Low Delay Burst Erasure Correction Codes

Emin Martinian
Massachusetts Institute of Technology
Cambridge, MA 02139

Carl-Erik W. Sundberg
SundComm
Chatham, NJ 07928

***Abstract*— In a variety of networks, packet losses occur in bursts. Conventional application of error correcting codes for packet recovery often requires interleaving and long decoder delays. These long delays are usually unacceptable in real-time multimedia communication applications such as Voice over IP (VoIP). Consequently erasure correction codes with severe limits on delay are desirable.**

We present various codes suitable for correcting bursts of lost packets. In addition, we show these codes have the shortest possible decoding delay. To demonstrate the practical benefits of these codes for (VoIP), we present results of informal mean opinion score listening tests conducted with various codes.

***Index Terms*—low delay coding, erasure channels, packet loss, maximally short codes, convolutional codes, VoIP,**

I. Introduction

In contrast to traditional data services such as email and File Transfer Protocol (FTP), many real-time communication protocols are emerging for packet switched networks such as the Internet. The strict delay requirements of applications such as Voice over IP (VoIP), video-conferencing, tele-medicine, etc. limit the use of ARQ (retransmission systems) and forward error correction (FEC) codes with extensive interleaving and decoding over long intervals. In addition, while most FEC codes are designed for memoryless channels, recent work suggests that packet losses in the Internet tend to occur in bursts [1].

Since packet losses due to congestion, errors, or other transmission problems can introduce significant distortions, we study the effectiveness of channel coding techniques to increase the quality of transmission across links with bursty losses. There are many ways to add redundancy to recover from packet losses, but a successful approach must consider the important features of real-time multimedia communication over packet networks. Since long pauses in interactive communication are unacceptable, a key requirement of these systems is low delay. A class of short, low delay, convolutional codes are presented in [2]. These codes have the property that for an erasure burst of a certain length, the required decoding delay and guard interval in terms of received packets are the shortest possible.

We begin by discussing some code constructions and a bound on decoding delay in Section II. In Section III we explore how these codes can be used to decrease overall distortion. To provide an empirical demonstration of the benefits of coding, we provide informal mean opinion score listening test results for a VoIP application in Section III-A. We close with a brief conclusion.

II. Code Constructions

According to standard terminology, we define a rate $R = k/n$ code to consist of an encoder and decoder with the following properties. The encoder is a causal, discrete-time system which at time step i, takes k symbols, $\vec{x}[i]$, as input and returns n symbols, $\vec{y}[i]$, as output. In this notation, $x_j[i]$ represents the jth input symbol at time i while $y_j[i]$ represents the jth output symbol at time i. The corresponding decoder takes a sequence of n symbols, $\vec{z}[i]$, as input and tries to recover the original sequence $\vec{x}[i]$. If the decoder recovers $\vec{x}[i]$ from $\vec{z}[i+T]$, $\vec{z}[i+T-1]$, $\vec{z}[i+T-2]$, ... , we say that the decoder operates with decoding delay T.

Such a code can be used for packet transmission in the following manner. A transmitter has a stream of data, $\vec{u}[i]$, which needs to be sent using n symbols per packet. To make room for redundancy, the original information is compressed into $k < n$ symbols per packet using an appropriate source coder. This creates room for $n - k$ redundant symbols to be added to each packet. Ideally the code rate, R, should be near 1 so that the original data does not have to be compressed much. Intuitively, though, lower rate codes have more redundancy which can be used to recover from lost packets.

A. A Simple Example

In order to introduce the problem we develop a rate $1/2$ example. For simplicity we consider packets carrying $2n$ bits of data. If each piece of information was sent in a separate packet of size $2n$, there would be no way to recover from lost packets. Consequently we need to add redundancy.

We can create room to add redundancy by compressing the original $2n$ information bits per packet into n bits per packet. We denote the compressed information as $\vec{x}[i]$ and put it in the first n bits of each packet. The second n bits in each coded packet can then be used to store redundant information such as parity check bits, check-sums, redundant copies, etc. We denote the ith encoded packet as $\vec{y}[i]$. Fig. 1 illustrates an example encoding scheme where packet i contains the original data, $x[i]$, as well as $x[i-3]$, a redundant copy from 3 packets ago.

This encoding scheme can correct a burst of up to 3 lost packets with a decoding delay of 3. To see this, imagine that packets $\vec{y}[i]$, $\vec{y}[i+1]$, and $\vec{y}[i+2]$ are lost. When packet $\vec{y}[i+3]$ is received both $x[i+3]$ and $x[i]$ are recovered. Since $\vec{x}[i]$ is received 3 packets after it was sent, the decoding delay is 3. Similarly when $\vec{y}[i+4]$ and $\vec{y}[i+5]$ are received $\vec{x}[i+1]$ and $\vec{x}[i+2]$ are recovered with decoding delay 3.

B. Decoding Delay Bound

The simple coding scheme illustrated in Fig. 1 illustrates the basic idea that adding redundancy in the form of time diversity allows recovery from lost packets. Is repetition the best way to add redundancy, though? The communication theory literature contains many examples of more complicated codes which usually perform better than repetition. Surprisingly, the simple

	Encoded Packet	Original Data	Redundant Data
	$\vec{y}[i] =$	$x[i]$	$x[i-3]$
	$\vec{y}[i+1] =$	$x[i+1]$	$x[i-2]$
	$\vec{y}[i+2] =$	$x[i+2]$	$x[i-1]$
packet lost	$\vec{y}[i+3] =$	$x[i+3]$	$x[i]$
packet lost	$\vec{y}[i+4] =$	$x[i+4]$	$x[i+1]$
packet lost	$\vec{y}[i+5] =$	$x[i+5]$	$x[i+2]$
recover $x[i+3]$	$\vec{y}[i+6] =$	$x[i+6]$	$x[i+3]$
recover $x[i+4]$	$\vec{y}[i+7] =$	$x[i+7]$	$x[i+4]$
recover $x[i+5]$	$\vec{y}[i+8] =$	$x[i+8]$	$x[i+5]$
	$\vec{y}[i+9] =$	$x[i+9]$	$x[i+6]$

Fig. 1. An example encoding scheme. Packet i contains the original data, $x[i]$ as well as a redundant copy, $x[i-3]$ from 3 packets ago. If packets $\vec{y}[i+3]$ through $\vec{y}[i+5]$ are lost the data can be recovered from the redundant part of later packets.

repetition code does in fact achieve the optimal decoding delay vs. correctable burst length trade-off for a rate $1/2$ code. Specifically, in [2] we prove the following theorem:

Theorem 1: If a rate R code can correct all erasure bursts of length s with decoding delay T then

$$T/s \geq \max\left[1, R/(1-R)\right]. \tag{1}$$

The decoding delay bound is similar to a well-known guard space bound [3]. The guard space bound states that if a rate R code can correct all erasure bursts of length s preceded and followed by an erasure free guard space of at least g symbols then $g/s \geq R/(1-R)$. In addition to the fact that T/s must be at least 1 while g/s can be less than 1, another difference between these bounds is that Reed-Solomon codes satisfy the guard space bound but not the delay bound. Specifically, imagine k symbols encoded with an (n, k) Reed-Solomon code. This rate $R = k/n$ code can recover any $n - k$ erasures in a burst or otherwise. However, when a symbol is erased, it can not be recovered until at least k symbols have been correctly received. Therefore the decoding delay can be much longer than Equation (1). See [3] and [2] for other codes meeting the guard space bound but not Equation (1).

The delay bound provides a practical tool to aid an engineer in system design. For example, consider a speech based communication system designed to operate over 64 kb/s links with a packet size of 10 ms where links are known to suffer occasional bursts of $s \leq 4$ lost packets. Since low delay is required for speech applications, a maximum of 120 ms or $T = 12$ packets is a reasonable decoding delay constraint. According to Equation (1), a code capable of correcting a burst of $s = 4$ losses with delay $T \leq 12$ requires $R \leq \frac{3}{4}$. Therefore if an appropriate burst erasure correction code is used then at most $\frac{3}{4} \cdot 64$ kb/s = 48 kb/s is available to encode the speech.

C. Construction and Analysis of Rate 2/3 Code

To construct a rate $2/3$ code we no longer have enough room to place a redundant copy of the information in each packet. Consequently we need to somehow combine the redundant information so it takes less space. One way to do this is using

	$\vec{y}[0] =$	$x_0[0]$	$x_1[0]$	$x_0[-1] \oplus x_1[-2]$
	$\vec{y}[1] =$	$x_0[1]$	$x_1[1]$	$x_0[0] \oplus x_1[-1]$
	$\vec{y}[2] =$	$x_0[2]$	$x_1[2]$	$x_0[1] \oplus x_1[0]$
packet lost	$\vec{y}[3] =$	$x_0[3]$	$x_1[3]$	$x_0[2] \oplus x_1[1]$
decode $x_0[3]$	$\vec{y}[4] =$	$x_0[4]$	$x_1[4]$	$x_0[3] \oplus x_1[2]$
decode $x_1[3]$	$\vec{y}[5] =$	$x_0[5]$	$x_1[5]$	$x_0[4] \oplus x_1[3]$

Fig. 2. Decoding a lost packet using a single parity check code.

	$\vec{y}[0] =$	$x_0[0]$	$x_1[0]$	$x_0[-2] \oplus x_1[-4]$
	$\vec{y}[1] =$	$x_0[1]$	$x_1[1]$	$x_0[-1] \oplus x_1[-3]$
	$\vec{y}[2] =$	$x_0[2]$	$x_1[2]$	$x_0[0] \oplus x_1[-2]$
	$\vec{y}[3] =$	$x_0[3]$	$x_1[3]$	$x_0[1] \oplus x_1[-1]$
packet lost	$\vec{y}[4] =$	$x_0[4]$	$x_1[4]$	$x_0[2] \oplus x_1[0]$
packet lost	$\vec{y}[5] =$	$x_0[5]$	$x_1[5]$	$x_0[3] \oplus x_1[1]$
decode $x_0[4]$	$\vec{y}[6] =$	$x_0[6]$	$x_1[6]$	$x_0[4] \oplus x_1[2]$
decode $x_0[5]$	$\vec{y}[7] =$	$x_0[7]$	$x_1[7]$	$x_0[5] \oplus x_1[3]$
decode $x_1[4]$	$\vec{y}[8] =$	$x_0[7]$	$x_1[8]$	$x_0[6] \oplus x_1[4]$
decode $x_1[5]$	$\vec{y}[9] =$	$x_0[8]$	$x_1[9]$	$x_0[7] \oplus x_1[5]$

Fig. 3. Decoding a burst of 2 lost packets using an interleaved version of the code in Fig. 2.

the well known idea of a parity check to construct the code. Specifically, the rate $2/3$ code, $\mathcal{C}_{\text{SPC-1}}$, can be constructed using the following encoding rule

$$\vec{y}[i] = (x_0[i], x_1[i], x_0[i-1] \oplus x_1[i-2]) \tag{2}$$

To see that this code can recover from a burst of 1 lost packet with a decoding delay of 2, consider the encoded sequence shown in Fig. 2 where the coded packet $\vec{y}[3]$ is lost. To recover $x_0[3]$ the decoder waits until

$$\vec{y}[4] = (x_0[4], x_1[4], x_0[3] \oplus x_1[2])$$

is received and recovers $x_0[3]$ using $x_0[3] = y_3[4] \oplus (-x_1[2])$. Similarly, $x_1[3]$ is recovered from $\vec{y}[5]$ using $x_1[3] = y_3[5] \oplus (-x_0[4])$.

Admittedly, a code which corrects 1 lost packet does not meet the intuitive definition of a burst erasure correction code. By using degree λ periodic interleaving, however, we can transform an s burst correcting code into a λs burst correcting code with decoding delay λT. For example, if we periodically interleave $\mathcal{C}_{\text{SPC-1}}$ by $\lambda = 2$ we obtain the code $\mathcal{C}_{\text{SPC-2}}$ shown in Fig. 3 If a burst of 2 lost packets starts at time $t = 4$, the lost data can be recovered in a manner similar to the previous example. The decoding delay for the interleaved code, $\mathcal{C}_{\text{SPC-2}}$, will be twice that required for the base code, $\mathcal{C}_{\text{SPC-1}}$. Therefore $\mathcal{C}_{\text{SPC-2}}$ is a rate $2/3$ code with $T/s = 4/2 = 2$.

By using single parity check codes in this manner we can construct rate $k/(k+1)$ codes which can correct bursts of λ erasures with a delay of λk. The encoding rule for such a code is

$$\vec{y}[i] = (\vec{x}[i], \sum_{j=0}^{k-1} x_j[i - \lambda(j+1)]). \tag{3}$$

As proved in [2], the decoding delay for these codes will be λk. According to the bounds developed in [2], these codes achieve the best trade-off between decoding delay and burst correction at rate $k/(k+1)$.

D. Construction and Analysis of Rate 3/5 Code

As with the rate $2/3$ code, constructing a rate $3/5$ code by simply repeating redundant copies is not possible. To generalize the rate $k/(k+1)$ codes to other rates we maintain the idea of using block codes computed across packets. However, instead of using single parity check codes as with the rate $2/3$ code, we move to using Maximum Distance Separable (MDS) block codes such as Reed-Solomon codes [4].

Linear, systematic, Reed-Solomon codes can be constructed for all values of $(n, k, d = n - k + 1)$ using q-ary symbols over GF($q = 2^r$) as long as $n \leq 2^r$. To construct a rate $3/5$ burst erasure correcting code we require, an $(n, k, d) = (6, 4, 3)$ block code. A $(6, 4, 3)$ MDS code, $\mathcal{R}S_{6,4,3}$, can be constructed by first constructing an $(8, 4, 5)$ systematic Reed-Solomon code over $GF(2^3)$ and then puncturing the last 2 parity check symbols. We denote the 2 remaining parity check symbols of $\mathcal{R}S_{6,4,3}$ corresponding to the 4 information symbols (x_0, x_1, x_2, x_3) as $P\{x_0, x_1, x_2, x_3\}$.

Using $\mathcal{R}S_{6,4,3}$, we can construct a rate $3/5$ code, $\mathcal{C}_{\text{MDS}}$, capable of correcting a 2 erasure burst with decoding delay 3. The encoding rule is

$$\begin{aligned} \vec{y}[i] &= (x_0[i], x_1[i], x_2[i], \\ &\quad P\{x_0[i-1], x_0[i-2], x_1[i-3], x_2[i-3]\}). \end{aligned} \tag{4}$$

Fig. 4 illustrates the encoding and decoding process for $\mathcal{C}_{\text{MDS}}$ with a 2 erasure burst.

The burst erases the coded symbols $\vec{y}[3]$ and $\vec{y}[4]$. The decoder waits until $\vec{y}[5]$ is received and then uses the decoding algorithm for $\mathcal{R}S_{6,4,3}$ to recover $x_0[3]$ and $x_0[4]$ from $y_{3,4}[5] = P\{x_0[4], x_0[3], x_1[2], x_2[2]\}$. Since $\mathcal{R}S_{6,4,3}$ has minimum distance 3 and neither $x_1[2]$ nor $x_2[2]$ were erased, $x_0[3]$ and $x_0[4]$ can be successfully recovered. Next the decoder receives $\vec{y}[6]$ and uses it to decode $x_{1,2}[3]$ using $y_{3,4}[6] = P\{x_0[5], x_0[4], x_1[3], x_2[3]\}$. Since $x_0[4]$ was recovered at the previous step and $x_0[5]$ was unerased, $x_{1,2}[3]$ can be successfully recovered. Finally, when the decoder receives $y_{3,4}[7] = P\{x_0[6], x_0[5], x_1[4], x_2[4]\}$, it recovers $x_{1,2}[4]$ since $x_0[5]$ and $x_0[6]$ were unerased.

By inspecting Fig. 4 we see that for a 2 erasure burst starting at time t, the decoding rule above requires that symbols $t+2$, $t+3$, and $t-1$ are unerased. Therefore we can see by inspection that the decoding delay will be $T = 3$, which meets the delay bound with equality. As with the rate $k/(k+1)$ codes, we can periodically interleave $\mathcal{C}_{\text{MDS}}$ by λ to obtain a rate $3/5$ code which corrects all length $s = 2\lambda$ erasure bursts with decoding delay $T = 3\lambda$. The interleaved code also meets the delay bound with equality.

1) Maximally Short Codes: In addition to the rate $3/5$ code and the rate $k/(k+1)$ codes discussed here, [2] presents code constructions for all rates of the form $R = (ms+1)/(ms+1+s)$ where m and s are non-negative integers. These codes can correct bursts of s erasures with decoding delay $ms+1$. Using degree λ periodic interleaving, these codes can be transformed to correct bursts of λs erasures with decoding delay $\lambda(ms+1)$. We refer to these codes as maximally short codes since, as shown in [2], they have the shortest possible decoding delay relative to rate and burst correcting capability.

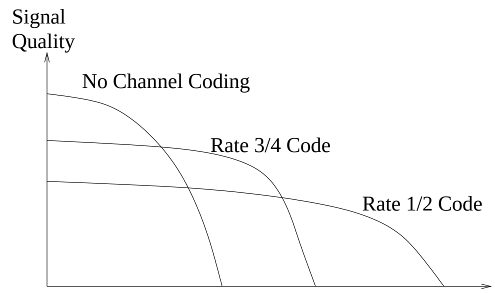

Fig. 5. Conceptual plot of the signal quality we expect for various levels of channel coding as a function of packet loss rate. We expect that for mild channels little or no coding will produce the highest quality signal at the receiver. As the packet losses become more pronounced, however, we expect that the added robustness of coded streams will produce better quality signals at the receiver.

III. Overall Distortion And Coding Gains

For a fixed capacity link, adding diversity via the codes discussed in Section II, decreases the bits available for source coding. For a channel with many erasures, however, we expect that the decrease in source coding rate is compensated for by the increased reliability introduced by the channel code. On the other hand, if the channel were perfect and no erasures occurred then little or no channel coding should be used. Thus the appropriate division of the link capacity between a channel code and a source code depends upon the severity of the channel.

A conceptual plot of this expected behavior is illustrated in Fig. 5 with some example channel coding rates. When no channel coding is used and the entire capacity is devoted to source coding, we expect good performance for channels with very little packet loss. As the packet loss increases, we expect the performance without channel coding to decrease rapidly. When some channel coding is included, such as a rate $3/4$ code, we expect the performance to be slightly worse when no packet loss occurs. As the packet loss increases, though, we expect the rate $3/4$ coded stream to be more robust than the uncoded stream. Finally, consider a stream with rate $1/2$ channel coding. When few packets are lost, we expect this stream to have the lowest quality. As the loss rate increases, however, we expect the rate $1/2$ code to suffer the least degradation. Systems with similiar trade-offs have been proposed for mobile radio [5] In [6] we consider transmitting a Gaussian source over a Gilbert-Elliot Channel and analytically derive results justifying the conceptual plot in Fig. 5.

A. Informal Mean Opinion Score Listening Tests

1) Purpose: In the previous sections we have described constructions of various burst erasure correcting codes and analyzed their robustness. Intuitively we expect that robust codes will mitigate the distortions caused by lost packets. In [6] we theoretically analyze the distortion reductions using such codes for some simple models. While these simple models allow us to

	$\vec{y}[0] =$	$x_0[0]$	$x_1[0]$	$x_2[0]$	$P\{x_0[-1], x_0[-2], x_1[-3], x_2[-3]\}$
	$\vec{y}[1] =$	$x_0[1]$	$x_1[1]$	$x_2[1]$	$P\{x_0[0], x_0[-1], x_1[-2], x_2[-2]\}$
	$\vec{y}[2] =$	$x_0[2]$	$x_1[2]$	$x_2[2]$	$P\{x_0[1], x_0[0], x_1[-1], x_2[-1]\}$
packet lost	$\vec{y}[3] =$	$x_0[3]$	$x_1[3]$	$x_2[3]$	$P\{x_0[2], x_0[1], x_1[0], x_2[0]\}$
packet lost	$\vec{y}[4] =$	$x_0[4]$	$x_1[4]$	$x_2[4]$	$P\{x_0[3], x_0[2], x_1[1], x_2[1]\}$
recover $x_0[3], x_0[4]$ here	$\vec{y}[5] =$	$x_0[5]$	$x_1[5]$	$x_2[5]$	$P\{x_0[4], x_0[3], x_1[2], x_2[2]\}$
recover $x_1[3], x_2[3]$ here	$\vec{y}[6] =$	$x_0[6]$	$x_1[6]$	$x_2[6]$	$P\{x_0[5], x_0[4], x_1[3], x_2[3]\}$
recover $x_1[4], x_2[4]$ here	$\vec{y}[7] =$	$x_0[7]$	$x_1[7]$	$x_2[7]$	$P\{x_0[6], x_0[5], x_1[4], x_2[4]\}$

Fig. 4. The rate $3/5$ code, $\mathcal{C}_{\text{MDS}}$, can correct a 2 erasure burst with a decoding delay 3.

obtain precise analytical results, they do not necessarily demonstrate the effectiveness of such codes in practice. For example, issues such as compression, error concealment, and the complicated nature of human perception of distortions need to be considered as well. Therefore, in order to demonstrate the application of these codes to increasing the quality of Voice over IP, we conducted various informal listening tests with some representative codes.

2) Hypothesis: The listening tests were designed to illustrate the benefits of coding over a fixed capacity bursty erasure channel. As illustrated by Fig. 5, for mild amounts of packet loss we expect little benefit from coding. As the loss rate increases, however, we expect the benefits of coding to become significant.

3) Procedure: To investigate the trade-off between channel coding and source coding rate allocation, we conducted listening tests using various levels of channel coding for an ADPCM coder [7]. In each test, the speech coder used an erasure concealment algorithm to mitigate the effect of lost packets. We simulated two channels: the Gilbert-Elliot Erasure Channel (GEEC) and a block loss channel shown in Figures 6 and 7 respectively.

In the Gilbert-Elliot Erasure Channel, packet losses always occur in the bad state (state 1), and never occur in the good state, (state 0). Transitions from good to bad or bad to good occur with probabilities α and β respectively. The probability of a packet loss given the previous packet was not lost is α, while the probability of a packet loss given the previous packet was lost is $1 - \beta$. Therefore when $1 - \beta$ is much greater than α the channel is bursty.

The block loss channel begins in the good state and the initial packet is not lost. Each time a packet is not lost there is a probability α that the next β packets will be lost. After the loss of β packets, the channel returns to the good state. Larger values of β correspond to larger levels of burstiness.

In each test, the total rate of the transmitted data is 64 kb/s. The original speech samples were quantized with 8 bit μ-law PCM, sampled at 8 kHz, and converted into 10 ms packets. The listening tests were conducted in two sessions over two days. For each channel condition and encoding format, the listeners were presented with 4 stimuli, each lasting 7-9 seconds. The 4 stimuli consisted of 2 male and 2 female speakers each reciting 2 sentences in a normal voice without noise. In each session, the first 20% of the stimuli consisted of an unscored training phase. After the training phase, the listeners were asked to score the

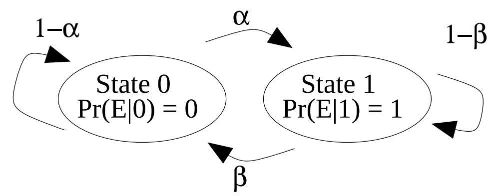

Fig. 6. In the Gilbert-Elliot Erasure Channel (GEEC) packet losses always occur in the bad state (state 1), and never occur in the good state, (state 0). Transitions from good to bad or bad to good occur with probabilities α and β respectively.

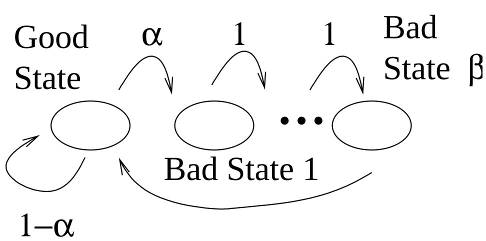

Fig. 7. Diagram of the block loss Channel. Packets losses never occur in the good state and always occur in the bad states. Each time a packet is transmitted and received correctly, the channel suffers a string of β losses with probability α. Thus losses always occur in bursts of β packets.

remaining samples according to the following range: 1 (bad), 2 (poor), 3 (fair), 4 (good), 5 (excellent).

We used four different encoding formats: no channel coding, a rate $3/4$ maximally short code similar to the rate $2/3$ code presented in Section II-C, the rate $1/2$ maximally short code in Fig. 1, and the rate $1/2$ maximally short code combined with a multidescriptive source coder. All of the coding formats, except for no channel coding, required a decoding delay of 6 packets which corresponds to 60 ms. Further details about the encoding formats are below:

- **No Channel Coding:** The entire 64 kb/s were allocated to the source coder. Unfortunately, the highest encoding rate supported by our encoder was 40 kb/s. Informal listening tests indicate that our 40 kb/s encoder performs almost as well as 8 bit μ-law PCM. Hence we used our 40 kb/s encoder as a proxy for a full 64 kb/s encoder. Despite this flaw we believe our results are still meaningful in understanding the qualitative behavior of channel coding in this scenario.
- **Rate 3/4 MS Code With $\lambda = 2$ Interleaving:** We allocated 48 kb/s to the source coder and used a rate 3/4 MS code with interleaving degree 2 as the channel code. As

in the previous format, we used our 40 kb/s encoder as a proxy for the proposed 48 kb/s encoder.

- **Rate 1/2 MS Code With $\lambda = 6$ Interleaving:** We allocated 32 kb/s to the source coder and used a rate 1/2 MS code (i.e. repetition coding) with interleaving degree 6 as the channel code.
- **Rate 1/2 Multiple Descriptions Code With $\lambda = 6$ Interleaving:** We described the source with a pair of 32 kb/s descriptions [7], transmitted in packets i and $i+6$. If either the description in packet i or packet $i+6$ is lost then a coarse reconstruction can be recovered from the remaining description. If neither packet is lost, then the two descriptions can be combined for a high quality description, [7].

4) Results: Tables I-VI show the mean opinion score (MOS) and standard deviation of the opinion scores (σ) for the informal listening tests on various channel conditions from mildest to most severe. The listening tests were conducted in two sessions. Tables II, III, IV, and VI were listened to in the first session and Tables I and V were listened to in the second session. The listeners for each test were Peter Kroon, Frank Baumgarte, Yair Ghitza, and Emin Martinian.

These results confirm the behavior we predicted in Fig. 5. In the case where no packet loss occurs, Table I shows that no channel coding produces the highest quality signal followed by the rate $3/4$ code and rate $1/2$ code. The multiple descriptions code also performs quite well since when no packet loss occurs, both descriptions can be combined for a high quality reproduction.

As the packet loss rate is increased in Table II, the performance of the uncoded stream drops precipitously while the rate $3/4$ and rate $1/2$ codes are mostly unaffected. The performance of the multidescriptive code also drops below the channel coded streams but not as far as the uncoded stream since the multidescriptive code still contains some redundancy to resist packet loss. This trend continues in Table III.

In Table IV, the packet loss rate increases enough that the rate $1/2$ coded stream has the highest quality. This trend continues in Tables V and VI. In addition, as the channel becomes more sever in Table VI, the multidescriptive code does better than the rate $3/4$ code.

IV. Conclusions

Motivated by the problem of real-time communication over packet networks, we discussed constructions of a number of new burst erasure correcting codes and analyzed their decoding delay. Citing a bound on the decoding delay developed in [2], we showed that these have the shortest possible decoding delay. In addition we presented results from informal listening tests using representative codes over various burst erasure channels. These results indicate that low delay coding provides significant benefits.

TABLE I

Listening tests with no erasures.

Encoding Format:	Loss Rate	MOS	σ
No Coding	0	3.75	0.75
Rate 3/4 MS Code	0	3.623	0.78
Rate 1/2 MS Code	0	3.19	0.53
Multiple Descriptions	0	3.69	0.46

TABLE II

Listening tests on GEEC with $\alpha = 0.05, \beta = 0.8$

Encoding Format:	Loss Rate	MOS	σ
No Coding	0.039	2.38	0.60
Rate 3/4 MS Code	0.004	3.56	0.70
Rate 1/2 MS Code	0.001	3.25	0.66
Multiple Descriptions	N/A	2.69	0.46

TABLE III

Listening tests on GEEC with $\alpha = 0.075, \beta = 0.85$

Encoding Format:	Loss Rate	MOS	σ
No Coding	0.065	2.06	0.56
Rate 3/4 MS Code	0.016	3.50	0.71
Rate 1/2 MS Code	0.005	3.19	0.73
Multiple Descriptions	N/A	2.56	0.70

TABLE IV

Listening tests on Block Loss Channel with $\alpha = 0.05, \beta = 2.$

Encoding Format:	Loss Rate	MOS	σ
No Coding	0.065	1.56	0.61
Rate 3/4 MS Code	0.011	2.88	0.78
Rate 1/2 MS Code	0	3.06	0.43
Multiple Descriptions	N/A	2.88	0.60

TABLE V

Listening tests on GEEC with $\alpha = 0.125, \beta = 0.75$

Encoding Format:	Loss Rate	MOS	σ
No Coding	0.105	1.44	0.61
Rate 3/4 MS Code	0.042	2.44	0.61
Rate 1/2 MS Code	0.011	2.88	0.70
Multiple Descriptions	N/A	2.19	0.53

TABLE VI

Listening tests on Block Loss Channel with $\alpha = 0.05, \beta = 6$

Encoding Format:	Loss Rate	MOS	σ
No Coding	0.167	1.06	0.24
Rate 3/4 MS Code	0.085	1.56	0.70
Rate 1/2 MS Code	0.007	2.75	0.66
Multiple Descriptions	N/A	2.38	0.70

References

[1] Vern Paxson, "End-to-end internet packet dynamics," *IEEE Transactions On Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

[2] Emin Martinian and Carl-Erik W. Sundberg, "Channel codes with low delay for burst erasure channels," submitted to the IEEE Transactions on Information Theory.

[3] G. D. Forney Jr., "Burst-correcting codes for the classic bursty channel," *IEEE Transactions on Communications Technology*, vol. COM-19, no. 5, pp. 772–781, October 1971.

[4] Shu Lin and Daniel J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1983.

[5] D. J. Goodman and C-E. W. Sundberg, "Combined source and channel coding for matching the speech transmission rate to the quality of the channel," *Bell Syst. Tech. J.*, vol. 62, no. 7, pp. 2017–2036, September 1983.

[6] Emin Martinian and Carl-Erik W. Sundberg, "Decreasing distortion using low delay codes for bursty packet loss channels," submitted to IEEE Transactions on Multimedia.

[7] Cheng-Chieh Lee, "Diversity control among multiple coders: A simple approach to multiple descriptions," in *2000 IEEE Workshop On Speech Coding*, Proceedings, Delevan, Wisconsin, September 2000, pp. 69–71.