Burst Erasure Correction Codes With Low Decoding Delay

Emin Martinian, *Member, IEEE,* and Carl-Erik W. Sundberg, *Fellow, IEEE*

***Abstract*— We present a new class of systematic, time-invariant, convolutional encoders suitable for low delay burst erasure correction. Specifically, we show that the new encoders have the shortest possible decoding delay required to correct all bursts of a given length with a fixed redundancy. By comparing the new encoders to Maximum Distance Separable (MDS) codes, we show that the latter generally require either more redundancy or more delay to correct bursts of a given length. In addition, we show that the new encoders can achieve better performance than MDS codes on a simple two-state Markov erasure channel. Thus, we demonstrate the advantages of using cross packet coding for delay sensitive applications such as Voice over IP, video-conferencing, etc., on bursty packet networks. Finally, we discuss suitable performance measures for encoders designed to correct both burst and random erasures and report the results of a computer search for such hybrid encoders.**

***Index Terms*— erasure channel, convolutional codes, maximally short codes, low delay coding,**

I. Introduction

Recent investigations suggest that, in a variety of networks, packet losses occur in bursts [3] [4] [5] [6] [7]. Conventional application of error correcting codes or retransmission (*e.g.*, ARQ) for packet recovery often requires interleaving and long decoder delays. Since long delays are usually unacceptable in real-time multimedia communication applications such as Voice over IP, video-conferencing, tele-medicine, etc., erasure correction codes with low decoding delays are desirable.

Specifically, consider a scenario where a source (*e.g.*, audio or video) must be transmitted over a packet channel (*e.g.*, the Internet). Specifically, imagine that every second one frame of the source consisting of k bits is provided to the encoder. Similarly, each second the transmitter can send a packet of n bits which is either received correctly at the receiver or erased (*e.g.*, due to congestion at an intervening Internet router). Due to delay constraints, the receiver must attempt to reproduce source packet i from the received packet stream with a delay of at most T packets. How should the transmitter encode each source packet into a channel packet such that the system is robust to bursts of packet losses?

Previous researchers have mostly focused on codes designed to maximize burst error correcting capability without regard to decoding delay (see [8] [9] [10] [11] [12] [13] [14] [15] [16] and references therein). An important metric in such work is the relationship between rate, correctable burst length, and guard space (the number of correctly received symbols required between bursts). In contrast, we consider codes for burst erasure correction and are primarily concerned with the relationship between rate, correctable burst length, and decoding delay.

Maximum Distance Separable (MDS) codes such as Reed-Solomon block codes, interleaved MDS codes and near-MDS codes [17] [18] have excellent burst correction capabilities but require long delays. Even though such codes have been proposed and implemented for various packet transmission applications, they do not generally achieve the best trade-off between delay, redundancy, and burst correction. The encoders presented in this paper are not only superior to MDS codes, but optimal in the sense that they achieve a lower bound on decoding delay.

We begin by discussing a bound relating the rate, correctable erasure burst length, and decoding delay in Section II. Next, in Section III, we present a construction yielding encoders which achieve this bound with equality. Specifically, we present a set of encoding rules yielding a class of time-invariant convolutional encoders which have the shortest possible decoding delay required to correct bursts of a given length. By comparing the proposed encoders to Maximum Distance Separable (MDS) codes in Section IV, we show that the new encoders are superior in the sense that they require a shorter decoding delay than MDS codes. Furthermore, we demonstrate that the new encoders can perform better on the simple two-state Markov channel model we refer to as the Gilbert Erasure Channel Model [19] [20]. In Section V, we discuss appropriate performance measures for hybrid encoders designed to correct both burst and random erasures. In addition, we present further encoders discovered via computer search. We close with some concluding remarks in Section VI.

II. Bounds

We are primarily interested in encoders for packet based applications. Specifically, we have in mind applications where at each time i, the transmitter is allowed to inject one packet of a fixed size into the channel. This packet is either received correctly or completely lost. Since most packet based systems typically include a sequence number for each packet we can consider such losses to be erasures. The goal of the encoder is to add redundancy to the transmitted packet stream to allow the decoder to recover from such losses. Furthermore the encoding should be done in such a way so that last packets can be recovered soon after the loss.

E. Martinian is with Mitsubishi Electric Research Labs and C-E. W. Sundberg is with SundComm. E-mail: emin@alum.mit.edu, cews@ieee.org. This work was initiated while both authors were with Lucent Technologies' Bell Labs and was presented in part at the International Conference on Communications in May 2002 [1] and at the International Symposium on Information Theory in Lausanne in July 2002 [2].

Since for packet applications both the loss and delay occur on a packet level and not a bit level we find it useful to consider codes defined over symbols with large alphabets (where each symbol in the code corresponds to a packet or fraction thereof if unequal size packets are desired.) Furthermore, instead of adding redundancy via bandwidth expansion (*i.e.*, increasing the number of symbols per time unit while keeping the symbol size fixed), we consider codes which add redundancy via packet expansion (*i.e.*, increasing the symbol size while keeping the number of symbols per time unit fixed).[1]

Specifically, we model the low delay packet transmission problem as follows. At each time i, a data source generates the packet of information

$$\vec{x}[i] = (x_0[i], x_1[i], ..., x_{k-1}[i]) \ ,$$

which contains k elements (*e.g.*, bits or bytes). An encoder causally maps $\vec{x}[i]$ along with previous packets into the coded packet

$$\vec{y}[i] = (y_0[i], y_1[i], ..., x_{n-1}[i]) \ ,$$

containing n elements. This mapping must be causal in the sense that $\vec{y}[i]$ can only depend on $\vec{x}[j]$ with $j \leq i$. The length n of the coded packet is typically specified by the underlying packet channel and cannot be changed while the source data length k is sometimes under the control of the system designer [24]. In any case, the rate of the code defined as $R = k/n$ has an important impact on the performance of the best possible code.

A burst of length B starting at time i is defined as the erasure (*i.e.*, loss) of one or more symbols from the set $\{\vec{y}[i], \vec{y}[i+1], ..., \vec{y}[i+B-1]\}$. If the earliest that $\vec{x}[i+j]$ (for $j \in \{0, 1, ..., B-1\}$) can be recovered due to such a burst is once $\vec{y}[i+B]$, $\vec{y}[i+B+1]$, ..., $\vec{y}[i+j+T]$ are received then the decoding delay required for recovering $\vec{x}[i+j]$ is T. Naturally, codes with lower rates contain more redundancy and should therefore be able to recover more quickly. This intuition leads to the following result.

Theorem 1: If a rate R encoder enables correction of all erasure bursts of length B with decoding delay at most T, then

$$T/B \geq \max\left[1, \frac{R}{(1-R)}\right]. \qquad (1)$$

Proof: Assume that the first B transmitted packets are lost in an erasure burst, and the next T packets are successfully received. At this point the decoder must be able to recover all lost packets. Therefore, if the next B packets are lost, they can be recovered once another T packets are successfully received. Repeating this pattern, we see that the decoder can completely recover from the alternating bursts in Fig. 1 each time T packets are received following B lost packets.

Since the ratio of received packets to transmitted packets is exactly $T/(T+B)$, the code rate must satisfy $R \leq T/(T+B)$ to ensure that no data is lost. Simple algebra shows that this condition is equivalent to $T/B \geq R/(1-R)$. Finally, note that regardless of the code rate, the causality restriction implies that no data can be recovered until at least one packet is successfully received following the burst. Therefore $T/B \geq 1$. Combining these lower bounds on T/B yields the desired result. ■

Theorem 1 is essentially a corollary of the asymptotic guard space bounds for finite-memory encoders [8] [25]. The guard space is the number of erasure free symbols both before and after a burst required to guarantee *eventual* recovery (*i.e.*, with unbounded decoding delay). Hence guard space is a looser requirement in the sense that encoders satisfying guard space bounds but not decoding delay bounds exist while any encoder which enables correction of bursts of length B with delay T allows correction of such bursts with guard space $G = T$ when infinite delay is allowed. As we will show briefly in the next section and in more detail in Section IV, Reed-Solomon block codes, interleaved Reed-Solomon codes [8], as well as other MDS and near MDS [17] [18] block codes achieve lower bounds on the required guard space, but do not achieve the lower bounds on decoding delay for a given rate and burst length.

III. Encoder Constructions

MDS codes (especially Reed-Solomon codes) have many good properties in correcting both erasures and bursts. Thus a natural question is whether such codes meet the decoding delay bound of Theorem 1. Imagine that we encode the 3 symbols $\vec{x}[0]$, $\vec{x}[1]$, and $\vec{x}[2]$ into the 5 symbols, $\vec{y}[0]$, $\vec{y}[1]$, . . ., $\vec{y}[4]$ using a Reed-Solomon block code. If the first two coded symbols, $\vec{y}[0]$ and $\vec{y}[1]$ are lost in a burst of length 2, $\vec{x}[0]$ can not be recovered until the remaining 3 coded symbols are received. Evidently, Reed-Solomon codes require a decoding delay of 4 to correct a burst of length 2 since $\vec{x}[0]$ is required only when $\vec{y}[4]$ is received.

But according to Theorem 1, the lower bound for the decoding delay of a rate $3/5$ code correcting a burst of length 2 is 3 and not 4. Is the bound in Theorem 1 loose, or are Reed-Solomon codes suboptimal? In this section we show the latter is true by constructing a family of codes which meet the lower bound on decoding delay for a wide range of rates.

Specifically, we construct a family of encoding rules $\mathcal{C}_{m,s}$ indexed by the integer parameters $m \geq 0$ and $s \geq 1$ having rate $R = (ms+1)/(ms+s+1)$. Since these encoders achieve the shortest decoding delay for a given burst length according to (1), we refer to the resulting codes as Maximally Short Codes.[2] Let the input for the encoder $\mathcal{C}_{m,s}$ consist of $(ms+1)$-tuples from the finite field $\mathrm{GF}(q)$ and let the output consist of $(ms+s+1)$-tuples from $\mathrm{GF}(q)$ with $q \geq 2s+ms$. Let $P\{u_1, u_2, ..., u_{ms+s}\}$ denote the s parity check symbols for a systematic, (n,k,d) = $(2s+ms, ms+s, s+1)$ Maximum Distance Separable encoder (*e.g.*, a Reed-Solomon encoder) with input $(u_1, u_2, ..., u_{ms+s})$. Then $\mathcal{C}_{m,s}$ is the rate

[1] Other authors have also considered encoders mapping inputs from one size alphabet to another. Such encoders are usually designed for use with orthogonal signaling or on partial-band interference channels [21] [22], or for multi-level codes [23] not low delay burst correction.

[2] Since decoding delay is an encoder property, Maximally Short Encoders might be technically more appropriate, but less pleasing to the ear.

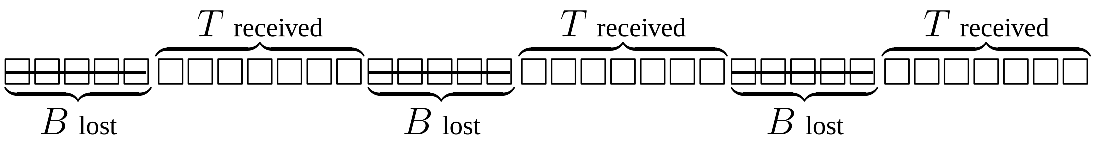

Fig. 1. An encoder enabling correction of all bursts of length B with decoding delay T must allow correction of the displayed alternating burst pattern.

$R = (ms+1)/(ms+s+1)$ encoder defined by the mapping

$$\vec{y}[i] = (x_0^{ms}[i], P\{x_0[i-1], x_0[i-2], ..., x_0[i-s], x_1^s[i-s-1], x_{s+1}^{2s}[i-2s-1], ..., x_{(m-1)s+1}^{ms}[i-ms-1]\}) \tag{2}$$

where $x_a^b[j]$ denotes $(x_a[j], x_{a+1}[j], ..., x_b[j])$. The rate $R = (ms+1)/(ms+s+1)$ encoder $\mathcal{C}_{m,s,\lambda}$ is obtained by periodically interleaving $\mathcal{C}_{m,s}$ by a factor of λ [16, chapter 14.3] to obtain

$$\vec{y}[i] = (x_0^{ms}[i], P\{x_0[i-\lambda], x_0[i-2\lambda], ..., x_0[i-s\lambda], x_1^s[i-(s+1)\lambda], x_{s+1}^{2s}[i-(2s+1)\lambda], ..., x_{(m-1)s+1}^{ms}[i-(ms+1)\lambda]\}). \tag{3}$$

As illustrated in Fig. 2, the encoding rule in (3) can be implemented using only delay elements and a systematic encoder for a Reed-Solomon code. Since delays are linear, time-invariant operations and the systematic Reed-Solomon encoder is a linear, memoryless transformation, Fig. 2 is in fact a linear, time-invariant, systematic encoder and $\mathcal{C}_{m,s,\lambda}$ is a time-invariant convolutional code [26] [27].

Before analyzing the general properties of such encoders we first consider some examples to illustrate the encoding and decoding process.

A. *A Rate 3/5 Example*

The rate $3/5$ encoder, $\mathcal{C}_{1,2}$, mapping a sequence of inputs consisting of 3-tuples to outputs consisting of 5-tuples, is illustrated in Table I. To show that this encoder enables correction of any burst of length 2 with decoding delay 3, imagine that $\vec{y}[i]$ and $\vec{y}[i+1]$ are lost. Decoding works as follows:

Time $i+2$: When $\vec{y}[i+2]$ is received, the decoder has $x_1^2[i-1]$ (since it was not lost in the burst) and $P\{x_0[i+1], x_0[i+0], x_1^2[i-1]\}$ (the 2 parity check symbols for the $(6,4,3)$ Reed-Solomon block code with input $x_0[i+1]$, $x_0[i]$, $x_1^2[i-1]$). Therefore the decoder recovers $x_0[i]$ and $x_0[i+1]$ at time $i+2$ by using a Reed-Solomon algorithm on $x_1[i-1]$, $x_2[i-1]$, and the 2 parity-check symbols $P\{x_0[i+1], x_0[i+0], x_1^2[i-1]\}$.

Time $i+3$: When $\vec{y}[i+3]$ is received, the decoder has received $x_0[i+2]$ (since it was not lost in the burst), $x_0[i+1]$ (which was recovered in the previous decoding step), and $P\{x_0[i+2], x_0[i+1], x_1^2[i]\}$ (the 2 parity check symbols for the $(6,4,3)$ code with input $x_0[i+2]$, $x_0[i+1]$, $x_1^2[i]$). Therefore the decoder recovers $x_1^2[i]$ at time $i+3$.

Time $i+4$: When $\vec{y}[i+4]$ is received, the decoder has received $x_0[i+3]$ and $x_0[i+2]$ (since they were not lost in the burst), and $P\{x_0[i+3], x_0[i+2], x_1^2[i+1]\}$. Therefore the decoder recovers $x_1^2[i+1]$ at time $i+4$.

Hence the decoder completely recovers $\vec{x}[i]$ at time $i+3$ and $\vec{x}[i+1]$ at time $i+4$ yielding a decoding delay of 3. Since the burst started at an arbitrary time, i, the encoder $\mathcal{C}_{1,2}$ allows correction of any length 2 burst with delay 3. According to the bound in (1), T/B must be at least $3/2$ for a rate 3/5 encoder. Thus, no rate 3/5 encoder can achieve a lower decoding delay than $\mathcal{C}_{1,2}$ to correct all length 2 bursts.

TABLE I

ENCODING EXAMPLE FOR THE ENCODER $\mathcal{C}_{1,2}$.

$$\begin{aligned}
\vec{y}[i-1] &= \left(x_0^2[i-1], P\{x_0[i-2], x_0[i-3], x_1^2[i-4]\}\right)\\
\vec{y}[i+0] &= \left(x_0^2[i+0], P\{x_0[i-1], x_0[i-2], x_1^2[i-3]\}\right)\\
\vec{y}[i+1] &= \left(x_0^2[i+1], P\{x_0[i+0], x_0[i-1], x_1^2[i-2]\}\right)\\
\vec{y}[i+2] &= \left(x_0^2[i+2], P\{x_0[i+1], x_0[i+0], x_1^2[i-1]\}\right)\\
\vec{y}[i+3] &= \left(x_0^2[i+3], P\{x_0[i+2], x_0[i+1], x_1^2[i+0]\}\right)\\
\vec{y}[i+4] &= \left(x_0^2[i+4], P\{x_0[i+3], x_0[i+2], x_1^2[i+1]\}\right)
\end{aligned}$$

B. *Rate $k/(k+1)$ Encoders*

For rates of the form $R = k/(k+1)$ (*i.e.*, $m = k-1$ and $s = 1$), the encoding rule in (3) takes on the particularly simple structure of a single parity check convolutional encoder:

$$\vec{y}[i] = \left(x_0^{ms}[i], \sum_{j=0}^{k-1} x_j[i-\lambda(j+1)]\right) \tag{4}$$

where additions are carried out in the appropriate finite field.

For example, Table II illustrates $\mathcal{C}_{1,1,2}$, the rate $2/3$ single parity check convolutional encoder with interleaving degree 2. If $\vec{y}[i]$ and $\vec{y}[i+1]$ are lost, successful decoding is accomplished via the parity check equations

$$\begin{aligned}
x_0[i] &= y_2[i+2] \oplus x_1[i-2] && \text{(5a)}\\
x_0[i+1] &= y_2[i+3] \oplus x_1[i-1] && \text{(5b)}\\
x_1[i] &= y_2[i+4] \oplus x_0[i+2] && \text{(5c)}\\
x_1[i+1] &= y_2[i+5] \oplus x_0[i+3], && \text{(5d)}
\end{aligned}$$

thus demonstrating that $\mathcal{C}_{1,1,2}$ allows recovery from a burst of 2 lost packets with decoding delay 4 meeting (1) with equality.

C. *Decoding Delay Analysis*

In general, maximum likelihood decoding using the Viterbi algorithm (with decisions on $\vec{x}[i]$ made at time $i+T$) minimizes the erasure probability subject to a decoding delay

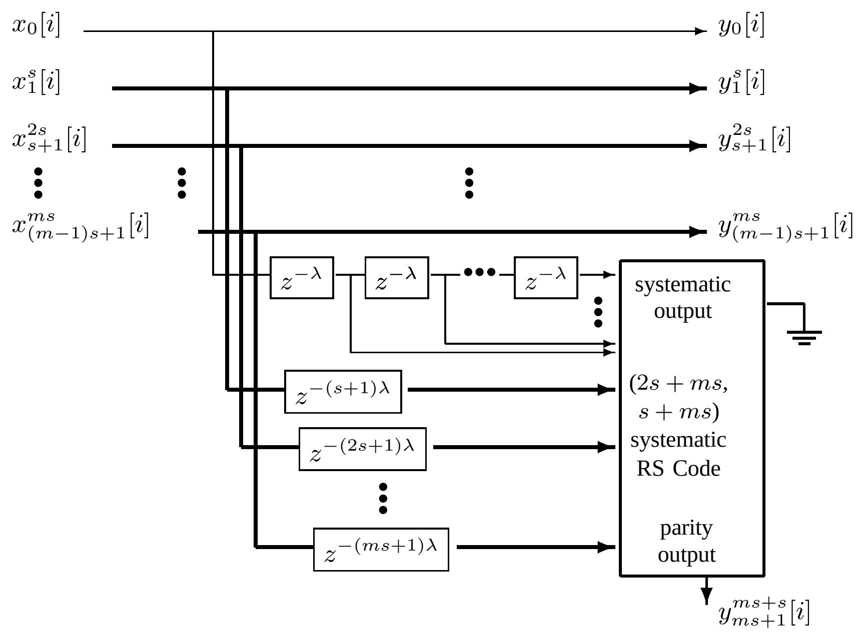

Fig. 2. Block diagram for the rate $R=(ms+1)/(ms+s+1)$ encoder, $\mathcal{C}_{m,s,\lambda}$, mapping the input symbol at time i, $\vec{x}[i]=(x_0[i],x_1[i],...,x_{ms}[i])$, to the output symbol at time i, $\vec{y}[i]=(y_0[i],y_1[i],...,y_{ms+s}[i])$. N unit delay elements are denoted via z^{-N} and the ground symbol indicates that the systematic output symbols from the Reed-Solomon encoder are not used. The thick bold lines represent s-tuples while the thin lines represent single values.

TABLE II

ENCODING EXAMPLE FOR THE ENCODER $\mathcal{C}_{1,1,2}$.

$\vec{y}[i-2]=\left(x_0^1[i-2],x_0[i-4]\oplus x_1[i-6]\right)$
$\vec{y}[i-1]=\left(x_0^1[i-1],x_0[i-3]\oplus x_1[i-5]\right)$
$\vec{y}[i+0]=\left(x_0^1[i+0],x_0[i-2]\oplus x_1[i-4]\right)$
$\vec{y}[i+1]=\left(x_0^1[i+1],x_0[i-1]\oplus x_1[i-3]\right)$
$\vec{y}[i+2]=\left(x_0^1[i+2],x_0[i+0]\oplus x_1[i-2]\right)$
$\vec{y}[i+3]=\left(x_0^1[i+3],x_0[i+1]\oplus x_1[i-1]\right)$
$\vec{y}[i+4]=\left(x_0^1[i+4],x_0[i+2]\oplus x_1[i+0]\right)$
$\vec{y}[i+5]=\left(x_0^1[i+5],x_0[i+3]\oplus x_1[i+1]\right)$

constraint of T symbols. Since such a decoder can be difficult to analyze, we consider a simple feedback decoding algorithm similar to the one used in Sections III-A and III-B. Specifically, for each $\vec{x}[i]$ lost in a burst of length s, recover $x_0[i]$ using the parity check terms in $\vec{y}[i+s]$ and previously decoded symbols and recover $x_{sj+1}^{(j+1)s}[i]$ using the parity check terms in $\vec{y}[(j+1)s+i+1]$ and previously decoded symbols.

Theorem 2: Using feedback decoding, the encoder $\mathcal{C}_{m,s}$ enables correction of any burst of s lost packets with decoding delay $ms+1$.

Note that the decoding delay versus burst length trade-off in Theorem 2 is optimal in the sense that it meets the bound in (1) with equality.

Proof: Without loss of generality, assume that all packets are successfully received until a burst of s packets are lost starting at time 0. Thus the information symbols which must be recovered are $\vec{x}[0]$ through $\vec{x}[s-1]$.

Note that the first symbol received after the burst,

$$\vec{y}[s]=(\vec{x}[s],P\{x_0[s-1],x_0[s-2],...,x_0[0],\\ x_1^s[-1],x_{s+1}^{2s}[-s-1],...,x_{(m-1)s+1}^{ms}[s-ms-1]\}),$$

contains s parity checks for the $(2s+ms,ms+s,s+1)$ Reed-Solomon code with input including $x_0[j]$ for $j\in\{0,1,...,s-1\}$ as well as ms terms which were correctly received before the burst. Therefore the $x_0[j]$ terms are recovered at time s using the Reed-Solomon decoding algorithm.

The second symbol received after the burst,

$$\vec{y}[s+1]=(\vec{x}[s+1],P\{x_0[s],x_0[s-1],...,x_0[1],\\ x_1^s[0],x_{s+1}^{2s}[-s],...,x_{(m-1)s+1}^{ms}[s-ms]\}),$$

contains the s parity checks for the Reed-Solomon code with input $x_1^s[0]$, $x_0[s]$ (which was received correctly after the burst), $x_0[1]$ through $x_0[s-1]$ (which were recovered in the preceding decoding step), and terms which were correctly received before the burst. Therefore $x_1^s[0]$ is recovered at time $s+1$. A similar argument shows that $x_1^s[1]$ through $x_1^s[s-1]$ can be recovered from $\vec{y}[s+2]$ through $\vec{y}[2s]$.

Next, $x_{s+1}^{2s}[0]$ is recovered using

$$\vec{y}[2s+1]=(\vec{x}[2s+1],P\{x_0[2s],x_0[2s-1],...,x_0[s+1],\\ x_1^s[s],x_{s+1}^{2s}[0],...,x_{(m-1)s+1}^{ms}[2s-ms]\}),$$

since the s terms $x_{s+1}^{2s}[0]$ were lost in the burst, while none of the terms

$$x_0[2s], x_0[2s-1], ..., x_0[s+1], \\ x_1^s[s], x_{2s+1}^{3s}[-s], ..., x_{(m-1)s+1}^{ms}[2s-ms]$$

were lost in the burst. Similarly, the only s terms in the parity check part of $\vec{y}[2s+2]$ which were lost in the burst are $x_{s+1}^{2s}[1]$. Therefore, $x_{s+1}^{2s}[1]$ is recovered at time $2s+2$. In fact, all the remaining symbols are recovered using this same procedure since for each $\vec{y}[(j+1)s+i+1]$, only the s terms corresponding to $x_{sj+1}^{(j+1)s}[i]$ were lost in the burst.

The decoding delay in recovering $\vec{x}[i]$ is determined by the delay in recovering the last part of each symbol, $x_{(m-1)s}^{ms}[i]$, from $\vec{y}[ms+i+1]$. Thus each $\vec{x}[i]$ lost in the burst of length s is recovered with decoding delay $ms+1$. Furthermore, once $ms+1$ packets after the burst are received the entire burst is corrected. ■

According to [16, chapter 14.3], degree λ periodic interleaving transforms an encoder allowing the correction of bursts of length B with guard space G into an encoder capable of correcting bursts of length λB with guard space λG. Periodic interleaving has a similar effect on the relationship between correctable burst length and decoding delay.

Corollary 2.1: Using feedback decoding, the encoder $\mathcal{C}_{m,s,\lambda}$ obtained by periodically interleaving $\mathcal{C}_{m,s}$ by λ can correct all bursts of length λs with delay $\lambda(ms+1)$.

Proof: The symbols for each of the λ deinterleaved streams, $\vec{y}[\lambda i + j]$, (where each stream corresponds to a fixed $j \in \{0, 1, ..., \lambda - 1\}$) are codewords of $\mathcal{C}_{m,s}$. Furthermore a burst of λs erasures effects only s symbols in each deinterleaved stream. Therefore, the decoding algorithm for $\mathcal{C}_{m,s}$ applied separately to each stream recovers $\vec{x}[\lambda i + j]$ with decoding delay $\lambda(ms+1)$. ■

IV. Comparison To Reed-Solomon Codes

We can gain some insight into the advantages of Maximally Short Codes obtained via (3) by comparing them to MDS block codes (*e.g.*, Reed-Solomon codes) [16] and near-MDS block codes (*e.g.*, Low Density Parity Check (LDPC) codes [17] [18]). While the latter can be decoded quickly in the sense that they require fewer computations than traditional MDS codes, for our purposes, LDPC codes are essentially equivalent to Reed-Solomon codes. This is because an (n, k) near MDS block code requires that at least roughly k of the n transmitted symbols are received before successful decoding can occur. Thus our comments regarding the decoding delay (in terms of packets received) for Reed-Solomon block codes apply equally well to LDPC codes.

A. Correctable Burst Length

MDS codes such as Reed-Solomon codes are inherently suboptimal in terms of decoding delay vs. correctable burst length due to their symmetric block structure. If enough symbols at the start of a block are lost, the first symbols in the block can not be recovered until the entire block is received. Specifically, to correct all bursts of length $n-k$, an (n, k) Reed-Solomon block code requires a decoding delay of $n-1$. To see this, assume the sequence $\vec{x}[0]$, $\vec{x}[1]$, ..., $\vec{x}[k]$ is encoded with an (n, k) Reed-Solomon code to form $\vec{y}[0]$, $\vec{y}[1]$, ..., $\vec{y}[n]$. If $\vec{x}[0]$, $\vec{x}[1]$, ..., $\vec{x}[n-k-1]$ are lost then decoding can not begin until the remaining symbols are correctly received. Thus, $\vec{x}[0]$ is recovered when $\vec{y}[n-1]$ is received. The ratio of decoding delay, T, to correctable burst length, B, for Reed-Solomon codes is [cf. (1)]

$$\frac{T}{B} = \frac{n-1}{n-k} \approx \frac{1}{1-R}. \tag{6}$$

Hence for $n \gg 1$, these codes require almost a factor of $1/R$ more decoding delay than Maximally Short Codes.

Note that, technically, the above use of a block code does not match the low delay packet transmission problem defined in Section II. Specifically, our problem model requires that at each time unit i the data source generates the information packet $\vec{x}[i]$ and the encoder produces the coded packet $\vec{y}[i]$. In contrast, block codes typically work in a "batch mode" where all the inputs are presented simultaneously to the encoder. This difference can be easily resolved by using periodic interleaving to turn a block code into an interleaved convolutional code matching our low delay packet transmission model and does not affect the conclusions about decoding delay.

For example, one way to employ an (n, k) RS code is to take each k-bit source packet $\vec{x}[i]$ and encode it with an (n, k) RS code to produce n coded bits. Clearly there is no point placing all n bits in the *same* coded packet since that achieves no robustness to the loss of the packet. Instead, the first coded bit is placed in $\vec{y}[i]$, the next coded bit is placed in $\vec{y}[i+1]$, and so on until the last coded bit is placed in packet $\vec{y}[i+n-1]$. As described in more detail by Forney [8], interleaving a block code in this manner has the effect of turning the block code into a convolutional code. For the purpose of analyzing robustness to erasures and decoding delay, however, we can still simply consider the properties of the block code. Specifically, convolutional code constructions combining B-by-N periodic interleavers with Reed-Solomon codes [8] also achieve the trade-off in (6).

B. The Gilbert Erasure Channel Model

The decoding delay vs. correctable burst length is a useful measure since it is not connected to any particular channel model. However, in order to gain some insight into the practical behavior of burst correcting codes we consider their performance on the Gilbert Erasure Channel model shown in Fig. 3. In this two state Markov model, erasures only occur in the bad state and never occur in the good state. Transitions from one state to another can occur each time a packet is transmitted as determined by the transition probabilities α, β.

The average packet loss probability for packets encoded using a Reed-Solomon code and transmitted over the Gilbert Erasure Channel can be determined using a straightforward calculation. Specifically, if we consider a systematic encoder

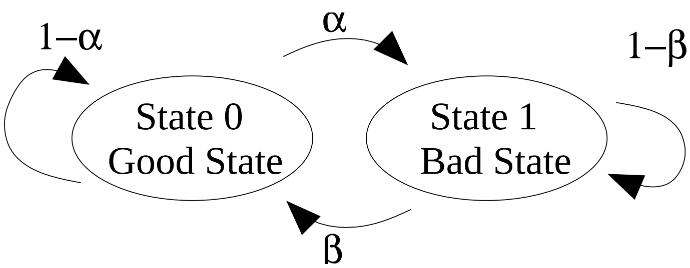

Fig. 3. The Gilbert Erasure Channel Model. Erasures always occur in the bad state, 1, and never occur in the good state, 0.

for an (n,k) Reed-Solomon code then

$$\Pr[\vec{x}[i] \text{ lost }] = \sum_{j=n-k}^{n-1} \Pr[\{\vec{y}[i] \text{ lost}\} \cap \{j \text{ more packets lost}\}]. \tag{7}$$

The expected packet loss rate for a Reed-Solomon code is denoted as $\mathrm{PLP}_{\mathrm{RS}}(n,k)$ and can be determined by averaging (7) over i

$$\mathrm{PLP}_{\mathrm{RS}}(n,k) = \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=n-k}^{n-1} \Pr[\{\vec{y}[i] \text{ lost}\} \cap \{j \text{ more packets lost}\}]. \tag{8}$$

The corresponding expected packet loss rate for the encoder $\mathcal{C}_{m,s,\lambda}$ is derived in the Appendix.

Figures 4 and 5 compare the expected packet loss probability for Reed-Solomon Codes and Maximally Short Codes as a function of rate. The channels considered have (ϵ,ρ) values of $(10^{-3},10^{3})$, $(10^{-2},10^{2})$, and $(5\cdot10^{-2},20)$ where $\epsilon=\alpha/(\alpha+\beta)$ represents the uncoded packet loss rate and $\rho=(1-\beta)/\alpha$ represents a measure of burstiness.[3] When a decoding delay of 6 symbols is allowed (Fig. 4), we see that Maximally Short Codes can achieve the same packet loss probability as Reed-Solomon Codes, but with higher rate (*i.e.*, less redundancy). When a longer decoding delay of 12 symbols is allowed (Fig. 5), Reed-Solomon Codes outperform Maximally Short Codes for the $(\epsilon,\rho)=(5\cdot10^{-2},20)$ channel, but Maximally Short Codes continue to prove superior for the more bursty channels. This occurs because, for certain channel parameters, significant losses occur due to non-bursty erasure patterns. Due to their superior distance profile, the Reed-Solomon Codes are able to recover from random losses which the Maximally Short Codes cannot. This suggests a hybrid encoder combining the superior burst correcting capabilities of Maximally Short Codes with the good distance properties of Reed-Solomon Codes might prove useful in practice.

In [24], we investigate the use of Maximally Short Codes in transmitting a Gaussian source over a Gilbert Erasure Channel. We show that the rate advantage obtained by Maximally Short Codes in Figures 4 and 5 translates into significant reductions in the average mean square distortion relative to systems employing Reed-Solomon Codes.

[3]The latter two channel conditions roughly correspond to measurements reported for wireless networks [28] and the Internet [7].

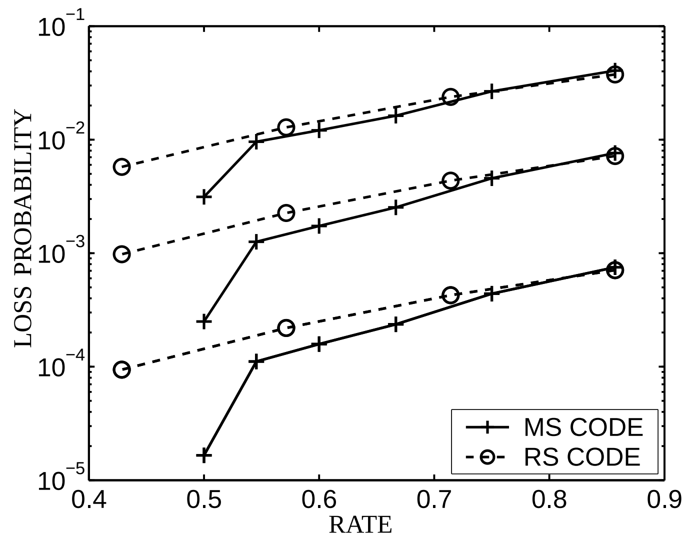

Fig. 4. Comparison of packet loss rates for Reed-Solomon Codes (broken lines) and Maximally Short Codes (solid lines) on the Gilbert Erasure Channel. The channel conditions for each pair correspond to (ϵ,ρ) values of $(10^{-3},10^{3})$, $(10^{-2},10^{2})$, and $(5\cdot10^{-2},20)$ from bottom to top where ϵ represents the uncoded packet loss rate and ρ represents the ratio of packet loss probabilities conditioned on whether the previous packet was lost or received. All codes require a decoding delay of 6 symbols.

V. Computer Encoder Search

In order to analyze the trade-off between burst and random erasure correcting capability, we consider a computer search. The key parameters of a code which determine burst and random erasure correcting capability without a delay constraint are the minimum span (*i.e.*, the minimum number of consecutive positions outside of which two codewords are the same) [8] and free distance [26] [16]. When a decoding delay constraint of T symbols is imposed, the relevant encoder parameters are the column distance [26] and the column span introduced in Section V-A.

A. Delay Constrained Performance Measures

An encoder has column distance d_j if every pair of input sequences which differ in the ith position results in output sequences which differ in at least d_j positions from time i to j. For such an encoder, the ith input symbol can be deduced from the output with delay at most $j-1$ provided that less than d_j erasures occur. This follows since even after the erasures, sequences with differing inputs at time i differ in at least one output position before time j.

An encoder has column span s_j if every pair of input sequences which differ in the ith position results in output sequences which differ in at least two positions s_j-2 symbols apart between time i to j. For such an encoder, the ith input symbol can be deduced from the output with delay at most $j-1$ provided that a burst of less than s_j consecutive erasures occurs. This follows since even after the erasure burst, two sequences with differing inputs at time i differ in at least one output position before time j.

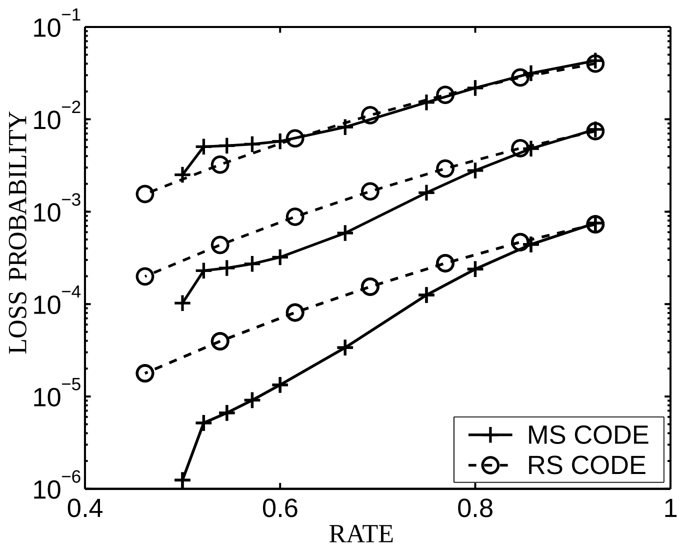

Fig. 5. Comparison of packet loss rates for Reed-Solomon Codes (broken lines) and Maximally Short Codes (solid lines) on the Gilbert Erasure Channel. The channel conditions for each pair correspond to (ϵ, ρ) values of $(10^{-3}, 10^{3})$, $(10^{-2}, 10^{2})$, and $(5\cdot 10^{-2}, 20)$ from bottom to top where ϵ represents the uncoded packet loss rate and ρ represents the ratio of packet loss probabilities conditioned on whether the previous packet was lost or received. All codes require a decoding delay of 12 symbols.

For a time-invariant convolutional encoder, it is well-known that the column distance can be determined by examining the column weights of the codewords. This fact leads to various fast algorithms for determining distance properties [29]. Similarly, for such an encoder, the column span can be determined by examining the column span of the codewords. Specifically, for a codeword, $\vec{c}$, corresponding to an input sequence where the first non-zero input occurs at time i, define the column span of the codeword $s_j(\vec{c})$ as two greater than the number of symbols between the first non-zero symbol after time $i-1$ and before time $j+1$. The column span of the encoder is the minimum column span of such codewords.

For example, consider the $\mathcal{C}_{1,1,2}$ encoder in Table II. Let $\vec{x}[0]$ be the first non-zero term in the input sequence $\vec{x}[i]$. Then $\vec{y}[0]$ is non-zero and so is either $\vec{y}[2]$ or $\vec{y}[4]$. Therefore $s_4(\vec{y}) = 3$ which (since $\mathcal{C}_{1,1,2}$ is linear and time-invariant) implies that the column span of the encoder is $s_4(\mathcal{C}_{1,1,2}) = 3$. Hence, any burst of 2 erasures can be corrected with delay 4 because the burst can only erase one of the two non-zero symbols in any non-zero codeword. Note that if $\vec{x}[0] = (0,1)$, the column weight of the corresponding codeword is $d_4(\vec{c}) = 2$. Hence there exists a pattern of two non-consecutive erasures which results in data loss.

B. *Encoder Structure*

To obtain an encoder decodable with delay T, we consider rate k/n, memory T encoders which map input sequences consisting of k-bit symbols to n-bit symbols. Such encoders can be easily implemented using k shift registers each of length n bits. The resulting generator matrix representation $\vec{y} = \vec{x}G$ [21], [26], [16], [30] specifies how the output bits depend on the contents of the shift registers [30]. The entry in row i and column j of G is denoted $g_{ij}[\tau]$ and indicates whether the input bit x_i from τ units in the past affects the current value of y_j. Specifically, the output symbol at time t, $\vec{y}[t] = (y_1[t], y_1[t], ..., y_n[t])$, is obtained from the current input symbol, $\vec{x}[t] = (x_1[t], x_1[t], ..., x_k[t])$, and the T preceding input symbols according to the convolution formula

$$y_j[t] = \sum_{i=1}^{k}\sum_{\tau=0}^{T} g_{ij}[\tau]x_i[t-\tau],\ j \in \{1,2,...,n\} \qquad (9)$$

where all additions are performed modulo 2. In the equivalent D-transform notation, (9) can be represented via

$$y_j(D) = \sum_{i=1}^{k} g_{ij}(D)x_i(D),\ j \in \{1,2,...,n\}. \qquad (10)$$

Tables III, IV, and V list the results of an exhaustive search of systematic encoders designed to maximize the column distance provided that the column span is at least s_T for various rates. [4] The entry in row i and column j of G corresponds to $g_{ij}[\tau]$ which is a binary vector. To conserve space, we represent these vectors in octal. For example, the rate $2/3$ encoder with $T = 9$, $s_T = 5$, $d_T = 3$ in Table IV has the row vector $[0000100100]$ as the third column of the first row which corresponds to $g_{13}[\tau] = \delta[\tau-4] + \delta[\tau-7]$. The equivalent generator polynomial for g_{13} is $D^4 + D^7$.

The rate $1/2$ encoders listed in Table III follow an interesting pattern: $s_T + d_T = T + 3$. Evidently, there is a direct trade-off between the burst and random erasure correcting capability. Also, encoders which can correct the longest possible bursts (*i.e.*, those meeting the bound in (1) with equality) all have $d_T = 2$. The rate $2/3$ encoders in Table IV demonstrate a similar trade-off. Furthermore, all the rate $2/3$ encoders meeting the bound in (1) with equality also have $d_T = 2$. Finally, we note that while the rate $1/3$ encoders in Table V demonstrate a trade-off between s_T and d_T, even encoders which meet the bound in (1) have $d_T > 2$.

VI. Concluding Remarks

In this work we present a class of time-invariant convolutional codes with systematic encoders requiring the shortest possible decoding delay to correct erasure bursts of a given length. We compare the new encoders to Reed-Solomon codes and show that the latter generally require longer delays or more redundancy to achieve comparable performance at a given decoding delay. Finally, we discuss performance measures for hybrid encoders designed to correct both burst and random erasures and report various encoders obtained through computer search.

Due to the growing demands for real-time applications in packet networks, techniques for low delay, reliable communications in the presence of bursts are becoming increasingly desirable. While Maximally Short Codes represent an attractive solution, many opportunities for future work remain. On the practical side, fast algorithms for evaluating the column

[4] Since it is always the case that $s_T \geq d_T$, encoders with $s_T = d_T$ are not shown.

TABLE III

RATE 1/2 CODES.

G	T	s_T	d_T
$\begin{pmatrix} 20 & 3 \end{pmatrix}_8$	4	4	3
$\begin{pmatrix} 20 & 1 \end{pmatrix}_8$	4	5	2
$\begin{pmatrix} 40 & 3 \end{pmatrix}_8$	5	5	3
$\begin{pmatrix} 40 & 1 \end{pmatrix}_8$	5	6	2
$\begin{pmatrix} 100 & 32 \end{pmatrix}_8$	6	5	4
$\begin{pmatrix} 100 & 3 \end{pmatrix}_8$	6	6	3
$\begin{pmatrix} 100 & 1 \end{pmatrix}_8$	6	7	2
$\begin{pmatrix} 200 & 67 \end{pmatrix}_8$	7	6	4
$\begin{pmatrix} 200 & 3 \end{pmatrix}_8$	7	7	3
$\begin{pmatrix} 200 & 1 \end{pmatrix}_8$	7	8	2
$\begin{pmatrix} 400 & 270 \end{pmatrix}_8$	8	6	5
$\begin{pmatrix} 400 & 154 \end{pmatrix}_8$	8	7	4
$\begin{pmatrix} 400 & 3 \end{pmatrix}_8$	8	8	3
$\begin{pmatrix} 400 & 1 \end{pmatrix}_8$	8	9	2
$\begin{pmatrix} 1000 & 315 \end{pmatrix}_8$	9	7	5
$\begin{pmatrix} 1000 & 332 \end{pmatrix}_8$	9	8	4
$\begin{pmatrix} 1000 & 3 \end{pmatrix}_8$	9	9	3
$\begin{pmatrix} 1000 & 1 \end{pmatrix}_8$	9	10	2
$\begin{pmatrix} 2000 & 473 \end{pmatrix}_8$	10	8	5
$\begin{pmatrix} 2000 & 667 \end{pmatrix}_8$	10	9	4
$\begin{pmatrix} 2000 & 3 \end{pmatrix}_8$	10	10	3
$\begin{pmatrix} 2000 & 1 \end{pmatrix}_8$	10	11	2

TABLE IV

RATE 2/3 CODES.

G	T	s_T	d_T
$\begin{pmatrix} 20 & 0 & 4 \\ 0 & 20 & 1 \end{pmatrix}_8$	4	3	2
$\begin{pmatrix} 100 & 0 & 10 \\ 0 & 100 & 1 \end{pmatrix}_8$	6	4	2
$\begin{pmatrix} 200 & 0 & 24 \\ 0 & 200 & 3 \end{pmatrix}_8$	7	4	3
$\begin{pmatrix} 400 & 0 & 20 \\ 0 & 400 & 1 \end{pmatrix}_8$	8	5	2
$\begin{pmatrix} 1000 & 0 & 44 \\ 0 & 1000 & 3 \end{pmatrix}_8$	9	5	3
$\begin{pmatrix} 2000 & 0 & 1737 \\ 0 & 2000 & 54 \end{pmatrix}_8$	10	5	4
$\begin{pmatrix} 2000 & 0 & 40 \\ 0 & 2000 & 1 \end{pmatrix}_8$	10	6	2

TABLE V

RATE 1/3 CODES.

G	T	s_T	d_T
$\begin{pmatrix} 20 & 6 & 1 \end{pmatrix}_8$	4	5	4
$\begin{pmatrix} 40 & 6 & 1 \end{pmatrix}_8$	5	6	4
$\begin{pmatrix} 100 & 56 & 1 \end{pmatrix}_8$	6	7	5
$\begin{pmatrix} 200 & 146 & 24 \end{pmatrix}_8$	7	7	6
$\begin{pmatrix} 200 & 46 & 1 \end{pmatrix}_8$	7	8	5
$\begin{pmatrix} 400 & 354 & 2 \end{pmatrix}_8$	8	8	6
$\begin{pmatrix} 400 & 26 & 1 \end{pmatrix}_8$	8	9	5
$\begin{pmatrix} 1000 & 603 & 51 \end{pmatrix}_8$	9	8	7
$\begin{pmatrix} 1000 & 474 & 1 \end{pmatrix}_8$	9	10	6
$\begin{pmatrix} 2000 & 1026 & 56 \end{pmatrix}_8$	10	10	7
$\begin{pmatrix} 2000 & 476 & 1 \end{pmatrix}_8$	10	11	6

span would allow for extensive computer searches hopefully yielding hybrid encoders with good burst and random erasure correction capabilities. On the theoretical side, a characterization of the trade-off between the burst and random erasure correction capability of an encoder would be useful both in measuring the success of computer code searches as well as possibly suggesting further analytic constructions. Some of these topics are explored in [31].

ACKNOWLEDGMENTS

The authors would like to thank the members of the Speech and Audio Communications group at Bell Labs, Lucent Technologies for their help, encouragement and advice. The authors would also like to thank the anonymous reviewers for their care and patience in reading the initial manuscript; their many valuable suggestions greatly improved the paper.

APPENDIX

To compute the symbol loss probabilities for Maximally Short Codes, we consider the simple feedback decoding algorithm in the proof of Theorem 2 (further erasure patterns may exist which are correctable if maximum likelihood decoding is used or if decoding delay longer than $T = ms+1$ is allowed). Let $\mathcal{L}_i$ denote the event that the ith packet is lost. Maximally Short Codes with $m = 0$ are rate $1/(s+1)$ repetition codes and hence $\Pr[\mathcal{L}_i]$ can be easily computed. For more complicated structures, we determine $\mathcal{L}_i$ by decomposing its complement, $\mathcal{L}_i^c$, into the mutually exclusive events $\{\mathcal{R}_{j,i}\}_{j=0}^{s}$, whence

$$\Pr[\mathcal{L}_i] = 1 - \sum_{j=0}^{s} \Pr[\mathcal{R}_{j,i}]. \qquad (11)$$

A. Loss Probability for $s \leq \frac{ms}{2} + \frac{1}{2}$

In the following analysis, we represent a state sequence for the Gilbert Erasure Channel as $\vec{e}$ and a subsequence from time i to time j as e_i^j. Also, we use a^b to denote the string a repeated b times (or the empty string if $b \leq 0$), and we use $a^{(b>c)}$ to denote the string $\{a\}$ when $b > c$ and the empty string otherwise.

First, let $\mathcal{R}_{0,i} = \{e_i = 0\}$ denote the event that the ith symbol is correctly received. Next, for any $1 \leq j \leq s$, let

$$\mathcal{R}_{j,i} = \{e_{i-j-ms+s}^{i+ms+1} = 0^{ms+1-s}, 1, *^{j-2}, 1^{(j>1)}, *^{s-j}, 0^{ms+1-s+j}\}, j \in [1, s] \qquad (12)$$

denote the event that the ith symbol is the jth loss in a burst of at most s symbols starting at time $i - j + 1$, preceded by $ms + 1 - s$ correctly received symbols and followed by $ms+1-s+j$ correctly received symbols. By construction, such a burst is correctable using feedback decoding as discussed in Theorem 2.

Theorem 3: When $s \leq \frac{ms}{2} + \frac{1}{2}$, the events $\{\mathcal{R}_{j,i}\}_{j=0}^{s}$ are mutually exclusive.

Proof: Showing that $\mathcal{R}_{j,i}$ and $\mathcal{R}_{j',i}$ are mutually exclusive requires showing that the corresponding erasure patterns, $\vec{e}(j)$ and $\vec{e}(j')$ are mutually exclusive. The event $\mathcal{R}_{0,i}$ is clearly exclusive of the other events since all events except $\mathcal{R}_{0,i}$

TABLE VI
USEFUL FORMULAS FOR THE GILBERT ERASURE CHANNEL.

Symbol	Meaning	Formula
$p_{0\to 0}(L)$	$\Pr(e_{i+L}=0\|e_i=0)$	$\frac{\alpha}{\alpha+\beta}\left[\frac{\beta}{\alpha}+(1-\beta-\alpha)^L\right]$
$p_{1\to 1}(L)$	$\Pr(e_{i+L}=1\|e_i=1)$	$\frac{\alpha}{\alpha+\beta}\left[1+\frac{\beta}{\alpha}(1-\beta-\alpha)^L\right]$
$p_{1\to 0}(L)$	$\Pr(e_{i+L}=0\|e_i=1)$	$1-p_{1\to 1}(L)$
$p_{0\to 1}(L)$	$\Pr(e_{i+L}=1\|e_i=0)$	$1-p_{0\to 0}(L)$
ϵ	$\Pr(e_i=1)$	$\frac{\alpha}{\alpha+\beta}$

have the ith symbol being erased. To show the other events are mutually exclusive we note that according to (12), when $s \geq j' > j > 0$,

$$e_{i-j-ms+s}^{i-j'+1}(j) = 0^{ms+2-s+j-j'}$$
$$e_{i-j-ms+s}^{i-j'+1}(j') = 0^{ms-s+1+j-j'}1 \quad (13)$$

Note that $\mathcal{R}_{j,i}$ requires that at least $ms+2-s+j-j'$ symbols before symbol i were not erased while $\mathcal{R}_{j',i}$ requires at least one of these symbols was erased. Thus provided that $ms+1-s \geq j'-j$ (*i.e.*, $\mathcal{R}_{j,i}$ requires at least one unerased symbol before time i), these events are mutually exclusive. Since $j'-j \leq s$ by assumption, $ms+1-s \geq s$ (or equivalently $s \leq ms/2+1/2$) guarantees that $\mathcal{R}_{j,i}$ and $\mathcal{R}_{j',i}$ are mutually exclusive. ■

The probabilities of the events, $\mathcal{R}_{j,i}$, can be determined by computing the probabilities of the state sequences in (12). Due to the Markov structure of the channel, this probability can be written completely in terms of the probability of going from state a to b in L steps (denoted by $p_{a\to b}(L)$) and the steady state loss probability (denoted by ϵ):

$$\Pr[\mathcal{R}_{j,i}] = (1-\epsilon)\cdot p_{0\to 0}(1)^{ms-s}\cdot p_{0\to 1}(1)\cdot p_{1\to 1}(j-1)\cdot p_{1\to 0}(s-j+1)\cdot p_{0\to 0}(1)^{ms-s+j}, j\in[1,s] \quad (14)$$

and $\Pr[\mathcal{R}_{0,i}] = 1-\epsilon$. Table VI lists the formulas required for (14).

For the interleaved encoder, $\mathcal{C}_{m,s,\lambda}$, the symbol loss probability can be computed by noting that interleaving transforms the base channel into λ parallel channels each with a separate $\mathcal{C}_{m,s}$ encoder [16, chapter 14.3]. Therefore each term a in (12) is replaced by the term $(a*)^{\lambda-1}$ to yield

$$\mathcal{R}_{j,i} = \left\{e_{i-\lambda(j+ms-s)}^{i+\lambda(ms+1)} = \left(0*^{\lambda-1}\right)^{ms+1-s}, \left(1*^{\lambda-1}\right), \left(*^{\lambda}\right)^{(j-2)}, \left(1*^{\lambda-1}\right)^{(j>1)}, \left(*^{\lambda}\right)^{(s-j)}, \left(0*^{\lambda-1}\right)^{ms+1-s+j}\right\} \quad (15)$$

for $j\in[1,s]$ with the corresponding probabilities

$$\Pr[\mathcal{R}_{j,i}] = (1-\epsilon)\cdot p_{0\to 0}(\lambda)^{ms-s}\cdot p_{0\to 1}(\lambda)\cdot p_{1\to 1}(\lambda(j-1))\cdot p_{1\to 0}(\lambda(s-j+1))\cdot p_{0\to 0}(\lambda)^{ms-s+j}, j\in[1,s]. \quad (16)$$

The probability that the ith symbol is lost is then obtained via (11).

B. Loss Probability for $m=1, s>0$

If the definition in Section A is used, the events $\{\mathcal{R}_{j,i}\}_{j=0}^{s}$ will not always be mutually exclusive when $m=1$ and $s>0$. For example, in the case where $m=1, s=4, R=\frac{5}{9}$, the events $\mathcal{R}_{1,i}$ and $\mathcal{R}_{4,i}$ are not mutually exclusive as shown in Table VII. Hence for $j\in[1,s]$, we instead define $\mathcal{R}_{j,i}$ as the event where the ith symbol is the jth symbol lost in a burst of length between j and s:

$$\mathcal{R}_{j,i} = \{e_{i-j}^{i+s+1} = 0, 1^j, *^{s-j}, 0^{1+j}\}, j\in[1,s]. \quad (17)$$

Such bursts are correctable using the feedback decoding rule in Theorem 2. Furthermore we retain the definition $R_{0,i} = \{e_i=0\}$ as before.

Theorem 4: The events in (17) are mutually exclusive for all j, j' such that $s \geq j' > j \geq 0$.

Proof: For $s \geq j' > j \geq 0$, (17) implies $e_{i-j}=0$ for $\mathcal{R}_{j,i}$ while $\mathcal{R}_{j',i}$ has $e_{i-j}=1$. Thus the events $\mathcal{R}_{j,i}$ and $\mathcal{R}_{j',i}$ differ in whether the symbol at time $i-j$ was lost and are therefore mutually exclusive. Furthermore, $\mathcal{R}_{0,i}$ is exclusive of the other events since all events except $\mathcal{R}_{0,i}$ have the ith symbol being lost. ■

Thus, to compute the expected erasure probability we evaluate $\Pr[\mathcal{R}_{j,i}]$ to obtain

$$\Pr[\mathcal{R}_{j,i}] = (1-\epsilon)\cdot p_{0\to 1}(1)\cdot p_{1\to 1}(1)^{j-1}\cdot p_{1\to 0}(s-j+1)\cdot p_{0\to 0}(1)^j \quad (18)$$

and $\Pr[\mathcal{R}_{0,i}] = 1-\epsilon$.

For the interleaved encoder, $\mathcal{C}_{m,s,\lambda}$, the relevant event definitions and probabilities are

$$\mathcal{R}_{j,i} = \left\{e_{i-\lambda j}^{i+\lambda(s+1)} = \left(0*^{\lambda-1}\right), \left(1*^{\lambda-1}\right)^j, *^{\lambda(s-j)}, \left(0*^{\lambda-1}\right)^{1+j}\right\}, j\in[1,s]. \quad (19)$$

and

$$\Pr[\mathcal{R}_{j,i}] = (1-\epsilon)\cdot p_{0\to 1}(\lambda)\cdot p_{1\to 1}(\lambda)^{j-1}\cdot p_{1\to 0}(\lambda(s-j+1))\cdot p_{0\to 0}(\lambda)^j. \quad (20)$$

The probability that the ith symbol is lost is then obtained via (11).

C. Loss Probabilities for $m=0, s>0$

The encoders $\mathcal{C}_{0,s,\lambda}$ correspond to interleaved repetition codes with rate $1/s$. Therefore, information symbol i is lost if and only if symbols $i, i+\lambda, i+2\lambda, ..., i+s\lambda$ are lost (i.e. $e_i^{i+\lambda s} = \{1*^{\lambda-1}\}^{s+1}$). Hence the expected loss probability is

$$\Pr[\mathcal{L}_i] = \Pr\left[e_i^{i+\lambda s} = \{1*^{\lambda-1}\}^{s+1}\right] = \epsilon\cdot p_{1\leftarrow 1}(\lambda)^s. \quad (21)$$

Using (21) for the $m=0$ case, (20) for the $m=1$ case, and (16) for the $m\geq 2$ case it is possible to determine the loss probability for any encoder $\mathcal{C}_{m,s,\lambda}$.

TABLE VII
THE EVENTS $\mathcal{R}_{1,i}$ AND $\mathcal{R}_{4,i}$ ARE NOT MUTUALLY EXCLUSIVE FOR THE ENCODER $\mathcal{C}_{1,4}$ BECAUSE THE ERASURE PATTERN FOR $\mathcal{R}_{1,i}$ IS A SUBSTRING OF THE ERASURE PATTERN FOR $\mathcal{R}_{4,i}$.

	e_{i-4}	e_{i-3}	e_{i-2}	e_{i-1}	e_i	e_{i+1}	e_{i+2}	e_{i+3}	e_{i+4}	e_{i+5}	
$\mathcal{R}_{1,i} = \{e_{i-4}^{i+5} =$	$*$	$*$	$*$	0	1	$*$	$*$	$*$	0	0	$\}$
$\mathcal{R}_{4,i} = \{e_{i-4}^{i+5} =$	0	1	$*$	$*$	1	0	0	0	0	0	$\}$

REFERENCES

[1] E. Martinian and C.-E. Sundberg, "Low delay burst erasure correction codes," in *Proc. International Conference on Communication*, vol. 3, 2002, pp. 1736–1740.

[2] ——, "Low delay burst erasure correcting codes for packet transmission," in *Proc. International Symposium on Information Theory*, Lausanne, Switzerland, 2002.

[3] J. Bolot, "Characterizing end-to-end packet delay and loss in the internet," *Journal of High Speed Networks*, pp. 289–298, September 1993.

[4] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proceedings of The Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, 1999, pp. 345–352.

[5] M. S. Borella, D. Swider, S. Uludag, and G. B. Brewster, "Internet packet loss: measurement and implications for end-to-end QoS," in *Proceedings of the 1998 ICPP Workshops on Architectural and OS Support for Multimedia Applications/Flexible Communications Systems/Wireless Networks and Mobile Computing*, 1998, pp. 3–12.

[6] M. Arai, A. Chiba, and K. Iwasaki, "Measrurement and modeling of burst packet losses in internet end-to-end communications," in *Proceedings of the 1999 Pacific Rim International Symposium on Dependable Computing*, 1999, pp. 260–267.

[7] V. Paxson, "End-to-end internet packet dynamics," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

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

[9] J. L. Massey, "Implementation of burst-correcting convolutional codes," *IEEE Trans. Inform. Theory*, pp. 416–422, July 1965.

[10] A. D. Wyner and R. Ash, "Analysis of recurrent codes," *IEEE Trans. Inform. Theory*, pp. 143–156, July 1963.

[11] Y. Iwadare, "A class of high-speed decodable burst-correcting codes," *IEEE Trans. Inform. Theory*, pp. 817–821, November 1972.

[12] ——, "On type-b1 burst-error-correcting convolutional codes," *IEEE Trans. Inform. Theory*, pp. 577–583, July 1968.

[13] E. R. Berlekamp, "Note on recurrent codes," *IEEE Trans. Inform. Theory*, pp. 257–258, July 1964.

[14] F. P. Preparata, "Systematic construction of optimimum linear recurrent codes for burst error correction," *Calcolo*, vol. 1, pp. 147–153, April 1964.

[15] R. T. Chien, "Burst-correcting codes with high-speed decoding," *IEEE Trans. Inform. Theory*, vol. 15, no. 1, pp. 109–113, January 1969.

[16] S. Lin and D. J. C. Jr., *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[17] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM 1998*, Vancouver Canada, September 1998, pp. 56–67.

[18] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Effi cient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 569–584, February 2001.

[19] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253–1265, September 1960.

[20] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Systems Technical Journal*, vol. 42, pp. 1977–1997, September 1963.

[21] J.-J. Chang, D.-J. Hwang, and M.-C. Lin, "Some extended results on the search for good convolutional codes," *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp. 1682–1697, September 1997.

[22] W. Ryan and S. Wilson, "Two classes of convolutional codes over GF(q) for q-ary orthogonal signaling," *IEEE Trans. Commun.*, vol. 39, no. 1, pp. 30–40, January 1991.

[23] J. Wu and D. Costello, "New multilevel codes over GF(q)," *IEEE Trans. Inform. Theory*, vol. 38, no. 3, pp. 933–939, May 1992.

[24] E. Martinian and C. E. W. Sundberg, "Decreasing distortion using low delay codes for bursty packet loss channels," *IEEE Trans. Multimedia*, vol. 5, no. 3, pp. 285–292, September 2003.

[25] R. G. Gallager, *Information Theory and Reliable Communication*. McGraw-Hill, 1968.

[26] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. New York, NY: IEEE Press, 1999.

[27] G. D. F. Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, November 1970.

[28] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A markov-based channel model algorithm for wireless networks," in *Proceedings of Fourth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2001)*, 2001.

[29] M. L. Cedervall and R. Johannesson, "A fast algorithm for computing distance spectrum of convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-35, pp. 1146–1159, November 1989.

[30] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Prentice-Hall, 1994.

[31] E. Martinian, "Dynamic information and constraints in coding," Ph.D. dissertation, Massachusetts Institute of Technology, 2004, in preparation.

Emin Martinian (S'2000-M'2004) earned a B.S. from the University of California at Berkeley in 1997 and an M.S. and Ph.D. from the Massachusetts Institute of Technology in 2000 and 2004 (all in Electrical Engineering and Computer Science). He has worked at OPC Technologies (which was later acquired by Mentor Graphics) from 1997-1998, Lucent Technologies' Bell Labs in 2000, Analog Devices in 2002, and has consulted for a variety of wireless technology companies before joining Mitsubishi Electric Research Labs in 2004. His current research includes data compression with distortion side information, streaming error correction codes, and graphical models, especially data compression with codes on graphs. More generally his interests span the areas of digital communications, information theory, signal processing, error control codes, and video processing, compression and authentication.

Carl-Erik W. Sundberg (S69-M75-SM81-F90) was born in Karlskrona, Sweden on July 7,1943.He received the M.S.E.E. and the Dr. Techn. Degrees from the Lund Institute of Technology, University of Lund, Lund, Sweden in 1966 and 1975 respectively.

Before 1976 he held various teaching and research positions at the University of Lund. During 1976 he an ESA Research Fellow at the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands. From 1977 to 1984 he was a Research Professor (Docent) in the department of Telecommunication Theory, University of Lund, Lund, Sweden. During this time he also held positions as consulting scientist at Ericsson and Saab. From 1984 to 2000 he was a Distinguished Member of Technical Staff (DMTS) at Bell Laboratories, Murray Hill, New Jersey, USA and during 2001 he was a DMTS at Agere Systems, Murray Hill, New Jersey, USA. He was a Senior Scientist at iBiquity Digital Corp, Warren, New Jersey, USA during 2002. Currently he is with SundComm, Chatham, New Jersey.

In 1978 he started his consulting company SundComm which has been involved in studies of error control and digital modulation techniques for Swedish Defense as well as a number of companies and international organizations. His research interests include source coding, channel coding, digital modulation, fault-tolerant systems, digital mobile radio, digital audio broadcasting, spread-spectrum, digital satellite systems and optical communications. He has published over 100 journal papers and contributed over 150 conference papers. He has 70 patents, granted and pending. He is a coauthor of Digital Phase Modulation, (New York: Plenum, 1986), Topics in Coding Theory, (New York: Springer-Verlag, 1989) and Source-Matched Digital Communications (New York: IEEE Press, 1996).

Dr Sundberg has been a member of the IEEE European-African-Middle East Committee (EAMEC) of Communications Society (ComSoc) from 1977 to 1984. He is a member of ComSoc Communication Theory Committee. He has also been a member of the Technical Program Committees for the International Symposium on Information Theory, St. Jovite, Canada, October 1983, for the International Conference on Communications, ICC84, Amsterdam, The Netherlands, May 1984, for the 5th Tirrenia International Workshop on Digital Communications, Tirrenia, Italy, September 1991, for the International Telecommunications Symposium, ITS94, Rio de Janeiro, Brazil, August 1994 and for the International Zurich Seminar on Broadband Communications,IZS2000, Zurich, Switzerland, February 2000.He has organized and chaired numerous sessions at a number of international meetings. He has been a member of the International Advisory Committee for the ICCS88, 90, 92 and 94 in Singapore. He served as Guest Editor for the IEEE Journal on Selected Areas in Communications (JSAC) in 1988-1989. He is a member of the Swedish URSI Committee (Svenska Nationalkommitten for Radiovetenskap). In 1986 he and his coauthor received the IEEE Vehicular Technology Societys Paper of the Year Award and in 1989 he and his coauthors were awarded the Marconi Premium Proc. IEE Best Paper Award. Two of his papers were selected for inclusion in the IEEE Communications Society 50th Anniversary Journal Collection Volume 2002. He is a Fellow of the IEEE since 1990 and is listed in Marquis Whos Who in America.