SCHEMES FOR ERROR RESILIENT STREAMING OF PERCEPTUALLY CODED AUDIO

Jari Korhonen and *Ye Wang*[1]

Speech and Audio Systems Laboratory
Nokia Research Center, Tampere, Finland

ABSTRACT

This paper presents novel extensions to our earlier system for streaming perceptually coded audio over error prone channels such as mobile IP. To improve error robustness while maintaining bandwidth efficiency, the new extensions combine the strength of an error resilient coding scheme in the sender, a prioritized packet transport scheme in the network and a compressed domain error concealment strategy in the terminal. Different concealment methods are used for each part of the coded audio data according to their perceptual importance and statistical characteristics. In our current implementation, we employed MPEG-2 Advanced Audio Coding (AAC) encoded bitstreams and an RTP/UDP-based test system for performance evaluation. Simulation results have shown that our improved streaming system is more robust against packet losses in comparison with conventional methods.

1. INTRODUCTION

Real-time multimedia streaming is one of the most rapidly evolving applications in IP networking. Streaming multimedia especially to mobile terminals sets contradictory requirements for transport delays and error robustness. Packet losses should be compensated efficiently to ensure high quality of service. On the other hand, real-time delivery of data and limitations for bandwidth consumption restrict usage of packet retransmissions. That is why different techniques for error correction, error resilience and concealment have been developed. The focus of this paper is on perceptually coded high quality audio.

An error resilient audio streaming system should be a well-balanced solution, which involves the sender, the network and the receiver. Redundancy is usually added from the sender for error protection. Retransmission is a network-level solution, and error concealment is usually considered as a receiver-based last resort. A good survey on this topic can be found in [1].

Perceptually coded audio bitstreams are generally more vulnerable against errors than their uncompressed counterparts. Variable-length coding, such as Huffman coding employed in AAC, is particularly vulnerable to individual bit errors, since it may cause error propagation from codeword to codeword.

MPEG-4 standard includes some error resilience tools to protect AAC bitstream against individual bit errors [2]. However, these tools are less effective in packet-switched streaming, because the network typically discards the entire packet if there are bit errors detected.

A frame in perceptually coded audio streaming is usually considered as the smallest decodable data unit. That is why existing packet loss recovery methods are usually based on the assumption of the loss of an entire frame [1].

To increase the error robustness against packet loss, we have proposed a method of shuffling AAC coded data elements among multiple packets [3]. However, existing error concealment methods and error resilience tools are not very effective for the new scenario. New tools are therefore needed. This need has led us to find some new solutions.

The paper is organized as follows. A brief discussion on the error sensitivity of AAC coded data is given in section 2. The key points of our earlier system are summarized in section 3. Then the novel extensions on packetization and scalefactor coding are presented, and error concealment of quantized Modified Discrete Cosine Transform (MDCT) spectral data is discussed in section 4. The simulation results are summarized and analyzed in section 5. Finally, section 6 concludes the paper.

2. ERROR SENSITIVITY IN AAC CODED DATA

Data in each AAC audio frame can be classified roughly into three categories according to their error sensitivity (see Figure 1).

2.1. Critical Data

The critical data part includes the most crucial information, such as MDCT window type and length (short or

[1] Ye Wang is currently with School of Computing, National University of Singapore

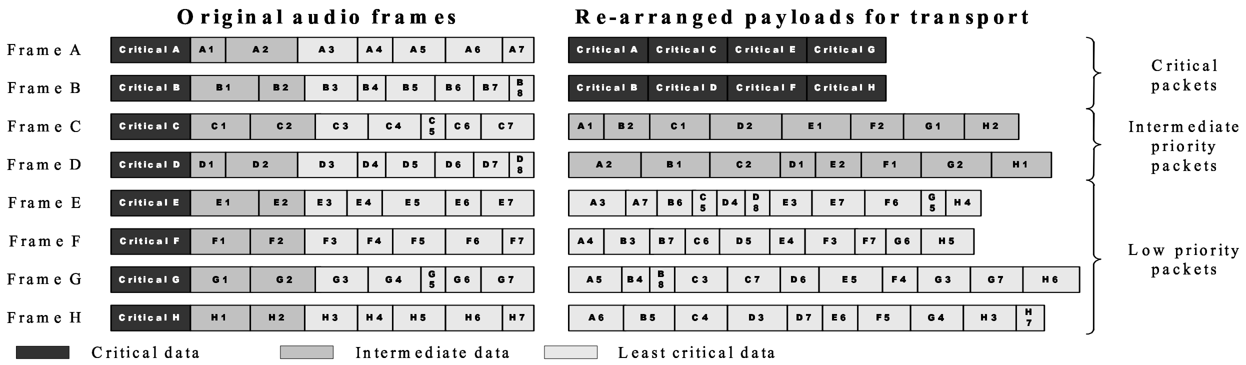

Figure 1. Splitting a set of audio frames into transport packet payloads in the proposed scheme.

long window), section data with Huffman codebook indices for different spectral data sections and global gain for scalefactors. Without the critical data it is extremely hard, if not impossible, to decode the remaining data in the frame.

2.2. Scalefactors

Scalefactors are used to define the range of spectral data in each scalefactor band in AAC. Delta Pulse Code Modulation (DPCM) is used to represent the relative values of individual scalefactors. Only the global gain is coded with its absolute value. The DPCM coded scalefactors are then coded with Huffman coding. This coding scheme is extremely vulnerable to errors: an error would propagate to all remaining scalefactors in that frame.

AAC error resilience tools specify an alternative coding for scalefactors based on Reversible Variable-Length Codes (RVLC) [2][4]. RVLC enables codewords decoded both forwards from the beginning and backwards from the end. This scheme is effective if there is only one individual bit error or one short burst error in that data section.

Losses in this part of the data are perceptually less severe than loss of the critical data, but more severe than loss of the quantized MDCT spectral data.

2.3. Quantized MDCT (QMDCT) Spectral Data

QMDCT spectral data in AAC are coded using Huffman coding with codebooks defined in the standard. Each Huffman codeword represents two or four adjacent QMDCT coefficients. To increase error robustness of the spectral data, AAC error resilience tools include virtual codebooks and Huffman code reordering [2]. Virtual codebooks allow more efficient detection of bit errors and the codeword reordering prevents error propagation.

In bit consumption perspective, the QMDCT spectral data is the largest data section in an AAC frame. However, losses in QMDCT spectral data are perceptually least critical.

3. INTERPACKET SHUFFLING OF AAC DATA ELEMENTS

The time domain window length of an AAC frame is 2048 PCM samples – a duration of ca. 46 ms if the sampling frequency is 44.1 kHz. Therefore error concealment is not a trivial task if the entire frame is lost. That is why some advanced methods were developed to improve the streaming audio quality [5]. If the more critical data can be preserved, and only a fraction of the less critical data is lost, it will be much easier to perform the error concealment task. That was the rationale behind the scheme proposed in [3], which allowed each AAC frame to be partially reconstructed in the case of packet loss.

Figure 1 illustrates the concept of the data element shuffling into transport packets [3]. Three different packet streams with different priorities are constructed: highest priority packets contain the critical data, intermediate and low priority packets contain the Huffman coded scalefactors and MDCT spectral data respectively. Because the proportion of the critical and scalefactor data to the entire frame size is low, typically less than 20%, network resources can still be utilized efficiently if more reliable transport mechanisms for the critical data packets are employed. Different levels of reliability in data packet delivery can be achieved by utilizing selective RTP retransmission [6][7], for example.

4. PROPOSED EXTENSIONS FOR ERROR RESILIENT STREAMING

The proposed extensions to our earlier system in [3] are explained in this section.

4.1. Huffman code packetization

If the critical data part is lost, there are no means to decode the Huffman coded scalefactors and spectral coefficients. This causes problems when shuffling Huffman coded data elements among different packets, because the boundary between the last Huffman codeword of the lost frame and the first Huffman codeword of the subsequent frame cannot be found in any packet. This problem can be alleviated with proper ordering of the Huffman codewords into packets.

In our first extension, we propose a reservation of a fixed-size slot for data from each frame in every packet. In case some codewords do not fit into the corresponding slot, the overflowing part is stored into a reservoir area in the end of each packet. This approach allows a partial rescue of the affected frames to become decodable, even in the case that some of the previous frames have to be dropped due to loss of the critical section. This extension is intended to put more protection for the first scalefactor sections (e.g. A1, B1 in Figure 1) and the first spectral data sections (e.g. A3, B3 in Figure 1), which correspond to the low frequencies. These data sections are perceptually more significant. Figure 2 depicts the new packetization scheme.

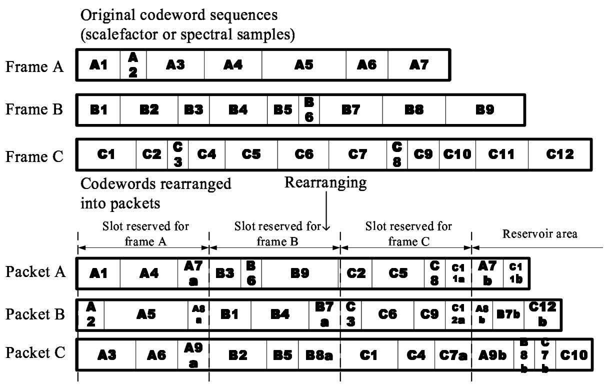

Figure 2. Packetization of entropy coded data elements.

4.2. Scalefactor coding

When the original DPCM coding in AAC is used, it is difficult to recover a missing scalefactor. Even worse, the error propagates to all the rest scalefactors in that frame. If RVLC is used, the decoder can decode from the end backwards, if one single error or a short error burst is detected in that data section. However, RVLC is much less effective when data element shuffling is applied, because missing elements due to packet loss are spread over that data section. RVLC cannot recover the codewords between the first and the last missing codeword. Another weakness of RVLC is the decreased compression efficiency.

To alleviate the problem, we propose a more error resilient method to code the scalefactors in the sender. Our new coding scheme is based on a simple linear model. We use a linear line to approximate the absolute scalefactor values:

$$s_i = d + \alpha \cdot i \,, \tag{1}$$

where s_i is the approximated scalefactor, i is the scalefactor index and d and α are two parameters.

The Minimum Least Squares (MLS) method is used for fitting the actual scalefactor values to the model. Parameters d and α are thus calculated with (2) and (3), respectively. Scalefactors in zero codebook sections are omitted in calculations.

$$d = \frac{\sum_{i=0}^{N} i^2 \sum_{i=0}^{N} s_i - \sum_{i=0}^{N} i \sum_{i=0}^{N} i s_i}{N \sum_{i=0}^{N} i^2 - \left(\sum_{i=0}^{N} i \right)^2} \tag{2}$$

$$\alpha = \frac{N \sum_{i=0}^{N} i s_i - \sum_{i=0}^{N} i \sum_{i=0}^{N} s_i}{N \sum_{i=0}^{N} i^2 - \left(\sum_{i=0}^{N} i \right)^2} \tag{3}$$

Original global gain is then replaced by parameter d and parameter α is quantized and coded with 5 bits and added to the critical data section to be delivered reliably. To achieve better compression efficiency, we have designed a new Huffman table, rather than using the Huffman tables in AAC standard, to code the residual between the actual scalefactor and the linear model. This coding scheme enables missing scalefactors to be replaced by the linear model with relatively small errors. Figure 3 shows an example of using a linear model to approximate the scalefactors.

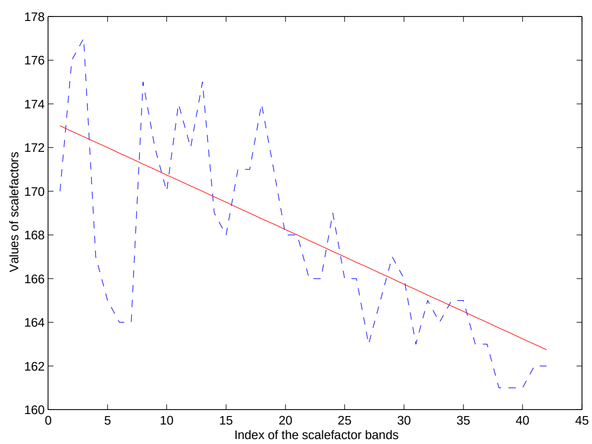

Figure 3. Scalefactors estimated by a linear model.

4.3. Compressed Domain Error concealment for the missing data

Due to the network based prioritized packet loss recovery scheme employed in our streaming system, the probability of losing critical data is very small. Most packet losses can be assumed to concern the packets containing scalefactors or QMDCT spectral data.

In the case of critical data loss, we simply mute the frame and rely on the subsequent time domain error concealment.

In the case of scalefactors data loss, we apply the proposed linear model to approximate the lost ones.

In the case of QMDCT spectral data loss, we apply linear interpolation from the corresponding coefficients in neighboring region to conceal errors. More complex prediction techniques can be applied in some applications where there is less limitation on memory and computation.

5. SIMULATION RESULTS

The proposed extensions were implemented into our earlier audio streaming test system. We have tested our streaming system with different packet loss rates for different priority packets. The main focus of this paper is on the loss of scalefactors and QMDCT spectral data. In the test system a set of 64 audio frames were distributed among 64 data packets so that the critical data was located in 8 high priority data packets, scalefactors in 8 intermediate priority packets and spectral data in 48 low priority packets. Test bitstreams were encoded with an AAC encoder. The sampling frequency is 44.1 kHz, the resulting bitrate is 128 kbit/s with stereo sound track.

The proposed scalefactor coding scheme significantly increased the robustness against packet loss of scalefactors. There is only slight subjective distortion in audio quality with packet loss rates up to 40% when our new scheme is used. In contrast, the original DPCM coding tolerates virtually no errors at all. The price to pay for our new scheme is slightly decreased compression efficiency. On average, the AAC frame size increased by up to 1.9 % for our test bitstreams. Table 1 summarizes a frame size comparison between the two scalefactor coding schemes.

Table 1. Comparison of average frame sizes (in bytes), using the original DPCM and the proposed method.

Test bitstream	Original coding	Proposed coding	Difference in size
Rock	364.7	371.8	1.9 %
Pop	364.6	368.1	1.0 %
Electronic	364.7	367.0	0.6 %
Classical	364.5	365.4	0.3 %

Table 2. Evaluated subjective quality degradation in proportion to the MDCT coefficient loss rate.

Data loss rate	Subjective quality degradation
< 20%	(Almost) imperceptible
~ 30%	Perceptible
~ 40%	Annoying
> 50%	Very annoying

We have tried different methods for concealing the QMDCT data. Our experience is that simple method such as muting or repetition works quite well with our streaming system due to the fact that only a fraction of QMDCT spectral data is lost. In general, there was no significant quality improvement with more complex schemes.

The authors and two colleagues at Speech and Audio Systems Laboratory, Nokia Research Center performed informal subjective evaluations. All four subjects have extensive experience in performing formal subjective listening tests. The results are summarized in table 2, where we have used simple muting for the missing QMDCT data.

6. CONCLUSIONS

In this paper, we have described extensions to our error resilient audio streaming system. We have evaluated the performance of the system in the case of packet losses containing scalefactors and QMDCT spectral data in AAC. It should be noted that our schemes are quite general and do not limit to using AAC only.

The simulation results have shown that the proposed techniques have improved error robustness against packet losses explicitly in comparison to the traditional schemes.

We plan to further optimize the system by integrating an advanced time-domain error concealment scheme into the system. This extension is useful for the case that the critical data are lost.

7. REFERENCES

[1] C. Perkins, O. Hodson, and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," IEEE Network, vol. 12 no. 5, pp. 40-48, 1998.

[2] ISO/IEC standard, "Coding of Audio-Visual Objects – Part 3: Audio", International Standard 14496-3, 2001.

[3] J. Korhonen, "Error Robustness Scheme for Perceptually Coded Audio Based on Interframe Shuffling of Samples", *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing 2002*, Orlando, Florida, pp. 2053-2056, May 2002.

[4] Y. Takishima, M. Wada, and H. Murakami, "Reversible Variable Length Codes", *IEEE Transactions on Communications,* vol. 43 no. 2, pp. 158-162, 1995.

[5] Y. Wang, and S. Streich, "A Drumbeat-Pattern based Error Concealment Method for Music Streaming Applications," *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing 2002*, Orlando, Florida, pp. 2817-2820, May 2002

[6] V. Varsa, and D. Leon, "RTP Retransmission Framework", Internet draft, June 2002.

[7] A. Miyazaki, H. Fukushima, K. Hata, T. Wiebke, R. Hakenberg, C. Burmeister, N. Takatori, S. Okumura, and T. Ohno, "RTP Payload Formats to Enable Multiple Selective Retransmissions", Internet draft, June 2002.