IEEE International Conference on Multimedia & Expo 2001, Tokyo, Japan, Aug. 2001.

SCALABLE AUDIO STREAMING OVER THE INTERNET WITH NETWORK-AWARE RATE-DISTORTION OPTIMIZATION

Jianping Zhou[1] and Jin Li[2]
[1]Microsoft Research China, 3F 49 Zhichun Road, Haidian, Beijing 100080, P. R. China.
[2]Microsoft Research, Signal Processing, One Microsoft Way, Bld. 113/3033, Redmond, WA 98052, USA.
Email: {i-jpzhou,jinl}@microsoft.com

ABSTRACT

Reliable audio streaming over the unreliable Internet is a challenging task. Because during a connected session, the bandwidth can vary greatly and the data packets can be lost in the transmission, it is difficult to guarantee a smooth quality of the delivered audio over the Internet. In this work, a system for streaming of scalable coded audio over the Internet is proposed. Scalable audio coding generates a bitstream consisting a number of data units (DUs), where a subset of the DUs can be extracted to reconstruct audio at a lower quality level. A network aware rate-distortion optimization model is proposed so that the server can easily adapt to the network bandwidth fluctuation by sending less packets when the network is congested. The server can also deal with severe packet loss by only retransmitting the more important data packets. By using the scalable audio coding and network aware delivery scheme, the quality of the audio streaming over severe network condition is improved dramatically.

1. INTRODUCTION

Reliable streaming of media (audio/video) content over the Internet is a challenging task. This is due to the fact that the Internet is a packet switch network with little quality of service (QoS) guarantee. The bandwidth of the Internet varies widely in term of the origin/destination and time of use. Moreover, the bandwidth of a single connected session also fluctuates during the session. The packet may be lost and delayed. Unlike data transmission, which can simply slow down and retransmit the loss packet in case of network congestion, media must be delivered within certain time constraint. All of these make it a challenging task to stream the medias reliably over the Internet.

In this work, we focus on audio streaming. Since audio requires less bandwidth than video, it is easier to be delivered over the Internet. Nevertheless, a high quality audio stream requires 64-128 kbps (kilo bits per second) bandwidth, which is still challenging to be delivered reliably over long-range, e.g., from one continent to another.

The simplest audio streaming scheme compresses the audio into a single bitstream, which is then sent sequentially over the Internet. Because the bitstream cannot be altered after it has been compressed, it is difficult to adapt to the network bandwidth fluctuation. The audio streaming schemes today, such as the Real Player™ and Media Player™, prepare for a single audio source several compressed bitstreams at different bitrates. For a client with a certain bandwidth, the server then selects the bitstream with the bitrate just below the bandwidth. In case the network bandwidth reduces or the packet loss ratio is high, the server may also switch to a lower bandwidth bitstream. Such multi-rate scheme improves the flexibility of the delivery, however, more server storage space is required. Moreover, switching between bitrate usually leads to audible artifacts. It is also a common practice to use the client buffer to smooth out the bandwidth fluctuation. When certain data packet is lost, the client can still play from the buffer while waiting for the data packet to be retransmitted. To combat occasional packet loss, various schemes have been developed[1]. One approach is the Automatic Retransmission Request (ARQ), which retransmits the lost packets after the server receives a negative acknowledgement from the client that the data packet is lost. Forward Error Correction (FEC) [2] is another approach, where additional parity packets are transmitted to protect the data packets. A third approach is interleaving and buffer management [3], which disperses the burst error caused by the lost packet into random errors in the bitstream that is further corrected by the FEC. These approaches can ensure reliable delivery with a low packet loss ratio. However, when the packet loss ratio is high or fluctuates widely during the Internet connection, reliable delivery cannot be ensured.

Scalable audio coding becomes popular recently[4][5]. For a scalable coded audio, the compressed bitstream can be classified into a number of data units (DUs). It is possible to use only a subset of the DUs to decode the audio with lower sampling resolution and/or quality level. In this work, we investigate the streaming of scalable encoded audio over the Internet. The key concept is to selectively deliver the DUs of a scalable encoded audio to adapt to the network bandwidth fluctuation and packet loss level. A rate-distortion based packet selection scheme has been proposed. In case the bandwidth of the connected session reduces, the DUs that offer less distortion decrease per coding rate are not sent. Similarly, when the network packet loss ratio is high, only the most important DUs in term of rate-distortion are retransmitted over the Internet. Using such a network-aware rate-distortion optimization strategy, we can efficiently delivery the audio bitstream over the Internet.

The rest of the paper is organized as follows. We first introduce the scalable audio coder in Section 2. The network-aware rate-distortion optimizated data unit selection is discussed in Section 3. Simulation results are given in Section 4. Finally, conclusions are given in Section 5.

2. SCALABLE AUDIO CODING

The framework of the scalable audio codec can be shown in Figure 1. The audio signal is first split into individual time slot, each of which is filtered by a PQF filter and down-sampled into four subbands. The subbands provide scalability in the audio sampling resolution. In case the computation power is tight, or the client device is at the low end, the high pass subband signals

IEEE International Conference on Multimedia & Expo 2001, Tokyo, Japan, Aug. 2001.

may be discarded. Modified DCT (MDCT) is performed on each decomposed subband. The transformed coefficients are then weighted according to a psychoacoustic mask. Finally, each weighted subband is bitplane encoded into an embedded bitstream. The embedded bitstream has the property that the resultant bitstream can be truncated at any point and still yields a decodable, albeit lower quality signal. We may assign several bitrate to the embedded encoded audio. For each coding bitrate, the available bits are assigned to the four subbands according to the rate-distortion criterion. The bitstreams of the subbands are then truncated and concatenated to form the encoded audio. For the lowest coding bitrate, a set of four truncated bitstream from each subband forms the base layer of the embedded audio. For a higher bitrate, we only need to transmit the difference between the truncated bitstream of this layer and that of the previous layer. Suppose there are a total of *n* bitrate points, each subband bitstream is thus truncated into *n* segments. We call a segment of the truncated bitstream of a subband as a Data Unit (DU). DU is the smallest unit in the delivery.

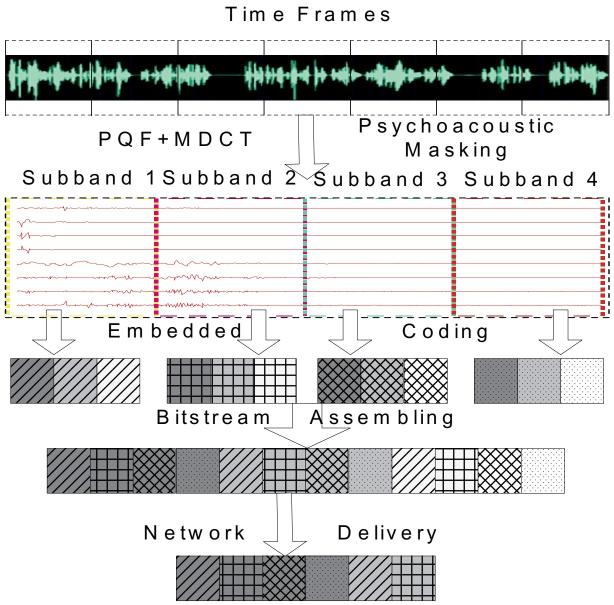


Figure 1 The framework of the scalable audio coder.

3. NETWORK-AWARE RATE-DISTORTION OPTIMIZATED DELIVERY

Unlike conventional audio coding, where the compressed audio bitstream is an integral unit, the scalable encoded audio is consisted of a set of data units (DUs) which can be flexibly organized in delivery. In this work, the server controls the DUs to be delivered to the client, while the client feeds back information about network bandwidth, packet loss ratio and delivered packets. The sending bitrate is controlled through a TCP-friendly Additive Increase Multiplicative Decrease (AIMD) algorithm [6]. To reduce packet overhead, multiple small DUs are packaged into a large network packet for delivery. When the data packet is delivered to the client, it is split into individual DUs. The DUs, i.e., the bitstream segments of each individual subband are then merged together to form the coding bitstream of each subband. Not all DUs are received, especially when the bandwidth is tight and the packet loss ratio is high. Nevertheless, the assembled bitstream of each subband forms a truncated bitstream, which can be decoded up to the truncation point. The original data frames and the regenerated data frames at the receiver may be substantially different, especially as some DUs may not be delivered to the receiver. However, it does not matter since the scalable decoder can still decode the audio, albeit low quality, from the delivered DU.

It is essential to identify the lost packets as soon as possible, so that the remedy can be applied. We mark each data packet with a sequence number. A packet is considered lost if a packet with a sequence number three higher than the missing one is delivered. A distinct acknowledgement mechanism termed joint-ACK-NACK feedback is utilized to feedback the packet loss information to the server. For the positive acknowledgement, we give a start sequence number and an end sequence number in the message, and acknowledge the receipt of all packets within the sequence number range. The sequence numbers of all packets that are considered lost are then listed afterwards, which constitutes the negative acknowledgement part. In this way, the positive and negative acknowledgement information is sent within a single message. Based on the number of delivered packets within a certain period of time, it is possible for the server to estimate both the network bandwidth and the packet loss ratio.

Armed with the feedback information, we use a network-aware rate-distortion optimization model to selectively transmit the DUs. Let the time window be N+1 time slots. Let *t = 0* be the current time slot, and *t = 1,2,3,…, N* be a sequence of *N* time slots in the future. There are four subbands within each time slot, and each subband is encoded into *L* DUs. There are altogether *4L(N+1)* DUs in the current time window. Let *PLR* be the current packet loss ratio of the network.

We index the DU as *DU(l,k,t)*, where *l, k* and *t* index the layer, subband and timeslot, respectively. Let the DU consume *s(l,k,t)* bits for coding. After decoding the DU, let the distortion of the reconstructed audio decreases by *d(l,k,t)*. Let *P(l,k,t)* be the probability that the DU be delivered to the client without sending it at the current timeslot. If *DU(l,k,t)* has never been sent in the previous time slots, or a negative feedback of the data packet containing *DU(l,k,t)* has been received, *P(l,k,t)=0,* i.e., the DU is definitely not received by the client. If a positive feedback of the data packet containing the DU has been received, *P(l,k,t)=1*, i.e, the DU has definitely been received by the client. If the DU has been sent, but a feedback has not been received from the client, *P(l,k,t)=1-PLR.* As a summary, we have:

$$P(l,k,t)=\begin{cases} 0 & \text{not sent or NACK received} \\ 1 & \text{ACK received} \\ 1-PLR & \text{sent, no acknowledgement} \end{cases}$$

We assign a gain factor *G(l,k,t)* for each DU, which is the expected coding distortion decrease per bit sent if the DU is sent at the current time slot. A network-aware rate-distortion optimal delivery strategy is thus to selectively transmit the DUs in the current time slot with the largest gain factor so that the long-term play back quality of the delivered audio is maximized. Since once the gain factors are calculated, selecting the DU with the largest gain factor is rather easy. The task becomes: to calculate the gain factor *G(l,k,t)* for each DU, based on the coding distortion *d(l,k,t),* DU length *s(l,k,t)*, delivery status *P(l,k,t)*, and current network packet loss ratio *PLR*, and a number of other factors. In this work, the gain factor *G(l,k,t)* of the DU is calculated through formula:

IEEE International Conference on Multimedia & Expo 2001, Tokyo, Japan, Aug. 2001.

$$G(l,k,t) = I(l,k,t) * R(l,k,t) * D(l,k,t) * ar(t) * pr(t) / s(l,k,t) \cdot$$

There are six items in the above formula. They are analyzed one by one below.

Item 1. Sent status *I(l,k,t)*.

We only send the DU that has not been sent before or has been negatively acknowledged. Therefore,

$$I(l,k,t) = \begin{cases} 1-PLR & \text{not sent or NACK received} \\ 0 & \text{otherwise} \end{cases}$$

Item 2. Reliance factor *R(l,k,t)*.

Since the embedded coder can only decode a truncated bitstream, not a bitstream with a missing central piece, the current DU can only be decoded if and only if all DUs of the former layers have been received. Therefore:

$$R(l,k,t) = \prod_{i<l} P(i,k,t)$$

Item 3. Distortion decrease *D(l,k,t)*.

The delivery of the current DU not only enables the current DU to be decoded, but also enables the DU in the hinder layer to be decoded. Therefore, the combined distortion decrease caused by sending the DU is:

$$D(l,k,t) = d(l,k,t) + \sum_{i=l+1}^{L} d(i,k,t) \prod_{j=l+1}^{i} P(j,k,t)$$

Item 4. On time delivery probability *ar(t)*.

We assign *ar(t)* to be the probability that the *DU(l,k,t)* will arrive at the client early enough for playback. We model *ar(t)* by observing the roundtrip time of packet delivery between the server and the client. The use of factor *ar(t)* prevents the sending of near future DUs which may not arrive on time for playback.

The product of the first four items thus analyzes the distortion decrease achieved by sending *DU(l,k,t)* in the current time slot.

Item 5. Balance factor *pr(t)*.

We assign *a prior* a balance factor *pr(t)* to address the importance of the near future time slots. Since the gain factor is calculated to select DUs to be sent in the current time slot, emphasis should be placed on DUs of the near future time slots, as DUs in the far future time slots can be sent later. It balances between the delivered audio quality versus error robustness.

Item 6. DU Bits *s(l,k,t)*

The longer the DU, the less number of DUs that can be sent in the current time slot. We thus divide the calculated gain factor by the number of bits in the DU.

After the gain factors of all DUs at the current time window have been calculated, the server may select to send out the DUs that offer the biggest gain. The server keeps sending the DUs until the allocated bandwidth of the current time slot has reached.

4. EXPERIMENTAL RESULTS

We optionally turn on-and-off a few features to show the effectiveness of the proposed scalable audio streaming framework. The following four schemes are compared:

(a) No rate-distortion optimization scheme
(b) Rate-distortion optimization without acknowledgement
(c) Rate-distortion optimization with flat balance factor *pr(t)*
(d) Rate-distortion optimization with balance factor *pr(t)* emphasizing the near future time slots

In all schemes, the scalable audio delivery system presented in the previous section is used. For scheme (a), we set *d(l,k,t)* and *s(l,k,t)* to be constant, therefore, no rate-distortion optimization is used to distinguish between different DUs. For scheme (b), no acknowledgement is sent by the client, therefore, the delivery probability *P(l,k,t)* will be 0 if a packet is not sent, and *(1-PLR)* if a packet has been sent. Scheme (c) and (d) differ only on the balance factor, i.e., whether the data units (DUs) in the near future time slots are emphasized.

The MPEG-4 standard audio clips horn23_2, trpt21_2 and vioo10_2 are used for test. The scalable audio coder encodes each audio clip at a coding rate of 64 kbps. The bitstream is then streamed from the server to the client. We illustrate the quality of the decoded audio through the noise-mask-ratio (NMR), where a lower value of the NMR shows a better quality decoded audio.

Assuming no packet loss, we first investigate the issue of bandwidth fluctuation. We deliver the audio clip horn23_2 without packet loss from the server to the client. During the streaming, the bandwidth of the connection increases from 20 to 64 kbps. The NMR curves for the four coders are shown in Fig. 2. Since there is no packet loss, acknowledgement information is of no-use, therefore, scheme (b) performs equal to scheme (d). It is observed that scheme (a) performs worst almost all schemes. The fact shows that the rate-distortion optimization does improve the quality of the delivered audio. With a flat balance factor, scheme (c) outperforms scheme (d) at the beginning 20 time slots. However, it lags behind scheme (d) for the remaining time slots. By preferring the delivery of DUs at the near future time slots, scheme (d) realizes the fact that the DUs in the far future timeslot may be delivered at a later time. It thus improves the overall quality of the delivery audio.

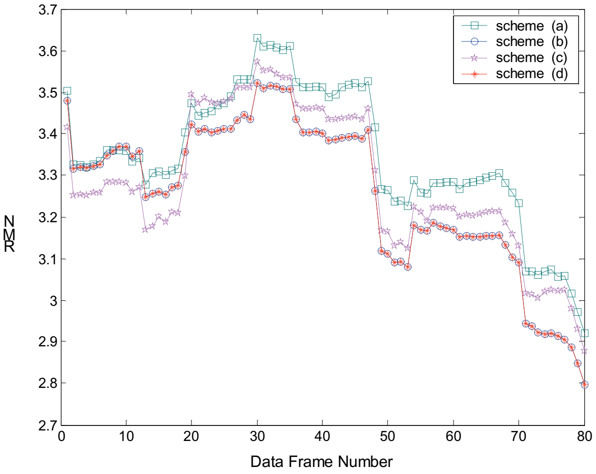
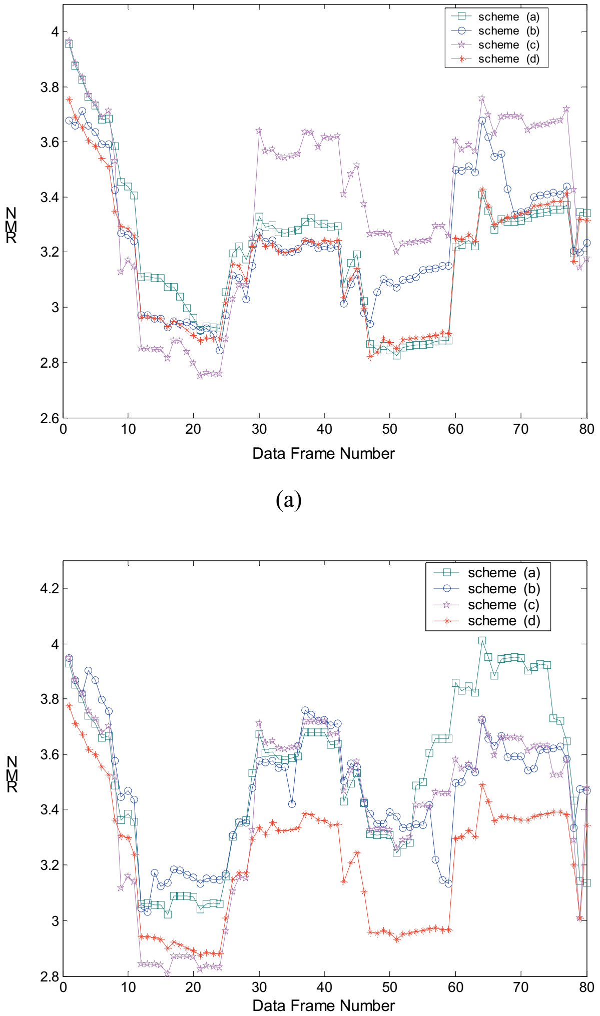


Fig 2. Comparison among four delivery schemes with bandwidth fluctuation. The audio clip is horn23_2 and there is no packet loss.

In the second experiments, we simulate a network environment with constant bandwidth of 64kbps but a non-zero packet loss ratio. The NMR of the four comparison schemes under the packet loss ratio of 5% and 20% are shown in Fig 3(a) and (b), respectively. In the beginning, scheme (c) works pretty fine. However, since the flat balance factor does not differentiate among near and far future DUs, it suffers later as important DUs in the near future time slot are lost. Overall, we observe that the scheme (d) performs the best.

IEEE International Conference on Multimedia & Expo 2001, Tokyo, Japan, Aug. 2001.

The numerical NMR results for streaming of audio clips honrn23_2, trpt21and vioo10_2 under constant network bandwidth (64kbps) and packet loss ratio of 5% and 10% are shown in Table 1. It is obvious that the scheme (d) outperforms all other approaches as it provides the lowest average NMR value for all clips and packet loss ratio. Subjective tests have also been conducted for the conditions listed in Table 1. Listeners prefer the audio delivered by scheme (d), while annoying artifacts are audible in the audio delivered by schemes (a), (b) and (c). This proves the fact that the rate-distortion optimization, network feedback, and balance between near and future time slots all play an important role in improving the quality of audio streaming.



(a)

(b)

FIG 3. Comparison among four delivery schemes with different packet loss ratios: (a) 5%; (b) 20%. The audio clip is horn23_2 and the bandwidth is 64 kbps.

5. CONCLUSIONS AND DISCUSSIONS

A scheme for the delivery of scalable coded audio over the Internet is presented. We encode the audio through a scalable audio coder into a set of data units. During streaming, the data units with the largest gains in expected distortion decrease per bit spent are selected for delivery. The proposed scalable audio delivery framework can easily accommodate the fluctuation of network bandwidth and high packet loss ratio, and improve the quality of audio streaming over the unreliable Internet.

6. ACKNOWLEDGEMENT

The authors would like to acknowledge the following individuals: Geoff Davis and Henrique (Rico) Malvar for providing the source code of the original scalable audio coder, Liexian Gu for improving the scalable audio decoder both in speed and performance, Jiangang Duan for programming the network interfance, and Qian Zhang for helpful discussions.

REFERENCE

[1] Colin Perkins, Orion Hodson, and Vicky Hardman, “A survey of Packet Loss Recovery Techniques for Streaming Audio”, IEEE Trans. Network, vol. 12, no. 5, Sept. 1998, pp. 40-48.

[2] Podolsky M., Romer C. and McCanne S. “Simulation of FEC-based error control for packet audio on the Internet,**”** INFOCOM '98, Seventeenth Annual Joint Conference of the IEEE Computer and Communication Societies, vol. 2, 1998, pp. 505-515.

[3] Shacham N. and McKenney P., “Packet recovery in high-speed networks using coding and buffer management”, INFOCOM '90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies, vol. 1, 1990, pp. 124-131.

[4] Moriya T., Iwakami N., Akio Jin and Mori, T., “A design of lossy and lossless scalable audio coding,” IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00), 2000, vol. 2, pp. 889-892.

[5] Kudumakis P.E. and Sandler M.B., “Wavelet packet based scalable audio coding”, IEEE International Symposium on Circuits and Systems (ISCAS '96), 1996, Vol. 2, pp 41-44.

[6] Qian Zhang, Wenwu Zhu and Ya-Qin Zhang, “Network-adaptive rate control with TCP-friendly protocol for multiple video objects”, IEEE International Conference on Multimedia and Expo (ICME 2000), 2000, vol 2, pp 1055-1058.

Table 1. Average NMR results for multiple audio clips with different delivery schemes.

Audio Clips Sch-emes	horn23_2		trpt21_2		vioo10_2	
	PLR= 5%	PLR= 20%	PLR= 5%	PLR= 20%	PLR= 5%	PLR= 20%
(a)	3.202	3.425	3.868	3.969	4.577	4.637
(b)	3.184	3.403	3.923	4.294	4.733	4.744
(c)	3.300	3.320	3.900	3.987	4.509	4.556
(d)	3.144	3.192	3.856	3.928	4.398	4.452