Transformation-Based Reconstruction for Real-Time Voice Transmissions over the Internet

Benjamin W. Wah, *Fellow, IEEE,* and Dong Lin

***Abstract*— In this paper, we explore the loss behavior encountered in transmitting real-time voice over the Internet and propose a new loss-concealment scheme to improve its received quality. One known technique to conceal loss is to send interleaved streams of voice samples and reconstruct missing or late samples by interpolation at the receiver. Based on this method, we propose a new transformation-based reconstruction algorithm. Its basic idea is for the sender to transform an input voice stream, according to the interpolation method used at the receiver and the predicted loss behavior, before interleaving the stream. The transformation is derived by minimizing reconstruction error in case of loss. We show that our method is computationally efficient and can be extended to various interleaving factors and interpolation-based reconstruction methods. Finally, we show performance improvements of our method by testing it over the Internet.**

***Index Terms*— Interleaving, Internet, linear interpolation, packet-loss behavior, real-time voice transmissions, reconstruction, transformation.**

I. Introduction

THE existing Internet protocol (IP) is a connectionless, best-effort protocol that does not provide quality-of-service guarantees like in traditional public switched telephone networks (PSTN). In order to achieve high-quality real-time voice transmissions with low delay in the Internet, effective loss-concealment mechanisms must be developed.

Existing software-based loss-concealment mechanisms can be classified into two categories: receiver-based, and sender- and receiver-based.

In receiver-based reconstruction schemes, lost packets are recreated by padding silence or white noise [1], or by repeating the last received packet [2], or by substituting lost packets by previously received packets after some form of pattern matching [3]. These strategies only work well when losses are infrequent and when frame sizes are small [4].

Sender- and receiver-based reconstruction schemes are usually more effective but more complex. A common way is for the sender to first process input streams in such a way that the receiver can better reconstruct missing data. Based on different ways of processing input data, these schemes can further be split into those that add redundant control and those that do not.

There are several methods for the sender to add redundancy. These include sending duplicate packets [5], or sending past packets coded in lower bit rate along with current ones [4], or sending error correction bits in voice packets using forward error correction (FEC) [6], [7]. All these methods require extra bandwidth or long end-to-end delays.

There are also algorithms that do not add redundancy but utilize inherent redundancies in source voice streams. A typical method interleaves voice samples into distinct packets and reconstructs lost samples by interpolation using their surviving neighbors. The simplest form is two-way interleaving that packetizes odd- and even-number samples separately [8], and interpolates lost samples by simple averaging in case one of the packets is lost [1]. We call the two packets with the corresponding even and odd samples an *interleaving pair*. Simple averaging generally works well but may not give high quality for voice segments containing nonnegligible high frequency components.

In this paper, we propose a sender- and receiver-based algorithm based on interleaving and interpolation. Simple averaging is used throughout this paper as an example of interpolation. Specifically, the sender in the new algorithm transforms an input stream according to the interpolation method used at the receiver and the predicted loss behavior, in order to enable better reconstruction quality. Fig. 1 outlines the algorithm.

To demonstrate the effect of transformation, consider a two-way interleaved stream based on a typical segment of voice data with 16 samples. Assuming that all odd samples were lost at the receiver and that the eight even samples were used to reconstruct the missing samples, the dotted line in Fig. 2 plots the reconstructed stream in which a missing odd sample was computed as the average of its two original adjacent even samples. The dashed line plots the reconstructed waveform using the transformed samples received. It is obvious that the reconstructed stream based on the transformed samples gives a better approximation to the original stream, based on the measure of signal-to-noise ratio (SNR) where

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

where s is the original signal and $\hat{s}$ is the reconstructed or received signal of s. Note that transformations at the sender and reconstructions by averaging at the receiver will introduce aliasing distortions to the signals.

Manuscript received April 2, 1999; revised September 10, 1999. This work was supported by National Science Foundation under Grant MIP 96-32316 and by a gift from Rockwell International. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mohammed Ghanbari.

The authors are with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: wah@manip.crhc.uiuc.edu; dlin@manip.crhc.uiuc.edu).

Publisher Item Identifier S 1520-9210(99)09939-3.

```
process sender
1. while (true) do
2.     Decide whether to transform a voice stream
           based on the reconstruction method used
           at the receiver and the loss statistics fed
           back from the receiver;
3.     Interleave the transformed stream;
4.     Packetize each substream;
5.     Compress each packet;
6.     Send the packets to the receiver
7. end-while
end

process receiver
1. while (true) do
2.     Decompress each packet;
3.     Group packets received into interleaving pairs;
4.     if both packets in an interleaving pair are received
5.         Deinterleave the stream;
6.         Decide whether to perform inverse
               transformation on the combined stream;
7.         Play the reconstructed stream
8.     else if one packet in an interleaving pair is lost
9.         Reconstruct the missing samples using simple
               averaging;
10.        Play the reconstructed stream
11.    else if both packets are lost
12.        Feed silence to the sound card
13.    end-if
14. end-while
end
```

Fig. 1. Pseudo-code showing the steps carried out by the sender and the receiver using two-way interleaving and reconstruction.

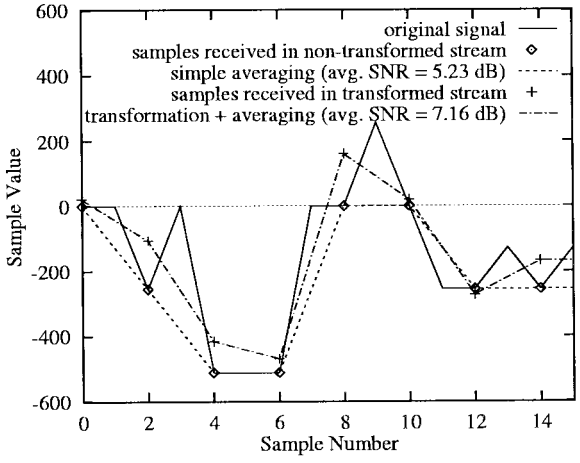


Fig. 2. Comparison of reconstruction quality between simple averaging and averaging based on transformed input data, assuming that only even samples are received.

The results in this paper is based on compression techniques that operate on a sample-by-sample basis, such as ADPCM [9], and not on low bit-rate techniques that operate on segments of samples at a time, such as CELP. The objective measure that we use, namely, SNR, is specific to ADPCM and not to CELP. There are two reasons for using ADPCM that may require higher bandwidth than CELP. First, our Internet experiments show that the main limitation in single-channel audio transmissions over the Internet is not the bandwidth but rather bursty losses. Hence, our main focus in this paper is to recover from bursty losses rather than reducing the bandwidth. Second, our prototype is software based and requires efficient implementations of its compression scheme. The transformation method proposed may still work when CELP is used, although this may require using a different objective and the integration of the transformation method with the coding method (rather than having transformation as a preprocessing stage).

TABLE I
HOSTS INVOLVED IN OUR INTERNET EXPERIMENTS

Host Domain Name	IP Address	Location
trace4.crhc.uiuc.edu	130.126.142.44	UIUC
cs.stanford.edu	171.64.64.64	Stanford U.
bart.cc.utexas.edu	128.83.40.7	U. of Texas
math.mit.edu	18.87.0.8	MIT
public.guangzhou.gd.cn	202.96.128.111	China
iasia1.iasi.rm.cnr.it	150.146.5.13	Italy
faraday.ee.uec.ac.jp	130.153.149.28	Japan

This paper has five sections. Section II studies packet-loss patterns to six Internet sites and concludes that packet losses can be concealed effectively by interleaving and reconstruction. Sections III and IV present in details our transformation-based reconstruction method and test results. Finally, Section V concludes this paper.

II. INTERNET TRAFFIC EXPERIMENTS

This section presents a series of experiments conducted between a University of Illinois at Urbana-Champaign (UIUC) computer and six Internet sites listed in Table I during the first week of September 1998. The experiments were used to identify the characteristics of packet losses and their bursty behavior.

During the experiments, the UIUC computer periodically sent 2000 probe packets, at a rate of 100 packets/s and 500 bytes/packet, at the beginning of each hour over a 24-h period to the echo port of each remote computer, and monitored the packets bounced back. (The sending rate and packet size were picked to reflect the upper bound on traffic in voice communications over the Internet. The packet size was picked to be smaller than the MTU of the Internet in order to avoid fragmentation. Results on other packet transmission rates can be found in the [10].) Statistics, such as sending and arrival times for each packet, was collected. To account for "delayed losses," each packet received had a scheduled "playback" time calculated from the arrival time of the first received packet and the difference of their sequence numbers. A packet was considered lost if it had been delayed by more than 200 ms of its scheduled playback time.

Our first set of experiments address the probability distribution of consecutive packet losses. Fig. 3 clearly demonstrates that burst lengths of 1 and 2 are predominant. In Fig. 3(a) and (b), bursts of length 1 account for more than 85% of the total losses. Even for the China-UIUC connection with high losses, more than 80% of the losses were of burst length of 1 or 2.

Table II lists for the UIUC-China connection the conditional probability distribution of the next burst length, given the current burst length. For both sending rates, losses with burst

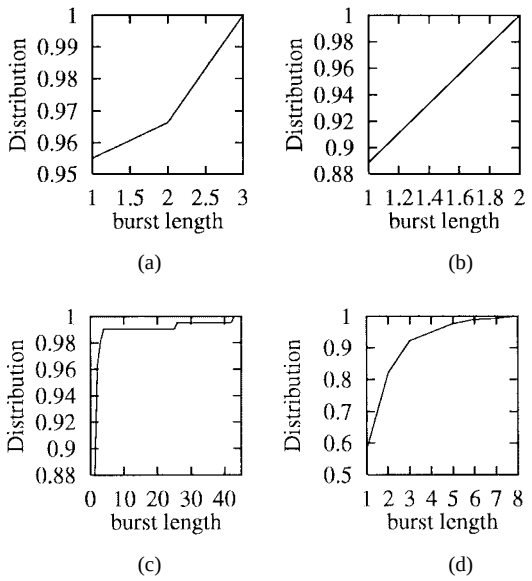


Fig. 3. Probability distribution function of consecutive packet losses. (The graphs showing the behavior of the connections to Texas and Stanford are similar to that to MIT and are not shown. The experiments were done in the first week of September 1998, with a sending rate of 100 packets per second and 500 bytes per packet.) (a) MIT-UIUC at 11 a.m. (b) Japan-UIUC at 11 a.m. (c) Italy-UIUC at 11 a.m. (d) China-UIUC at 11 a.m.

TABLE II
THE CORRELATION OF BURST LENGTHS FOR $n_p = 20000$ PACKETS SENT FROM UIUC TO CHINA AND BACK. THE DATA IN EACH ROW REPRESENTS THE CONDITIONAL DISTRIBUTION OF THE NEXT BURST LENGTH, GIVEN THE CURRENT BURST LENGTH LISTED IN THE FIRST ELEMENT IN THAT ROW

Burst length	Frac. of occ.	Conditional prob. dist. of next burst length						
		1	2	3	4	5	6	≥7
Sending rate: 10 ms/packet and 500 B/packet ($n_s = 7732$)								
1	0.657	0.663	0.865	0.935	0.965	0.975	0.977	1.000
2	0.208	0.638	0.866	0.941	0.966	0.986	0.988	1.000
3	0.071	0.614	0.843	0.932	0.948	0.956	0.968	1.000
4	0.027	0.667	0.885	0.938	0.968	1.000		
5	0.012	0.733	0.889	0.933	0.978			1.000
6	0.002	0.625	0.875		1.000			
7	0.002	0.800		1.000				
8	0.001	0.750		1.000				
9	0.001	0.667	1.000					
≥10	0.019	0.635	0.904	0.981				1.000
Sending rate: 60 ms/packet and 500 B/packet ($n_s = 6409$)								
1	0.689	0.691	0.864	0.911	0.930	0.940	0.949	1.000
2	0.176	0.683	0.871	0.917	0.935	0.945	0.950	1.000
3	0.048	0.665	0.811	0.915	0.927	0.939	0.951	1.000
4	0.021	0.704	0.887	0.915	0.930	0.944	0.972	1.000
5	0.010	0.515	0.848		0.939		1.000	
6	0.008	0.690	0.793	0.931	0.966			1.000
7	0.011	0.784	0.973				1.000	
8	0.004	0.800	0.933		1.000			
9	0.007	0.750	0.917	1.000				
≥10	0.026	0.674	0.880	0.946	0.956			1.000

length longer than 3 happened very infrequently, and one long burst did not imply that the next burst would also be long. For example, when packets were sent every 10 msec, the unconditional probability for the current burst length to be 4 and the next burst length to be greater than or equal to 4 is only $0.027 \times (1 - 0.938) \doteq 0.002$.

The fact that burst lengths are usually small (similar results have been shown in [11]) indicates that interleaving can be a good method to ease reconstruction. When the burst length is less than the interleaving factor, there are always parts of information received that can be used to recover the lost parts. For instance, with an interleaving factor of 2, a bursty loss of length 1 and a bursty loss of length 2 with samples belonging to different interleaving pairs can be recovered approximately. With an interleaving factor of 4, a bursty loss of length less than or equal to 3 and a burst length of 4, 5, and 6 with lost packets belonging to different interleaving sets can be recovered. In general, with an interleaving factor of i, it is possible to recover a bursty loss of length less than or equal to $i-1$ and some of the bursty losses of length in the range $[i, (2i-2)]$.

Let the total number of packets sent be n_p and the interleaving factor be i. Over all the interleaving sets, assuming that consecutive losses of length j, $j \leq i$, happen m_j^i times, the total number of packets lost is n_s (independent of i):

$$n_s = \sum_{j=1}^{i} j \times m_j^i. \tag{2}$$

We can derive $\Pr(\textit{fail} \mid \textit{loss}, i)$, the conditional probability that a packet cannot be recovered for interleaving factor i. This happens when all the packets in an interleaving set are lost. From (2),

$$\Pr(\textit{fail} \mid \textit{loss}, i) = \frac{i \times m_i^i}{n_s}. \tag{3}$$

$\Pr(\textit{fail} \mid i)$, the unconditional probability that a packet cannot be recovered for interleaving factor i, can be computed as follows:

$$\Pr(\textit{fail} \mid i) = \frac{i \times m_i^i}{n_s} \times \frac{n_s}{n_p} = \frac{i \times m_i^i}{n_p}. \tag{4}$$

Fig. 4 plots $\Pr(\textit{fail} \mid i)$ for various interleaving factors and connections. $\Pr(\textit{fail} \mid i)$ drops quickly when the interleaving factor increases. For all times and all six connections, $\Pr(\textit{fail} \mid i)$ is negligible when the interleaving factor is equal to or greater than 4. Moreover, except for the China-UIUC connection, an interleaving factor of 2 works well in general, achieving $\Pr(\textit{fail} \mid i)$ well below 3%. For the China-UIUC connection [see Fig. 4(a) and (b)], an interleaving factor of 2 is not always enough because about 10–15% of the total losses will not be recoverable.

The above experimental results suggest that a small interleaving factor (between 2 to 4) is adequate. In most cases, an interleaving factor of 2 leads to good recovery.

III. TRANSFORMATION-BASED RECONSTRUCTION ALGORITHMS

In this section, we develop transformations to be applied to voice samples before interleaving, with a goal of minimizing reconstruction error. We begin with an interleaving factor of two and extend the method later to cope with larger interleaving factors.

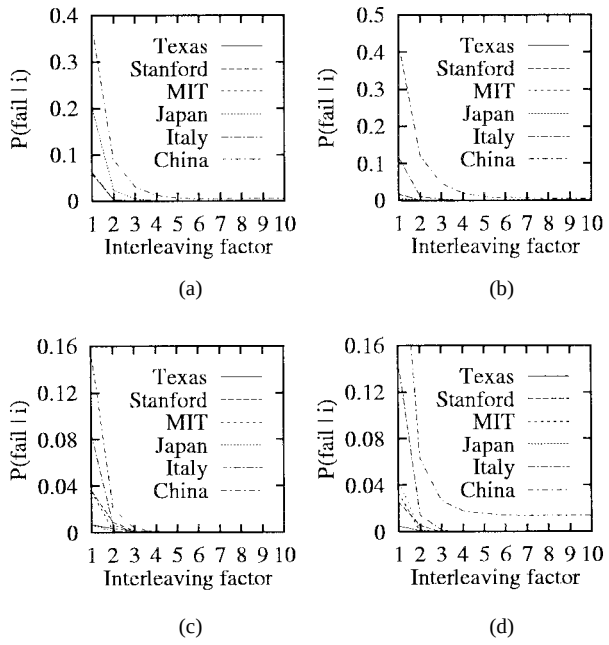


Fig. 4. $\Pr(\textit{fail} \mid i)$, probabilities of bursty losses that cannot be recovered, under various interleaving factors. (The packet transmission rate was 100 packets per second at 500 bytes/packet. The experiments were done in the first week of September 1998.) (a) UIUC 12 a.m. (b) UIUC 8 a.m. (c) UIUC 12 p.m. (d) UIUC 8 p.m.

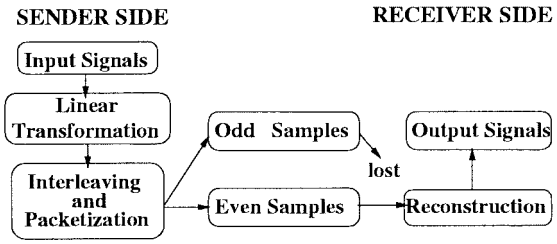


Fig. 5. Process of transformation and reconstruction in two-way interleaving.

A. Transformations and Inverse Transformations in Two-Way Interleaving

In Section I, we mention that our proposed method is based on interleaving and average reconstruction. That is, the sender performs interleaving and distributes related information in different packets, and the receiver reconstructs lost samples using the average of adjacent samples received [1]. The method improves the quality in the presence of isolated losses. However, the amount of aliasing distortions may be large when voice signals have high frequency response and cannot be reconstructed accurately by simple averaging.

To improve the quality of the above method, the sender transforms each original voice sample into a new sample before interleaving, packetization, and transmission (Fig. 5). It transforms the samples in such a way that the reconstructed samples at the receiver will be the best approximation to the original ones on the average with respect to SNR, in case that half of the samples are lost.

There are two cases to be considered at the receiver: when only one of the two packets in an interleaving pair is received, and when both packets are received.

Case I: One packet lost in an interleaving pair. Assume that the original data stream at the sender, $\vec{x} = x_0, x_1, \ldots, x_{2N-1}$, is transformed into stream $\vec{y} = y_0, y_1, \ldots, y_{2N-1}$ by transformation $\mathbf{T}$ (unknown yet), and that the receiver only receives half of the stream. Without loss of generality, assume that all even samples $\vec{y}_{\text{even}} = y_0, y_2, \ldots, y_{2N-2}$ are received. After average reconstruction, the reconstructed stream, denoted by $\hat{\vec{y}} = \hat{y}_0, \hat{y}_1, \ldots, \hat{y}_{2N-1}$, is calculated as follows (assuming $x_{2N} = 0$):

$$\hat{y}_i = \begin{cases} y_i, & i \text{ even} \\ \dfrac{y_{i-1} + y_{i+1}}{2}, & i \text{ odd and } i \neq 2N-1 \\ \dfrac{y_{2N-2}}{2}, & i = 2N-1. \end{cases} \tag{5}$$

$\mathcal{E}_o$, the reconstruction error, is defined as

$$\mathcal{E}_o = \sum_{i=0}^{2N-1} (x_i - \hat{y}_i)^2. \tag{6}$$

To minimize $\mathcal{E}_o$, y_i, for any even i, needs to satisfy the following equations:

$$\frac{\partial \mathcal{E}_o}{\partial y_i} = 0, \quad i = 0, 2, \ldots, 2N-2. \tag{7}$$

After substituting $\mathcal{E}_o$ in (6) into (7), we get the following matrix transformation:

$$\vec{y}_{\text{even}} = \begin{pmatrix} y_0 \\ y_2 \\ \vdots \\ y_{2N-4} \\ y_{2N-2} \end{pmatrix} = \mathbf{T} \times \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2N-2} \\ x_{2N-1} \end{pmatrix}. \tag{8}$$

where

$$\mathbf{T} = \mathbf{A}^{-1} \times \mathbf{B} = \begin{pmatrix} 1 & \frac{1}{5} & 0 & & & & & \\ \frac{1}{6} & 1 & \frac{1}{6} & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ & & & & \frac{1}{6} & 1 & \frac{1}{6} & \\ & & & & 0 & \frac{1}{6} & 1 & \end{pmatrix}^{-1} \times \begin{pmatrix} \frac{4}{5} & \frac{2}{5} & & & & & & & & \\ 0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & \\ & & & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix}. \tag{9}$$

The condition for (8) to have one and only one solution is when matrix $\mathbf{A}$ in (9) is invertible. The proof for the condition can be found in [10] and is not shown here.

In a way similar in deriving $\vec{y}_{\text{even}}$ in (8), the transformation for $\vec{y}_{\text{odd}}$ can be obtained.

Fig. 2 illustrates the result of applying the transformation presented in this subsection on a segment of 16 voice samples. The SNR based on the transformed (resp. original) even samples and the reconstructed odd samples is 7.16 dB (resp. 5.23 dB).

Case II: No packet lost in an interleaving pair. If both packets in an interleaving pair are received, we can apply the following inverse transformation to restore $\vec{x}$ by combining the transformation for even and odd samples:

$$\vec{x} = \begin{pmatrix} \frac{4}{5} & \frac{2}{5} & & & & & & & & \\ \frac{1}{3} & \frac{2}{3} & & & & & & & & \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ & & & & & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & & \\ & & & & & & \frac{1}{3} & \frac{2}{3} & & \\ & & & & & & & \frac{1}{3} & \frac{2}{3} & \\ & & & & & & & \frac{2}{3} & \frac{1}{3} & \\ & & & & & & & & \frac{1}{3} & \frac{4}{5} \end{pmatrix}^{-1} \times \begin{pmatrix} 1 & 0 & \frac{1}{5} & 0 & & & & & & & \\ 0 & 1 & 0 & \frac{1}{6} & & & & & & & \\ \frac{1}{6} & 0 & 1 & 0 & \frac{1}{6} & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & & & \frac{1}{6} & 0 & 1 & 0 & \frac{1}{6} \\ & & & & & & & \frac{1}{6} & 0 & 1 & 0 \\ & & & & & & & & \frac{1}{5} & 0 & 1 \end{pmatrix} \vec{y}. \quad (10)$$

After computing $\vec{x}$ using (10) at the receiver, the original stream $\vec{x}$ can be restored if $\vec{y}$ does not have any precision loss during processing and transmission. The proof for the existence and uniqueness of the solution is similar to that in *Case I*.

The computational complexity of our proposed transformation is low, since all the matrices as well as their inverses can be computed beforehand. The actual online computation only involves about 23 multiplications per sample [10] and is analyzed in Appendix A.

B. Transformations and Inverse Transformations to Cope with Longer Bursts

Statistics in Section II demonstrates that domestic sites generally have bursty losses of one packet in duration, whereas international sites may have bursty losses of three or more packets. Hence, an interleaving factor of 2 is not always enough.

When designing multiway interleaving, not only should bursty losses be considered but also isolated losses. For example, two related packets separated by interleaving with factor larger than two may both be lost when there are two isolated losses. To allow better reconstruction under different loss patterns using our transformation method, we construct M-way interleaving by recursive two-way interleaving. As an example, we describe four-way interleaving below.

Fig. 6 shows the four streams created at the sender. The sender first interleaves the original stream $\vec{x}$ to $\overrightarrow{x^1}$ $(=x_0, x_2, \ldots.)$ and $\overrightarrow{x^2}$ $(=x_1, x_3, \ldots.)$. Next, it transforms $\overrightarrow{x^1}$ and $\overrightarrow{x^2}$ using the transformation procedure discussed in last section and interleaves each of the two transformed streams to get four streams $\overrightarrow{y^1_{\text{even}}}$, $\overrightarrow{y^1_{\text{odd}}}$, $\overrightarrow{y^2_{\text{even}}}$, and $\overrightarrow{y^2_{\text{odd}}}$.

When there is no loss at the receiver, we can construct $\overrightarrow{x^1}$ by deinterleaving $\overrightarrow{y^1_{\text{even}}}$ and $\overrightarrow{y^1_{\text{odd}}}$ and by applying the inverse transformation on the de-interleaved stream. Similarly, $\overrightarrow{x^2}$ can be reconstructed from $\overrightarrow{y^2_{\text{even}}}$ and $\overrightarrow{y^2_{\text{odd}}}$.

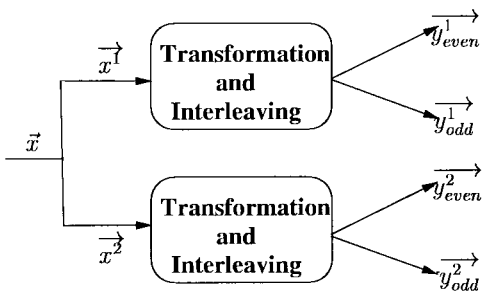


Fig. 6. Constructing four-way interleaving using recursive two-way interleaving.

When some packets are lost at the receiver, there are four possibilities. In the first case, three packets in an interleaving set were lost. For example, assume $\overrightarrow{y^1_{\text{odd}}}$, $\overrightarrow{y^2_{\text{even}}}$, and $\overrightarrow{y^2_{\text{odd}}}$ were lost. By our explanation in Section III-A, we can optimally reconstruct $\overrightarrow{y^1_{\text{odd}}}$ from $\overrightarrow{y^1_{\text{even}}}$ received. After reconstructing $\overrightarrow{x^1}$, we can reconstruct each sample in $\overrightarrow{x^2}$ 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 $\overrightarrow{y^1_{\text{odd}}}$ is optimally reconstructed.

In the second case, two packets that are both in either $\overrightarrow{x^1}$ or $\overrightarrow{x^2}$ were lost. For instance, assume $\overrightarrow{y^2_{\text{even}}}$ and $\overrightarrow{y^2_{\text{odd}}}$ were lost. We can apply the inverse transformation in *Case II* in Section III-A on $\overrightarrow{y^1_{\text{even}}}$ and $\overrightarrow{y^1_{\text{odd}}}$ to obtain $\overrightarrow{x^1}$, and recover $\overrightarrow{x^2}$ by averaging. The reconstruction quality should be similar to the case without transformations because $\overrightarrow{x^1}$ was not optimized for the reconstruction of $\overrightarrow{x^2}$.

In the third case, two packets, one in $\overrightarrow{x^1}$ and one in $\overrightarrow{x^2}$, were lost. Using our method, both lost packets can be optimally reconstructed based on the packets received. Hence, we 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 $\overrightarrow{y^2_{\text{odd}}}$ was lost. Using our method, we can optimally reconstruct $\overrightarrow{y^2_{\text{odd}}}$ from $\overrightarrow{y^2_{\text{even}}}$ received. We can further reconstruct $\overrightarrow{x^1}$ by applying our inverse transformation on $\overrightarrow{y^1_{\text{even}}}$ and $\overrightarrow{y^1_{\text{odd}}}$. Without any loss of computational precision, the resulting reconstruction quality is guaranteed to be better than the case without transformations.

The above analysis shows that our proposed reconstruction method is general enough to work under a variety of interleaving factors and bursty loss conditions.

Obviously, a large 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 long bursty losses, but has shorter delay

TABLE III
AUDIO FILES USED IN OUR EXPERIMENTS

File	Source of Voice Data	Size(KB)	Type
1	www.geocities.com/Hollywood/Hills/6498/ateyours.wav	127.4	movie clip (talk)
2	www.geocities.com/Hollywood/Hills/6376/heat.wav	115.6	movie clip (man)
3	admii.arl.mil/~fsbrn/phamdo/speech_demo.html	65.6	woman's speech
4	Data recorded from microphone	1797.9	music

TABLE IV
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 IN WHICH ONE OF THE TWO INTERLEAVED STREAMS WAS LOST. *NO LOSS* REPRESENTS THE CASE IN WHICH BOTH STREAMS WERE RECEIVED. A NUMBER IN BOLD, ONE FOR THE CASE OF LOSS AND ANOTHER FOR THE CASE OF NO LOSS, REPRESENTS THE BEST SNR AMONG THE FIVE TRANSFORMATION-MATRIX SIZES

	SNR (dB)									
Sound	$N=32$		$N=40$		$N=48$		$N=56$		$N=64$	
File	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

that is sometimes critical in real-time applications. The specific M used should be determined from feedback information sent by the receiver (see Section IV-C).

IV. EXPERIMENTAL RESULTS

In this section, we evaluate our proposed reconstruction method under three scenarios: controlled losses with no compression, controlled losses with compression effects included, and tests on the Internet under realistic loss and compression conditions.

A. Tests under Controlled Losses and No Compression

The tests in this and the next subsection were conducted on the four sound files listed in Table III. This subsection is focused on the scenario with controlled losses and no compression.

The first set of experiments is for *Case I* in Section III-A in which only one packet in an interleaving pair was lost. Without loss of generality, we consider the case when all odd samples were lost. Fig. 7 compares the performance between the method in Section III-A and the case without transformation. For all the voice and music files, transformations help improve the reconstruction quality by about 1 to 2 dB, corresponding to lowering the reconstruction error by about 20%–30%.

For every sound file, we tested different sizes of the transformation matrix N: $4, 8, 16, 32, 64, 128$, and 256. Fig. 7 shows that N does not have any significant effect on quality when $N > 64$. Hence, we do not use transformation matrices with size larger than 64.

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

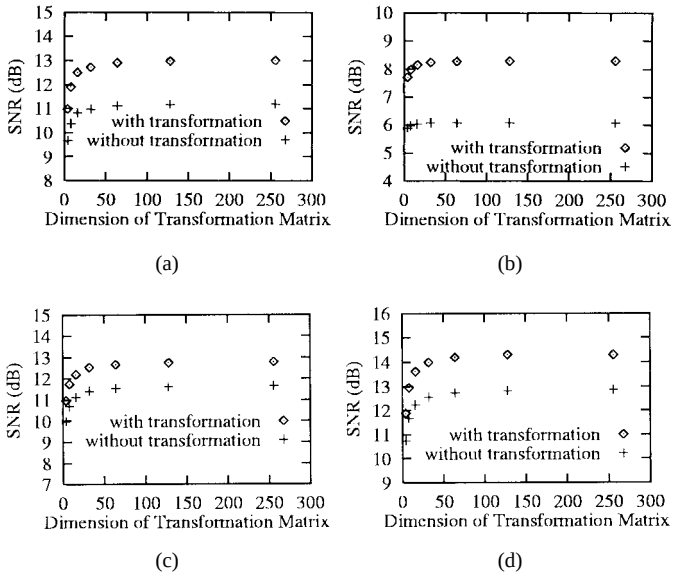


Fig. 7. Comparison of reconstruction quality between using transformation and no transformation under various sizes of transformation matrix. (a) Sound file 1. (b) Sound file 2. (c) Sound file 3. (d) Sound file 4.

We observe that the SNR's for cases of no loss are consistently decreasing when N increases, 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 have to be solved in order to get $\vec{x} = x_0, x_1, \ldots, x_{2N-1}$ from $\vec{y} = y_0, y_1, \ldots, y_{2N-1}$ using (10). This causes each $x_i, i \in [0, 2N-1]$, to be related to all the other $2N-1$ values, and numerical errors will be accumulated in the process of solving (10) when some y_i have rounding errors. Perturbations in $\vec{x}$ caused by numeric errors in $\vec{y}$ are usually measured by the condition number of $\mathbf{S}$, where $\vec{x} = \mathbf{S}^{-1}\mathbf{R}\vec{y}$ after rewriting (10). For example, the condition number of $\mathbf{S}$ is about 1712 for $N = 32$ and increases to 6744 when N increases to 64. The dramatic increase in condition number requires smaller N to be chosen.

TABLE V
RECONSTRUCTION QUALITY FOR FOUR-WAY INTERLEAVING BASED ON RECURSIVE TWO-WAY INTERLEAVING FOR CASES WITH AND WITHOUT TRANSFORMATION. A NUMBER IN BOLD REPRESENTS THE BETTER SNR BETWEEN THE METHOD WITH TRANSFORMATION AND THAT WITHOUT FOR EACH LOSS CASE

	SNR (dB)							
	I		II		III		IV	
File	With	W/O	With	W/O	With	W/O	With	W/O
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**

I: three out of four interleaved streams were lost.
II: one of the two groups of the four streams was lost.
III: two streams, each in a different group, were lost.
IV: one out of four interleaved streams was lost.

Another observation of Table IV is that the degradation for the case of loss is only around 0.2 dB when N drops from 64 to 32, whereas the improvement in reconstruction quality for the case of no loss is over 9 dB. Hence, smaller transformation matrices are preferred.

Finally, the third set of experiments deal with four-way interleaving. In the first three cases of Table V in which two or three packets in an interleaving set were lost, our proposed transformation method has better or equal reconstruction quality as compared to the case without transformation. In the last case in which one out of four streams was lost, our proposed method is worse due to precision loss. However, this case is expected to happen less frequently than the other three cases when four-way interleaving is used.

B. Tests Under Controlled Losses and with Compression

In this subsection, we consider the second scenario in which voice data is compressed as well as transformed. We chose the adaptive differential pulse-code modulation (ADPCM) compression method in our experiments over other low bit-rate speech coding methods because ADPCM has low overhead and allows us to be executed in software in real time [9].

The introduction of compression has greatly degraded the performance of our transformation-based reconstruction method. Table VI shows that transformation can improve reconstruction quality when one of the streams is consistently lost, but degrade the quality when both streams are received. In the latter case, with compression included, we should not perform inverse transformation when all the packets in an interleaving set are received. That is, Step 6 of the receiver process in Fig. 1 should not be carried out, and the interleaved streams should just be deinterleaved. The reason for this is that compression and decompression may modify some signal values by a significant amount, leading to errors that propagate to other signals when the inverse transformation in (10) is applied. For instance, several samples in Fig. 8 were introduced errors in signal values of over a hundred after compression and decompression. As shown in the last subsection, the condition number for $\mathbf{S}$ in (10) is quite large for even small N. Hence, when errors in elements of $\vec{y}$ are on the order of a hundred, deviations of $\vec{x}$ will be on the order of ten thousands or even larger.

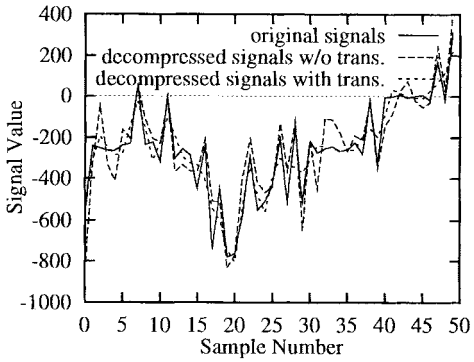


Fig. 8. Example illustrating the effects of compression/decompression on transformation under the assumption of no loss.

TABLE VI
COMPARISION OF RECONSTRUCTION QUALITY AMONG USING TRANSFORMATION, DYNAMIC TRANSFORMATION, AND NO TRANSFORMATION, AFTER INCORPORATING COMPRESSION/DECOMPRESSION. *LOSS* REPRESENTS THE CASE IN WHICH ONE OF THE TWO INTERLEAVED STREAMS WAS LOST. *NO LOSS* REPRESENTS THE CASE IN WHICH BOTH STREAMS WERE RECEIVED. N WAS SET TO 64. A NUMBER IN BOLD REPRESENTS THE BEST SNR AMONG THE THREE SCHEMES

	SNR (dB)					
	Loss			No Loss		
File	With	Dyn.	W/O	With	Dyn.	W/O
1	**12.54**	11.7	10.92	14.45	19.29	**20.96**
2	**7.97**	6.63	5.92	9.37	17.66	**19.14**
3	8.64	**8.94**	7.74	9.77	**10.07**	9.76
4	**13.84**	13.77	12.54	16.23	20.50	**22.46**

To alleviate degradations in reconstruction quality when all the packets of the interleaved streams are received, we propose a procedure to let the sender perform transformation only when there is no significant degradation in both cases of loss and no loss. Without loss of generality, assume two-way interleaving. The sender compares two alternatives, the first involving the interleaving of two consecutive packets, the compression and decompression of each, the reconstruction of all the samples assuming one interleaved stream was lost, and the computation of the reconstruction quality. The second alternative is the same as the first except that the signals are first transformed. By comparing the reconstruction qualities of the two alternatives, the sender decides whether to transform the input data or not. The results of the above dynamic transformation algorithm are shown in the third and sixth columns of Table VI. By using dynamic transformations, the reconstruction quality can be improved when one stream was lost, without sacrificing significantly reconstruction quality when both streams were received.

C. Tests on the Internet

Finally, we present experimental results on tests on the Internet under realistic loss behavior and the effects of compression. In our experiments, in addition to measuring reconstruction quality in terms of SNR, we measure end-to-end delay, jitter, and subjective quality by mean opinion score.

Fig. 9 shows the components of our real-time voice transmission system that has silence detection [12], ADPCM com-

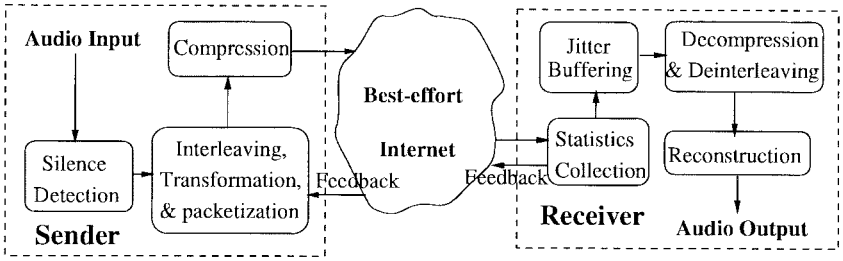
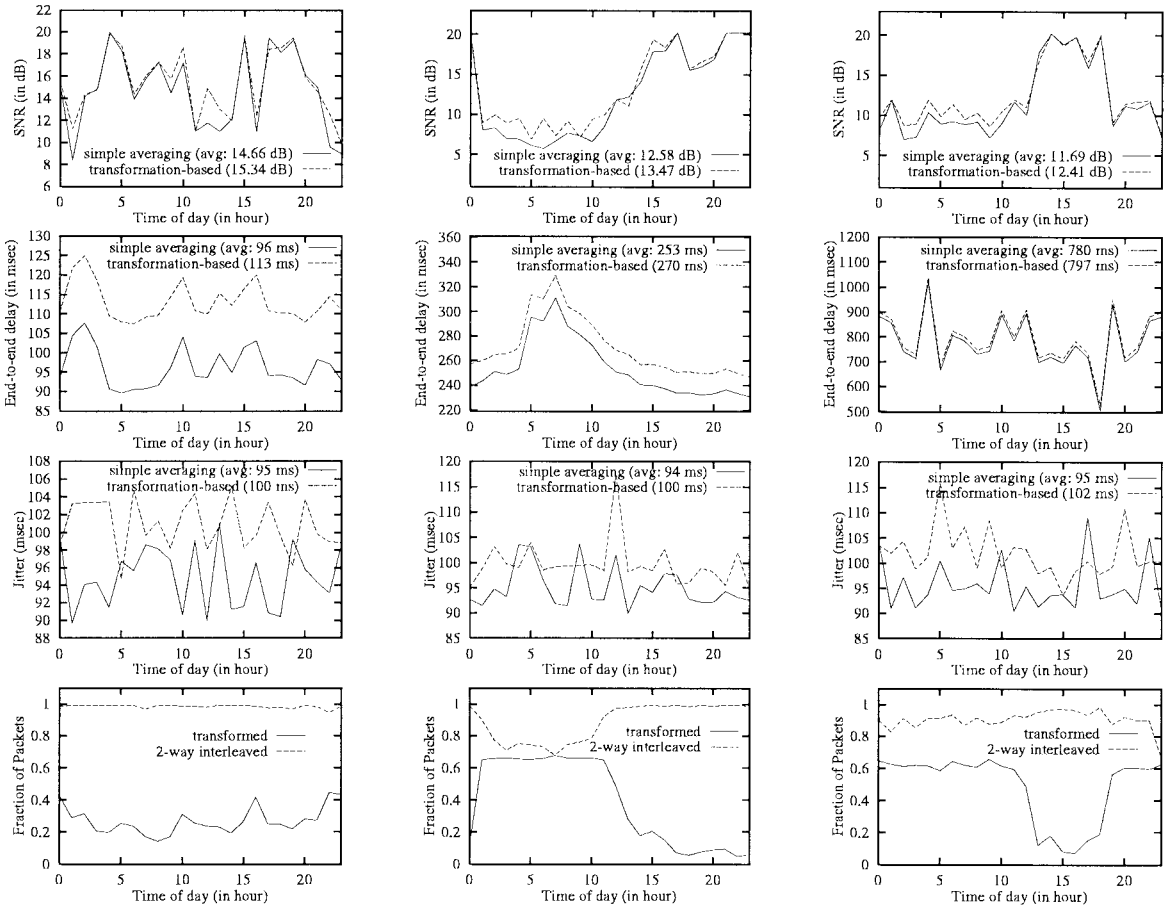


Fig. 9. Voice transmission prototype with feedback on loss statistics.



(a) (b) (c)

Fig. 10. Top three graphs of each connection compare the real-time transmission quality in terms of reconstruction errors (SNR), end-to-end delays (ms), and jitters (ms) between simple averaging and reconstruction based on transformed voice data for the six Internet connections. The bottom graph of each connection shows the fraction of packets transformed and those that were two-way interleaved. The results for the Texas-UIUC and Stanford-UIUC connections are similar to those of the MIT-UIUC connection and are not shown. Likewise, the results for the Japan-UIUC connection are similar to those of the Italy-UIUC connection. (a) MIT-UIUC. (b) Italy-UIUC. (c) China-UIUC.

pression [13], interleaving, reconstruction, statistics collection, and buffering to remove jitters [14].

As noted in Section IV-B, reconstruction quality depends on the loss rate and whether transformation was performed by the sender. When losses are low, the sender should determine a priori whether to transform voice samples before they are sent, whereas when losses are high, transformation almost always helps reduce reconstruction error, provided that a suitable interleaving factor is chosen. Hence, the receiver in our prototype sends run-time statistics on loss and burst length to the sender periodically. Based on this information, the sender chooses the best interleaving factor and whether to perform transformation before sending data to the receiver.

In our experiments, we used the six hosts listed in Section II and carried out our experiments once per hour during a 24-h period in the first week of November, 1998. Due to the use of echo ports, the end-to-end delays measured were longer than the corresponding one-way delays.

The top graphs in Fig. 10 compare the SNR between simple averaging and reconstruction based on transformed input data for the six connections. Using transformed input data, the reconstruction quality was almost always better. For connec-

TABLE VII
MEAN OPINION SCORE OF RECONSTRUCTED STREAMS WITH AND WITHOUT TRANSFORMATION

	Group 1		Group 2		Group 3	
Connection	W/O	With	W/O	With	W/O	With
China-UIUC	2.25	2.75	2.125	3	2.125	2.375
Italy-UIUC	2	2.125	2.375	3	2.625	3.125
Japan-UIUC	3	3.125	3.125	3.75	3.625	4.25
MIT-UIUC	3	3.375	3.75	4	3.75	3.875
Stanford-UIUC	3.625	3.625	3.375	3.375	3.625	4.125
Texas-UIUC	4.125	4.5	4.75	5	4.625	4.625

tions with low losses, reconstruction based on transformed data can achieve about the same quality as reconstruction based on simple averaging. In this case, most of the quality loss is due to compression. For connections with high losses, such as the China–UIUC connection, reconstruction based on transformation can improve over reconstruction based on simple averaging by about 0.7 dB on the average, with a peak improvement of over 2 dB.

The second graphs of Fig. 10 compare the average end-to-end delays over the same 24-h period. Transformations and reconstructions based on transformed data increase the average end-to-end delay by about 20 ms over reconstructions based on simple averaging. The low overhead demonstrates the low complexity of our proposed method.

The third graphs of Fig. 10 show that jitters are increased slightly under transformations. The additional jitters are caused by increased variance in processing times.

Last, the solid lines in the bottom graphs of Fig. 10 measure the fraction of packets that were transformed. For connections with low losses, such as the MIT-UIUC connection, only around 20% of the packets were transformed. In contrast, for connections with high losses, up to 65% of the packets were transformed. The dashed lines plot the fraction of packets that were two-way interleaved (the remaining packets were four-way interleaved). For domestic connections, almost all packets were two-way interleaved, whereas 10%–30% of the packets might be four-way interleaved for international connections when loss rates were high or burst lengths were larger than 2.

We have carried out subjective tests to measure the reconstruction quality using mean opinion score (MOS) [15]. In our tests, we asked three listeners to rate the reconstructed streams with and without transformation. For each connection and each test group, we picked eight streams randomly from the data collected over the 24-h period, leading to $6 \times 3 \times 8 \times 2$ streams that were randomized before tested by each listener. Table VII shows that the perceptual quality of reconstruction based on transformed input data is almost always better than that without transformation.

V. CONCLUSION

In this paper, we have proposed a new transformation-based reconstruction algorithm for real-time low-delay voice transmissions over the Internet. Our algorithm first transforms input signals into another form based on the way that signals are reconstructed at the receiver and the loss behavior in the network. We tested the new method in experiments over the Internet. Reconstructions based on transformed signals can improve SNR by about 0.7 dB for connections with heavy losses, and maintain high transmission quality for connections with low losses. Subjective tests in terms of MOS also indicate quality improvements. We show that our algorithm has negligible computational overhead, allowing it to be implemented efficiently in real time. Finally, our algorithm is general and can be extended easily to other interpolation-based reconstruction algorithms. Our future work is focused on extending the method to low-bit rate compression methods, such as CELP, that involves new objectives not based on SNR and the integration of transformation and coding.

APPENDIX A: COMPUTATIONAL COMPLEXITY OF TRANSFORMATION

To illustrate the transformation complexity, we use (8) as an example. Substituting $\mathbf{T}$ in (8) by $\mathbf{A}^{-1}\mathbf{B}$, we have $\vec{y}_{\text{even}} = \mathbf{A^{-1}} \times (\mathbf{B}\vec{x})$. The computational cost for $\mathbf{B}\vec{x}$ is $6N$ multiplications plus $4N$ additions. To multiply $\mathbf{B}\vec{x}$ by $\mathbf{A^{-1}}$, N^2 multiplications plus $(N-1)\times N$ additions are needed. The overall complexity to get $\vec{y}_{\text{even}}$ is $(N+6)\times N$ multiplications and $(N+3)\times N$ additions for every N samples, leading to a complexity of $N+6$ multiplications and $N+3$ additions per sample.

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 sampled data for processing is in 16-bit linear PCM wave format and in the range $[-32768, 32\,767]$ 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 $\vec{x'}$ $(=\mathbf{B}\vec{x})$ is less than $M = \frac{4}{3} \times 32\,768$ 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

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

If $|\mathbf{A}_{n,k}^{-1}| < \frac{1}{2MN}$, then the difference with or without $\mathbf{A}_{n,k}^{-1} \times x'_k$ is at most $\frac{1}{2N}$. Thus, $\sum(\mathbf{A}_{n,k}^{-1} \times x'_k)$ is less than 0.5 for all $|\mathbf{A}_{n,k}^{-1}| < \frac{1}{2MN}$. Based on our assumption, all these items 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, $6 + 17 = 23$ multiplications and a few additions for each sample when N is greater than 17.

REFERENCES

[1] J. Suzuki and M. Taka, "Missing packet recovery techniques for low-bit-rate coded speech," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 707–717, June 1989.

[2] R. C. F. Tucker and J. E. Flood, "Optimizing the performance of packet-switch speech," in *IERE Confe. Digital Processing of Signals in Communications*, Loughborough Univ., U.K., Apr. 1985, no. 62, pp. 227–234.

[3] O. J. Wasem, D. J. Goodman, C. A. Dvordak, and H. G. Page, "The effect of waveform substitution on the quality of PCM packet communications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 342–348, Mar. 1988.

[4] V. Hardman, M. A. Sasse, M. Handley, and A. Watson, "Reliable audio for use over the internet," in *Proc. Int. Networking Conf.*, June 1995, pp. 171–178.

[5] Telogy Networks, http://www.webproforum.com/, Int. Eng. Consortium Web Proforums, Voice over packet tutorial, Nov. 1997.

[6] N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management," in *Proc. IEEE INFOCOM*, May 1990, pp. 124–131.

[7] J. C. Bolot and P. Hoschka, "Adaptive error control for packet video in the Internet," in *Proc. Int. Conf. Image Processing*, Sept. 1996, vol. 1, pp. 25–28.

[8] 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 Trans. Commun.*, vol. COMM–29, pp. 101–110, Feb. 1981.

[9] A. S. Spanias, "Speech coding: A tutorial review," *Proc. IEEE*, vol. 82, pp. 1441–1582, Oct. 1994.

[10] D. Lin, "Real-time voice transmissions over the internet," M.Sc. thesis, Dept. Elect. Comput. Eng., Univ. Illinois, Urbana-Champaign [Online] Available http://manip.crhc.uiuc.edu/pub/papers/Dirs/TM16/, Dec. 1998.

[11] J. C. Bolot, "Characterizing end-to-end packet delay and loss in the internet," *High-Speed Networks*, vol. 2, pp. 305–323, Dec. 1993.

[12] S. Jacobs, A. Eleftheriadis, and D. Anastassiou, "Silence detection for multimedia communication," *Multimedia Syst. J.*, vol. 7, no. 2, pp. 157–164, Mar. 1999.

[13] N. S. Jayant, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[14] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *Proc. 13th Annu. Joint Conf. IEEE Computer and Communications Societies on Networking for Global Commmunication*, 1994, vol. 2, pp. 680–688.

[15] P. E. Papamichalis, *Practical Approaches to Speech Coding*, Englewood Cliffs, NJ: Prentice-Hall, 1987.

Benjamin W. Wah (S'74–M'77–SM'85–F'91) received the Ph.D. degree in computer science from the University of California, Berkeley, in 1979.

He is currently the Robert T. Chien Professor of Engineering and a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute, University of Illinois at Urbana-Champaign. Previously, he had served on the faculty of Purdue University, West Lafayette, IN (1979–1985), as a Program Director at the National Science Foundation (1988–1989), as Fujitsu Visiting Chair Professor of Intelligence Engineering, University of Tokyo, Tokyo, Japan, (1992), and McKay Visiting Professor of Electrical Engineering and Computer Science, University of California, Berkeley (1994). His current research interests are in the areas of nonlinear search and optimization, knowledge engineering, multimedia signal processing, and parallel and distributed processing.

Dr. Wah was the Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 1993 to 1996, and is the Honorary Editor-in-Chief of *Knowledge and Information Systems*. He currently serves on the editorial boards of *Information Sciences*, *International Journal on Artificial Intelligence Tools*, *Journal of VLSI Signal Processing*, and *Parallel Algorithms and Applications*. He had chaired a number of international conferences and is currently serving as the International Program Committee Chair of the IFIP World Congress in 2000. He has served the IEEE Computer Society in various capacities, is the elected First Vice President for Publications in 1999, and will be President-Elect in 2000. He is a Fellow of the Society for Design and Process Science. In 1989, he was awarded a University Scholar of the University of Illinois, and in 1998, he received the IEEE Computer Society Technical Achievement Award.

Dong Lin was born in Tianjin, China, on March 9, 1973. She received B.Sc. degree from the Department of Electrical Engineering and Information Science, University of Science and Technology of China in June 1996 and the M.S. degree from the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (UIUC) in 1999. She is currently pursuing the Ph.D. degree at UIUC.