimedia.

JACEK GRABIEC received a B.S. degree from the Department of Electrical Engineering, Automation and Electronics of the University of Mining and Metallurgy, Krakow, Poland. He is currently a software engineer at the 3Com Advanced Technologies Research Center in Mount Prospect, Illinois. His research interests include modeling and simulation of communication systems, network performance evaluation, and data traffic analysis.

JERRY MAHLER received B.S. and M.S. degrees in electrical engineering from Northwestern University, Evanston, Illinois, in 1995 and 1997, respectively. He is currently a research engineer at the 3Com Advanced Technologies Research Center in Mount Prospect, Illinois. His research interests include networking, performance evaluation, multimedia, and wireless communications.