Loss Concealment for Multi-Channel Streaming Audio

Rishi Sinha
University of Southern California
941 W. 37th Place, PHE 335
Los Angeles, CA 90089
+01-213-740-1604
rishisin@usc.edu

Christos Papadopoulos
University of Southern California
941 W. 37th Place, SAL 238
Los Angeles, CA 90089
+01-213-740-4780
christos@imsc.usc.edu

Chris Kyriakakis
University of Southern California
3740 McClintock Ave., EEB 432
Los Angeles, CA 90089
+01-213-740-8600
ckyriak@imsc.usc.edu

ABSTRACT

With the advent of high-speed networks such as Internet2, high quality uncompressed transmission of multi-channel audio streams has become possible. For interactive applications, such as a distributed musical performance, minimizing latency is of paramount importance. Given the strict latency requirements, error recovery (via either retransmission or FEC) may not always be successful, and thus concealment is frequently required. In this paper we propose a novel concealment algorithm, based on inter-channel redundancy, for multi-channel, professional quality, uncompressed audio streams, with particular emphasis on an experimental 10.2 audio standard, which provides an immersive experience for the audience and the players in a performance. We also propose a smoothing method based on Bezier curves. We focus on interactive applications; thus we investigate concealment techniques that can be performed in real time. Our algorithms are implemented in a testbed capable of streaming up to 24 uncompressed audio channels with end-to-end latency of less than 6 ms. Our results show that our techniques outperform existing methods. We expect that our protocol will become an important part of a distributed immersive musical performance system currently being developed at our university.

Categories and Subject Descriptors

C.2.2 [**Computer Communication Networks**]: Network Protocols – *applications.*

General Terms

Measurement, Design, Experimentation.

Keywords

Loss concealment, immersive audio, streaming, real-time, multi-channel.

1. INTRODUCTION

The Internet is already being used for high-end applications such as multi-channel audio and HDTV streaming [3][4][5], which require bandwidth reaching into several hundreds of megabits and low latency. While not yet common in the commodity Internet, such experiments abound in Internet2, a high performance network linking together hundreds of academic institutions, research labs and government agencies.

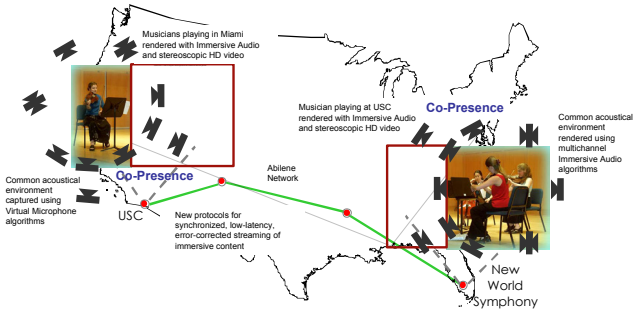


Figure 1: Distributed immersive musical performance.

One such project is taking place at USC and aims to link together via the Internet musicians located in different cities to create a distributed immersive musical performance (Figure 1). The goal of this project is to investigate technical and perceptual factors that influence musicians participating in a distributed performance. Such factors include accurate recreation of a concert hall environment to preserve the audio and visual cues musicians rely on in a performance, and minimizing latency to ensure players remain synchronized. To achieve the highest possible quality with the lowest latency, audio streams are transmitted among musicians as uncompressed PCM data, sampled at 48 kHz and 16 or 24-bit resolution.

This research has been funded (or funded in part) by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

NOSSDAV'03, JUNE 1-3, 2003, Monterey, California, USA.

The problem of minimizing latency is particularly important. With propagation latencies reaching 35ms (one-way) in coast-to-coast transmission, applications and protocols must avoid incurring any additional latency. Interestingly, the coast-to-coast network transmission latency is similar to the latency musicians experience in large orchestras due to sound propagation delays, which can reach 40 ms. This implies that performers may be able to adapt. Informal experiments carried out in our lab with two musicians in separate rooms and digitally variable delay, support this evidence. The musicians were able to play together with latencies up to 100 ms, depending on the musical piece.

Accurate audio reproduction without dropouts is critical to ensure musicians in a distributed performance feel as if they are in the same space. Given the Internet characteristics, however, packet loss or re-ordering is inevitable. While error recovery methods such as forward error correction (FEC) add little latency for random loss, they are susceptible to burst loss. Making FEC robust to burst loss requires additional buffering, which increases latency beyond what can be tolerated for interactive applications such as distributed musical performance. Therefore, some means of low-latency error concealment must be provided.

In this paper we explore loss concealment algorithms suitable for high-quality, multi-channel, interactive applications over the Internet. We assume that the application is not bandwidth limited, and the network is well provisioned to support the needs of the application. We assume the presence of multiple channels, transmitted in uncompressed PCM format, and that our protocol has easy access to the uncompressed data. Minimizing latency is of paramount importance, but not at the expense of audio quality, so our goal is to devise concealment algorithms that can be performed in real time, with minimal or no impact in latency.

Thus, our goal is the following – upon detection of one or more consecutive lost packets, leading to a gap in the waveform, to quickly generate a set of samples that can be inserted into the gap to effectively conceal the loss. In immersive applications, real-time demands combined with a high sampling rate, a large number of channels and a large sample size lead to the constraint that only a few packets can be buffered at the receiver, and the emphasis is thus on fast loss recovery upon detection of a gap. In the past, loss concealment methods have exploited the self-similarity of the transmitted waveform to repeat in the gap a part of the received waveform that is estimated to be similar to the lost part. A multi-channel immersive environment presents more opportunities for exploiting redundancy, this time among channels rather than in the same channel. The major contributions of this paper are a novel loss-concealment method that exploits inter-channel redundancy in addition to intra-channel redundancy, and a fast smoothing procedure based on Bezier curves that eliminates amplitude discontinuities at the loss boundaries with a small, constant number of simple operations.

The rest of this paper is organized as follows. Section 2 discusses the related work; Section 3 describes particular features of the target application and environment and presents an overview of our method; Section 4 presents our technique for using inter-channel redundancy; Section 5 presents our Bezier curve stitching method to smooth amplitude discontinuities at loss boundaries; in Section 6 we describe the results of evaluation of our methods in terms of perceptual quality and performance; we conclude in Section 7.

2. RELATED WORK

To the best of our knowledge, ours is the first work that explores techniques for loss concealment in streaming multi-channel, immersive sound. The area of single-channel loss concealment for packet voice has seen a lot of research. The techniques developed therein are applicable to the multi-channel immersive environment, but do not exploit the relationships between the various channels and the inter-channel redundancy. Both these families of techniques, namely, the multi-channel redundancy-based substitution technique developed in this paper and the single-channel techniques developed earlier, need to be applied to our problem. While the first exploits inter-channel redundancy, the second exploits intra-channel redundancy.

Here we describe the related work in the area of single-channel loss concealment applicable to waveform-coded audio. Though many of the schemes have been evaluated on packet voice applications, the principles can be applied to packetized music too, with considerations that will be discussed in Section 3. Loss-concealment and other techniques for error recovery in streaming media have been surveyed in [2] and [16].

Much of the work in this area is derived from the waveform substitution methods developed in [8] and evaluated in [7]. These methods are based on the substitution of a part of the received waveform into the lost section. Two methods are proposed to choose the fragment to substitute into the loss. The first method, based on pattern matching, searches the received waveform for a match on the shape of the part of the waveform just before the loss. The fragment following the match is chosen as the substitution. (Subsequent work in [11] and [12] has also explored pattern matching both before and after the lost section.) In the second method, based on pitch detection, the waveform is continuously scanned to detect positive and negative peaks, which gives an estimate of the pitch. Substitution is then performed by repeating a pitch period in the lost section. This work also proposes a stitching algorithm for smoothing the amplitude differences at the boundaries of the missing section, based on a weighted addition of overlapping waveforms.

Because of changes in pitch over the lost section, it is possible that the end of the substitution will not be phase-matched with the subsequent received waveform. Solutions to this problem have been proposed in [11] and [13]. The method proposed in [13] is to adjust the time-scale of the substituted waveform in order to compress or stretch it appropriately such that the phase difference at the end of the lost section is eliminated. The solution proposed in [11] is based on repeating in the lost section appropriate numbers of two different pitch periods, one from the section before the loss, and the other from the section after the loss. The phase difference is eliminated in the center of the substitution.

It is argued in [14] that waveform substitution based on repeating a pattern-matched section or a pitch period leads to audible distortions. The authors propose a new method for filling gaps by stretching neighboring sections into the loss by adjusting their time-scales. Phase discontinuity remains a problem with this method. An application of this principle of time-scale

modification to packet voice is described in [1], but this work is specific to properties of speech that are not shared by music.

These methods are receiver-based and do not rely on the sender. Though this is the class of solutions we target, it may be used in conjunction with sender-based sample-interleaving methods. The odd-even interleaving method proposed in [10] was one of the earliest. By sending odd- and even-numbered samples in different packets, this method thins out packet losses, and the missing samples can be interpolated by a simple average or more elaborate methods, such as the pattern-matching method for interpolating isolated missing samples proposed in [9]. The same idea can be used with other periods of interleaving, waveform substitution techniques being used to conceal the shrunken loss. However, a high degree of interleaving comes at the cost of increasing the latency in the stream, and is therefore not desirable in our application. For an interleaving-based interpolation scheme like this, a transformation-based scheme has been proposed [15] where the sender applies a transformation to the data that depends on the interpolation method used and the predicted loss, attempting to improve the quality of reconstruction.

Some of the goals of our work are similar to those of offline restoration of digital audio, techniques for which are presented in [6]. Specifically, offline restoration deals with removal of clicks from audio material, which are analogous to the packet losses we encounter. The performance of these offline methods in a real-time environment has not been evaluated, and it remains to be seen if these methods are applicable under that constraint.

3. BACKGROUND AND OVERVIEW

In this section we describe the characteristics of the target environment in terms of the network conditions we expect to encounter and the characteristics of the data we carry. We describe how these factors motivate our solution, and give an overview of the solution. Finally, we define some terminology.

3.1 Network Properties

Our target network is a well-provisioned network like Internet2, with capacity to reliably support uncompressed audio streams in multiple channels. We conducted a number of measurements at Internet2 sites in order to determine the loss and jitter characteristics in this environment. We find that most packet losses are isolated single-packet losses, with burst losses happening very infrequently. The distribution of inter-arrival times of the packets shows a peak at the expected value, but a few packets experience very high delays.

For the network measurements, we transferred data with the same rate, packet sizes and inter-packet time as a 16-channel stream with 2-byte samples at 48000 samples per second per channel. We used the audio hardware to generate this data at the correct rate. From four different sites, we streamed data to a receiver in Los Angeles, California. These sites are at Miami (Florida), College Park (Maryland), Atlanta (Georgia) and Amherst (Massachusetts). All five sites are Internet2 sites. Messages are generated every 1.333 ms, and each message is divided into two packets, sent out back-to-back. We collect measurements of loss and jitter for the purpose of characterizing the demands placed on any loss concealment technique.

Table 1. Network loss characteristics

		Loss %age	Number of consecutive losses		Single loss %age
			Mean	Max.	
FL	I	0.07%	1.000	1	100.00%
	II	0.09%	1.260	187	99.68%
	III	0.08%	1.086	71	99.75%
MD	I	0.07%	1.042	45	99.83%
	II	0.08%	1.175	200	99.91%
	III	0.07%	1.007	9	99.91%
GA	I	0.07%	1.014	12	99.84%
	II	0.07%	1.028	14	98.81%
	III	0.08%	1.125	148	99.03%
MA	I	0.07%	1.063	21	98.54%
	II	0.07%	1.059	8	98.85%
	III	0.07%	1.083	135	99.48%

Table 2. Network jitter characteristics

		Inter-message time		
		Mean (ms)	Max. (ms)	S.Dev. (ms)
FL	I	1.332	50.574	0.399
	II	1.332	14.636	0.392
	III	1.332	31.435	0.406
MD	I	1.332	5.673	0.322
	II	1.332	134.661	0.341
	III	1.332	12.844	0.331
GA	I	1.332	14.394	0.328
	II	1.333	10.394	0.317
	III	1.333	99.946	0.378
MA	I	1.332	12.464	0.321
	II	1.333	9.540	0.302
	III	1.332	12.552	0.406

Table 1 and Table 2 show numbers obtained from these measurements. The rows show data from three runs for each of the sites in Miami, College Park, Atlanta and Amherst, respectively. In Table 1 the first column lists the percentage of packets that were lost and the second, third and fourth columns list the average, maximum and standard deviation of the number of packets lost consecutively. The last column shows the percentage of losses that involved only one packet. The total number of packets in each run was 900,000. Table 2 lists the mean, maximum and standard deviation of the message inter-arrival times at the receiver.

We can see from Table 1 that the overwhelming majority of losses are single-packet losses. In addition, these single-packet losses can be qualified as truly being isolated rather than occurring in close proximity to each other, because we have observed that these losses are well distributed, separated by hundreds of packets in most cases. However, we do suffer from burst losses running into hundreds of packets. Thus, the distribution of losses is such that these are the two classes of loss that need to be engineered for – solitary packet losses and long outages involving hundreds of packets.

An important observation we make is that losses of packets transmitted back-to-back (i.e. packets belonging to the same message) are not correlated, as might be expected if a drop-tail model for losses were postulated. There are negligibly few losses that involve only the two packets belonging to the same message.

Reordering of packets was not observed to be problem. Not only was the number of reordered packets extremely small (ranging from 3 to 20), almost all reordered packets arrived soon enough to be acceptable within the limits of buffering. In the event that packets are reordered beyond recovery, this is equivalent to a loss and is dealt with as such by the loss concealment techniques.

The delays shown in Table 2 are not affected by packet losses in our measurement, as we have discounted the inter-message delays caused by lost packets from this measurement. The jitter introduced is within the range of tolerability by the application, but the loss concealment algorithm needs to be employed to deal with the occasional large delay introduced by the network. In a real-time application with less buffering than the delay, this is equivalent to a loss.

3.2 Content Properties

Figure 2 shows the placement of channels in our immersive audio environment. Recording is done by microphones placed similarly.

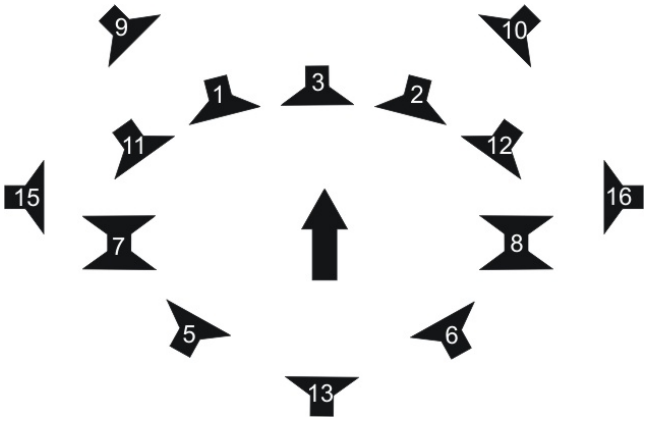


Figure 2: Placement of speakers in immersive environment.

We transmit 14 channels of uncompressed signed linear PCM audio. The sample size is 16 bits. Sampling frequency is 48kHz, so each sample is about 20µs. The bandwidth required by each stream is 750 kbps, so the total required bandwidth is about 10 Mbps. Each packet is 1400 bytes long. We place samples from seven channels in each packet, for a total of 700 samples per packet and 100 samples from each channel. Loss of a packet creates a 2ms gap in each of seven channels.

Compare the above packetization scheme with packetizing 50 samples from each of the 14 channels in one packet. Loss of a packet would cause a 1 ms gap in all channels. There are two reasons why putting all channels in one packet is not desirable. Firstly, the method for loss concealment that we describe in the next section relies on the presence of data from certain channels to substitute for contemporaneous data lost in other channels. Secondly, even if this method were not used, a gap in half the channels is easier to conceal than a gap in all channels, notwithstanding the above difference in gap sizes. Since our recording latency exceeds 2 ms, the chosen packetization does not increase the latency.

The current audio type being investigated is live orchestral music. Note that we do not use synthesized audio in this application; live sound is captured using microphones, and thus each channel has a rich and complex harmonic structure. Nevertheless, there is short-term self-similarity in each channel that can be exploited. Also, there is a high degree of correlation between the content of neighboring channels.

3.3 Overview

In the environment described above, most packet losses are singular, and result in loss in half the channels only. The advantage of a low-latency, high-sampling-rate, multi-channel stream is that each packet represents a very short duration of audio. Nevertheless, each of these instances of loss, unless concealed in some way, produces a loud and annoying click, which gets amplified to unacceptable levels in large audio equipment. Ignoring the gap and splicing the ends of the gap together is not an acceptable solution as it introduces a timing mismatch between the producer and consumer sides in this real-time application. Thus, for isolated packet losses and certainly for multiple consecutive packet losses, the gap needs to be filled in with samples approximating the original waveform.

There are three problems that need to be overcome when a new waveform fragment is substituted into the gap: the amplitude differences at the boundaries of the gap, between the new and original waveforms may be large enough to cause a loud click; the waveform that is substituted must resemble the original; and the substituted waveform should join either end of the original waveform at the correct phase.

When there is loss in all channels, concealment in each channel must be performed by one of the classical methods described in Section 2, using information from within the same channel to conceal the loss in any given channel. We have seen in the network characteristics presented earlier that though relatively much less frequent than smaller losses, this kind of loss scenario can be expected to arise.

However, for any single packet loss, and for certain two consecutive packet losses, only half the channels are affected, and here we propose the novel inter-channel method described in Sections 4, along with the new fast smoothing method we introduce in Section 5. The older methods are applicable to this loss regime too, but we believe that in this loss regime ours is a much better method. This is because our experiments have

shown that finding a suitable replacement for the loss and phase mismatch at the end of the replacement are significantly hard problems for orchestral music, with its complex structure.

Note that the loss regime under which the inter-channel method is applicable is not restricted to single-packet losses. Depending on the packetization, interleaving and latency, the method is applicable to longer losses, as will be explained.

3.4 Terminology

Here we define some terminology that will be used in the rest of the paper. C*n* denotes channel number *n*. Let *L* denote a section of *n* contiguous samples lost from a channel. Let S_j denote the (*j-1)*-th sample in *L*, $j = 0, \ldots, n-1$. For $j < 0$ and $j > n-1$, let S_j denote the corresponding sample before or after *L*. Let s_j denote the numerical value of sample S_j. Let v_{max} and v_{min} denote the maximum and minimum values any sample can take. A (j, s_j) pair is viewed as a point on the plane.

4. INTER-CHANNEL METHOD

We propose a technique that interpolates among different channels in order to derive a waveform substitution for a lost section, instead of using samples from the same channel, as existing methods do. The method is based on the observation that the signal values in channels placed physically close together in Figure 2 are closely similar, and at during a short interval, the signal in one channel may be approximated by combining signals from other channels. Thus, the waveform substitution for a missing interval in a channel is derived from the sample values of other channels during the same interval. This section describes the rules for this derivation, requirements for mutual exclusion of dependent channels and the kinds of losses that can be concealed with this method.

Table 3. Substitution rules

Missing channel	First choice substitution	Second choice substitution	Third choice substitution
C1	(C11+C3)/2	C11	(C9+C3)/2
C2	(C12+C3)/2	C12	(C10+C3)/2
C3	(C1+C2)/2	(C1+C12)/2	(C2+C11)/2
C5	C7	C13	C6
C6	C8	C13	C5
C7	C5	C13	C8
C8	C6	C13	C7
C9	(C11+C1)/2	(C7+C1)/2	(C13+C11)/2
C10	(C12+C2)/2	(C8+C2)/2	(C13+C12)/2
C11	(C9+C1)/2	(C7+C1)/2	C9
C12	(C10+C2)/2	(C8+C2)/2	C8
C13	(C5+C6)/2	(C7+C8)/2	(C5+C8)/2
C15	C16	-	-
C16	C15	-	-

4.1 Substitution Rules

We use spatial (acoustical) proximity as the main criterion for selecting candidate channels for interpolation. After considering the spatial relationships among all 14 channels in our experimental 10.2 channel immersive audio system, we derived the substitution rules shown in Table 3. The table shows up to three choices for substituting missing samples in a channel from other, proximal channels (channels 4 and 14 are not used). Each choice is either a direct substitution or the result of interpolating two other channels. In the latter case the new sample is derived as the average of the donor channels.

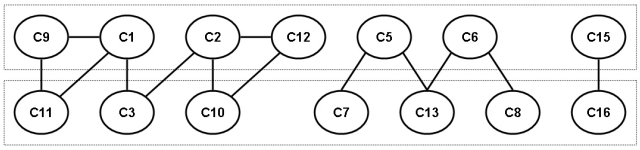


Figure 3: Channel dependency graph contains cycles.

4.2 Channel Groups

As mentioned earlier, for channel interpolation to work, it is essential that data from the donor channels be available when data from the recipient channel is lost. Thus, the distribution of channels must be done in such a way that mutually dependent channels are not grouped together in the same packet. For example, if we are using channels 11 and 3 to substitute in place of channel 1 (according to the first choice rule), neither channel can occur in the same packet as channel 1. However, dividing the set of 14 channels into mutually independent groups is not trivial, since there are dependency cycles among the channels, which we describe next.

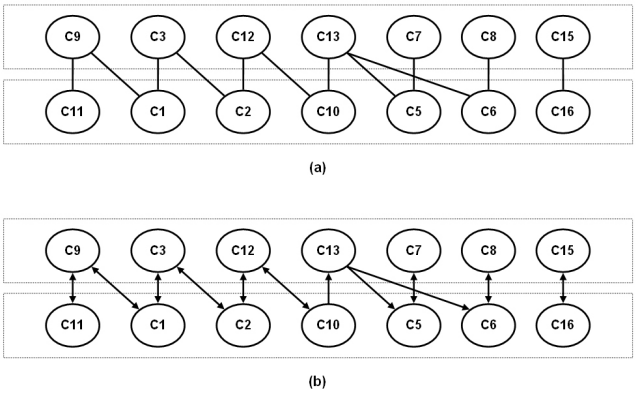


Figure 4: Dividing channels into independent groups.

We define a dependency relation on the channels as follows: for a chosen set of substitution rules, one for each channel, if channel *X* needs channel *Y* for substitution or channel *Y* needs channel *X* for substitution, then channels *X* and *Y* have a dependency relationship. The problem is to divide the channels into groups such that there are no dependency relationships between the members of a group. For example, taking the first choice for each

of the channels, we can build a graph where each undirected edge represents a dependency relationship. This is shown in Figure 3.

This graph is cyclic. We cannot divide the 14 channels into two mutually independent sets of seven channels each, because the cycle(s) will always ensure that there is at least one dependency within a group. However, since we have up to three choices for each substitution, this is not the only graph available to us. In fact, since we have three choices each for 12 of the channels, the number of graphs is 3^{12}, and we find that some of these are indeed acyclic. An example of a dependency graph that divides channels into independent groups is shown in Figure 4(a).

We have described one example of how channels can be divided into groups, but there are of course many other possibilities. Two isomorphic ways of enabling the stream to tolerate a greater number of consecutive packet losses are: having more channel groups and increasing the latency in order to generate multiple packets per channel group. They are isomorphic because an increase in the number of channel groups should be accompanied by an increase in the latency rather than a decrease in the packet size, since doing otherwise would be an injudicious use of network resources to achieve a lower packet loss rate. Also, when we increase the latency in order to generate multiple packets per channel group, it would be wise to perform further levels of channel grouping within each channel group. Thus, the two ways are convergent.

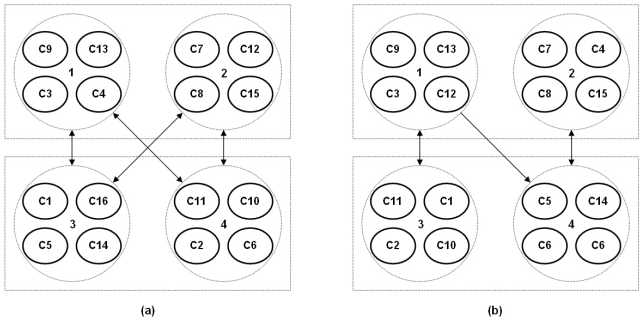


Figure 5: Channel groups of size four.

In this subsection, we briefly describe the considerations in splitting the channel set into more than two channel groups, and in the next subsection we will see how the results of this can be applied to the convergent technique of increasing latency. Since we are considering a highly latency-sensitive application, increasing the latency can be done only within tight bounds, and this discussion serves to generalize the channel grouping approach rather than suggest a path for indefinite extension of its loss tolerance.

For the purpose of the following discussion, we assume for symmetry that all 16 channels are in use, channels 4 and 14 having no dependencies. Once a channel grouping like the one in Figure 4(a) has been obtained, it is easy to divide it into a larger number of smaller groups simply by dividing each channel group into smaller ones. Since the larger channel groups are internally dependency-free, the smaller ones will be too. Figure 5 shows two such divisions, with individual dependencies collected into inter-group dependencies, and directed dependencies being shown this time. Each of the groups 1-4 is intended to be sent in a separate packet. An arrow from group *M* to group *N* indicates some channel in *M* needs some channel in *N* for substitution. The group dependencies are based on the channel dependencies shown in Figure 4(b), where directed edges have the obvious meaning. This figure illustrates the principle to be followed in choosing smaller channel groups – in Figure 5(a), only two sets of double-packet loss can be recovered from, while in Figure 5(b) one more double-packet loss (groups 2 and 3) can be fully recovered from and yet another double-packet loss (groups 1 and 4) can be partially recovered from, because of the fewer edges. This is the best we can do with this set of substitution rules.

4.3 Generalized Channel Group Method

To the extent that the application can accommodate extra latency, which may not be very much in all cases, one channel group may be allowed to occupy multiple packets. An example of this is shown in Figure 6.

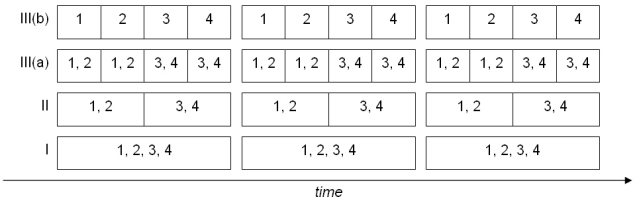


Figure 6: Packetization options.

Each row of this figure shows continuously generated data along a timeline clubbed into blocks of a particular size. The channel groups from Figure 5(b) are used here, and numbers inside blocks refer to the contents of the blocks in terms of channel group number. We may operate at any row, the block size of that row being equal to the full network packet size, which is a constant (1400-1500 bytes is generally considered a "full" or close to "full" packet). The block size of row I represents the latency required for this interleaving.

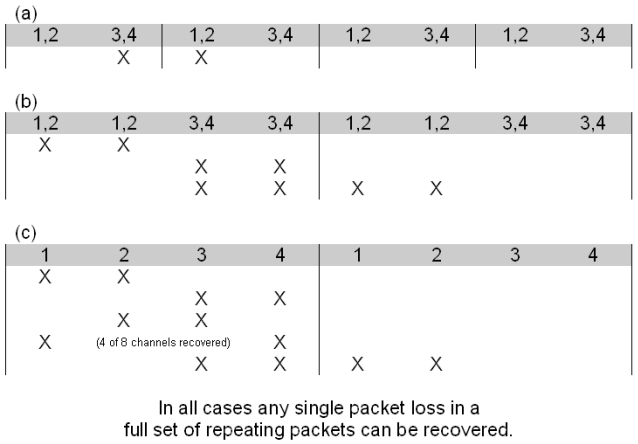


Figure 7: Tolerable losses for various groups and packets.

The losses that can be tolerated at row II are shown in Figure 7(a). Each column represents a packet, and an 'X' marks a packet loss and each row gives a pattern of packet losses that can be concealed. With twice the amount of latency (or half the packet size), we can operate in row III(a) (of Figure 6), for which the tolerable packet losses are shown in Figure 7(b). However, we can do better with iterative channel grouping. Under the dependencies of Figure 5(b), and with the same latency as row III(a), we can operate in row III(b) with the losses shown in Figure 7(c). In our experiments we operate in row II.

5. BEZIER CURVE STITCHING

It is not enough to replace a substituted waveform into a missing section, even if the substitution represents a very good approximation for the original data. It is very likely that there will be discontinuities at the boundaries of the lost section because of amplitude differences between adjacent samples of the substituted and original waveforms. We have observed that even modest amplitude differences produce very loud clicks. We present here a new method for smoothing these discontinuities using the family of curves known as Bezier curves. The Bezier curve stitching method for smoothing amplitude discontinuities is applicable to all substitution-based loss concealment methods and in all loss regimes. It is not specific to the inter-channel loss-concealment method.

5.1 The Problem of Smoothing

Our aim is to develop a lightweight and reliable smoothing method that is suited for a real-time stream with very low latency. Instead of methods based on overlapping boundaries [7][8] that do not exist naturally, or relying on adjusting the amplitude of the entire substitution [11][13], we seek a method that performs a few simple operations and specifically targets the amplitude discontinuity at the boundary. Straight-line interpolation for consecutive missing samples has been proposed in the past [9], and we evaluated this method as a means of reducing the slope of the imaginary line joining adjacent samples. Straight-line stitching is shown in Figure 8. We pick a number m of samples in L such that the straight line extends over samples S_{-1} through S_{m-1} on the left and over samples S_{n-m} through S_n on the right. However, this method does not solve the problem as it does not extend the waveform smoothly and does not preserve the characteristics of the signal. The problem is solved using Bezier curves, which can naturally be guided smoothly between two points.

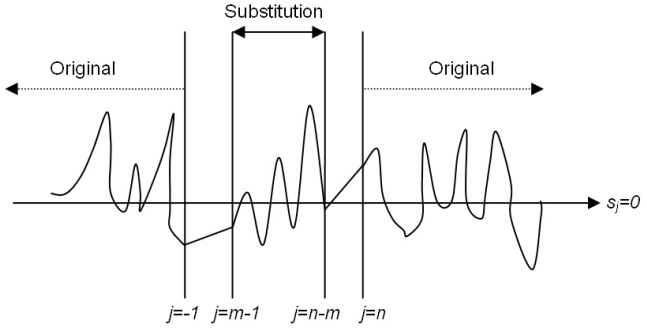


Figure 8: Straight-line stitching.

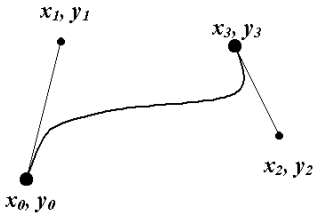


Figure 9: A Bezier curve and its control points

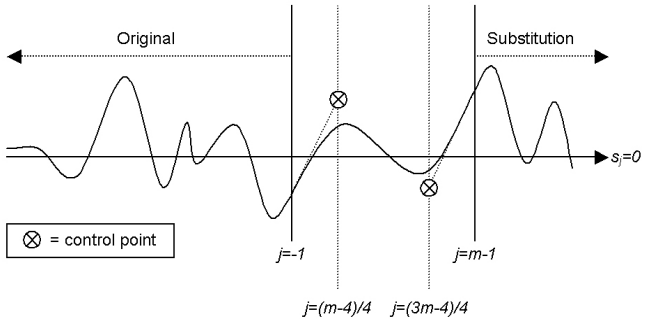


Figure 10: Bezier curve stitching (left side)

5.2 Basic Method

Originally developed in the 1970s for CAD/CAM applications, Bezier curves are used extensively in graphics applications and are also used in the PostScript drawing language. A Bezier curve is computed as follows: given end points (x_0, y_0) and (x_3, y_3), and control points (x_1, y_1) and (x_2, y_2), the points on the curve are defined by the following equations (where $0 <= t <= 1$):

$$x(t) = a_x t^3 + b_x t^2 + c_x t + x_0$$
$$y(t) = a_y t^3 + b_y t^2 + c_y t + y_0$$
$$c_x = 3(x_1 - x_0)$$
$$b_x = 3(x_2 - x_1) - c_x$$
$$a_x = x_3 - x_0 - c_x - b_x$$
$$c_y = 3(y_1 - y_0)$$
$$b_y = 3(y_2 - y_1) - c_y$$
$$a_y = y_3 - y_0 - c_y - b_y$$

A Bezier curve ensures that the constructed curve is tangential to the line connecting the end point with its control point. Figure 9 shows a Bezier curve and the control points used to construct it.

By choosing proper control points we can ensure a smooth transition between the two ends of the waveform to be stitched. As shown in Figure 10, we place control points such that the Bezier curve has the same slope as the waveform at either end of the Bezier curve – control points are chosen along the tangents on either side. The Bezier curve spans m samples at either end of L. At the beginning of L, the first control point is at the intersection of the line $j = (m-4) / 4$ and the line joining $(-2, s_{-2})$ to $(-1, s_{-1})$ and the second control point is at the intersection of the line $j = (3m-4) / 4$ and the line joining $(m-1, s_{m-1})$ to (m, s_m).

Similarly, at the end of L, the first control point is at the intersection of the lines $j = (4n-3m) / 4$ and the line joining $(n-m-1, s_{n-m-1})$ to $(n-m, s_{n-m})$ and the second control point is at the intersection of the line $j = (4n-m) / 4$ and the line joining (n, s_n) to $(n+1, s_{n+1})$. If the s-coordinate of any control point is outside the range $[v_{min}, v_{max}]$, it is reassigned the closer boundary value.

Now we describe the computation of the curve itself. First the straightforward way is stated, and in the next subsection, we describe the fast approximation. The straightforward way to plot this curve on the continuous plane would be to insert a number of different values of t into the functions $x(t)$ and $y(t)$ and plot the points thus obtained. On a plane with discrete x- and/or y-axes, the method remains the same, but with the addition of a rounding step. In our case, however, the curve needs to be plotted on a discrete plane at predetermined values of x, because x is the sample index. In this case, we need to solve for t given $x(t)$, and plug the correct value of t (between 0 and 1) into the expression for $y(t)$. This involves solving a cubic equation once for every Bezier point desired, which is $2m$ times for every substitution. This involves many time-consuming floating-point operations and trigonometric functions.

5.3 Fast Approximation

As a fast approximation to the above time-consuming process, we use the following method. We replace the cubic function $x(t)$ with a linear function. The sample indexes at which we desire the Bezier function computed are $x_0, x_0+1, \ldots, x_0+m-1$.

Thus, the new function $x(t)$ is

$$x(t) = x_0 + \lfloor t(m-1) \rfloor$$

which can be solved for t to get

$$t = j/(m-1)$$

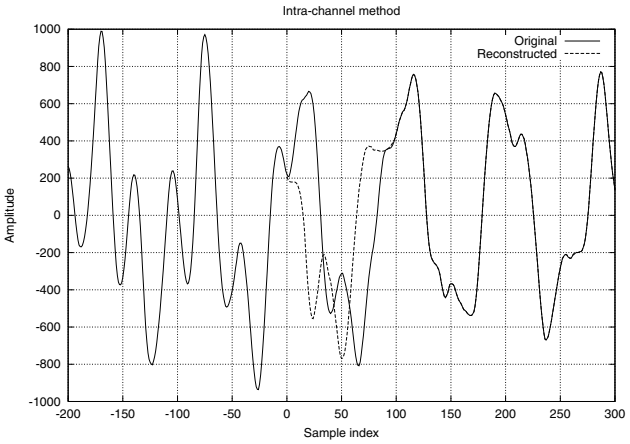
where $j=x(t)-x_0$.

Thus, for values of j from 0 to $m-1$, we obtain t from the above expression, and substitute them into $y(t)$ to obtain the approximate Bezier. Instead of dynamically computing powers of t in $y(t)$, values of t, t^2 and t^3 are pre-computed and stored in a table. This leads to the fact that the curve can be computed with a much smaller number of floating-point and integer operations.

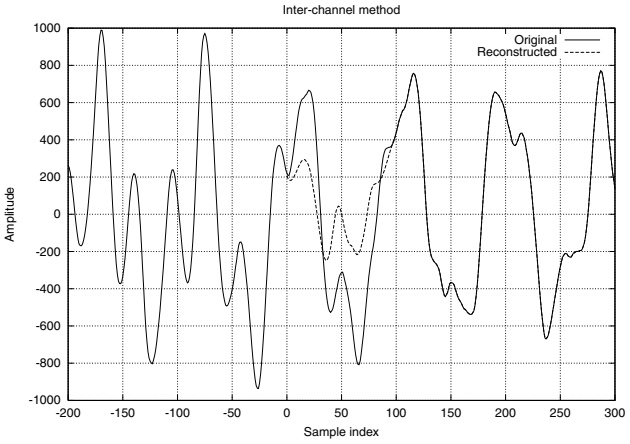
The effect of this approximation is to smoothly "redistribute" the true Bezier over the stitching region. We have found this to be a good approximation.

6. EVALUATION

We evaluated our loss-concealment method and Bezier stitching method by introducing losses into a 14-channel recording of a classical piece and recovering those losses using two methods – the inter-channel method proposed by us, and an intra-channel method based on pitch detection and pitch-period replication proposed earlier [7][8]. The losses introduced were of one-packet duration, which is 100 samples (about 2 ms) in this case. With the inter-channel method, we used the channel groups of Figure 4 (a). With both methods, we employed Bezier stitching with $m=16$ for smoothing at the loss boundaries.



(a)



(b)

Figure 11: Predictability in the waveform.

6.1 Inter-Channel Substitution

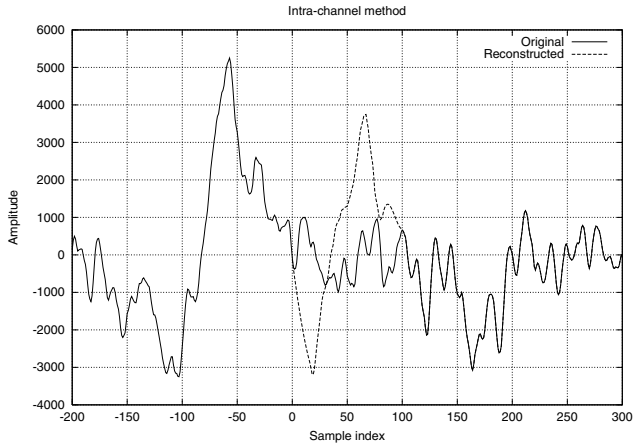
Figure 11 illustrates a simple case of loss concealment by the two methods. This shows the loss and recovery on a part of channel 3 of the recording that is recovered using channels 1 and 2 in the inter-channel method. The horizontal axis is labeled with sample indexes relative to the beginning of the loss. Loss and recovery are between sample indexes 0 and 100.

Due to the stationary nature of the waveform, an infrequent condition in our observation, the intra-channel method provides a good approximation for the lost section (Figure 11(a)). However, a typical classical performance like this recording is dominated by variability in the waveform illustrated in Figure 12. The inter-channel method frequently produces distortions in the attempted recovery of the loss due to this variability.

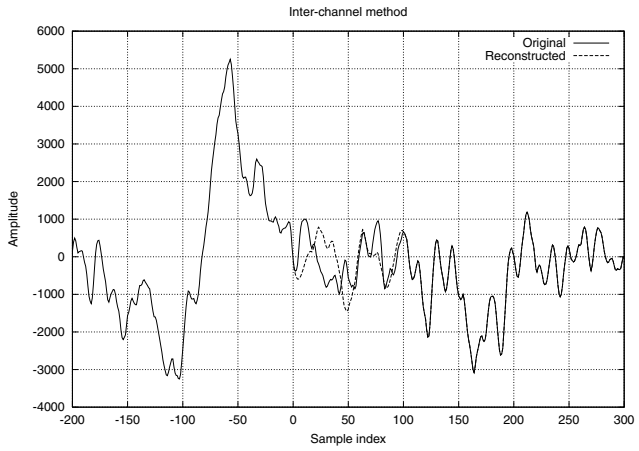
Using inter-channel substitution provided a substantial improvement in perceptual quality due to a reduction in the number of clicks and other distortions compared to the intra-channel method. The pitch period in the waveform does not remain stable in most sections, except those few sections that contain a relatively pure note. This leads to the effect that the

pitch-detection method produces a substitution that is frequently out of phase with the beginning and/or end of the loss section.

Figure 12 (a) shows the difficulty of pitch detection and phase matching when the shape of the waveform is variable enough that it is not possible to predict the pitch and phase with great accuracy. The intra-channel method is unable to model the loss. However, since the inter-channel method has a good estimate available from the repairing channels for the period of loss, the task of estimating the characteristics of the waveform is solved quite simply and reliably (Figure 12 (b)).



(a)



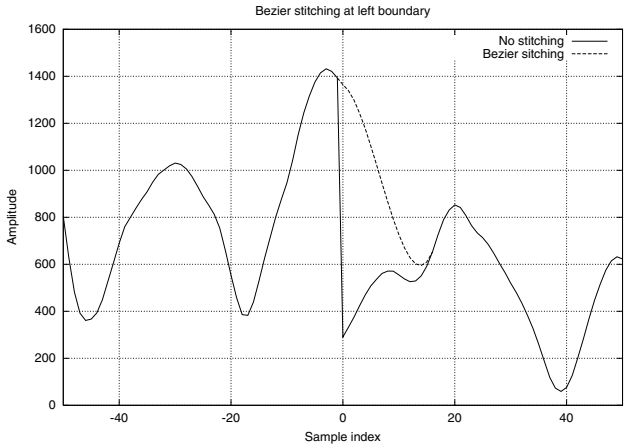
(b)

Figure 12: Effect of sudden changes in the waveform.

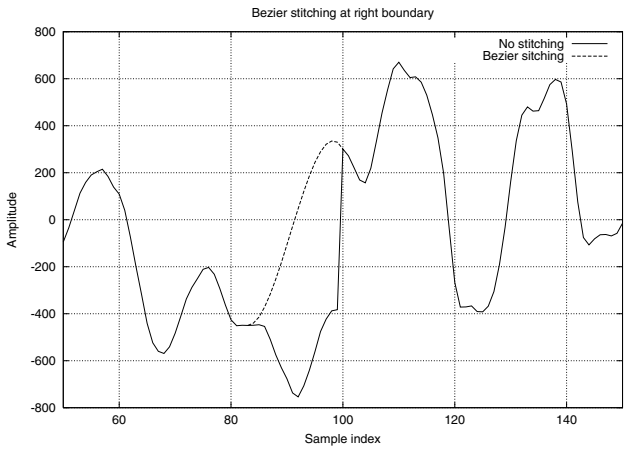
The inter-channel method is computationally much simpler than methods that rely on searching the waveform to match a pattern or scanning it in order to estimate its peaks or zero-crossing rate in order to obtain an estimate of the pitch. In addition, the inter-channel method is inherently more reliable than other methods in replicating the original waveform. Thus we believe it is an extremely simple and effective method of loss concealment in multi-channel streaming when there is data present from neighboring channels.

6.2 Bezier Curve Stitching

With the fast approximation of Bezier stitching, this is a quick and efficient way to stitch the adjacent portions at a loss boundary. Figure 13 shows the way Bezier curves solve the problem of discontinuities while preserving the shape of the waveform. The figures illustrate loss boundaries for two different loss sections in channel 3. The loss boundaries are at 0 and 100 in (a) and (b) respectively. The Bezier curve is computed over a small number of samples (16) in each case. In each case, the solid curve in the range [0, 100] indicates the substitution obtained using the inter-channel method. The dotted curve indicates the Bezier stitching applied to the ends of the substitution. We found in our experiments that Bezier stitching almost completely eliminates the problem of discontinuities.



(a)



(b)

Figure 13: Stitching at loss boundaries with Bezier curves.

We also find that the approximation of the Bezier curve computation described in Section 5 is a good one and does not affect the quality of the stitching obtained. It also decreases the computation time of the curve from the order of microseconds to hundreds of nanoseconds per point.

7. CONCLUSIONS

We have presented a loss concealment algorithm for streaming immersive multi-channel audio. This consists of (a) a method for exploiting inter-channel redundancy in order to conceal losses in a channel with substitutions from other channels by appropriately packetizing the stream and (b) a Bezier-curve method for quickly smoothing the discontinuities at the boundaries of the substituted and original sections. The method of substitution is suitable for single packet losses, which are dominant in the networks we target. Even single-packet losses can be very disruptive in a concert environment, so this represents a substantial gain. The Bezier stitching algorithm is generally applicable to the problem of boundary smoothing and can be used with other substitution-based loss concealment methods and in case of burst losses too. Our method compares with existing methods that are based on redundancy within a single channel. We demonstrate that our method exploits inter-channel redundancy to provide better substitutions than intra-channel methods. We show that given the variability of complex waveforms, current intra-channel methods are unable to provide substitutions that correctly approximate the loss, and the inter-channel method performs much better. Though an intra-channel method must be relied on for burst losses that cannot be corrected by the intra-channel method, we believe that existing methods are not sufficient for the highly demanding environment of real-time immersive streaming, and we are currently working on techniques that can effectively conceal burst losses. Such a method will be based on a model for predicting the properties of the required substitution.

8. ACKNOWLEDGEMENTS

We would like to thank Adnan Mahmud for help in preparing this paper.

9. REFERENCES

[1] F. Liu, J.W. Kim and C.-C. J. Kuo. Adaptive Delay Concealment For Internet Voice Applications with Packet-Based Time-Scale Modification. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. 1461--1464, May 2001.

[2] C. Perkins, O. Hodson and V. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Media. IEEE Network Magazine, pp. 40--48, September/October 1998.

[3] USC Internet HDTV Almost Ready For Prime Time. http://www.digitaltelevision.com/2002/april/news0425_1.shtml.

[4] IMSC Presents Landmark Internet Concert Event. http://imsc.usc.edu/news/symphony.html.

[5] IHDTV Workshop. http://www.washington.edu/ihdtv/

[6] S.J. Godsill and J.W. Rayner. Digital Audio Restoration. Springer-Verlag London Ltd., 1998.

[7] 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, pp. 342--348, November 1998.

[8] 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, pp. 1440--1448, December 1986.

[9] M. Yuito and N. Matsuo. A New Sample-Interpolation Method for Recovering Missing Speech Samples in Packet Voice Communications. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 381--384, May 1989.

[10] N.S. Jayant and S.W. Christensen. Effects of Packet Losses in Waveform Coded Speech and Improvements due to Odd-Even Sample-Interpolation Procedure. IEEE Transactions on Communications, vol. 29, no. 2, pp. 101--109, February 1981.

[11] W.-T. Liao, J.-C. Chen and M.-S. Chen. Adaptive Recovery Techniques for Real-Time Audio Streams, Proceedings of IEEE INFOCOM, vol. 2, pp. 815--823, April 2001.

[12] J. Tang. Evaluation of Double Sided Periodic Substitution (DSPS) Method for Recovering Missing Speech in Packet Voice Communications. Proceedings of Tenth Annual International Phoenix Conference on Computers and Communications, pp. 454 --458, March 1991.

[13] R.A. Valenzuela and C.N. Animalu. A New Voice-Packet Reconstruction Technique. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol 2, pp. 1334--1336, May 1989.

[14] H. Sanneck, A. Stenger, K. Ben Younes and B. Girod. A New Technique for Audio Packet Loss Concealment. Global Internet 96 (Global Telecommunications Conference 1996), pp. 48--52, November 1996.

[15] B. Wah and D. Lin. Real-Time Voice Transmissions over the Internet. IEEE Transactions on Multimedia, vol. 1 no. 4, pp. 342--351, December 1999.

[16] B. W. Wah, X. Su, and D. Lin. A Survey of Error-Concealment Schemes for Real-Time Audio and Video Transmissions over the Internet. Proceedings of International Symposium on Multimedia Software Engineering, pp. 17--24, December 2000.