V. Hardman and O. Hodson. “Internet/Mbone Audio”
Handbook of Emerging Communications Technologies: The Next Decade.
Ed. Saba Zamir
Boca Raton: CRC Press LLC, 2000

13 Internet/Mbone Audio

V. Hardman and O. Hodson

CONTENTS

Interest in Internet audio has grown with the popularity of the Internet. The technology presented in this chapter covers a range of Internet audio systems: Mbone audio tools, Internet telephony, and audio streaming applications. Mbone audio represents the superset of the functionality of Internet audio applications and includes many aspects of the other two. This chapter therefore focuses on Mbone audio tools, and in particular discusses RAT (Robust Audio Tool). RAT is a second generation tool, incorporates an extremely rich set of functionality, and represents our own experience in Internet audio.

This chapter begins with an overview of technology that underpins Internet audio systems, audio compression algorithms and the Internet. It considers both speech and music algorithms, and the Internet, particularly with regard to the transport and application layer protocols. An overview of the components of an Mbone audio tool leads to an analysis of the current problems: gaps in the output audio and the effect on participants of an unnatural acoustic environment.

13.1 INTRODUCTION

Internet audio is a relatively new commercial application area that has grown with the popularity of the Internet and the availability of low cost audio hardware for multimedia PCs. For the purposes of this chapter, we consider Internet audio as covering three distinct areas: multimedia conferencing, broadcast applications such as music-on-demand, and Internet telephony. In comparison to existing technologies, such as the Public Switched Telecommunication Network (PSTN), Internet audio has the advantage that it can be integrated with other media, it can be mixed to provide multiway conferencing, and audio compression can be performed in software (and easily updated as algorithms improve).

The first packet audio system transmitted speech packets between a few sites in America over the ARPANET network. A follow-up project used a satellite connection to interconnect two sites outside the US (one of which was UCL, London) with the ARPANET network in the late seventies [Gold 73]. These early projects led to much subsequent research on both sides of the Atlantic (including the Universe, Admiral, and Unison projects [Clarke 90] in the UK). Research during the eighties developed individual media systems — an audio system and a video system — which used separate hosts and were integrated together using a Local Area Network (LAN). The recent proliferation of multimedia PCs, capable of compressing audio and video in real time, and the emergence of the global Internet have led to a rapid increase in the use of desktop collaborative multimedia systems running on a single host.

Despite many standardization activities in compression algorithms for the PSTN, most speech is still transmitted uncompressed. This situation is because of the need for efficient inter-working with other telecommunications operators and countries, and because telephone networks grow to cater for demand. In the

Internet, the situation is reversed, and there is often insufficient capacity to match demand. Internet audio competes with other traffic for a share of bandwidth, and compression is necessary to avoid congestion. Compression is also used to enable machines to be connected to the Internet through low capacity modem links, and to enable the heterogeneous Internet to carry all types of audio, such as toll quality speech and high quality music. Congestion in the Internet results in packet loss, because congested routers discard excess packets, and many schemes have been developed to overcome the effects of packet loss on audio. In the future, these schemes could be used to reduce costs, as services are likely to be charged for on a bandwidth-usage basis.

To transmit audio over the Internet, all a user needs is a multimedia PC, a headset, and network access. Internet audio tools are written as separate programs that a user starts from the UNIX command line or a windows interface. An audio tool smoothes the flow of arriving packets and converts them to a continuous stream of digital audio samples for playback to the audio device. The large processing capabilities of multimedia PCs mean that many techniques can be employed to improve the system at different stages in the pipeline: software compression algorithms can be swapped on an as-desired or as-needed basis to compress music as well as speech; error protection strategies can be matched to the quality requirements of the application and the current characteristics of the Internet; and digital audio techniques can be used to enhance the acoustic *feel* of the system.

Because of the global deployment of the Internet and the availability of Internet audio systems on low-cost general purpose PCs, the impact of Internet audio will be vast. An interactive aspect can easily be added to many existing applications, such as web-based distance learning, computer games, and mailing lists. Desktop multimedia conferencing applications which use multiway audio, video, and shared text will also evolve into more demanding applications, such as distance learning, video-on-demand, and telepresence. Current research in Internet audio is considering ways of improving the received quality of audio transmitted over a datagram network. It is also promoting congestion avoidance using adaptive applications that back-off transmission rate in the event of loss.

This chapter begins with a survey of the technology of Internet audio tools, looking first at compression algorithms and Internet protocols. Combining these two aspects into an audio tool is examined in terms of the structure of a typical Internet audio tool, with problem solutions for packet loss, and considering in detail options for enhancing the acoustic feel of the system. The chapter then discusses the impact Internet audio will have, and it concludes with an assessment of the current focus of the research community.

13.2 CHARACTERISTICS OF THE TECHNOLOGY

Networked-audio application development relies upon knowledge of both audio and networks. When the network is a packet network, and especially a best-effort one such as the Internet, this dual understanding is especially important because there is an interaction between network effects and the operation of the audio compression algorithm.

13.2.1 Characteristics of Audio

At the heart of any networked audio system is a compression algorithm which is used to reduce the bandwidth consumed. There are two distinct classes of audio compression schemes, voice and music, which address sound compression in complementary ways. Speech compression algorithms exploit the fact that the speech production process depends on anatomical movement (therefore changes relatively slowly) and has a limited frequency range. Music compression schemes are often based on a model of the human auditory system, which responds to a wider range of frequencies than encountered in speech but has decreasing sensitivity outside this part of the spectrum.

13.2.1.1 Characteristics of Speech and Human Conversation

Speech is limited in frequency range to less than 10 kHz, with most of the signal energy residing between 300 and 3400 Hz [Rabiner 78]. Higher frequencies ease listener identification but contribute little to comprehension. Speech is composed of a limited set of individual sounds, known as phonemes, which can be broadly split into two main groups: voiced and unvoiced. Voiced sounds are generated by air from the lungs being forced over the vocal cords, which vibrate in a quasiperiodic manner and produce a series of pulses of air. These air pulses pass along the vocal tract that allows certain frequencies to resonate, while attenuating others. For unvoiced sounds, the vocal tract produces a constriction, and air is forced through it, producing a turbulent flow. The two types of sounds have different types of frequency spectra. Voiced sounds, such as vowels, show a characteristic high average energy level, with distinct formant (resonant) frequencies. Unvoiced sounds show much smaller average energy, with a noise-like appearance in the higher parts of the frequency domain. Both groups of sounds are shaped by the dynamics of the vocal tract and interactions of airflow with the lips, teeth, tongue, and nasal cavity. Compression advantage in most modern speech compression algorithms is achieved by modeling the speech production process as the combination of a linear digital filter fed by an excitation signal. The digital filter models the vocal tract, and its parameters change quite slowly. The excitation signal is either a stream of regularly spaced pulses to represent voiced sounds, or a series of random pulses (a noise-like sequence) to imitate unvoiced sounds. For a greater understanding of the acoustic analysis of speech, see Kent and Read [Kent 92].

Conversation between two or more people usually consists of one person talking with the others listening. Changes of speaker occur when the speaker stops speaking (mutual silence) or the original speaker is interrupted (double-talk) [Brady 69]. Double-talk happens infrequently and lasts only for short periods of time. Periods of mutual silence typically occupy up to 50% of conversation time, and it is reasonable to assume that a two party conversation generates solitary talk-spurts for the other 50% of the conversation time. Because of these gaps in conversation, speech systems can save bandwidth by transmitting only when speakers are active. This technique is known as silence suppression and is widely deployed in telephony networks and Internet audio tools.

Interactive conversation also needs low end-to-end delay over the channel, if conversation patterns are not to break down. In the local analog PSTN, the delay has to be much smaller than this because of the generation of echoes, perception of which increases with increasing delay. With an end-to-end digital network, these echoes do not arise, and the delay value can be set at the maximum for interactive conversation (less than 250 ms end-to-end). [Montgomery 83]

13.2.1.2 Speech Compression Algorithms and Standards

Speech compression algorithms were originally standardized for use over the PSTN. The frequency response of the analog transmission parts of the telephone network has a frequency range which covers the greater part of the spectrum of human speech (300Hz to 3.4 kHz). Speech coding algorithms covering this range are known as telephone or toll quality.

The representation of discretely sampled audio as binary codes is pulse code modulation (PCM). The simplest scheme is to linearly quantize samples over the range of amplitudes. For speech, it has been found that 11 bits of precision are necessary for linear quantization not to be noticeable [Rabiner 78], but most computer systems round this up to 16 bits/sample. Linear quantization results in a varying signal-to-noise ratio with the amplitude of the signal. A widely used alternative is logarithmic companding which scales the signal logarithmically. The International Telephony Union (ITU) standard G.711 defines two companding schemes (A-law and μ-law), which both represent samples by 8 bits. Both schemes have near constant signal-to-noise ratios across the signal amplitude range and are near transparent to the human listener.

The mechanics of speech production are such that there is a high degree of correlation between adjacent samples. Further reductions in the number of bits per sample can be achieved by using a predictor in conjunction with an adaptive quantizer. This approach is described as adaptive differential pulse-coded modulation (ADPCM). ITU standard G.726 represents speech samples with 2, 3, 4, and 5 bits per sample using such a scheme. Another standard, G.722, covers speech in the range 0-7kHz (wideband speech) using this technique on sub-bands within the frequency range.

An alternative way of representing speech signals is linear predictive coding (LPC). LPC coders model the speech production process as the linear sum of earlier samples (digital synthesis filter), which is fed by an excitation (residual) signal (see Figure 13.1). The encoder algorithm estimates the coefficients that will be used in the synthesis filter (usually 8-10 filter coefficients) and tries to identify the pitch period, which will be used to generate the pulses in the excitation or residual signal. Since the method of pitch prediction is not very accurate,* modern techniques are hybrid arrangements, where the synthesis filter is fed by an excitation signal that is a preserved version of the real vocal tract excitation signal, not an estimate of its properties averaged over a window of samples. Linear prediction-based compression algorithms are such as LD-CELP (G.728, which gives an output bit-rate of 16kbps), and standards developed for wire-less communication, such as RPE-LTP (GSM speech coder, 13 kbps bit-rate), CS-ACELP (G.729 standard, 8kbps output bit-rate).

* Some sounds, such as voiced fricatives, cannot be represented by this model.

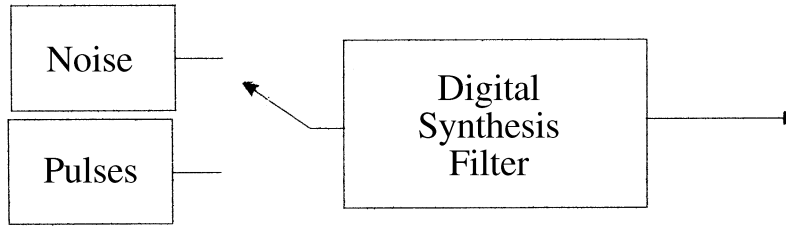

Figure 13.1 The linear predictive coding model of the speech production process

LPC-based codecs are named according to the representation of the residual; vector quantisation uses codes to represent the excitation signal, and the compression algorithm is called code-excited linear prediction (CELP). A comparison of algorithm against bit-rate can be found in Table 13.1.

For more information on speech coding see Spanias.

13.2.1.3 Characteristics of the Human Auditory System

The human auditory system is sensitive to frequencies from 20Hz up to 20kHz, and is able to sense sounds with a dynamic range of 130 dB. These properties are largely attributable to the anatomy of the ear, which consists of mechanisms to pass vibration to the inner ear oval window and on to the cochlea, a tapered tube wound into a helix and divided along its length by the basilar membrane. Sensitive hair cells along the length of the basilar membrane convey pressure variations to the aural nerves. The shape of the cochlea and the position of the hair cells along it determine the frequencies the hair cells are sensitive to and, if sensitive, the degree of sensitivity. Groups of hair cells act as overlapping band pass filters, and this behaviour is represented in the critical band model [Barnwell 96].

Popular music coders use the critical band model and also exploit the hearing phenomena of aural masking whereby a tone in a critical band is able to mask another tone in the same band; a musical signal in one band will mask noise in that

TABLE 13.1
A Comparison of Toll Quality Speech Coding Standards

Coding scheme	Sample/Frame based	Frame length (ms)	Bit-rate (kbps)
Linear 16	Sample	N/A	128
G.711	Sample	N/A	64
G.726	Sample	N/A	16/24/32/40
G.723.1	Frame	(4*7.5) 30	5.3/6.3
G.728	Frame	(8*2.5) 20	16
G.729/729A	Frame	10	8
GSM Full rate	Frame	20	13

band, but not noise in another band. Masking occurs because of the presence of higher level signals near the masked signal. It can occur in both the time (both forward and backward temporal masking occurs) and frequency domains. In music coders, spectral analysis of a small section of the incoming audio (along the lines of the critical band model) allows masked tones to be not represented in the output coded stream. Spectral analysis in this way also allows noise levels in bands where music tones exist to be higher than in bands where there are no music tones. In this way the compression algorithms produce transparent quality, with no perceived noise having been added.

13.2.1.4 Music Compression Algorithms

Two widely deployed music compression algorithms used on the Internet are MPEG-1 and AC-3. MPEG-1 exists as three variants: layers I, II, III, with each layer having a lower bit-rate and higher complexity for a given quality. A diagram of the principles of music compression can be seen in Figure 13.2.

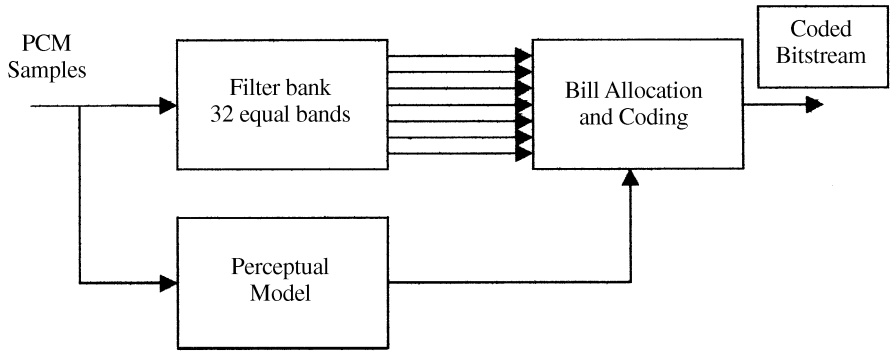

Figure 13.2 Principle of MPEG-1 Audio Encoder

MPEG-1 uses the perceptual model to determine how to allocate bits to the output of the filter bank. In layer I the perceptual model handles frequency masking. Layer II improves encoding efficiency by encoding across larger groups of samples, using improved coding techniques, and adding temporal masking to the perceptual model. Layer III improves the resolution of the filter bank by performing a modified discrete cosine transform (MDCT) on the output of the filter bank. It is also able to perform improved temporal resolution by changing the length of the sample blocks passed through the filter bank. Layer III also has a greatly improved bit-allocation scheme and uses entropy coding to provide additional bit-rate reduction. The bit-rates of the MPEG-1 layers can be seen in Table 13.2.

For a more detailed description of the principles of music coding in general, see Pan [Pan 93].

TABLE 13.2
Per Channel Bit Rates for Each of the MPEG1 Layers. If Stereo is Desired, Twice the Output Bit rate is Needed

Layer no.	Per channel bit rate (kbps)
Layer I	192
Layer II	96
Layer III	64

13.2.2 Characteristics of the Internet

The Internet transports information over a wide range of networking technologies. Central to the Internet philosophy is the concept of protocol layering and encapsulation, in which each successive protocol layer adds a more refined service to that offered by the layer below. The Internet Protocol (IP) conceals details of the underlying network technologies and offers a best-effort delivery service to the layer above. The basic unit of IP transmission is the datagram, which is carried independently from all other datagrams — the network is connectionless and there are no reception guarantees. For compressed continuous media, such as audio, the best-effort service of the Internet represents a serious problem for successful transmission. Audio compression algorithms are designed to operate on a stream of samples, not a series of unrelated packets; in order for the stream of samples to be successfully decoded, they must be presented to the decoder in the same order and essentially without loss.

This section provides an overview of Internet network characteristics, focusing specifically on those that relate to real-time audio transmission. The operation of the connectionless datagram switch (IP router) leads to effects that are especially important in real-time audio transmission: jitter and packet loss. Also, transport and application layer protocols build a set of tailor-made facilities for each application, and this section focuses on those relating to real-time media. In addition, multiway delivery mechanisms that are being used in commercial and research applications are considered. A better understanding of the Internet in general can be found in Tannenbaum and Peterson [Tannenbaum 96][Peterson 96].

13.2.2.1 Routers and Network Effects

The Internet consists of a large number of routers, which forward datagrams between hosts; forwarding is on the basis of the destination address in each packet. Routers make routing decisions on a per packet basis, which enables the network to be more robust, albeit at the expense of efficiency. Internet routers offer a best-effort service with no guarantees on timely delivery or even actual delivery.

Variations in the delay encountered by datagrams across the network occur because they are buffered in routers; buffering is used to absorb traffic burstiness. The amount of delay experienced at any router varies with the handling of routing updates, the commencement and termination of streams, the handling of one-off

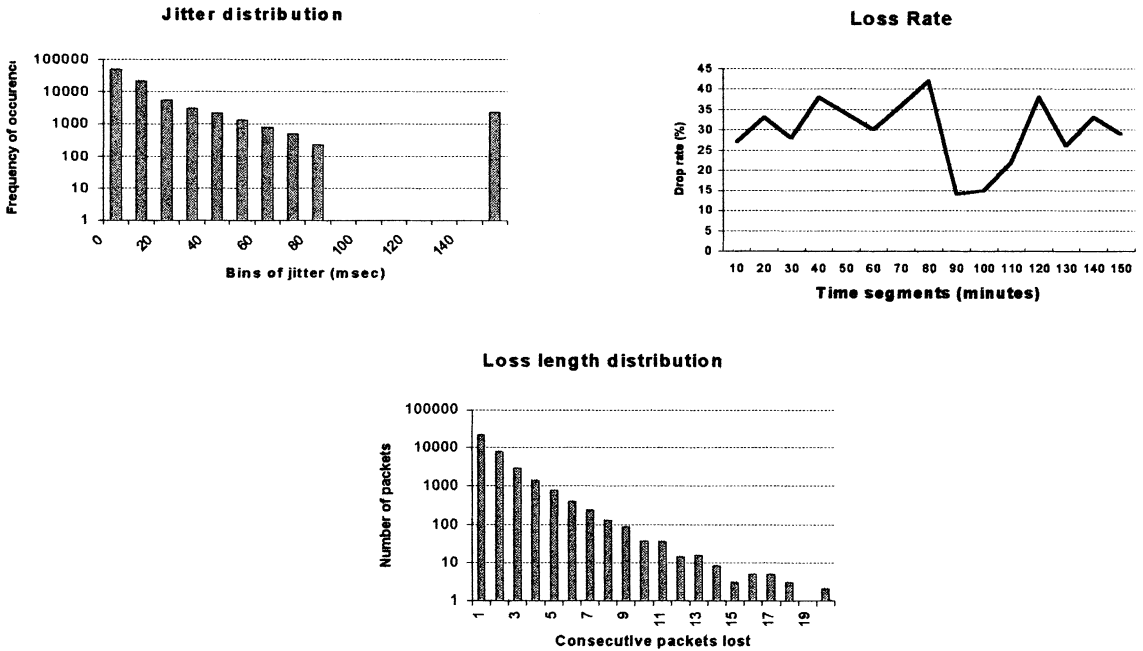

Figure 13.3 Graph showing sample loss and delay variation for real-time audio between two machines[OH1].

requests, etc. This means that the end-to-end delay between any two hosts is perpetually changing.

Routers suffer congestive losses in two ways. First, they might be overwhelmed by the volume of datagrams they receive and resort to discarding the excess. Second, the volume of incoming traffic might be such that traffic directed to an outgoing interface is greater than the bandwidth of the output link. The current mechanism for servicing packets buffered at a router is on a FIFO basis, and the tail of the buffer is dropped during congestion; consequently packets can be lost singularly or in bursts.

Real-time audio tools need to be able to adapt to both of these problems, and that issue is discussed in Section 13.3.2.2.

13.2.2.2 Internet Protocol Stack

The IP protocol provides a best effort service using a minimal set of facilities, including source and destination identification, packet fragmentation and reassembly, header checksum, and type of service (widely ignored by routers) etc. Any further facilities required by an application are provided by transport and application layer protocols. The diagram below (Figure 13.4) shows how an application can build up a set of facilities by using different transport and application layer protocols.

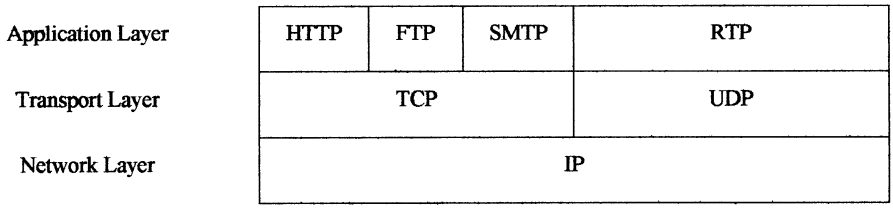

Figure 13.4 An Illustration of the IP protocol stack with a selection of application layer protocols.

13.2.2.2.1 Transport Level Protocols

Transport protocols, such as the Transport Control Protocol (TCP) and the User Datagram Protocol (UDP), essentially provide a multiplexing service to applications. TCP also provides congestion control by scaling the rate of transmission, and reliability achieved via re-transmissions. TCP is inappropriate for real-time applications because the delay required for retransmission in the event of loss is too large for interactivity considerations. The arrival of a retransmission also means that applications cannot determine the amount of instantaneous available bandwidth. In contrast, UDP provides no delivery guarantees, or congestion control, and is consequently more amenable to real-time data. Providing bandwidth scalability and packet-loss robustness to real-time applications is an open research issue.

13.2.2.2.2 Multimedia Application Protocols

Application layer protocols provide services to applications by enhancing the service provided by the transport layer. Real-time media-specific application layer protocols can be further subdivided into those responsible for the streaming multimedia information and those responsible for negotiating and controlling the flows. Control protocols use TCP at the transport layer (because they need to receive all of the control data), and the streaming protocols use UDP (because they need timely delivery of data rather than guaranteed receipt).

A widely used multimedia streaming protocol is the RTP, Real-time Transport Protocol [RTP], which has an associated control protocol — RTCP, the Real-time Transport Control Protocol. RTP essentially provides applications with sequence number information for ordering purposes, media timestamping for calculating rendering times, a payload descriptor for codec identification, and a marker bit for application-defined purposes, such as talkspurt starts. RTCP conveys sender and receiver statistics and participant information. The RTCP bandwidth is limited to five percent of the total consumed by both RTP and RTCP streams. RTP is designed primarily for audio and video, and it specifies statically assigned payload types for popular compression algorithms (others can be defined through an external codec mapping mechanism). Some companies, such as Microsoft and Apple, are now working on generic coding schemes which can multiplex multiple inputs into a single stream and include not only codec-type information, but also the codec itself [ASF][RTP Generic].

A number of proprietary streaming protocols have found prominence in the marketplace with products such as RealNetwork's RealAudio and Xing Technology's Streamworks. These protocols are designed for streaming applications, which are tolerant to delay, and consequently include retransmission capabilities in the event of loss. RealNetwork's RTSP, Real-time Streaming Protocol [RTSP] has met with approval within the Internet Engineering Task Force (IETF) and has wide industry support. Despite its name, the primary purpose of RTSP is to control media streams, and it includes functionality, such as rewind/fastforward, and synchronization.

There are two competing standards for conference negotiation: the IETF's Session Invitation Protocol (SIP) [SIP] and the ITU's H.323 protocol [H.323]. In addition to call negotiation and control, H.323 also specifies which audio and video codecs may be used, transport mechanisms, and signaling. In contrast, SIP is a lightweight protocol that is not bound to specific audio and video compression algorithms.

13.2.2.3 Multiway Media Delivery

It is often necessary to deliver media data from one-to-many or from many-to-many. IP Multicast [Deering 89] was specifically designed for this purpose, but it is not yet deployed in all parts of the network. A widely used interim solution is one of packet reflectors, which forward and replicate packets between hosts. The reflector approach is used for commercial streaming products, such as RealAudio. However, it has a number of inherent drawbacks as a multiway delivery mechanism: the reflector is a single point of failure, reflector placement is ad-hoc, and the delivery tree is extremely likely to be suboptimal as a result. Reflector schemes might

eventually be replaced as more Internet Service Providers (ISPs) support multicast. Several streaming products actually support multicast but currently use reflectors.

13.2.2.3.1 Multicast Backbone (Mbone)

The Mbone provides multiway communication facilities, by setting up a distribution tree from a sender to all interested receivers. In its simplest forms, there is a separate distribution tree for each sender. Receivers can join or leave a conference at will, and there is no explicit means of membership control. A receiver that wishes to join a conference sends an IGMP (Internet Group Management Protocol) join message to the local router. The local router then sends a graft message back towards the distribution tree. A receiver that wishes to leave a conference sends an IGMP leave message. If there are no other receivers attached to the router, the router will then send a prune message back towards the root of the tree. The early forms of multicast routing protocols were based on source-based distribution trees. More recently, newer multicast routing protocols set up shared distribution trees, which may be slightly suboptimal, but which provide other advantages. To use the multiway communication facilities provided by multicast, the sender merely has to use a class D address (in the range 239.225.225.225 to 225.0.0.0), assuming the local router is multicast capable. For a fuller discussion of multicast routing, see Deering [Deering 89] and Peterson [Peterson 96].

13.3 STRUCTURE AND TECHNOLOGY OF PACKET AUDIO SYSTEMS AND TOOLS

The availability of cheap multimedia capable PCs means systems that are able to handle real-time audio processing are widely deployed. Current processor performance means that quite highly complex compression may be performed in software, and, because many audio and video cards are now supporting compression with onboard DSPs, many more algorithms can be run in real-time. Networked audio tool programs can be developed with a minimal set of requirements: an Internet connection, headset or loudspeakers, and an audio card, all of which are standard components on many PCs. These factors have encouraged the rapid adoption of the new technology, and have led to a proliferation of different types of audio tool. There are three audio tool types that stem from different application requirements: Internet telephony (point-to-point, low delay), audio-on-demand (point-to-point, delay tolerant), and Mbone audio (multiway, low delay).

This section focuses on the technology of an Mbone audio tool and is based on our experience with the Robust Audio Tool (RAT) developed at UCL [Hardman 98], but other examples exist, such as Vat [Jacobson 92]. The same techniques are employed in point-to-point and audio-on-demand applications. An Mbone audio tool is a program that interfaces to the audio hardware, the Internet protocol stack, and the user interface. Writing a successful audio tool requires knowledge of network characteristics, audio card hardware, headsets, microphones, acoustics, and compression algorithms.

Internet/Mbone audio tools are used in many applications piloting projects and by users worldwide (research software source code and executables are often made

freely available). This has allowed researchers to identify and solve some of the problems with current audio tools — those relating to transmission over the best-effort Internet and to manipulation of real-time audio on a general purpose host.

13.3.1 Low-Cost General Purpose Hosts Suitable for Multimedia Internet Access

Internet audio tools collect samples from the audio hardware and put them into packets. Most general purpose workstations and PCs have the IP stack implemented in the operating system (including selection of the transport layer protocol, UDP or TCP/IP, and unicast or multicast facilities). Multimedia-specific application layer protocols (e.g., RTP) are implemented in the user program, in-line with the recommendations of application layer framing [Clark 90]. In a multimedia PC or UNIX workstation, the usual method of communication with an audio card is the operating system device driver that collects samples from the hardware, and a chunk of memory containing perhaps 20 ms of linear PCM audio samples is given to the user program.

Our audio tool is supported on a range of general purpose workstation and PC operating systems (Solaris, SunOS, IRIX, FreeBSD, Linux, HP-UX, Windows 95/98, and Windows NT). The program code is written in C, with the user interface written in TCL/TK [Ousterhout 94], a portable code scripting language. Hardware-specific code that interfaces with the audio hardware is determined at linkage time and is hidden from the program by a standard audio interface.

On a UNIX workstation, reading audio from within a user program is via the *select* system call, which indicates that one or more buffers of audio are ready. The number available depends on how long it has been since audio was last read from the device, and the size of each buffer is fixed (perhaps 20 ms each — the value can be set). From within a program operating under Windows reading audio is different. When an application opens the audio device under Windows, it creates a multimedia task that generates messages every time a block of audio is available. The audio application uses a callback to handle these messages, and the audio application can sleep between messages. Both methods of reading audio mean that the audio tool does no processing while waiting for audio from the device and that the CPU can be used by other programs.

The preferred features of an audio card are full-duplex operation (standard on all workstations and available on PCs), 16-bit sample audio resolution, and multiple sampling frequency support. A number of legacy cards that deliver 8-bit samples are usable, but they result in lower quality, especially because they are often compressed by the audio card using μ-law PCM, which leaves artifacts in the audio. In order to cater to the full complement of audio codecs, the card should support a range of sampling frequencies (8, 16, 32, 44.1, 48 kHz). For more computer-intensive facilities, such as fully localised 3D audio, audio cards with on-board DSPs are emerging.

Some audio/video hardware supports on-card compression through high capacity digital signal processors, which can be utilized by an audio tool if an appropriate API is available. Most of the major processor families have single-instruction multiple data (SIMD) operations, such as Intel MMX and Sun VIS architectures,

that enable general purpose processors to enjoy strong DSP performance. SIMD instructions allow many audio operations to be optimized, such as filtering and sample rate conversion.

Operating systems (such as Microsoft Windows) provide audio compression algorithms that can be accessed through a standard interface, the audio compression manager (ACM), that provides access to a range of codecs (G.711, G.721, GSM, MPEG-I layer-3). In addition the ACM has the ability to perform sample format conversion and sample rate conversion.

13.3.2 Technology of an Internet/Mbone Audio Tool

The basic structure of an audio tool can be seen in the block diagram of Figure 13.5. The user interface presents controls, such as volume, together with run-time statistics reporting mechanisms. A silence detection algorithm determines if the buffer is speech or silence, because Mbone audio tools conserve bandwidth by transmitting audio only when the user is speaking. Buffers determined to contain speech samples are compressed and formatted into packets with an RTP header. The packet is then passed to the operating system for standard UDP/IP transmission using a unicast or multicast address. The address is specified as a command-line option at program start.

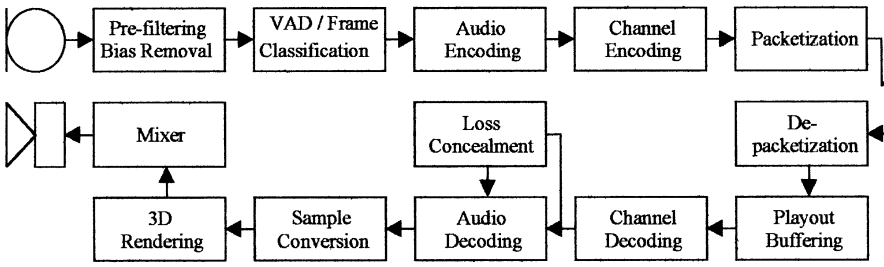

Figure 13.5 Block diagram of an audio tool

If the compression algorithm generates a continuous bit-rate, packets are generated at regular intervals. Because of the characteristics of the Internet, jitter will be added to datagrams as they traverse the network. At the receiver, a buffer is needed to smooth out the effects of jitter. After software decompression, mixing takes place before the audio is passed to the device driver for play.

13.3.2.1 User Interface

The user interface of our audio tool (see Figure 13.6) consists of a list of participants, volume controls, and activity bars. The names of active speakers are highlighted to help participants identify who is talking; when a participant stops talking, the highlighting slowly fades. Fading helps users identify who has spoken recently, which is especially useful in a multiway conference. An options page (launched by

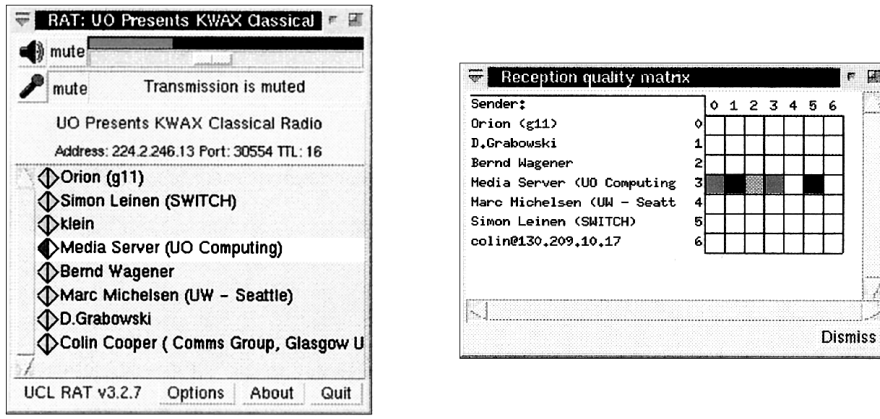

Figure 13.6 RAT user interface

clicking a button on the main window) is available to allow the more experienced user to change compression level/packet rate.

13.3.2.2 Buffering

Transmitting continuous media in a flow over a packet network incurs delay. That delay has the following components:

$$\text{Delay at receiver} = \text{packetisation delay} + \text{network propogation} + \text{buffering at routers}$$

Delay is associated with packetisation because the transmitter must collect enough samples to fill a packet before it can send the packet onto the network. This delay is usually fairly stationary during the lifetime of a flow. Network propagation delay is often fixed or is more variable when retransmission at the data-link level occurs, such as over mobile links. Buffering at routers introduces a delay component that depends upon other traffic in the network and varies considerably.

Buffering is needed at the receiver to remove the effects of jitter from an incoming flow. The buffer adds delay to early datagrams, and correspondingly less to later datagrams, to increase the likelihood of many packets arriving in time for playback. Each incoming datagram of audio therefore has a play-out point associated with it. Average jitter levels are different for every flow, and they change with congestion on the network which means that an adaptive play-out point calculation is performed for each source. Because some packets might have spuriously high levels of jitter added to their arrival time or might not arrive at all, a budget of slightly less than 100% is commonly used to calculate the amount of buffering required. Recommendations for calculating the play-out point can be found in Schulzerine, et al. [RTP], Moon, Kurose, and Towsley [Moon 98], and Diot, Huitema, and Turletti [Diot 95].

13.3.2.3 Compression

The most computational intensive part of an audio tool is the compression algorithm. As a rule of thumb, compression algorithms increase logarithmically in complexity for a doubling of compression advantage. Because of the high level of complexity in many compression algorithms, the performance of an individual codec is optimized for a range of platforms. Within the audio tool, a compression algorithm will be called to compress a buffer, rather than individual samples of audio, from the audio hardware. Between calls to compress (and decompress at the receiver), the internal variables of the codec are stored for the next time the compression module is called.

Early research into the effects of loss in interactive packet audio shows that small packets should be used, preferably carrying no more than 16 ms of audio [Minoli 79]. This use of small packets balanced the impact of packet rate on the network, with the likely perception of loss. Within the Internet, the desire to maintain a reasonable packet header-to-payload ratio has meant that the nominal minimum size of a packet over the Internet is 160 bytes, which translates into 20 ms of toll-quality PCM. If the packet rate needs to be reduced to ease loss rates at the receiver(s), then ADPCM might be used to code 40 ms of audio into a datagram (or RPE-LPC to code 80 ms), while keeping the payload size the same.

13.3.2.4 Talking in a Multiway Environment and Silence Detection

Mbone audio sessions commonly do not use floor control mechanisms and are based on dynamic group membership (participants can join and leave a conference at will without any explicit reconfiguration of the audio tool). Chaos is largely avoided by reliance on the protocol of polite conversation to facilitate meaningful multiway discourse.

Silence detection is commonly used in multiway conferences to restrict traffic generation to roughly one half of a PSTN bearer circuit, regardless of the number of participants. Packets are generated only when someone talks, and this means that a series of packets will be generated, interspaced by gaps (the series of packets is called a talk spurt). The detection of speech in the presence of background noise is nontrivial, and silence detection algorithms must be adaptive, because background noise levels vary [Rabiner 78]. A rough energy measure is commonly used, and this can be increased by the addition of spectral analysis algorithms to identify low energy fricatives, which frequently fall below the threshold of the background noise. Silence detection algorithms should be disabled for broadcast type applications and for music, as these algorithms frequently lead to clipping effects.

13.3.3 Audio Tool Problems

By far the most annoying problem with audio transmission is the gaps in the audio, and significant effort has been directed towards identifying the causes of this problem and eradicating them. Other audio tool problems and causes of frustration are associated with an unnatural auditory environment.

13.3.3.1 Cause Identification of Gaps in the Output Audio

Gaps in the output audio can be caused by any of the following situations, and the cause is not always obvious:

- incorrectly operating silence detection algorithm
- actual datagram loss over the Internet
- excessive jitter on the datagram arrival time
- packet(s) dumped by the audio tool (because of scheduling anomalies)
- faulty microphone or headsets

Some of the problems (such as faulty microphone or headsets) can be isolated by the activity bars provided by the user interface. The identification of other problems relies on detailed examination of loss statistics from the network (such as actual datagram loss over the Internet) and from the audio tool itself (packet dumped by the audio tool because of scheduling anomalies). Excessive jitter on packet arrival times is an infrequent event, catered to by the adaptive play-out point calculation, and it is displayed in the network statistics window. The design of successful silence detection algorithms is nontrivial and requires examination of the frequency spectrum to perform well. This extra facility is commonly not available within most audio tools because of the large amount of processing power taken to analyze frequency spectra.

Of the causes of gaps in the output audio, the loss of datagrams over the network and the occurrence of scheduling anomalies are by far the most common.

Solutions for datagram loss over the Internet — Packet loss is especially a problem for speech, where the use of relatively large packets means that the impact of loss is severe [Hardman 95]. Packet loss is also a problem for music, although delay tolerance means that a greater range of repair techniques are feasible than is the case for interactive communication [Hodson 98]. Possible approaches to alleviating the impact of packet loss are receiver-only repair, channel coding at transmitter, and combined source and channel coding at transmitter [Perkins 98].

Receiver-only solutions rely on the assumption that the audio characteristics have not changed over the duration of the packet and are usually taken to work for packet sizes of 16 ms. Different approaches to tackling this problem include repeating the audio from before the loss [Goodman 86], applying pattern-matching techniques on one or both sides of the loss [Waseem 88], interpolating over the gap [Sanneck 96], and applying repair in the transform domain (Figure 13.7). A number of codecs, such as the GSM, specify transform domain repair mechanisms, which suffer less from discontinuities at the boundaries of loss. Receiver-only techniques don't work very well for the larger packet sizes commonly used over the Internet because with an increasing packet size there is an increasing probability that the source characteristics will have changed (phoneme sizes are 5-100 ms, with the average being about 80 ms) [Hardman 95]. Interleaving components of audio across multiple datagrams can produce improvements because gaps are then significantly reduced in size, and receiver-only solutions are then used at the receiver. The disadvantage associated with this method, however, is an increase in delay, because multiple packets have to be collected at the transmitter before they can be sent.

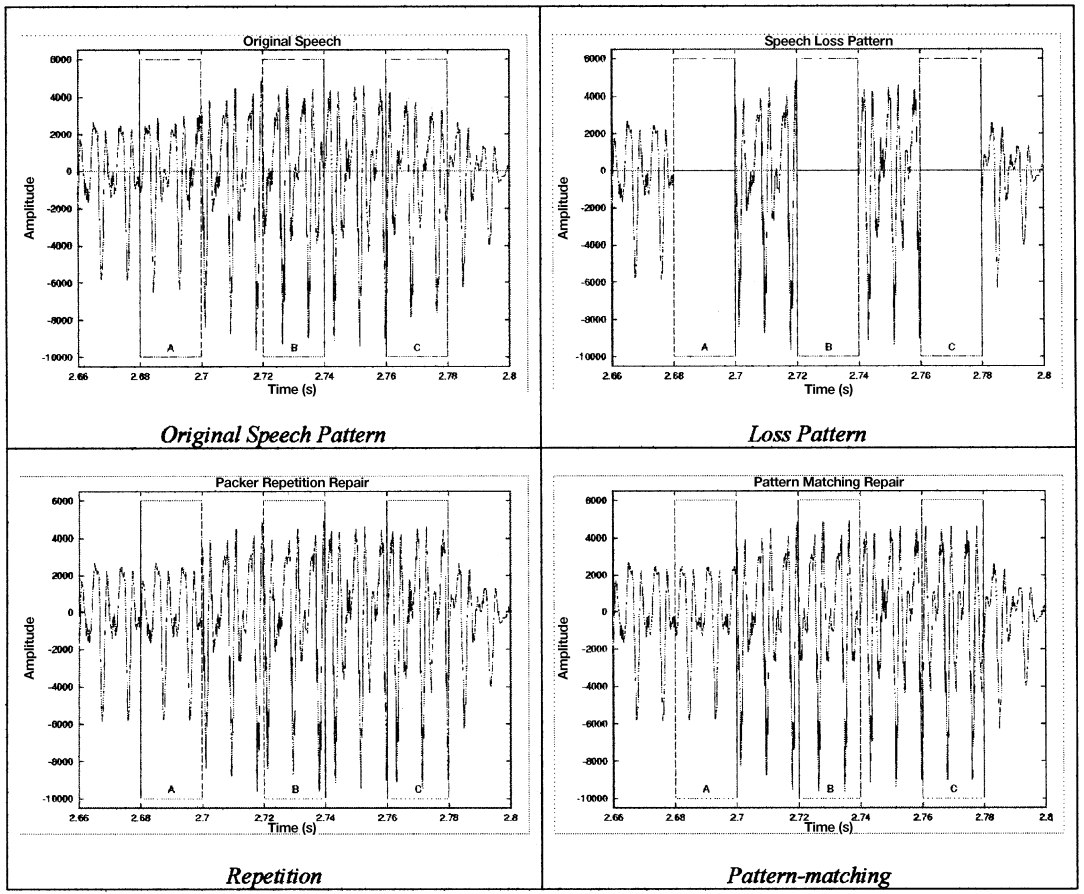

Figure 13.7 Examples of error concealment

Channel coding is the traditional approach to the problem of packet loss, where the source is assumed to produce a bit-stream that is preserved at the bit level. Common channel coding techniques are forward error correction (FEC) and interleaving and retransmission [Perkins 98]. FEC takes k blocks data and transforms it into n blocks of data, and, as long as the number of blocks lost is no more than $(n - k)/2$, all of the original k blocks can be determined. The inherent problem with these techniques is that preservation of speech at the bit level is not necessary, and these techniques tend to fail at a particular error level, rather than gracefully deteriorating. For natural conversation, the end-to-end delay needs to be less than 250 ms [Montgomery 83]. Interleaving, retransmission, and forward error correction increase the amount of buffering required by receivers and increase the end-to-end delay of the audio [Perkins 98]. Over local area networks, the additional delay required by retransmission means that it may be a viable repair method, but it is not suitable for interactive audio over wide area links.

Combined source and channel approaches to packet loss repair use analysis results from the compression algorithm to gauge the effect of loss and to transmit extra information to help repair at the receiver. In mobile communications, FEC is applied to parts of speech frames that are most sensitive to loss (e.g., GSM) [Vary 88]. The use of FEC in this manner is necessary because mobile communication suffers from large bursts of bit errors. If not corrected, bursts of errors introduce serious distortion into the output speech and cause the decoder to mis-track for a period after the loss. Using FEC for the important parts of speech or music frames is also a technique being employed in the Internet. Efforts by Mbone audio tool researchers have developed redundancy [Hardman 95], which piggy-backs redundant information onto subsequent datagrams in the stream in order to repair single losses (which are the most common), as shown in Figure 13.8.

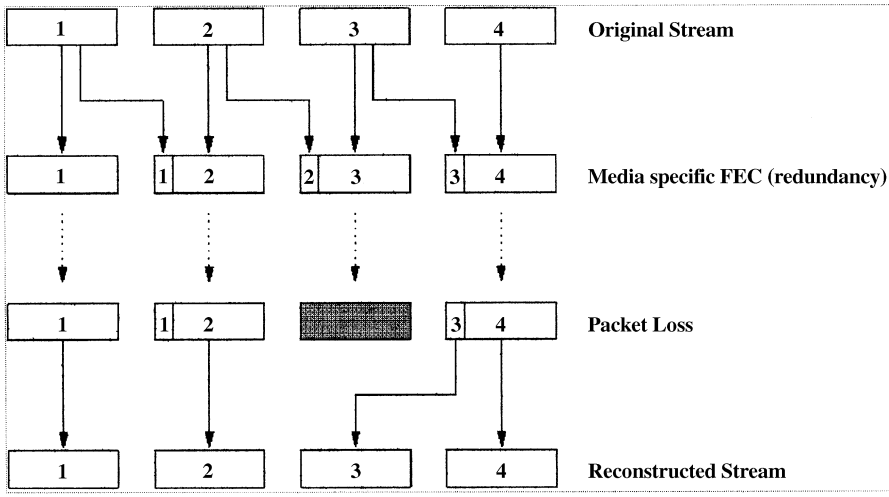

Figure 13.8 The use of redundancy to repair packet losses over the Internet

Redundancy can be extended to recover from bursts of losses, by lagging the redundancy from the original packet by different amounts [Kouvelas 97a]. The redundant information can be the output of the same codec (the most important consideration in reducing loss as a result of congestion is to reduce the packet rate), or can be the output of a different codec (that uses a high level of compression, or produces reduced quality speech). This scheme utilizes the fact that an exact fill-in to cover the loss is not necessary, and short degradations in sound quality are almost transparent. In addition, it incurs only extra delay at the receiver and none at the sender, because the piggybacked copy is from an earlier instant.

When interactive operation is not required, retransmission can be used. Retransmission is viable for streaming applications, because clients can ensure they extend the receiver buffering enough for every packet to cover the cost of sending the retransmission request and waiting for the data. Effort has recently been directed towards defining loss repair mechanisms for the Internet [Hardman 95], [Perkins 98].

Solution for lack of real-time scheduling support — Lack of real-time scheduling support means that an audio tool process may not be able to write enough audio to the device driver to cover the next period when the audio tool process is suspended. Similarly, it might be that the audio tool cannot read packets from the network interface socket before they have been overwritten. Both of these effects manifest themselves as packet loss.

Our RAT assumes that the clocking rate of the input audio from the audio card is in phase with the output (which is true if 1 crystal is used to provide input and output sampling frequencies). The audio tool reads whatever is available at the input and writes-out a corresponding amount. Using this mechanism it is possible to estimate the elapsed time since the last read and to estimate if the host machine is lightly or heavily loaded. If the host is heavily loaded, scheduling intervals might get large, and the output buffer in the device driver might run dry, leading to a gap in the output audio. To avoid this, RAT uses an adaptive algorithm that maintains a history of gaps and determines whether or not to increase the amount of samples in the device driver buffer to minimise gaps [Kouvelas 97b].

13.3.3.2 Cause Identification of an Unnatural Audio Environment

The acoustic feel of an Mbone audio tool is disrupted by many factors that can be categorized as two-way and multiway effects:

Two-way effects

- Difficulty in setting microphone and loudspeaker amplitudes correctly
- Silence suppression between sentences — high levels of background noise appear to cut in and out
- System appears dead because of a lack of reverberation
- Echoes because of the closed audio loop

Multiway effects

- No localisation abilities means it is difficult to separate voices from each other and to recognise individuals
- Multiple speakers have as many different background noise environments, and the change among them is disconcerting

13.3.3.2.1 Two-way Effects

Difficulty in setting microphone and loudspeaker amplitudes correctly — The acoustic part of the system suffers from problems when used with general purpose hosts. The need to support a wide range of headsets or microphones and loudspeakers means that impedance level mismatches result, and volume levels from participants are different. Input gain control is provided at the transmitter for this purpose, but participants cannot easily monitor their output volume at the far end. Because of this difference in volume between individual speakers, participants have to change the receiver volume gain on a regular basis as different people talk. This problem also causes input audio not to occupy the full dynamic range of the compression algorithm, which means that larger values of quantisation noise will be added to the signal than might be expected in the ideal case.

A solution to this problem is to use automatic gain control (AGC) at the transmitter. The AGC algorithm tracks the energy levels in the input talkspurt and multiplies every sample by a suitable scaling factor. Unfortunately, AGC suffers from problems, in particular, rapid gain changes during words reduce the intelligibility of what is being said. A system that is too slow to react will result in either greater quantisation noise being added by the compression algorithm. However, if the scaling factor is too slow to react, overloading results. Despite these problems, an AGC mechanism can be made to work reasonably well in an Mbone audio tool.

Silence suppression between sentences — Silence suppression is needed in multicast audio because it reduces the bandwidth consumed to manageable levels over the Mbone. It is also needed in order to sensibly mix multiple incoming flows from different speakers, because to mix the audio from hundreds of participants would mean that the few (usually one) speech streams would be masked out by many contributions of background noise.

Silence detection causes speech to be transmited in talk spurts and cuts intertalkspurt audio. This means that background noise cuts in and out, which has been shown to reduce the intelligibility of what is being said. In mobile point-to-point network connections, the traditional approach has been to transmit comfort noise, which is an average version of background noise at the transmitter, to the receiver [Hanzo 94]. Bandwidth is restricted because rough spectral information is sent, which is then used at the receiver to simulate frequency-correct noise for intertalkspurt gaps.

System appears dead — In the natural world, reverberation is caused when sounds are delayed and reflected off the walls of the environment the listener is in. The speaker hears his or her own speech, plus added reverberation. The addition of reverberation to speech actually improves intelligibility [Olson 57]. It also gives the

speaker some idea of the size of the room they are in and how loudly they need to speak to enable other people to hear them (compare the acoustic environment of a small room to that of a cathedral).

When using headsets in a multiway conference, the audio channel itself is perceived as dead; no reverberant information is present for the speaker to determine the acoustic map of the space into which they are speaking; the result is that people tend to shout. A cheap version of reverberation commonly used in telephone handsets is side-tone [Richards 73] — the speech input leaks through the handset to the earpiece, with essentially a very low delay. This technique is imitating the first reflection in the reverberation, which tends to be the most prominent anyway. Full reverberant audio can easily be generated using special audio cards, often designed to give the ability to localize individual speakers.

Previous work has also shown that a feedback loop is required between the AGC algorithm and the reverberation part of the system because it is possible to influence how loudly a person speaks by altering the level of reverberation [Sharifi 96].

Echoes — When an audio conference system is used with loudspeakers and microphones, there is often feedback through the system. This echo is extremely annoying for a remote participant, as their own voice is echoed back to them with often quite a large delay, and the perception of echoes increases with time after the original sound [Brady 71]. Frequently, feedback continuously circulates around the closed audio loop, and if the gain is greater than unity the system will howl-round. The use of more directional microphones eases the problem because then it is less likely that a microphone's pick-up pattern will overlap with the loudspeaker's directional output. A better solution is to use echo cancellation techniques, which adaptively estimate the delay path through the system and subtract the delayed version from the input audio. An alternative to echo cancellation is to use headsets, but these are neither comfortable for long conferences, nor do they provide a natural acoustic environment.

Some simple echo management functionality is available in audio tools to attenuate the microphone if the loudspeaker is active and vice versa, but the speech then suffers from clipping effects; parts of received speech are lost if the listening participant coughs, for example.

13.3.3.2.2 Multiway Effects

Lack of localisation ability — In a natural environment, a human's ability to localize sounds is used to focus attention and gaze on a speaker and to improve the audibility of a single conversation in a busy room (the cocktail party effect) [Cherry 53]. In general, this effect is associated with the urge to tip and turn the head to improve the localisation of the sound source. The visual image of the speaker is then matched with auditory input, and the combination of image and sound direction increases the intelligibility of what is being said [Begault 94].

The lack of auditory clues, when heard over a mono channel, makes individuals' speech sound as if it is coming from the same position and direction — the back of the listener's head, if the listener is using a headset. This is especially problematic in multiway conferences, in which the participants are suffering from visual overload and cannot focus on the list of speakers in the audio tool window to determine who is talking.

Different background noise environments for different speakers — Even if the rest of the causes of an unnatural acoustic environment are eliminated, a multiway conversation will never sound natural because multiple different acoustic environments are being switched among. What is really needed is elimination of a speaker's acoustic environment from the audio signal and addition of either an average environment at the receiver, or addition of the actual listeners' environment at the receiver. Such a solution is extremely difficult or impossible to produce.

13.4 IMPACT OF LOW-COST AUDIO COMMUNICATION

The impact of low-cost Internet audio communication will be vast. The deployment of Internet and Mbone audio is expected to increase rapidly with the availability of the global Internet and inexpensive personal computers in business and the home. Not only will the cost of existing applications (such as point-to-point telephony) be substantially reduced, but new applications will enhance interactive facilities well beyond audio-only communication.

Packet audio applications can be split into two broad groups — people-to-people and people-to-systems. Low delay and relatively small, but guaranteed, bandwidth requirements in both directions characterize people-to-people applications, such as Internet telephony and Mbone audio. In contrast, people-to-system applications (such as audio streaming applications) are much more delay tolerant and can function with substantially less bandwidth than is needed for real-time communication.

13.4.1 Internet Telephony

Internet telephony servers and software are already commercially available, and their popularity is expected to increase [Minoli 98]. Such systems also need to be integrated with the existing PSTN networks, and most commercially available products provide a comprehensive solution, including end application software, and gateway cards and software. Mechanisms for remote service location are also being standardised for the Internet and enhanced to provide capabilities such as the remote location of a gateway based on cost [Rosenberg 98].

Point-to-point Internet telephones use unicast transmission facilities. Multiway audio tools, such as CuSeeMe and Netmeeting, use unicast transmission facilities and manually set-up reflectors to achieve multiway communication. When multicast becomes more widely available in the Internet, many more people will use Internet audio to add an interactive dimension to existing applications, for example mailing lists, news groups, and computer games (some games already use multipoint interactive audio communication over the Internet).

13.4.2 Mbone Audio and Multimedia Conferencing

Many-to-many audio tools use multicast, a more efficient evolution of unicast and reflectors. Mbone audio is regularly used as part of multimedia conference sessions, and the Mbone is moving from a pilot to a service [Perkins 97][Buckett 95]. Multicast

multimedia conferencing uses a combination of media to provide communication, often audio, video, and shared text. The availability of low-cost multimedia conferencing also means that distance learning applications are becoming popular, and telepresence applications are being researched. Not only will further deployment and development of the Mbone benefit multiway communication, but the increased integration of multimedia conferencing tools (such as the integration of audio with video to provide lip synchronisation [Kouvelas 96]) will substantially improve the attractiveness of this method of communication.

13.4.3 Audio and Video on Demand

Initially delivered via file transfer, music is now streamed over the web using unicast transmission and reflectors (e.g., RealAudio). Similar applications might use AC-3 as a compression algorithm and error protection to successfully stream audio via point-to-point communication (at 44.1 or 48 kHz sampling frequency). Mbone audio tools are also widely used to multicast live radio stations. These tools use toll-quality speech compression algorithms (8 kHz sampling frequency), although support for multiple sampling rates up to 48 kHz is now available in some audio tools [Hodson 98].

Interactive systems also frequently need to access stored clips of material — in a business meeting, this could be a record of previous meetings; in distance learning, this could be access to other educational material. Multimedia jukeboxes are also being developed in the research community to provide stored clip access [Lambrinos 98]. Such streams would be controlled using RTSP, an Internet draft standard protocol that allows hosts to remotely control playback and recording of streamed audio and video [RTSP].

13.5 CONCLUSION AND FUTURE DIRECTIONS

This chapter presents an overview of Internet audio. A full understanding of Internet audio systems can be gained only from a comprehensive assessment of compression algorithms, acoustics, Internet protocols, and general purpose host implications. Analysis of these aspects leads to Internet audio tools and the problems associated with them.

Speech compression algorithms exploit the characteristics of speech production to gain a compression advantage and quite low bit-rates. Unfortunately, we can hear better than we can talk, and because music compression algorithms rely on the characteristics of hearing, the bit-rates produced are substantially larger than those for speech. The standards that relate to speech and music compression have been developed primarily for networks other than the Internet and tend to suffer substantially from packet loss over the Internet.

Real-time applications use UDP over IP because they do not need the reliable service offered by TCP. The IETF has standardised an application layer protocol, RTP, to provide facilities such as timestamps and sequence numbers for real-time applications. Of particular importance for audio is the ability to convert point-to-point applications into multiway ones. Multiway facilities are currently provided over the Internet by unicast transmission and reflectors. However, the Internet can

support a more efficient method of multiway communication — multicast — which has distinct advantages over reflector-based systems. In the future, it is likely that multiway applications will use multicast for multiway multimedia delivery.

This chapter focused on a particular Mbone audio tool, rather than consider examples of each of the application classes, because Mbone audio represents a superset of the technology. Mbone audio tools are made up of a pipeline of components designed to reduce environmental problems, such as the effect of the best-effort Internet, the use of compression algorithms designed for guaranteed bit-rate transmission, and the use of general purpose workstations that provide no support for real-time applications. These problems manifest themselves primarily as gaps in the output audio, and this chapter identified their many causes, and suggested solutions. Another complex audio tool problem addressed here was an unnatural acoustic environment, which is associated with the transmission of one channel of audio to single and multiple participants.

Internet audio can be split into three types of applications: Internet telephony, music-on-demand, and Mbone audio. In the future, it is likely that many current Internet applications will have an interactive edge added to them because Internet audio has a low cost. The advantage of transmitting audio over the Internet is not just that it is low cost, or that multiway transmission is possible to hundreds of receivers, but that audio can easily be integrated with other media, such as video, graphics, and text. From this we can conclude that the impact of Internet audio will be vast.

13.5.1 Future Directions

The best-effort Internet will evolve to be able to support some guaranteed quality of service (QoS) levels. An initial attempt at providing QoS over the Internet produced the reservation protocol (RSVP), which allows receivers to specify that a guaranteed QoS be used over the links in that part of the distribution tree. Routers will provide QoS by employing queuing algorithms such as class-based queuing [Floyd 95], which might process delay-sensitive (or higher QoS) traffic with preference to other traffic.

In tandem with the drive to enable the Internet to support QoS levels is a desire to encourage all traffic to share evenly the available bandwidth and to back-off when congestion levels increase. Some applications are currently fair in their use of bandwidth (such as TCP/IP), but interactive applications do not behave so well. It is probable that current router buffering will change from a FIFO discipline to using techniques such as random early drop [Floyd 94], which might give advanced notification of congestion to receivers and penalize badly behaving sources. Audio applications might become adaptive to congestion [Kouvelas 98], although this might be restricted to applications that transmit higher quality audio. In such situations, the audio compression algorithm must be able to deliver multiple, different qualities and therefore bit-rates.

Multimedia conferencing and telepresence applications will evolve far better interaction capabilities than can currently be provided by independent media tools. Those tools will be integrated by a framework, which will support both remote

configuration and control capabilities [McCanne 98] and increased media integration. Instead of monolithic tools, it is likely that software will be in the form of a series of reusable objects in a pipeline [DirectX].

ACKNOWLEDGMENTS

RAT is the work of a group, and we acknowledge Colin Perkins and Isidor Kouvelas, amongst others. The work is funded by EPSRC projects RAT (#GR/K72780) and MEDAL (#GR/L06614) and British Telecom plc (JAVIC project). In addition we thank Phil Lane for comments on the text.

REFERENCES

[ASF] Online documentation http://www.microsoft.com/asf/

[Barnwell 96] T. Barnwell III, K. Nayebi, C. Richardson, *Speech Coding: A computer laboratory Textbook*, Georgia Tech. Digital Signal Processing Series, 1996.

[Begault 94] D.R. Begault, *3D Sound for Virtual Reality and Multimedia*, Academic Press, 1994.

[Brady 71] P.T. Brady, "Effects of Transmission delay on Conversational Behaviour on Echo-Free Telephone Circuits," *Bell System Tech. J.*, Vol. 50, No. 1, 115–134, Jan 1971.

[Brady 69] P.T. Brady, "A model for generating on-off speech patterns in two-way conversation" *Bell System Tech. J.*, Vol. 48, No. 9, 2445–2472, Sep 1969.

[Buckett 95] J. Buckett, I. Campbell, T.J. Watson, M.A. Sasse, V.J. Hardman, A. Watson, "ReLaTe: Remote Language Teaching over SuperJANET" *Proc. UKERNA 95*.

[Gold 73] B. Gold, "Digital Speech Networks," *Proc. IEEE*, Vol. 65, No. 12, 1636–58, 1973.

[Cherry 53] C.E. Cherry, "Some Experiments on the Recognition of Speech, with One and Two Ears," *J. Acoust. Soc. Am.*, Vol. 25, 975–979, 1953.

[Clarke 90] D. Clarke, "Final Report from the Unison Project," ALVEY Projects of the DTI, 1990.

[Clark 90] D. Clark, D. Tennenhouse, "Architectural Considerations for a new Generation of Protocols," *Proc. ACM SIGCOMM*, 1990.

[Deering 89] S. Deering, "Host Extensions for IP Multicasting," RFC1112, Internet Engineering Task Force, Aug 1989.

[Diot 95] C. Diot, C. Huitema, T. Turletti, "Multimedia Applications should be adaptive," *Proc. High Performance Comp. Syst. '95*, 1995.

[DirectX] http://www.microsoft.com/directx.

[Floyd 95] S. Floyd, V. Jacobson, "Link sharing and resource management models for packet networks," *IEEE/ACM Trans. Networking*, Vol. 3, No. 4, Aug 1995.

[Floyd 93] S. Floyd, V. Jacobson, "Random early detection for congestion avoidance," *IEEE/ACM Trans. Networking*, Vol. 1, No. 4, Aug 1993.

[Goodman 86] D. Goodman, G. Lockhart, "Waveform Substitution Techniques for Recovering Missing Speech Segments in Packet Voice Communications," *IEEE Trans. Acous., Speech and Signal Processing*, Vol. ASSP-34, No. 6, 1440-1448, Dec 1986.

[H.323] "Recommendation H.323 (02/98) Packet-based multimedia communications systems," http://www.itu.int/.

[Hanzo 94] L. Hanzo, R. Steele, "The Pan-European Mobile Radio System Part II" *Eur. Trans. Telecommun. and Relat. Tech.*, Vol. 5, No. 2, Mar-Apr 1994.

[Hardman 95] V.J. Hardman, M.A. Sasse, A. Watson, M. Handley, "Reliable Audio for Use over the Internet," *Proc. INET95*, 1995.

[Hardman 98] V. Hardman, M.A. Sasse, I. Kouvelas, "Successful Multi-party Audio Communication over the Internet," *Communi. ACM*, May 1998.

[Hodson 98] O. Hodson, S. Varakliotis, V. Hardman, "A software platform suitable for multiway audio distribution over the Internet," in *Audio and Music technology: the challenge of creative DSP*, IEE Colloquium, London, U.K., 1998.

[Jacobson 92] V. Jacobson, "VAT manual pages," Lawrence Berkeley Laboratory (LBL), Also http://www-nrg.ee.lbl.gov/vat/, Feb 1992.

[Kent 92] R.D. Kent, C. Read, "The Acoustic Analysis of Speech," Whurr Publishers, 1992.

[Kouvelas 96] I. Kouvelas, V. Hardman, A. Watson, "Lip Synchronisation for use over the Internet: Analysis and Implementation," *Proc. Globecom96*, 893–898, 1996.

[Kouvelas 97a] I. Kouvelas, O. Hodson, V. Hardman, J. Crowcroft, "Redundancy Control in Real-Time Internet Audio Conferencing," International workshop on Audio-Visual Services over Packet Networks, 1997.

[Kouvelas 97b] I. Kouvelas, V.J. Hardman, "Overcoming Workstation Scheduling Problems in a Real-Time Audio Tool," *Proc. USENIX Ann. Tech. Conf.*, 1997.

[Kouvelas 98] I. Kouvelas, V.J. Hardman, J. Crowcroft, "Network Adaptive Continuous-Media Applications Through Self Organised Transcoding," *Proc. NOSDAV*, 1998.

[Lambrinos 98] L. Lambrinos, P.T. Kirstein, V.J. Hardman, "The Multicast Multimedia Conference Recorder," *7th Int. Conf. Communi. and Networks*, IEEE, 1998.

[McCanne 97] S. McCanne, et al., "Toward a Common Infrastructure for Multimedia-Networking Middleware," *Proc. 7th Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video* (NOSSDAV 97), 1997.

[Minoli 98] D. Minoli, E. Minoli, *Delivering Voice over IP Networks*, Wiley Computer Publishing, 1998.

[Minoli 79] D. Minoli, "Optimal packet length for packet voice communication," *IEEE Trans. Commun.*, Vol. COM-27, 607–611, Mar 1979.

[Moon 98] S. B. Moon, J. Kurose, and D. Towsley, "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms," ACM / Springer Multimedia Systems, Vol. 6, 17–28, Jan 1998.

[Montgomery 83] W.A. Montgomery, "Techniques for packet voice synchronization," *IEEE J. Sel. Areas Commun.*, SAC-1 (6), 1022–28, Dec 1983.

[Olson 57] H.F. Olson, *Acoustical Engineering*, Van Nostrand, 1957.

[Ousterhout 94] J.K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, 1994.
[Pan 93] D.Y. Pan, "Digital audio compression," in *Digital Tech. J.* Vol. 5, No. 2, 1993.
[Perkins 98] C.S. Perkins, O. Hodson, V. Hardman, "A Survey of Packet-Loss Recovery Techniques for Streaming Audio," *IEEE Net.*, Sept/Oct 1998.
[Perkins 97] C. Perkins, J. Crowcroft, "Real-time Audio and Video Transmission of IEEE Globecom 96 over the Internet," *IEEE Commun. Mag.*, Apr 1997.
[Peterson 96] L. Peterson, B. Davie, *Computer Networks: A systems Approach*, Morgan Kaufmann Publishers, Inc., 1996.
[Rabiner 78] L.R. Rabiner, R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.
[RealAudio] 'RealAudio' http://www.real.com/.
[Richards 73] D.L. Richards, *Telecommunication by Speech*, Butterworth and Co. 1973.
[Rogerson 97] D. Rogerson, *Inside COM*, Microsoft Press, 1997.
[Rosenberg 98] J. Rosenberg, B. Suter, "Wide Area Service Location," IETF Internet Draft, draft-ietf-srvloc-wasrv-00.txt.
[RSVP] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource Reservation Protocol," *IEEE Net*, Vol. 7, No. 5, 8–18, Sept 1993.
[rtp] "RTP: A Transport Protocol for Real-Time Applications," Audio-Video Transport WG, RFC 1889.
[RTP] H. Schulzerine, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC1889, IETF, Jan 1996.
[RTP Generic] A. Periyannan, D. Singer, M. Speer, "Delivery Media Generically over RTP," ftp://ftp.nordu.net/internet-drafts/draft-periyannan-generic-rtp-00.txt.
[RTSP] http://www.real.com/library/fireprot/rtsp/ Also rfc2326.txt.
[Sanneck 96] H. Sanneck, K. Stenger, B. Younes, B. Girod, "A new technique for audio packet loss concealement," *IEEE Global Internet*, 48–52, 1996.
[Sharifi 96] H. Sharifi, "An investigation of Acoustic Enhancing Features for Use in Audio Tools," MSC Thesis Report, 1996.
[SIP] M. Handley, H. Schulzerinne, E. Schooler, J. Rosenburg, "SIP: Session Invitation Protocol," http://www.cs.columbia.edu/~jdrosen/sip/drafts/draft-ietf-mmusic-sip-10.txt.
[Spanias 94] A. Spanias, "Speech Coding: A tutorial review," *Proc. IEEE*, 82(10):1541–1582, 1994.
[Tannenbaum 96] A. Tannenbaum *Computer Networks*, Prentice-Hall Int. Ed., 1996.
[Vary 88] P. Vary, R. Hofmann, K. Hellwig, R.J. Sluyter, "A Regular-Pulse Excited Linear Predictive Codec," *Speech Communication* 7, 209–215, 1988.
[VIS] http://www.sun.com./microelectronics/vis/.
[Waseem 88] O.J. Wasem, D.J. Goodman, C.A. Dvorak, H.G. Page, "The effect of waveform substitution on the quality of PCM packet communications," *IEEE Trans. Acoust., Speech, and Signal Processing*, 36 (3), 342–8, 1988.