A SOFTWARE PLATFORM FOR MULTIWAY AUDIO DISTRIBUTION OVER THE INTERNET

Orion Hodson, Socrates Varakliotis, and Vicky Hardman
{O.Hodson, S.Varakliotis , V.Hardman}@cs.ucl.ac.uk
Department of Computer Science, University College London, Gower Street, London, WC1E 6BT

1. Introduction

The Robust Audio Tool (RAT) allows users to achieve real-time multi-way communication over the Internet. It was initially intended for use in multiway conferences, but more recently is being used as an Internet audio broadcast application, by radio stations in the US and elsewhere. RAT can also be used in a point-to-point manner, and as a transcoder between networks of differing capabilities, e.g. for mobile access to the Internet. It is one of several tools that have been developed within the research community, some of which are also available commercially.

Interest in Internet audio has grown with the deployment of the Internet, and the widespread availability of full duplex audio hardware on PC's and workstations. Despite packet audio being one of the first ARPANET / SATNET research investigations in the early 1970's [Gold73], the present Internet is not well suited to real-time audio delivery, since it only offers a best-effort service with no bounds on transit times, or data delivery guarantees.

The emphasis of work in RAT has been on maximising the audio quality despite inherent problems of packet transport, processor scheduling and audio capabilities of the end system. It also serves as a platform for audio research. For maximum flexibility, RAT is designed as a modular application and many of the underlying subsystems have alternative complexity / capability implementations. RAT can also leverage off platform-specific features, such as software and hardware audio codecs, various audio devices within a platform, and alternative protocol stacks. The important features of RAT, in comparison to other Internet audio tools, is that it is able to support multirate processing, has no restrictions on audio frame duration, and supports multi-channel audio, and both fixed and variable size audio frames.

In this paper we briefly discuss methods of real-time multimedia delivery, and identify issues of particular importance for music transmission over the Internet. For music coding researchers interested in using RAT to exploit their research, we present an overview of the architecture of the Robust Audio Tool and specifically focus on codec integration into RAT. Finally, we present some off-line performance measurements of a public domain MPEG1 music codec that has recently been integrated into RAT, and illustrate Internet performance in terms of packet loss, and variable transit delays.

2. Music and Multimedia Delivery on the Internet

Most current commercial Internet audio streaming applications use application-level reflectors to forward packets from source to receivers, or to other reflectors. Cascades of reflectors are used for broadcasting to large audiences, but since the reflectors are manually placed on amenable hosts, rather than in the network routers, the distribution tree may be sub-optimal. Reflector schemes are bandwidth inefficient, and they do not scale well, nor are they robust to network outages.

Instead of using reflectors to provide multi-way communication facilities, a router-implemented mechanism of efficiently replicating streams can be used, which is called multicast [RFC1112]. The collection of multicast-capable routers in the Internet is called the Multicast backbone (Mbone) [Macedonia94]. A variety of Mbone audio tools exist, such as RAT [Hardman 98][Hardman 99][RAT URL], but the majority of multicast-capable tools are used for multicast conferencing applications (including H.323-based systems). Some streaming applications also support multicast, but since many Internet Service Providers (ISPs) do not support it, there currently is no alternative to reflector schemes. In the future, it is reasonable to assume that multi-way multimedia delivery over the Internet will be via multicast.

In multicast, the network conspires to deliver data to multiple clients in an efficient manner, and hosts express an interest in data by subscribing to a group address. Senders do not have to set-up a call to join a conference, and the sender is not aware of which or even how many receivers are listening, unless the application protocol provides this information. This de-coupling of hosts from the routing implementation means that multicast is inherently scalable and robust to network outages.

The Internet protocol (IP) suit is a layered stack, with different layers refining the functionality provided by the layer below, and with protocols above IP being used on an as-needed basis. For real-time audio transmission, RAT (and most real-time applications) uses a lightweight transport protocol layer, the User Datagram Protocol (UDP), on top of IP. UDP provides no loss resilience or congestion control, and so for real-time media transmission, the Internet Engineering Task Force (IETF) has standardised a specific protocol to be used at the application layer, called the Real-time Protocol (RTP).

2.1 Internet / Mbone Protocol for Real-time Multimedia (RTP)

RTP was developed to provide a lightweight application layer protocol for time-dependent data and to provide interoperability among conferencing tools [RTP]. RTP has a sister protocol, the Real-Time Control Protocol (RTCP), which is used for distributing sender and receiver statistics and participant details (name, location, email, etc). Applications have the responsibility of encapsulating data in the RTP payload, of adding the RTP header, and of providing RTCP control functionality. Strictly, RTP is network protocol independent, though it is commonly used on top of UDP/IP.

The pertinent features of RTP for multimedia delivery are:

- *Marker Bit*: for speech audio streams; if set, denotes the start of a new talkspurt.
- *Payload type*: a unique number to denote an audio codec. These numbers are defined in a separate standards document [RTP-Profile] and include a wide range of audio codecs, although only a few are implemented in the current Mbone audio tools.
- *Sequence Number*: This field is a packet counter, starting from a random number to minimise the risk of external interference. It is used to detect loss, packet disordering, and to distinguish talkspurts when packets with marker bits are lost.
- *Timestamp*: Conveys the sampling timestamp of the data in the RTP packet.

RTCP is used to convey user, quality of service, and application specific information. RTCP reports vary depending on whether a participant has sent data since the last report or not. If a participant has sent no data, they send a receiver RTCP report. The RTCP report includes the following information: the fraction of packets lost since the previous report; the cumulative number of packets lost; the highest sequence number received; an estimate of the inter-arrival jitter; the timestamp of the last received sender report; and the delay since the last received report. Senders send the same report timestamp, plus a transmitted packet count. The protocol profile recommends that RTCP data from all participants should not exceed 5% of the total bandwidth.

3. Issues for Internet Multimedia Applications

The key problems for multimedia delivery over the Internet are the average end-to-end transit time, jitter in end-to-end transit times, and the loss of packets during transit. Applications must be able to cope with these problems and also out-of-sequence and duplicated deliveries.

End-to-end delay consists of the time taken to sample and encode the data, the network propagation time, (including network access and the time spent in router queues), operating system imposed delays in process scheduling and receiver buffering. The most significant sources of variation are the time spent by packets queuing at routers and the operating system process scheduling and network interaction. For interactive conferences, the end-to-end delay should ideally be less than 250ms [Montgomery83], but this constraint can be relaxed for music delivery. Adaptive playout buffering at receivers is essential to absorb the end-to-end delay variations [Moon98, Diot95], and to maintain the relative timing relationship between sender and receivers. A graph of typical jitter distribution values can be seen in Figure 1a.

Packet loss (see figure 1b, 1c) is the most serious problem for multimedia applications, because many codecs rely on state continuity between frames; packet losses disrupt the state, leading to mistracking at the decoder. A number of alternative approaches to this problem are available, considered in more detail in [Perkins98]:

- *Loss concealment* - receivers can implement loss concealment algorithms, such as frame repetition or audio pattern matching [Goodman86, Wasem88]. These may either be performed in the waveform or in the transform domain.
- *Channel coding* - channel coding schemes such as retransmission, forward error correction, or interleaving can be applied to mitigate the effects of packet losses. In large groups, retransmission requires a separate protocol to determine suitable hosts from whom to obtain data [Xu97]
- *Adaptive coding* - the type of audio or channel coding applied may be varied in response to explicit receiver feedback. Eliciting timely feedback information can be difficult due to the transient behaviour of the Internet [Nee97].
- *Layered coding* - addresses the issues of network link heterogeneity, as receivers can subscribe to layers that are appropriate to their link capacity [McCanne96]. Layering has the disadvantage of sending more packets and makes playout calculation more difficult.

Of the above, adaptive and layered coding are two approaches to congestion avoidance, rather than error robustness. Adaptive coding uses extra explicit information to determine a suitable transmission rate, and the major draw-back for multi-way delivery is that it restricts audio quality for all receivers, not just those on congested links. In layered transmission, each receiver (or group of receivers) makes their own decision about the layers they receive and can tailor their own reception to suit any of network bandwidth, congestion levels, receiver processing power and desired cost.

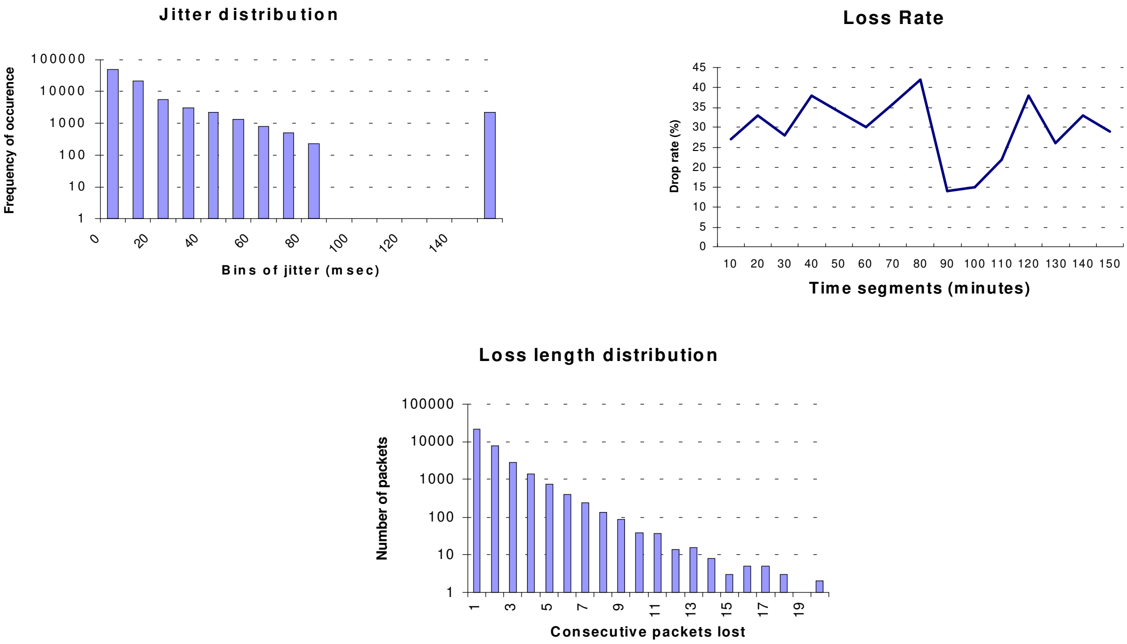

Figure 1: Illustrative loss graphs for 3-hour Mbone session. a) Jitter distribution b) Loss rate c) Loss length distribution (note that the last jitter bin includes all instances of jitter greater than 150ms)

In addition to the basic problems of delay and loss, hosts also need to be aware of constraints such as the maximum transport unit (MTU) size of the underlying network, in order to avoid additional delays and multiplicative loss effects from fragmentation. For low bit-rate encoding schemes, IP is an inefficient transport mechanism, where, for instance, 30ms of G.723.1 audio data in a packet has an associated header of 12 bytes for 24 bytes of data.

4. RAT Program Code

RAT can be divided into two distinct entities: the audio engine and the user interface. These are isolated from each other and communicate via a conference bus. The audio engine is responsible for audio processing, coding, packetization, and for maintaining RTCP statistics. The user interface presents the user with information on conference participants, their reception quality, and allows the user to configure all aspects of the audio engine. The conference bus is a locally scoped multicast group that allows applications to communicate with each other.

The audio engine consists of approximately 32,000 lines of C code, and the user interface of 2000 lines of Tcl/Tk, plus a further 2000 lines of supporting C code. The choice of Tcl/Tk [Ousterhout94] for the user interface was motivated by the ease of construction of interfaces with Tk, and the wide range of platforms that support it. A number of composable toolkits are now available for constructing multimedia applications, and these are often more generic than the modules used in RAT. To maximise platform support and performance, and also because development started before many of these toolkits were available, we have not yet attempted to use any of the composable toolkits. Examples of such toolkits are Microsoft's DirectX/Media framework, the Java Media Framework (JMF), and Berkeley's MASH toolkit [Mash97].

5. RAT Architecture

The key stages in the audio processing are shown in figure 2. All of these stages are written to be independent of rate and channels (mono and stereo only), and are clocked by a master clock at 96 kHz. In order to reduce implementation complexity, only sampling rates that are integer multiples of 8 kHz are supported

Audio processing starts (and ends) with the audio device, which provides a common interface to waveform audio functions on the available hardware, and converts the available sampling modes into the internal format used in the tool (16-bit linear samples). When the tool starts, it probes the audio devices and the sampling modes they offer. Two pseudo audio devices also exist: the null audio device and the transcoding audio device. The null audio device is used when no real audio device is obtainable, it simply discards audio it receives and passes silence when audio is requested. The transcoding audio device acts as a crossover point for audio streams when the tool is acting as a transcoder between unicast-unicast, unicast-multicast, or multicast-multicast sessions.

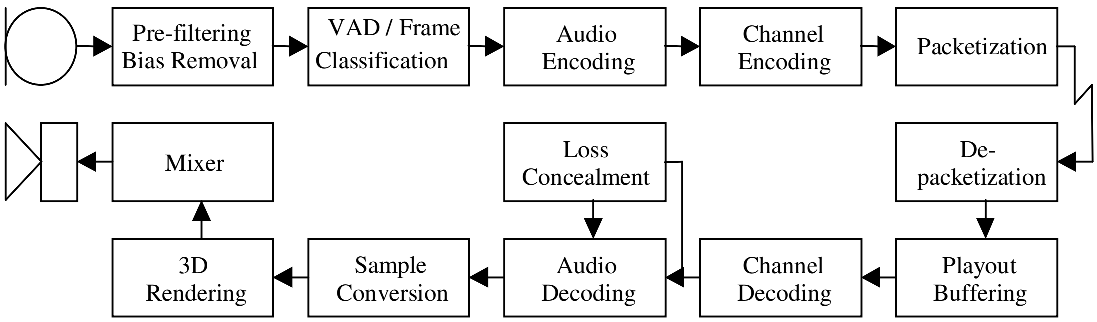

Figure 2: Audio Processing Paths within the Robust Audio Tool

Audio from the device is sampled in discrete blocks that match the duration of encoding scheme to be applied for transmission. If RAT is being used for interactive conferences, then facilities such as voice activity detection and automatic gain control can be used. For music delivery, however, these features are not normally needed. The audio is encoded, the resulting blocks are passed to a channel coder, then packetised and transmitted. At present, three variations of channel coding exist: interleaving, redundant audio, and no channel coding. With interleaving, adjacent blocks in the original stream are transmitted in different packets, so that the effects of a single packet loss are dispersed – it is easier to conceal several short losses than one large one. In the redundant audio scheme, multiple copies of encoded audio frames are transmitted in separate packets, so that there are at least 2 chances that audio will be available at the decoder for any particular frame.

When packets arrive at the receiver, their arrival time and source clock timestamp are used to update the estimate of the differences between the receiver and source clocks and the amount of jitter in the arrival times. Each packet is then placed in the playout buffer using the last calculated playout delay. The playout delay gets recalculated at the start of new talkspurts/transmissions, and if not recalculated will become too small or too large. The drift arises from frequency differences in source and receiver sampling clocks, or changes in the offset between clocks. At playout time, the packet is passed to the channel decoder to recover the coded frames, and then onto the audio decoder (or loss concealment module). Using late decoding, it is possible to ensure that all streams are decoded in the correct order and that as much as possible of the received audio can be utilized.

Following decoding, there exist two optional processing stages: sample rate conversion and 3D-audio rendering. Sample rate conversion is only necessary when the sender is using a codec based on a different sampling rate to that of the receiver. 3D-audio rendering is primarily aimed at multiway conferencing, and it allows the receiver to position different sources of audio at different angles and distances relative to the listener's head position [Hardman96]. 3D audio is only particularly effective at 16kHz and above. Multi-channel capabilities are also supported in RAT for music applications, where multiple channels are encoded at the source. Sample rate conversion and 3D audio are computationally expensive, and multiple schemes are implemented for both, giving a cost-quality trade-off.

For multi-way interactive conferencing applications, the last stage is mixing the audio streams. For this, a circular buffer is used and samples are added into the buffer. An entry is stored, which indicates the last audio instant that audio was mixed from a given sender. This is required so that if the playout point adapts audio from each sender, consecutive audio blocks from the same source are not mixed together.

6. The Codec Interface in RAT

Of particular interest to anyone who wishes to use RAT to exploit the capabilities of a new audio compression algorithm, is the interface to the codec. The codec interface in RAT is a table, which contains entries for all aspects of the encoding and decoding processes. The table has pointers to desired encoding / decoding functions, and numeric information indicating the configurations where the codec is suitable. The table is built dynamically when the application starts, which allows platform-specific codecs (e.g. Windows ACM codecs) to be included, as well those compiled into the application.

Each codec has initialisation, transformation, and 'clean-up' functions that are used by each encoder instance (per source) and initialisation and 'clean-up' functions to manage any shared information used by all instances of the encoder / decoder. Two optional functions may be present: a codec specific repair routine and frame size calculator. The codec specific repair routine is used to perform transform domain loss concealment. The frame size calculator is used to assist in fragmenting and validating received audio from variable frame size codecs.

The numeric information includes the payload identifier, the sample rate and number of channels, the number of samples in a coded audio frame, the size of the coded audio frame, the size of any per packet state, and the state sizes of data held by the encoder and decoder. The payload identifier is used to select the appropriate encoder and decoder at the sender and receiver.

7. Typical performance measurements of a music codec

We have conducted a number of encoding and decoding experiments in order to evaluate the performance of reference implementations of high quality codecs. Mbone audio tools have to-date been only used to provide single channel toll quality speech facilities, and the ability to support multi-channel music compression algorithms is very recent. In contrast to the usual processor consumption of RAT of 5% or so on most PCs and workstations, stereo CD-quality audio compressed using a suitable music compression algorithm to about 350kbps per channel uses approximately 10 times the processing power as for a typical speech application. This currently represents a sizeable chunk of the total processing time available, and effort to optimise execution time in a codec is well spent.

The data shown here was generated using a 30-second stereo PCM audio data file, sampled at both 44.1 and 8 kHz. We have measured MPEG-I (Layer II) encoding and decoding times for the advanced psycho-acoustic model (model 2), for a number of output bit rates (32 to 448 kbps), on a Pentium II processor running at 266 MHz. The results are shown in table 1 below:

Bit-rate	Encoding		Decoding	
Kbps	**44.1kHz**	**8kHz**	**44.1kHz**	**8kHz**
32	0.64	3.00	1.76	8.57
192	0.54	2.86	1.50	7.50
288	0.53	2.72	1.43	6.67
448	0.52	2.61	1.36	6.67

Table 1: Off-line MPEG-I Encoding/Decoding performance shown as a fraction of real-time performance. Results taken using public domain MPEG code

Code profiling of the encoder reveals that, not surprisingly, the most significant fraction of time (20-30%) is spent in FFT functions. The code tested is not optimised and we hope to achieve significant performance increases through the use of signal processing libraries with single-instruction-multiple data (SIMD) optimisations [MMX, VIS]. An alternative approach would be to allow the use of off-line encoding, but sometimes music applications will require the ability to encode material as well as to decode.

8. Conclusion and Future Work

Mbone audio tools are being used for multi-way music distribution over the Internet. This paper has presented an overview of the RAT architecture, focussing specifically on multi-channel multi-rate music distribution. In particular, the codec interface is described, together with the routines that need to be written if 2nd party audio codecs are to be integrated into RAT. The paper also includes off-line performance measurements of a public domain MPEG1 codec, as current estimates of processing power consumption by RAT running such a codec are approximately 60% of the CPU. We also presented some illustrative graphs of current Mbone loss rates, loss length distribution and jitter distribution to give someone designing error robustness strategies an idea of current typical behaviour of the Internet.

In the future, we intend to develop facilities for automatic feature selection by estimation of CPU loading, and to refine RAT for use with layered coding algorithms, focussing on both the network protocol and the system aspects.

Acknowledgements

Vicky Hardman and Isidor Kouvelas developed the first version of the Robust Audio Tool. Orion Hodson, Colin Perkins, and Isidor Kouvelas developed the multirate version described in this paper. Markus Iken contributed the 3D-rendering module. The development of this software has been aided by funding from the EPSRC (projects #GR/K72780, and #GR/L06614) and British Telecommunications PLC (JAVIC project). Hardware and software contributions are from Hewlett Packard, Microsoft, and Sun Microsystems.

References

[Diot95] C. Diot, C. Huitema, and T. Turletti, "Multimedia Applications should be adaptive", *Proceedings of High Performance Computer Systems'95*, Mystics (CN), August 23-25, 1995.

[Gold73] B. Gold, "Digital Speech Networks", *Proceedings of the IEEE*, Vol. 65, No. 12, pp.1636-58.

[Goodman86] D. J. Goodman, G. B. Lockhart, O. J. Wasem, and W. C. Wong, "Waveform substitution techniques for recovering missing speech segments in packet voice communications*", IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34 (6), pp.1440-48, December 1986.

[Hardman96] Hardman V., Iken M., 'Enhanced Reality Audio in Interactive Networked Environments', *Framework for Interactive Virtual Environments (FIVE) conference*, Pisa, Italy December 1996.

[Hardman98] Hardman, V., Sasse, M.A., Kouvelas, I., "Successful Multi-party Audio Communication over the Internet", in *Communications of the ACM*, May 1998.

[Hardman99] Hardman, V.J., Hailes S.M.V., 'Multimedia Mobile Access for the Internet / Mbone', to appear in *Insights into Mobile Multimedia*, edited by David Bull, Nishan Canagarajah and Andrew Nix, published by Academic Press

[Macedonia94] Macedonia M.R., Brutzman D.P., 'Mbone provides audio and video over the Internet', IEEE Computer, April 1994., pp. 30-36.

[McCanne96] S. R. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast", *Proceedings of SIGCOMM96*, pp.117-30, August 1996.

[McCanne97] S. R. McCanne *et al*, "Toward a Common Infrastructure for Multimedia-Networking Middleware", *Proceedings of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'97),* Washington University in St. Louis, May 1997.

[MMX] http://www.intel.com/drg/mmx/manuals/index.htm

[Moon98] S. B. Moon, J. Kurose, and D. Towsley, "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms", *ACM / Springer Multimedia Systems*, Vol. 6, pp.17-28, January 1998.

[Montgomery83] W.A. Montgomery, "Techniques for packet voice synchronization", *IEEE Journal on Selected Areas of Communication*, SAC-1 (6), pp.1022-28, December 1983.

[Nee97] P. Nee, K. Jeffay, and G. Danneels, "The performance of two-dimensional media scaling for internet videoconferencing", *Proceedings of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'97),* Washington University in St. Louis, May 1997.

[Ousterhout94] J. K. Ousterhout, "Tcl and the Tk Toolkit", Addison-Wesley, 1994.

[Perkins98] C. S. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet-Loss Recovery Techniques for Streaming Audio", *IEEE Network*, September/October, 1998.

[RAT URL] Http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[RFC1112] S.Deering, 'Host Extensions for IP Multicasting' rfc1112, Internet Engineering Task Force, August 89.

[RTP] H. Schulzerine, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, Internet Engineering Task Force, January 1996.

[RTP-Profile] H. Schulzerine, "RTP Profile for Audio and Video Conferences with Minimal Control", RFC1890, Internet Engineering Task Force, January 1996.

[VIS] http://www.sun.com./microelectronics/vis/

[Wasem88] O. J. Wasem, D. J. Goodman, C. A. Dvorak, and H. G. Page, "The effect of waveform substitution on the quality of PCM packet communications", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36 (3), pp.342-8, March 1988

[Xu97] R. X. Xu, A. C. Myers, H. Zhang, and R. Yavatkar, "Resilient multicast support for continuous media applications*", Proceedings of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'97),* Washington University in St. Louis, May 1997.