

OBJECT-BASED AUDIO STREAMING OVER ERROR-PRONE CHANNELS

Stuart K. Marks, Ruben Gonzalez

School of Information Technology
Griffith University

s.marks@griffith.edu.au, r.gonzalez@griffith.edu.au

ABSTRACT

This paper investigates the benefits of streaming autonomous audio objects over error-prone channels instead of encoded audio frames. Due to the nature of autonomous audio objects such a scheme is error resilient and has a fine-grain scalable bitrate, but also has the additional benefit of being able to disguise packet loss in the reconstructed signal. This paper proposes object-packing algorithms which will be shown to be able to disguise the presence of long bursts of packet loss, removing the need for complex error-concealment schemes at the decoder.

1. INTRODUCTION

Audio streaming is traditionally done using a frame-based approach. Typically a frame of n samples is taken from the audio source, encoded, sent across the channel, decoded and played back. This allows the latency to be minimized, and enables the data rate and quality to be determined by the design of the audio coder due to Shannon's separation theorem [1]. However, it has become apparent that Shannon's separation theorem does not apply to many modern communication networks, including the Internet and mobile networks [2], which have variable bandwidths and high, bursty packet loss rates [3,4].

For audio to be streamed over these error-prone, variable bandwidth networks it is necessary to employ joint source/channel coding. In this case the source coder must take into account the effects of the channel, and must be prepared to deal with possible corrupted data at the decoder. Joint source/channel coding techniques which have been investigated, include bitrate scalable coders [5,6] which attempt to avoid packet loss by not congesting over the channel, error resilient coders [7] which can do not crash when errors occur, but rather resynchronize and continue to decode when the errors subside, and error concealment schemes [7] which create replacement frames for the frames lost from dropped packets.

This paper examines the benefits of shifting away from frame-based to object-based streaming over these variable bandwidth and error-prone networks for wideband audio. The object-based streaming proposed in this paper packs autonomous audio objects into packets instead of an entire encoded frame. This paper will show that object-based streaming can provide an error-resilient and scalable audio stream, but more importantly removes the need for error-concealment by disguising packet loss, reducing the complexity of the decoder.

A sinusoidal model coder will be used to generate the audio objects. While error concealment and protection schemes have been investigated for sinusoidal model coders, all have done so in a frame-based manner [8,9,10]. This paper will show that using an object-based approach simplifies and improves the performance of the streaming system.

The paper begins by outlining the details of object-based audio streaming. The following section discusses the simulator which is used to measure the performance of both frame-based and object-based audio streaming. The results from these experiments are presented and analyzed in the penultimate section, with the final section outlining the conclusions of the paper.

2. OBJECT-BASED AUDIO STREAMING

Object-based audio streaming utilizes an audio coder which can generate autonomous audio objects. An object represents a portion of an audio signal; the object has an onset time and duration for the portion of the audio signal it represents. Autonomous objects are objects which can be decoded without reference to other objects, providing error robustness and resilience, and the flexibility required for object-based audio streaming.

Model coders, also known as parametric coders, have been recognized for their ability to create such autonomous audio objects [11]. Model coders may include any number of models, including sinusoidal, harmonic, transient and noise models. This paper uses the sinusoidal model to generate autonomous objects from sinusoidal tracks [12].

The benefits of encoding the sinusoidal tracks as autonomous audio objects has previously been overlooked, with the focus being on the pursuit of low bitrate coders [6,13,14]. These coders split the sinusoidal tracks into frames, with the timing information being implied. To provide tracks which are autonomous the entire track should be encoded in one object, with the explicit encoding of the timing information. The sinusoidal coder used in this paper is able to efficiently encode the timing information using Run Length Encoding (RLE).

This method of autonomous object generation increases the latency of the streaming, as the object cannot be encoded till it has been fully analyzed. This limits object-based streaming to non-interactive streaming applications, where the audio is pre-encoded and stored on the server for streaming. This is a common scenario for streaming of wideband audio.

For streaming of a set of pre-encoded objects there is a single task: to get as many objects received at the decoder before their playback time. This flexibility is the key benefit of object-based streaming, and allows algorithms to be developed which optimize the packing of the objects.

There are a number of practical considerations which should be considered. The latency should be reduced, so objects need to be sent in rough chronological order. To reduce memory usage at the decoder objects should be received close to their playback time. To reduce the effects of packet loss, objects should be spread amongst packets, so a dropped packet does not create a lost frame.

The focus of this paper is to demonstrate that packing objects in an appropriate manner can disguise the effects of packet loss. Algorithms which reduce latency and provide fine-grain bitrate scalability are also covered in the following subsections.

2.1 Chronological packing

The chronological packing algorithm simply packs the objects into the packet in chronological order, i.e. the object with the earliest onset time is packed first, and then successive objects are added till the packet is full. This algorithm ensures the latency is kept to a minimum, while being able to disguise low packet loss rates.

2.2 Prioritized packing

The prioritized packing algorithm improves the performance during long packet loss bursts. This is achieved by spreading the objects with high perceptual importance evenly throughout the packet stream, while the less perceptually important objects are packed into the remaining space.

Objects can be prioritized based on a number of parameters, such as object duration, average amplitude, or Signal-to-Mask Ratio (SMR), but it was found priority based on the object's energy performed best during experimentation.

The algorithm begins by determining how many high priority objects should be placed in the current packet to evenly spread across future packets. The number of objects in the high priority queue and the time spanned by these objects determines this value.

The algorithm then packs these objects, interleaving adjacent objects between packets provided there is sufficient space. Remaining space in the packet is used to pack low priority objects in chronological order. This approach works well as low priority objects characteristically require less space than the high priority objects.

2.3 Dynamic Packing

The problem of packing objects into a packet of fixed space is known as the Knapsack problem, and dynamic algorithms exist for finding optimal solutions. The complexity of using such a dynamic algorithm becomes justified when the characteristics of the channel are highly dynamic.

In this case a bandwidth estimation algorithm can determine the size of the packet. The cost of each object is the space required to encode it. While the benefit of each object can be determined by the perceptual importance of the object, with scaling according to onset time, so that an object is not sent too early or late.

Investigations into these algorithms is left as further work as the focus of this paper is to demonstrate the benefits of object-based streaming over error-prone channels, which can be achieved with the previous algorithms.

3. SIMULATION

A simulated streaming environment was used to measure the performance of frame-based and object-based streaming over error-prone channels. The simulation consisted of two slightly different parts, one for frame-based streaming part using MPEG.1 Layer 3 (MP3) [15] and G.723 [16] coders, and an object-based part using a multiresolution sinusoidal coder [17].

The frame-based streaming part encodes and decodes the audio on-the-fly, with audio taken from the source file (mono signal sampled at 44.1kHz with 16bits per sample) in frames of 1024 samples and encoded using MP3 at 32kbps or G.723 with 2 bits per sample. The encoded packet is then passed across the simulated channel, which simulates a single packet loss burst at a random time and for a given number of packets. Received packets are

decoded and placed in the reconstructed signal. Frames from dropped packets are replaced with a silent frame-insertion based error-concealment scheme, which was found to be the most perceptually tolerable insertion-based scheme for long packet loss bursts.

For object-based streaming the audio source is pre-encoded into a set of objects. The selected packing algorithm places the objects into packets, which are sent over the same simulated channel as with the frame-based simulation. The inter-departure times of the packets is set to the same frame rate as the frame-based streaming part, and the size of the packets is set to obtain the average bitrate of the sinusoidal coder. Received objects are decoded and synthesized in 1024 sample frames and placed into the reconstructed signal. Objects contained within drop packets are lost, and therefore are not synthesized.

For each simulation a segmented-SNR (SNR calculated on a frame basis) trace is generated for objective testing and visual inspection of performance, while the reconstructed signal is placed in a sound file for perceptual testing, and the ultimate validation of performance.

4. RESULTS

The simulator was used to generate results for a 20 second harp signal for both frame-based and object-based streaming with packet loss bursts of varying. The segmented-SNR traces are plotted in Figure 1 for the frame-based streaming part and in Figure 2 for the object-based streaming part. Note that the location of the beginning of the burst and duration were fixed with these cases to aid comparison.

Figure 1 demonstrates the error resilient nature of G.723 coder when state information is encoded with each packet, allowing the G.723 decoder to recover immediately after the packet loss burst. The MP3 bitstream is not so error-resilient, and audible artifacts can be heard briefly after the packet loss burst. The decoder is then able to begin decoding again but is no longer synchronized with the source audio. This results in a low and erratic segmented-SNR after the packet loss burst, but was not found to be perceptually detrimental.

The quality of the reconstructed audio from the sinusoidal model is not high as it is unable to capture transient and noise portions of the signal. This also creates the high variation evident in the segmented-SNR traces in Figure 2. When the packet loss burst begins the chronological packing algorithm is able to maintain the signal for a brief period, and then gracefully degrades. This is due to objects with long durations beginning just before the packet loss.

The prioritized packing algorithm improves upon this, as there are no audible dropouts; instead signal

components are lost, reducing quality. This can be seen when examining the segmented-SNR traces at a scale finer than is possible here.

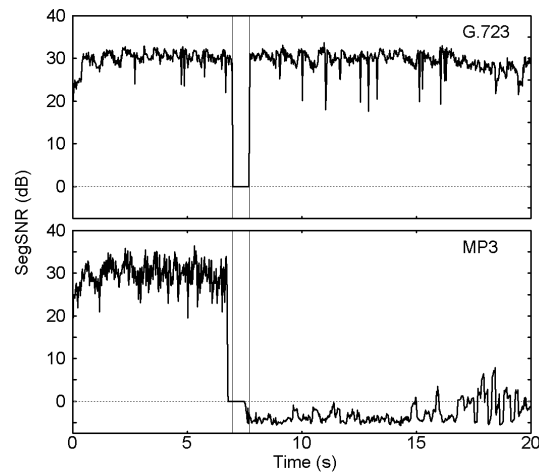


Figure 1. Segmented SNR traces for the frame-based streaming simulations for a packet loss burst of 32 packets (0.743 seconds) beginning at around seven seconds into the signal, indicated by the vertical lines.

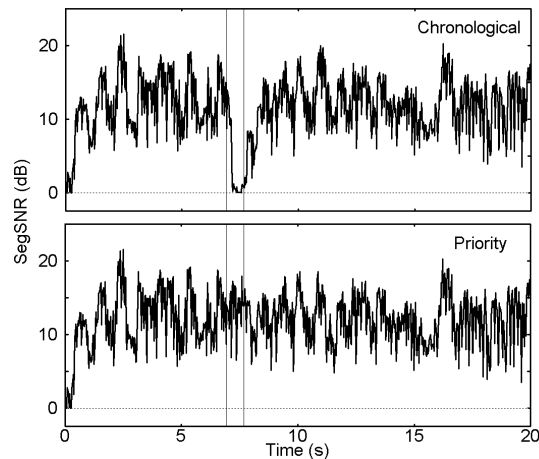


Figure 2. Segmented SNR traces for the object-based streaming simulations for a packet loss burst of 32 packets (0.743 seconds) beginning at around seven seconds into the signal, indicated by the vertical lines.

Perceptual tests were conducted on the reconstructed signals using the MUSHRA test procedure [18], and verified the outcomes from the segmented SNR traces. The results of the perceptual tests are summarized in Figure 3, and show that listeners find audio drop outs, synonymous with frame-based streaming, more annoying than the gracefully reduction of audio quality evident with object-based streaming. The results show that object-based streaming using the prioritized packing algorithm is able to maintain quality during long periods of packet loss.

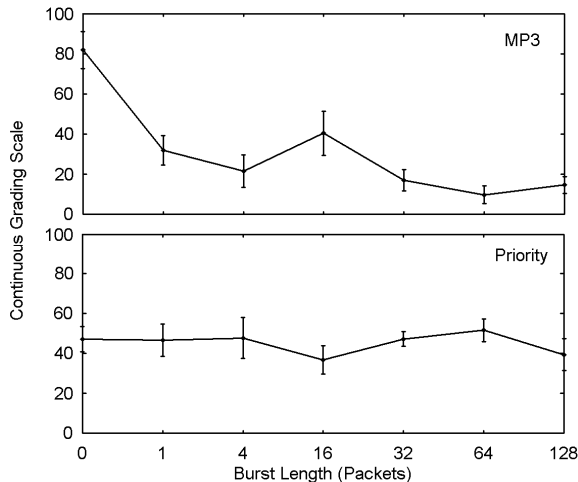


Figure 3. MUSHRA test results. Error bars indicate 95% confidence intervals. The spike for the MP3 results at a burst length of 16 packets is due to the packet loss burst occurring near the end of the audio sample.

5. CONCLUSION

This paper has shown the benefits in streaming audio in an object-based manner instead of the traditional frame-based approach. Object-based streaming provides the flexibility required for fine-grain bit-rate scalability, and allows objects to be packed to disguise packet loss, avoiding the use of complex error concealment schemes.

6. ACKNOWLEDGEMENTS

Australian Research Council's Spirt Scheme, Sun Microsystems for the G.723 source code and the LAME project for the MP3 source code.

7. REFERENCES

[1] Shannon, C., "A mathematical theory of communication", Bell Technical Journal, vol. 27, pp. 379-423,623-656, 1948.

[2] Vembu, S., Verdu, S. and Steinbert, Y., "When does the source-channel separation theorem hold?", IEEE International Symposium on Information Theory, p. 198, Jul. 1994.

[3] Kouvelas, I., "Redundancy Control in Real-Time Internet Audio Conferencing", AVSPN'97, Aberdeen, Scotland, Sep. 1997.

[4] Hagenauer, J. and Stockhammer, T., "Channel coding and transmission aspects for wireless multimedia", Proceedings of the IEEE, vol. 87, is. 10, pp. 1764-1777, Oct. 1999.

[5] Feiten, B., Schwalbe, R. and Feige, F., "Dynamically scalable internet audio transmission", 104th AES Conference, Amsterdam, May 1998.

[6] Verma, T. and Meng, T., "A 6Kbps to 85Kbps scalable audio coder", IEEE ICASSP '00, vol. 2, pp. 877-880, 2000.

[7] Perkins, C., Hodson, O. and Hardman, V., "A survey of packet loss recovery techniques for streaming audio", IEEE Network, vol. 12, is. 5, pp. 40-48, Sep. 1998.

[8] Purnhagen, H., Edler, B. and Meine, N., "Error protection and concealment for HILN MPEG-4 parametric audio coding", 110th AES Convention, May 2001.

[9] Lindblom, J. and Hedelin, P., "Error protection and packet loss concealment based on a signal matched sinusoidal vocoder", IEEE ICASSP'03, vol. 1, pp. 100-103, Apr. 2003.

[10] Rodbro, C., et al., "Compressed domain packet loss concealment of sinusoidally coded speech", IEEE ICASSP'03, vol. 1, pp. 104-107, Apr. 2003.

[11] Verma, T., "A perceptually based audio signal model with application to scalable audio compression", PhD Thesis, Stanford University, Oct. 1999.

[12] McAulay, R. and Quatieri, T., "Speech Analysis-Synthesis Based on a Sinusoidal Representation", IEEE Transactions on Acoustics, Speech, Signal Processing, vol. 34, no. 4, pp. 744-754, Aug. 1986.

[13] Purnhagen, H. and Meine, N., "HILN - The MPEG-4 Parametric Audio Coding Tools", IEEE International Symposium on Circuits and Systems, ISCAS 2000, Geneva, May 2000.

[14] Levine, S. and Smith, J., "A switched parametric and transform audio coder", IEEE ICASSP'99, vol. 2, pp. 985-988, Mar. 1999.

[15] "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbps - IS 11172-3 (audio)", ISO/IEC JTC1/SC29, 1992.

[16] "5, 4, 3 and 2 bits sample embedded Adaptive Differential Pulse Code Modulation (ADPCM)", ITU Recommendation G.727, <http://www.itu.org>.

[17] Levine, S., Verma, T. and Smith, J., "Alias-free, multiresolution sinusoidal modeling for polyphonic, wideband audio", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, USA, 1997.

[18] "Method for the subjective assessment of intermediate quality level of coding systems", ITU Recommendation BS.1534-1, <http://www.itu.org>