UDC 621.397.3

Experiments of Real-Time MPEG Audio over the Internet

●Jey-Hsin Yao ●Yao-Min Chen

(Manuscript received June 5,1997)

This paper reports the development of an audio-on-demand system operating across the Internet. A shorter version of the report can be found in Ref. 1). The software-only real-time playback system uses the MPEG-1 Audio format as compression method and a client-server architecture was assumed. To ensure the quality of playback under adverse network conditions of the Internet, packet-level interleaving and loss compensation were implemented. Two approaches were taken to test the performance of the system; first by continually trafficking the packets over the Internet and secondly by connecting to a network simulator with suitable competing traffic. The network simulator is very useful in studying the behavior of the system with more control of the desired conditions and without interrupting the actual network. Further work was done to incorporate forward correction codes to improve the performance under severe network congestion.

1. Introduction

Multimedia services over the Internet have become popular in the last few years and with these applications arise new challenges for the network industry and research community. While the IP network utilizes the best-effort delivery for conventional data communication and thus has no mechanism to ensure the quality of service (QoS), audio/video applications often have more stringent requirements due to their sensitivity to loss and delay of packets. Under such circumstances, robustness of the system has to be enhanced to counteract the negative effect of excessive delay or possible loss of data packets. Research efforts[2)-4)] in the past mostly focused on video conferencing applications and attempted to solve problems such as audio/video synchronization and rate control. Ref. 5) is devoted to audio but still under the framework of video conferencing. These works mainly address low bandwidth audio with low complexity in order to satisfy the real-time encoding requirement. Our paradigm, on the other hand, is audio-on-demand so that we focus on higher quality, higher bandwidth audio without the real-time encoding constraint. MPEG-1 Audio[6)] was chosen as the method of compression because its Layer 2 encoding offers good audio quality with moderate complexity for real-time software-only decoding and since it is an international standard, a large number of music files are already stored in this format. MPEG Audio has a choice of bit rates between 32 to 192 kbit/s per monaural channel and these bit rates are often sustainable in many parts of the Internet. Lately the MPEG-2 AAC (Advanced Audio Coding)[7)] has also become an international standard and achieves high audio quality at the bit rate of 64 kbit/s. The AAC is attractive also because it outperforms the MPEG-1 coders at half the bit rate and at 64 kbit/s, the audio data can be transmitted through ISDN networks.

2. Framework

We implemented the client-server system on UNIX workstations. The encoded MPEG Audio files reside in the server machine and client programs, running at various locations can play the audio files through the Internet by requesting the data stream from the server.

Figure 1 illustrates the target applications. The clients might be low-end PC's or workstations while the servers must be more powerful and ca-

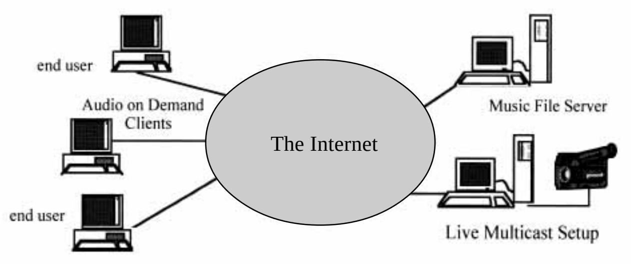


Fig.1– Client-server configuration.

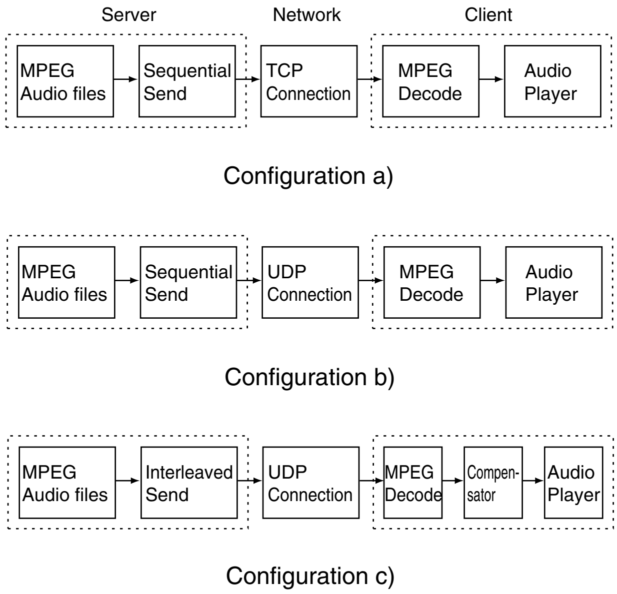


Fig.2– Block diagrams of client-server.

pable of handling multiple connections. In addition to serving preprocessed MPEG Audio data, it is also possible to provide live audio multicast, which may require very high computing power or certain special hardware for real-time encoding on the server side. The client, however, needs only to satisfy the decoding requirement. Such client-server paradigm is similar to that of RealAudio.[8)]

3. Evolution of the playback system

Figure 2 illustrates three different configurations that were tested. At the first stage, we used TCP (Transmission Control Protocol) for all the traffic including MPEG Audio data delivery, as shown in Fig. 2a), and found that it works well in a LAN (Ethernet) environment when the contention level is low. However, excessive delay occurs when the audio packets have to traverse multiple routers and frequent fragmentation of audio playback is observed. In configuration b), we use TCP only for the initial setup of communication (exchange of IP address, port number, file name, etc.) and User Datagram Protocol (UDP), instead, for the MPEG Audio data. A time-stamping function was added to resolve the problem of out-of-sequence arrival of packets. Since the quality of service (QoS) of UDP is by no means guaranteed, some data packets may be lost or fail to reach the client program in time for real-time playback. Consequently, the configuration b) could usually play back the MPEG files in real time but the audio quality is often impaired severely because of missing packets. To cope with this drawback of UDP, a module performing lost packet compensation is necessary. We investigated several methods of lost packet compensation and found by experiments that the loss of a single packet, which corresponds to a frame of 24 ms in MPEG Audio, can be easily compensated by repeating the last frame without much degradation to subjective quality. This method is adopted in our current system because it also requires very little additional computation and the complexity issue is extremely important for our software-only premise. A loss of consecutive packets, on the other hand, cannot be easily compensated and may result in much degradation of audio quality. In Fig. 2c), packet level interleaving, together with a compensator, was incorporated to alleviate the adverse effect of consecutive losses. The next chapter explains the function of such interleaving.

4. Packet-level interleaving

Since packet loss is often caused by congestion of routers over the Internet, there is a correlation of packet loss among packets arriving in the same time interval, thus a loss of consecutive packets is not uncommon if the packets are transmitted in the same order as they are played back. To improve the subjective quality, we have designed an interleaving scheme for the audio server to disperse possible burst errors. Basically the server buffers up a block of audio data packets and transmits them in a certain order such that the probability of consecutive packet loss is reduced after the client received and rearranged the packets. **Figure 3** explains the concept of packet interleaving.

Informal subjective listening confirmed that interleaving combined with simple packet-loss compensation significantly improves the performance of our system. More sophisticated compensation scheme was also investigated but low complexity requirement puts a tight constraint on available options.

5. Design principle for interleaving

Interleaving is implemented on the server and in essence the idea is to manipulate the sending order of data packets. A quick method to generate an interleaved sequence is to write an ordered sequence of numbers in rows and then read the elements out in columns. For example, in a buffer of size 24, we may construct a matrix like this:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24

and produce sequence {1, 7, 13, 19, 2, 8, 14, 20, 3, 9, 15, 21, 4, 10, 16, 22, 5, 11, 17, 23, 6, 12, 18, 24}. However, we can increase the average distance of adjacent packets by swapping the columns and obtain the following matrix.

2	6	4	1	3	5
8	12	10	7	9	11
14	18	16	13	15	17
20	24	22	19	21	23

Reading out by columns, we get the sequence {2, 8, 14, 20, 6, 12, 18, 24, 4, 10, 16, 22, 1, 7, 13, 19, 3, 9, 15, 21, 5, 11, 17, 23}, which is the order the server sent MPEG Audio packets in experiments reported in Chapters 6 and 7.

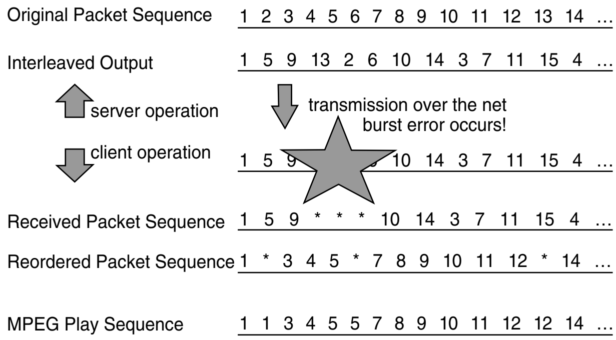


Fig.3– Interleaving and compensation.

While the above heuristic design method worked well in simulations, we also considered a more general problem of theoretical interest. Let s denote sequence (1,2, ..., n). A *permutation sequence* for *s* is a sequence that forms a mathematical permutation of *s*. For example, (8,4,7,1,5,2,6,3) is a permutation sequence for (1,2,3,4,5,6,7,8). Note that a permutation sequence defines a mapping π from the elements of *s* onto themselves. In the example, $\pi(1)=8$, $\pi(2)=4$, $\pi(3)=7$, $\pi(4)=1$, etc. In a buffer of size n, there exist n-factorial permutations and the design goal of interleaving is to find one (may not be unique) permutation that maximizes the average Hamming distances for all packets adjacent in the original (playback) sequence. We attempted to solve this problem analytically but did not obtain significant breakthroughs. If n is small, we could resort to full search to determine the optimal permutation or sequence. Nonetheless, experiments using such sequences do not show noticeable improvement in performance in comparison with ones using sequences heuristically designed from matrix manipulation. Thus for practical purposes, we simply relied on the heuristic method and did not investigate further the theoretical solution. Another design issue is the buffer size. Naturally the longer the buffer is, the larger Hamming distance can be achieved and consequently, the more effective burst packet loss can be removed. However, a larger buffer also implies higher memory requirement and longer initialization delay and this is a tradeoff worth further investigation.

To test our interleaving strategy, we conducted experiments both over the real Internet and on a simple network simulator. In the experiments, we found that the system performs satisfactorily with a buffer size of 24, corresponding to a delay of approximately 0.6 seconds and with the sending order described earlier in this chapter.

6. Experiment over the Internet

For a 2-day period, a 96-kbit/s MPEG Audio data stream was continually transmitted from University of Hawaii to Fujitsu Laboratories of America (Santa Clara, California). We switched between interleaving and non-interleaving transmission every five minutes. At the end of each five-minute interval, we collected packet loss statistics which included the overall loss rate (the number of lost data packets divided by the number of packets transmitted by the sender) and the rate of consecutive loss at different lengths.

In **Fig. 4**, the loss rate (the ratio of the data packets that failed to arrive in time for real-time playback and the total number of packets sent) over time is drawn. The solid curve corresponds to the case where interleaving was used, while the dash-dotted curve represents the non-interleaving one. The curves reflect the typical traffic load of the Internet, which is light at night (low loss rate) and heavy during the day (high loss rate). Not surprisingly, the two curves are close since interleaving should affect how loss is distributed but not the overall rate (based on the assumption that a similar degree of congestion occurs equally likely to an interleaved or a non-interleaved packet stream in a given short time frame).

In **Fig. 5a**) and **b**), we consider packet loss of different lengths. The quantity plotted in the figures is the number of burst loss occurrences in a five-minute interval. Figure 5a) shows the values corresponding to loss of lengths 1, 2 and 3, and Figure 5b) shows the values corresponding to lengths 4, 5 and 6. From the figures, we observe that interleaving indeed reduces the frequency of long burst loss as desired.

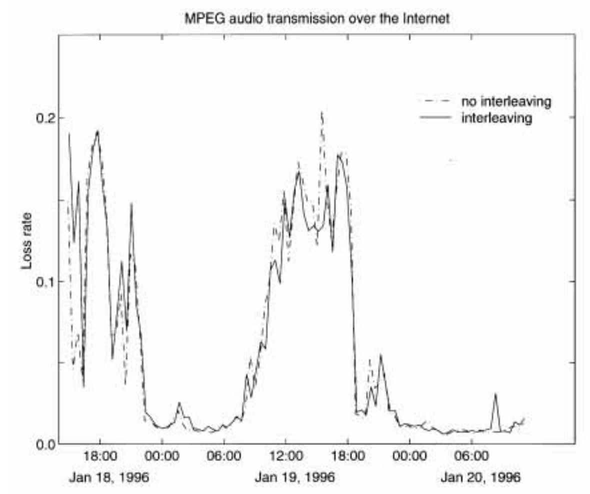
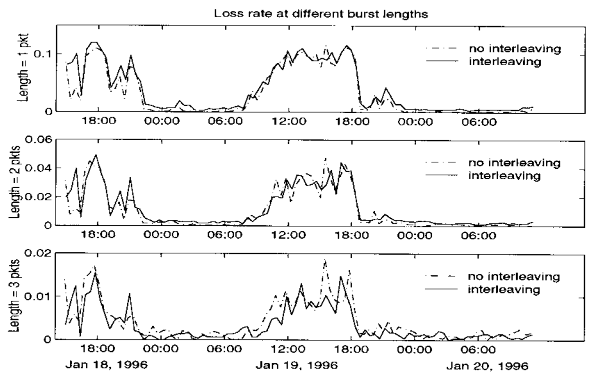
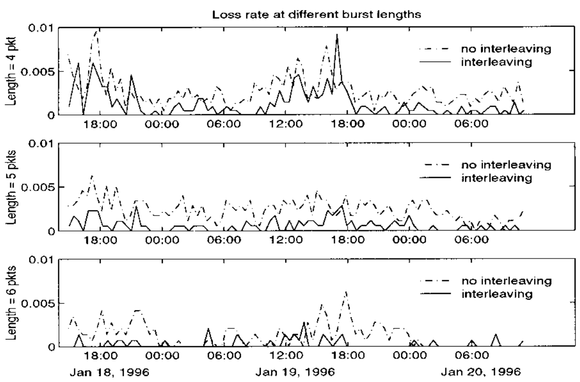


Fig.4– Loss rate for MPEG Audio over the Internet.



a) Lengths 1, 2 and 3



b) Lengths 4, 5 and 6

Fig.5– Packet loss.

It should also be noted that according to the figures, the interleaved transmission has higher rates at some instances even for consecutive packet losses. This is due to the time difference the simulations were conducted. Although they are only five minutes apart, the traffic load can vary widely over the Internet.

7. Experiment over a Network Simulator

We have also implemented a network simulator to model internetworks where packet loss rate is better controlled.[9] The simulator uses queuing systems to model network elements, such as routers or switches and simulates packet loss, delay and jitter that may occur when audio packets are transmitted over the Internet. Instead of sending MPEG Audio over the real Internet, we sent packets through the simulator. Within the simulator, artificial traffic was generated to compete with the MPEG Audio traffic for limited network resources including buffer and transmission capacity. The artificial traffic has Poisson interarrival times and an exponential packet-size distribution. Since the MPEG Audio traffic was routed through the simulator in real time, subjective listening test can be conducted at the receiving end to evaluate the perceptive quality of the received audio. A diagram of the simulator is provided in **Fig. 6**. Note that it is possible to concatenate several single-stage simulator and form a multiple-stage one but our experiment was based on the single-stage model.

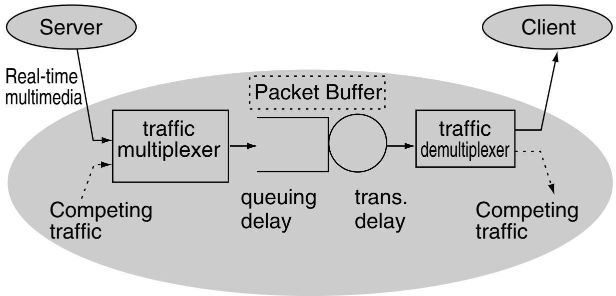


Fig.6– Single-Stage Network Simulator.

Our experiment used the following parameters. First, the transmission capacity of the FIFO queue was 1.5 Mbit/s. Second, the buffer size is 10 Kbytes. Finally, the competing traffic has an average packet size equal to the size of an MPEG Audio packet, where the MPEG Audio data frame has 432 bytes.

We varied the intensity of the competing traffic and considered the corresponding burst loss distribution. We compared the distributions resulting from our experiment with a reference distribution, defined below. Let N denote the total number of frames sent. Let p denote the overall loss rate. Note that if the loss probability of each frame were independent of other frames, the amount of packet loss with burst length k would be $N(1–p)^2p^k$; this distribution is the ideal case when the correlation among packet loss probabilities is completely removed. In **Tables 1-3**, the statistics collected from our experiment are compared against the corresponding values if the packet loss process were to follow the reference distribution. The percentage that an experiment statistic deviates from its reference value is listed right next to the experiment statistic (inside the parentheses), where a plus (resp., minus) sign means that the statistic is larger (resp., smaller) than the reference value.

From the tables, it is not hard to conclude that our interleaving strategy results in burst loss distribution much closer to the reference distribution than the distribution when no interleaving is used. In this sense, interleaving effectively removes the packet loss correlation.

8. Conclusion

We present a working audio-on-demand system that can operate over the Internet. The system uses MPEG Audio but its design is not tied to any specific compression scheme. The technique of packet-level interleaving was introduced to disperse consecutive packet loss and thus improved the performance. Experiments conducted over the Internet validate the use of interleaving and we further verified the results by connecting the system to a network simulator. Theoretical work also

Table 1. In this table we show the distribution of burst loss when the arrival rate of the competing traffic was 1.5 Mbit/s. Here, the total number N of audio frames was 970,567. The overall loss rate p resulting from the experiment was 0.124.

Burst length k	No Interleaving		Interleaving		Reference value $N(1-p)^2 p^k$
1	76,985	(−6.94%)	81,453	(−1.39%)	82,605
2	11,984	(+17.0%)	10,613	(+4.02%)	10,203
3	1,834	(+44.6%)	1,354	(+7.46%)	1,260
4	278	(+77.1%)	164	(+5.12%)	156
5	34	(+78.9%)	26	(+36.8%)	19
6	4		0		0
7	0		0		0
8	0		0		0
9	0		0		0
10	0		0		0

Table 2. In this table we show the distribution of burst loss when the arrival rate of competing traffic was 1.65 Mbit/s. The number N of data frames saent was 1,183,894. The experiment resulted in overall loss rate $p = 0.124$.

Burst length k	No Interleaving		Interleaving		Reference value $N(1-p)^2 p^k$
1	104,879	(−6.76%)	113,212	(−0.76%)	112,355
2	16,209	(+16.4%)	13,416	(−3.34%)	13,881
3	2,484	(+44.3%)	1,684	(−1.81%)	1,715
4	376	(+76.5%)	250	(+17.9%)	212
5	41	(+57.7%)	22	(−15.4%)	26
6	7		7		3
7	0		2		0
8	0		0		0
9	0		0		0
10	0		0		0

Table 3. In this table we show the distribution of burst loss when the arrival rate of competing traffic was 1.35 Mbit/s. The number of audio frames sent, N, was 896,682. The overall loss rate p was 0.005.

Burst length k	No Interleaving		Interleaving		Reference value $N(1-p)^2 p^k$
1	3,612	(−16.2%)	4,378	(−0.41%)	4,396
2	349	(+1562%)	31	(+40.9%)	22
3	0		0		0
4	0		0		0
5	0		0		0
6	0		0		0
7	0		0		0
8	0		0		0
9	0		0		0
10	0		0		0

showed that interleaving removes the correlation among the loss probabilities of neighboring packets, which suggests that interleaving may be beneficial to the transport of other stream media as well. However, the buffering delay caused by interleaving may preclude its use for most real-time interactive applications. In a related research project, the combination of interleaving and forward error correction based on the current system was probed and found effective in retaining audio quality at low to medium packet loss rate and achieving graceful quality degradation at a higher loss ratio.[10] For future research, the system is currently being modified to generate RTP[11] standard compliant streams, so that the server may obtain feedback information from the client and other network elements to adapt its function accordingly.

Acknowledgment

We would like to thank Professor Galen Sasaki of University of Hawaii who let us access computer facilities there.

References

1) Yao, J-H., Chen, Y-M. and Verma, T. : MPEG-Based Audio-On-Demand Experiment for the Internet. Interworking 96, Nara, Japan, Global Information Infrastructure (GII) Evolution, ed. Rao, S., Uose, H., and Luetchford, J. C., IOS Press, Amsterdam, Netherlands, pp.503-511, October 1996.
2) Chen, Z., Tan, S., Campbell, R., and Li, Y. : Real-Time Video and Audio in the World Wide Web. Proceedings of the 4th World Wide Web Conference, Boston, MA, pp.333-348, December 1995.
3) Chaddha, N., Wall, G., and Schmidt, B. : An end-to-end software only scalable video delivery system. Proceedings of NOSSDAV 95, Durham, NH, pp. 139-150, April 1995.
4) Jeffay, K., Stone, D. L., Talley, T., and Smith, F. D. : Adaptive Best-Effort Delivery of Digital Audio and Video Across Packet-Switched Networks. Proceedings of NOSSDAV 93, pp.3-14.
5) Bolot, J-C. and Vega-Garcia, A. : Control Mechanisms for Packet Audio in the Internet. Proceedings of INFOCOMM 96, San Francisco, CA, pp.232-239, March 1996.
6) Brandenburg, K. and Stoll, G. : ISO-MPEG-1 Audio: A Generic Standard for Coding of High Quality Digital Audio. *Journal of Audio Engineering Society,* October 1994.
7) ISO/IEC, JTC/SC29/WG11, MPEG Audio subgroup : Informational Text on Advanced Audio Coding. *http://www.chips.ibm.com/.mpeg/Sevilla/N1518.*
8) Progressive Network : About RealAudio. *http://www.realaudio.com/prognet.*
9) Chen, Y-M : Internet Simulator for Testing Networked Multimedia. *FUJITSU Sci. Tech. J.* **33***,* 2, pp. 127-137(1997).
10) Chen, Y-M., Wang, H., Taniguchi, T., and Sasaki, G. : Robust MPEG Audio for the Internet. IEEE Global Telecommunications Conference 97.
11) Schulzrinner, H., Casner, S., Frederick, R., and Jacobson, V. : RTP: A Transport Protocol for Real-Time Applications. Internet RFC 1889, *ftp://ds.internic.net/rfc/rfc1889.txt.*

Jey-Hsin Yao received his B.S. degree in Electrical Engineering from National Taiwan University in 1984. He also received M.S. and Ph.D. degrees from University of California, Santa Barbara in 1987 and 1992, respectively. After completing his degrees, he joined Fujitsu Network Transmission Systems, Inc. and transferred to Fujitsu Laboratories of America in 1994. His research interests include speech coding, digital signal processing and multimedia networking issues.

Yao-Min Chen is a Member of Research Staff with Fujitsu Laboratories of America, Santa Clara, California, USA. He received the Ph.D and M.S. degrees in Electrical and Computer Engineering from the University of Texas at Austin, in 1994 and 1991 respectively, and the B.S. degree in Electrical Engineering from National Taiwan University in 1987. In 1995, he conducted post-doctoral research in University of Hawaii at Manoa before joining Fujitsu. His research interests are in communication networks, multimedia, and algorithms.