Transformation-based Reconstruction for Audio Transmissions over the Internet*

Benjamin W. Wah and Dong Lin
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
E-mail: {wah,dlin}@manip.crhc.uiuc.edu

Abstract

This paper studies the design of data transformation algorithms for audio data transmitted over the Internet, with a goal of reconstructing the original signals by the receiver with little distortions in the presence of bursty loss of packets. It assumes that a single audio stream is interleaved into multiple packets, and a lost sample at the receiver is reconstructed as the interpolation of adjacent samples received. We propose a non-redundant transformation-based reconstruction algorithm that can minimize the reconstruction error for any fixed, interpolation-based reconstruction algorithm. Its basic idea is that the sender transforms the input audio stream optimally, based on the reconstruction method used at the receiver before sending the data packets. Consequently, the receiver is able to recover much better from losses of packets than without any knowledge of what the signals should be. In particular, we study our transformation algorithm based on one popular linear interpolation-based reconstruction algorithm. We found that our scheme can improve the signal-to-noise ratio (SNR) by 1 to 2 dB with very little extra computation efforts as compared to the scheme without transformation.

1. Introduction

Real-time multimedia data communications, such as real-time Internet phone, have different characteristics from traditional data communications. The first significant characteristic is their high delay sensitivity. Given strict end-to-end and inter-frame delay requirements, packets delayed over a certain time limit are considered lost [9] and need not be retransmitted by the sender [2]. The second significant characteristic is that most multimedia applications do not require data to be 100% precise [1]. Hence, high packet loss (up to 50%) [4] does not always imply significant performance degradation. These two characteristics must be considered in developing algorithms to cope with loss.

In coping with loss, either the receiver can reconstruct the missing data, or the sender can send redundant data. The first approach is feasible because there is inherent redundancy in multimedia data, and the receiver can reconstruct missing data based on the packets received. In the following, we describe these approaches in details.

Many existing reconstruction algorithms are heuristic and can handle packet loss but provide no guarantee on the quality of reconstructed data. The first class of non-redundant reconstruction methods for concealing loss involves the receiver only. A common strategy is to replay the last packet received during the interval when the lost packet is supposed to be played back. Other strategies are to replace the lost packets using a segment of silence or a segment of white noise. These schemes only work well when both the packet size and the length of a bursty loss are small [5].

The second class of reconstruction methods for concealing loss involves both the sender and the receiver. Based on different ways of processing input data, these schemes can further be split into two subclasses: one by adding redundant control information and one does not. There are several methods for the sender to add redundancy to cope with loss, including the sending of duplicate packets [7], forward error correction (FEC) [6, 3], and sequence loss protection [5]. All these methods cannot guarantee reconstruction quality, require extra bandwidth, and increase end-to-end delay. There are also quite a few algorithms that do not add redundancy but utilize the inherent redundancy of the

*Research supported by National Science Foundation Grant MIP 96-32316 and by a gift from Rockwell International.
Proc. 17th IEEE Symposium on Reliable Distributed Systems, 1998.

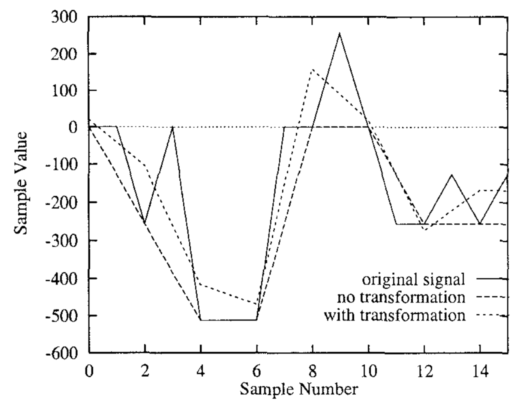

Figure 1. Comparison of the quality of reconstruction between no transformation and transformation, assuming that the odd samples were reconstructed by taking the average of its two adjacent even samples.

source input stream. A typical method based on interleaving transmits interleaved audio samples in one packet and reconstructs lost samples using the average of their surviving neighbors contained in other packets [8]. This method is easy, fast, and does not require extra bandwidth. However, it may fail when signals are rapidly changing.

In short, a combined sender and receiver reconstruction technique generally improves significantly over a receiver-only reconstruction technique. In this paper, we propose a new sender- and receiver-based reconstruction algorithm that transforms input data so that distortions are minimized when some interleaved packets are lost and the missing samples are reconstructed using a given interpolation-based reconstruction method. When all the interleaved packets are received, the receiver must be able to reconstruct the original data stream. This is not as straightforward as it seems because only transformed samples are received.

To illustrate the algorithm, consider a simple two-way interleaved data stream based on a typical segment of voice data with 16 samples. Assuming that the odd samples were lost at the receiver and that the eight even samples were used to reconstruct the missing samples, Figure 1 plots the reconstructed stream in which a) a missing odd sample is computed as the average of its two adjacent even samples, and b) the even samples were first transformed at the sender, and the receiver used the transformed even samples to reconstruct the missing odd samples. It is obvious that the reconstructed stream based on transformed samples can give a better approximation to the original stream, based on

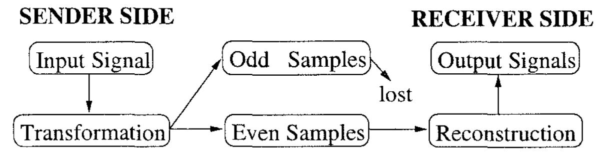

Figure 2. The process of transformation and reconstruction in two-way interleaving

the measure of SNR (Signal-to-Noise Ratio), where:

$$SNR = 10\, log_{10} \frac{\sum(s^2)}{\sum(s - \hat{s})^2} \qquad (1)$$

where s is a sample in the original stream and $\hat{s}$ is the reconstructed sample of s. As shown in the 4^{th} and the 12^{th} samples in Figure 1, the transformation smoothes out rapidly changing parts of the original waveform but tries to preserve its trend so that the reconstructed samples based on averaging is more accurate.

This paper has five sections. Section 2 gives a detailed discussion of our optimization algorithm for two-way interleaving. Section 3 extends the algorithm to cope with longer bursty losses. Section 4 shows our experimental results, and conclusions are drawn in Section 5.

2. Transformation-based Reconstruction for 2-Way Interleaving

In this section, we analyze the problem for 2-way interleaving and the reconstruction algorithm based on the average of adjacent samples (Figure 2).

The method works as follows. Assume the input audio stream to be $x_0, x_1, \cdots$, and each packet to be of N samples. Further assume packet P_1 has even samples $x_0, x_2, \cdots, x_{2N-2}$, and P_2 has odd samples $x_1, x_3, \cdots, x_{2N-1}$. We call P_1 and P_2 an *interleaving pair*. The receiver reconstructs P_2 as follows when it receives P_1 but not P_2:

$$\hat{x}_{2j-1} = \frac{x_{2j-2} + x_{2j}}{2} \qquad \text{where } j = 1, \ldots, N \qquad (2)$$

In this method, the sender only performs interleaving and groups related information into different packets, hence preventing an isolated loss to cause the loss of an entire segment of information. Intuitively and experimentally, the method improves the quality of information transmitted in the presence of isolated losses. However, it does not guarantee the quality of the reconstructed signals because the waveform of audio signals can sometimes be very rugged, and an average approximation may be inaccurate.

To overcome this problem without increasing network bandwidth, the sender transforms each original sample into a new sample before decomposition, interleaving, packeting, and transmission. It performs the transformation in such a way that the reconstructed samples at the receiver will be the best approximation to the original ones on the average, where the reconstruction criterion is defined as the mean-square error between the reconstructed signals and the original signals.

In the following subsections, we present the details of our transformation. There are two cases to be considered at the receiver, the first case is when only one of the packets in an interleaving pair is received, and the other when both packets in the interleaving pair are received.

2.1. Case I: One Packet Loss in an Interleaving Pair

Suppose, $\vec{x} = x_0, x_1, ..., x_{2N-1}$, the original data stream at the sender, is transformed into stream $\vec{y} = y_0, y_1, \cdots, y_{2N-1}$ by transformation $\mathbf{T}$ (unknown yet). Suppose further the receiver only receives half of the stream. Without loss of generality, assume all even samples $\vec{y}_{even} = y_0, y_2, \cdots, y_{2N-2}$ are received. After reconstruction using the average reconstruction method, the reconstructed stream, denoted by $\vec{\hat{y}}_{even} = \hat{y}_0, \hat{y}_1, ..., \hat{y}_{2N-1}$, is calculated as follows (assuming $x_{2N} = 0$):

$$\hat{y}_i = \begin{cases} y_i & i \text{ even} \\ \frac{y_{i-1}+y_{i+1}}{2} & i \text{ odd and } i \neq 2N-1 \\ \frac{y_{2N-2}}{2} & i = 2N-1 \end{cases} \tag{3}$$

RE, the reconstruction error, is defined as follows:

$$RE \quad = \sum_{i=0}^{2N-1} (x_i - \hat{y}_i)^2 \tag{4}$$

To minimize *RE*, we need to compute y_i to satisfy the following equations, for any even i:

$$\frac{\partial RE}{\partial y_i} = 0, \quad i = 0, 2, \cdots, 2N-2. \tag{5}$$

After substituting *RE* in (4) into (5) and simplifying the expression, we get the following matrix transformation:

$$\vec{y}_{even} = \mathbf{T} \times \vec{x} \tag{6}$$

where

$$\mathbf{T} = \mathbf{A}^{-1} \times \mathbf{B} \tag{7}$$

$$= \begin{pmatrix} 1 & \frac{1}{5} & 0 & & & & \\ \frac{1}{6} & 1 & \frac{1}{6} & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & \frac{1}{6} & 1 & \frac{1}{6} \\ & & & & 0 & \frac{1}{6} & 1 \end{pmatrix}^{-1} \times \begin{pmatrix} \frac{4}{5} & \frac{2}{5} & & & & & & & & \\ 0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & \\ & & & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix} \tag{8}$$

It is easy to prove that (5) has one and only one solution because matrix $\mathbf{A}$ in (8) is invertible.

From (6) we obtain y_{2n}, $n = 0, 1, \cdots, N-1$. In order to get the complete transformation from $\vec{x}$ to $\vec{y}$, $\vec{y}_{odd} = y_1, y_3, \cdots, y_{2N-1}$ are needed, which can be obtained by following a similar procedure for deriving $\vec{y}_{even}$.

Now the transformation from $\vec{x}$ to $\vec{y}$ is complete. The sender can send the transformed signals over the Internet, and if the loss is isolated, the receiver can reconstruct the original signals near optimally. The reason why optimality is only "near" is due to the fact that $\vec{y}$ takes floating point values after transformation and is generally cast to lower precision (like integers) to reduce bandwidth in transmission. Luckily, the loss of precision has minor effect on the reconstruction quality since only an average rounding error of ± 0.5 per sample is introduced.

Figure 1 illustrates the result of applying the transformation procedure presented in this subsection on a segment of voice samples. The SNR based on the transformed even samples and reconstructed odd samples is 7.16 dB (Figure 1b), whereas the SNR based on the original even samples and reconstructed odd samples is 5.23 dB (Figure 1a).

2.2. Case II: No Packet Loss in an Interleaving Pair

In case that both packets in an interleaving pair are received, we obtain the following inverse transformation to restore $\vec{x}$.

$$\vec{x} = \begin{pmatrix} \frac{4}{5} & \frac{2}{5} & & & & & & \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ & & & & & & \frac{2}{5} & \frac{4}{5} \end{pmatrix}^{-1} \times \begin{pmatrix} 1 & 0 & \frac{1}{5} & 0 & & & & & & & \\ 0 & 1 & 0 & \frac{1}{6} & & & & & & & \\ \frac{1}{6} & 0 & 1 & 0 & \frac{1}{6} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & \frac{1}{6} & 0 & 1 & 0 & \frac{1}{6} \\ & & & & & & & \frac{1}{6} & 0 & 1 & 0 \\ & & & & & & & & \frac{1}{5} & 0 & 1 \end{pmatrix} \vec{y} \tag{9}$$

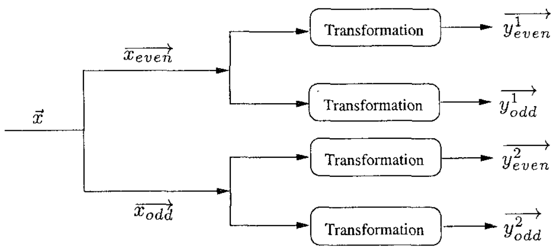

Figure 3. Constructing 4-way interleaving using two 2-way interleaving.

After computing $\vec{x}$ using (9) at the receiver, the original stream $\vec{x}$ can be restored if $\vec{y}$ does not have any precision loss during processing and transmission.

If there is precision loss when we cast $\vec{y}$ from floating-point values to 16-bit signed integers, then the reconstruction quality may obviously be not as good as using floating point values. The reason is that the inverse transformation defined in (9) is actually a linear transformation, and the loss of precision in every component of $\vec{y}$ will be accumulated. Fortunately, by carefully choosing the size of the transformation matrices, the original signals can be reconstructed quite well. Section 5 shows that the SNR of the reconstructed signal can reach around 30 dB.

In summary, our method always performs better than the method without transformation in the presence of loss. It gives near optimal reconstruction in case that one packet of an interleaving pair is lost, and can reconstruct the original signals if no packet is lost.

3. Coping with Longer Bursty Losses

After reassembling the packets received in the order sent, two-way interleaving can overcome packet loss with burst length of one and can accommodate a burst length of two as long as the two lost packets are in different interleaving pairs. However, burst lengths in the Internet can be larger than one. Statistics from our experiments and [4] illustrates that domestic sites generally have bursty losses of one packet, whereas international sites may have bursty losses of three or more packets.

We can construct M-way interleaving using a combination of 2-way interleaving. Consider 4-way interleaving as an example. Figure 3 shows the four streams created at the sender. The sender first interleaves the original stream $\vec{x}$ to even ($\overrightarrow{x_{even}} = x_0, x_2, \cdots$.) and odd ($\overrightarrow{x_{odd}} = x_1, x_3, \cdots$.) streams. It then transforms $\overrightarrow{x_{even}}$ and $\overrightarrow{x_{odd}}$ to get four streams $\overrightarrow{y^1_{even}}$, $\overrightarrow{y^1_{odd}}$, $\overrightarrow{y^2_{even}}$, and $\overrightarrow{y^2_{odd}}$ using the transformation procedure discussed in Section 2. We call the corresponding four packets *an interleaving set.*

At the receiver, when there is no loss, $\overrightarrow{x_{even}}$ can be reconstructed by inverse transformation of $\overrightarrow{y^1}$, which is constructed by de-interleaving $\overrightarrow{y^1_{even}}$ and $\overrightarrow{y^1_{odd}}$. Similarly, by inverse transformation of $\overrightarrow{y^2}$, which is constructed by de-interleaving $\overrightarrow{y^2_{even}}$ and $\overrightarrow{y^2_{odd}}$, $\overrightarrow{x_{odd}}$ can be reconstructed.

At the receiver, there are four possibilities of loss on the packets received. In the first case, three packets in an interleaving set were lost. For example, assume $\overrightarrow{y^1_{odd}}$, $\overrightarrow{y^2_{even}}$, and $\overrightarrow{y^2_{odd}}$ were lost. By our explanation in Section 2, we can optimally reconstruct $\overrightarrow{y^1_{odd}}$ from $\overrightarrow{y^1_{even}}$ received. Since $\overrightarrow{y^1}$ is now reconstructed, we can reconstruct each sample in $\overrightarrow{x_{odd}}$ by computing the average of its adjacent even samples. In this case, the reconstruction quality is guaranteed to be better than that without transformation, because $\overrightarrow{y^1_{odd}}$ is optimally reconstructed.

In the second case, two packets that are both in either $\overrightarrow{y^1}$ or $\overrightarrow{y^2}$ were lost. For instance, assume $\overrightarrow{y^2_{even}}$ and $\overrightarrow{y^2_{odd}}$ were lost. We can apply the inverse transformation discussed in Section 2.2 on $\overrightarrow{y^1_{even}}$ and $\overrightarrow{y^1_{odd}}$ to obtain $\overrightarrow{x_{even}}$, and recover $\overrightarrow{x_{odd}}$ by averaging. The reconstruction quality may not always be better than the case without transformations because $\overrightarrow{x_{even}}$ was not optimized for the reconstruction of $\overrightarrow{x_{odd}}$.

In the third case, two packets, one in $\overrightarrow{y^1}$ and one in $\overrightarrow{y^2}$, were lost. Since our method can optimally reconstruct both lost packets based on the packets received, it can guarantee better performance than the case without transformations.

In the last case, only one packet in an interleaving set was lost. For example, assume $\overrightarrow{y^2_{odd}}$ was lost. Using our method, we can optimally reconstruct $\overrightarrow{y^2_{odd}}$ from $\overrightarrow{y^2_{even}}$ received. We can further reconstruct $\overrightarrow{x_{even}}$ by applying our inverse transformation on $\overrightarrow{y^1_{even}}$ and $\overrightarrow{y^1_{odd}}$. Without any loss of computational precision, we can also guarantee better reconstruction quality than the case without transformations.

In short, without precision loss, our scheme always guarantees better performance than the case without transformation for 4-way interleaving.

Obviously, a larger M is able to tolerate more consecutive packet losses, at the expense of longer delays, because the receiver will have to assemble all the interleaved streams before playing the combined stream. On the other hand, a small M cannot overcome many packet losses, but has shorter delay that is sometimes critical in real-time applications. A suitable M for real-time audio transmissions by the sender should be driven by feedback information from the receiver.

4. Complexity for 2-way Interleaving

The computational complexity of our proposed transformation is low, as all the matrices as well as their inverses can be computed beforehand if their dimensions are known. The actual online computation only involves a few multiplications and additions.

To illustrate the computational complexity for the sender, we use transformation (6) for 2-way interleaving as an example. Substituting $\mathbf{T}$ in (6) with $\mathbf{A}^{-1}\mathbf{B}$, we have $\vec{y} = \mathbf{A}^{-1} \times (\mathbf{B}\vec{x})$. If every element of $\mathbf{A}$ is nonzero, then the computational cost for $\mathbf{B}\vec{x}$ is $3N$ multiplications plus $2N$ additions. To get y, the computational cost is N^2 multiplications plus $(N-1) \times N$ additions. The overall complexity for solving the equations is, therefore, $(N+3) \times N$ multiplications and $(N+1) \times N$ additions for every N samples. In other words, the complexity for each sample is $N+3$ multiplications and $N+1$ additions.

In practice, the complexity for large N is far less than the above because not all entries of $\mathbf{A}^{-1}$ are effective in calculating $\vec{y}$. A simple explanation is given as follows.

Assuming the audio data for processing is in 16-bit linear PCM wave format in the range $[-32768, 32767]$ and a ± 1 change of sample value within the range will not be perceptible to human ears. We observe that the maximum absolute value in vector $\vec{x'}$ ($\vec{x'} = \mathbf{B}\vec{x}$) is less than $M = \frac{4}{3} \times 32768$ because the summation of any row of matrix $\mathbf{B}$ is less than $\frac{4}{3}$. A transformed signal y_{2n} can be computed using:

$$y_{2n} = \lfloor \sum_{k=1}^{N} (\mathbf{A}^{-1}_{n,k} \times x'_k) \rfloor$$

If $|\mathbf{A}^{-1}_{n,k}| < \frac{1}{2MN}$, then the difference with or without $\mathbf{A}^{-1}_{n,k} \times x'_k$ is at most $\frac{1}{2N}$. Thus, $\sum(\mathbf{A}^{-1}_{n,k} \times x'_k)$ is less than 0.5 for all $|\mathbf{A}^{-1}_{n,k}| < \frac{1}{2MN}$. Based on our assumption, all the items in the summation can be neglected. Hence, only those elements in matrix $\mathbf{A}^{-1}$ whose value is greater than $\frac{1}{2MN}$ are effective. For other elements, we can simply set them to zero. Obviously, the computational cost can be greatly reduced if such elements occurs frequently in the matrix. After calculating $\mathbf{A}^{-1}$, we found that only about 17 items out of each line are larger than $\frac{1}{2MN}$ for $256 > N > 17$. The total complexity is, therefore, reduced to $3 + 17 = 20$ multiplications and a few additions for each sample when N is greater than 17.

The computational complexity at the receiver needs to be analyzed separately for the two cases presented in Section 2. For *case I*, the computational cost at the receiver is low because it only fills in the missing samples by taking the average of adjacent samples received. For *Case II*, each sample will need $2N$ multiplications and $2N-1$ additions (9). Our experimental results in the next section show that N is generally small, between 32 and 64; hence, the computational cost per sample is low.

5. Experimental Results

In this section, we apply our transformation algorithms to real audio data streams to test their performance. Our SNR performance measure is calculated using (1). We apply our algorithms to four different audio files: two were selected from movie clips with woman's and man's voice, respectively, the third is a segment of a speech, and the last is a segment of music.

The first set of experiments is for *Case I* in Section 2 in which only one packet in an interleaving pair is lost. For simplicity, consider the case when all the odd samples are lost. The performance comparisons between the method presented in Section 2 and the case without transformation are shown in Figure 4. (We found similar results when all even samples were lost.) For every file, we tried different sizes of the transformation matrix: $4, 8, 16, 32, 64, 128, 256$. Figures 4a-4c compare our algorithms when applied to voice files, whereas Figure 4d compares our algorithms when applied to a music file.

The results show that our transformation-based reconstruction method is applicable to both voice and music files and improves over the case when no transformation is performed. Moreover, in case of loss, the reconstruction method based on transformed samples is consistently better than the method without transformation by about 1 to 2 dB, which means the reconstruction error of the method based on transformation is as low as $79\% - 63\%$ of the reconstruction error introduced by the method without transformation.

Another observation on the results is that N does not have any significant effect on the reconstruction quality when $N > 64$. Hence, there is no need to use very large matrices to do the transformation. This is important in our implementation because it takes less space to store smaller transformation matrices. Moreover, we show in the next set of experiments that a smaller transformation matrix achieves less reconstruction error when no packets are lost.

The absolute SNR values differ in these figures due to several reasons. Both the audio volume and quality can affect the absolute value of SNR. For example, if the audio quality is poor and the signals contain a lot of random and discontinuous noise, then it is hard for the missing samples to be reconstructed using the average of neighboring samples. Also, if the audio volume is low, then noise may mislead the reconstruction procedure.

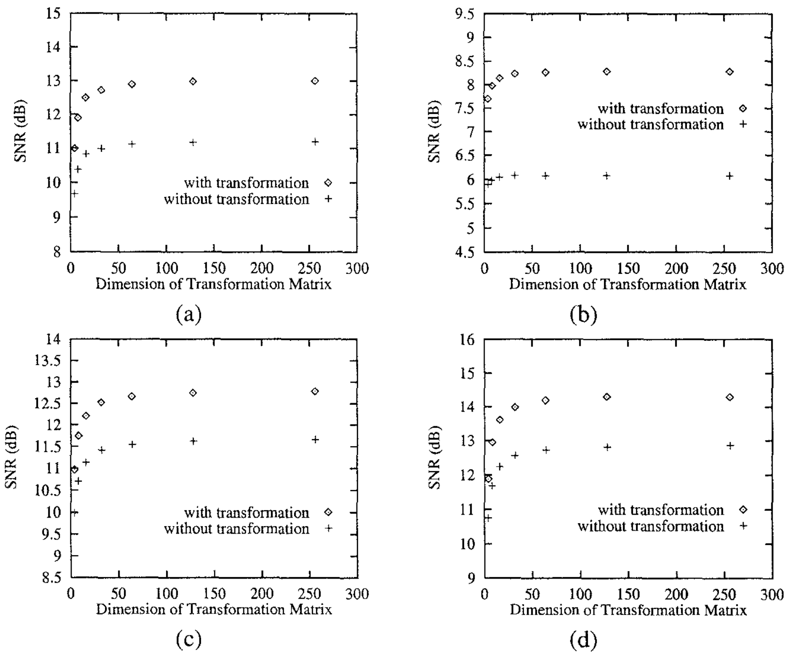

Figure 4. Comparison of reconstruction quality between using transformation versus no transformation on (a) a movie clip of a woman's voice, (b) a movie clip of a man's voice, (c) a segment of speech, and (d) a segment of music.

The second set of experiments is for *Case II* in Section 2 when both packets in an interleaving pair are received by the receiver. Table 1 shows the reconstruction quality for different sizes of the transformation matrix. The largest size of the transformation matrix chosen is 64, at which point near optimal reconstruction quality can be achieved. For every size of the transformation matrix, we compute the SNR's for the cases with and without loss.

The table shows that the SNR's for cases of no loss are consistently decreasing when the size of the transformation matrix is increasing, whereas the SNR's for cases with loss are consistently increasing. The reason for this behavior is that for a given N, $2N$ simultaneous equations are to be solved in order to get $\vec{x} = x_0, x_1, \cdots, x_{2N-1}$ from $\vec{y} = y_0, y_1, \cdots, y_{2N-1}$ using (9). This causes each x_i, $i \in [0,\ 2N-1]$ to be related to all the other $2N-1$ values and to accumulate numerical errors in the process of solving (9) when some y_i has rounding errors. Our results show that larger N's introduce more rounding errors and worse reconstruction quality.

Another observation from Table 1 is that the degradation in reconstruction quality is only around 0.2 dB for the case of loss when N changes from 64 to 32, whereas the improvement in reconstruction quality for the case of no loss is over 9 dB. When SNR reaches around 30 dB at $N = 32$, the reconstruction quality is usually satisfactory.

The choice of the proper value of N may also depend on the loss behavior of the connection. If the connection has very few losses, then smaller N's are preferred, whereas if losses are frequent, then larger N's are better.

Finally, Table 2 shows the reconstruction quality for 4-way interleaving based on two 2-way interleaving. In the first three cases in which 2 or 3 packets in an interleaving set were lost, our proposed transformation method has better or equal reconstruction quality as compared to the case without transformation. The reconstruction quality in the last case is worse due to precision loss. Since 4-way interleaving is mostly applied when we experience a high percentage of long bursty losses, we conclude that our method improves the reconstruction quality.

6. Conclusions

In this paper, we have proposed a new transformation-based reconstruction algorithm that transforms the input signals into another form in such a way that facilitates better reconstruction at the receiver. Our experiments show that our algorithm generally works well for both voice and

Table 1. The quality of reconstructed information based on transformed voice samples for two-way interleaving. N is the size of the transformation matrix. *Loss* **represents the case when one of the two interleaved streams was lost.** *No Loss* **represents the case when both streams were received.**

Sound File	SNR (dB)									
	$N = 32$		$N = 40$		$N = 48$		$N = 56$		$N = 64$	
	Loss	No Loss	Loss	No Loss	Loss	No Loss	Loss	No Loss	Loss	No Loss
1	12.71	28.37	12.82	25.19	12.85	23.19	12.83	21.02	12.90	19.14
2	8.23	31.09	8.23	28.11	8.25	26.12	8.25	23.96	8.26	22.13
3	12.57	32.12	12.57	28.87	12.63	26.47	12.71	24.39	12.70	23.25
4	13.99	34.56	14.06	31.54	14.12	29.02	14.18	27.31	14.19	25.62

Table 2. Reconstruction quality for 4-way interleaving based on three 2-way interleaving for cases with and without transformation; (I) three out of the four interleaved streams were lost; (II) one of the two groups of the four interleaved streams was lost; (III) two streams, each in a different group, were lost; and (IV) one stream out of the four interleaved streams was lost.

Sound File	SNR (dB)							
	I		II		III		IV	
	With	Without	With	Without	With	Without	With	Without
1	7.93	6.68	11.19	11.23	9.17	7.77	13.06	14.28
2	4.55	4.09	6.08	6.08	8.05	6.90	11.20	9.10
3	7.37	6.18	11.66	11.68	8.25	6.74	12.99	14.60
4	9.39	8.28	12.89	12.90	10.53	9.12	14.63	15.92

music data. With no packet loss, our algorithm can reconstruct the original signals with very small reconstruction error. When there are 50% isolated packet losses, it improves the SNR by 1 to 2 dB when compared to the algorithm with no transformation. Its computational overhead is negligible, allowing it to be implemented efficiently in real time. It is also general and can be extended to other interpolation-based reconstruction algorithms.

References

[1] T. Strayer B. Dempsey and A. Weaver. Adaptive error control for multimedia data transfers. In *Proceedings. International Workshop on Advanced Communications and Applications for High Speed Networks.*, pages 279–289, March 1992.

[2] J.C. Bolot and A.V. Garcia. Control mechanisms for packet audio in the Internet. In *Proc. IEEE Infocom '96*, pages 232–239, April 1996.

[3] J.C. Bolot and P. Hoschka. Adaptive error control for packet video in the Internet. In *Proceedings. International Conference on Image Processing (Cat. No.96CH35919).*, volume 1, pages 25–28, 9 1996.

[4] Z. G. Chen. *Coding and Transmission of Digital Video on the Internet.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[5] V. Hardman, M.A. Sasse, M. Handley, and A. Watson. Reliable audio for use over the Internet. *http://www.cs.ucl.ac.uk*, pages 1–13, October 1997.

[6] P. McKenney N. Shacham. Packet recovery in high-speed networks using coding and buffer management. In *Proc. IEEE Infocom '90, San Fransisco, CA*, pages 124–131, May 1990.

[7] Technology Network. Voice over packet tutorial, 1997.

[8] S.W. Christensen N.S. Jayant. Effects of packet losses in waveform coded speech and improvements due to odd-even sample-interpolation procedure. *IEEE Trans. on Communications*, COM-29(2), February 1981.

[9] D. Towsley R. Ramjee, J. Kurose and H. Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the 13th Annual Joint Conference of the IEEE Computer and Communications Societies on Networking for Global Commmunication*, volume INFOCOM94-2, pages 680–688, 1994.