

# Delay-Optimal Streaming Codes under Source-Channel Rate Mismatch

Pratik Patil, Ahmed Badr and Ashish Khisti  
Electrical and Computer Engineering  
University of Toronto  
Toronto, ON, M5S 3G4  
Email: {ppatil, abadr, akhisti}@comm.utoronto.ca

Wai-Tian Tan  
Mobile and Immersive Experience Lab  
Hewlett Packard Laboratories  
1501 Pagemill Road  
Palo Alto, CA, 94304

**Abstract**—We study low-delay error correction codes for streaming-recovery over a class of packet-erasure channels. In our setup, the encoder observes one source frame every  $M$  time slots, but is required to transmit a channel packet in each time slot. The decoder is required to reconstruct each source frame within a playback delay of  $T$  source frames. The collection of  $M$  transmitted channel packets between successive source frames is called a (channel) macro-packet.

For a certain class of burst-erasure channels, we characterize the associated capacity and develop explicit codes that attain the capacity. We recover as a special case, the capacity when  $M = 1$ , studied in earlier works. Our proposed code constructions involve splitting each source frame into two groups of sub-symbols, applying unequal error protection and carefully allocating source and parity-check sub-symbols within each (channel) macro-packet. Our constructions are a non-trivial extension of the previously proposed codes for  $M = 1$ . Simulation results indicate significant gains over baseline error correction codes for the Gilbert model for burst erasures.

## I. INTRODUCTION

Emerging applications such as video/audio conferencing, mobile gaming and cloud computing impose stringent end-to-end latency constraints and are inherently streaming in nature. The sender terminal must encode a source stream in real-time, and the destination must output each source frame within a fixed playback deadline. The end-to-end latency is generally less than 250 ms [1, Table 1, pp. 7]. The round-trip time in traditional networks can alone approach such limits. Thus we need advanced techniques for error correction, rate control, and scalable compression optimized for the delay-constrained and streaming nature of such applications.

In this paper we propose a novel class of delay-optimized error correction codes for real-time streaming over burst-loss channels. Commonly used error correction codes operate on message blocks. To apply them to streaming data, we need to either buffer data packets at the encoder or accumulate all packets at the decoder before any recovery is possible. To reduce delay we need to keep the codeword lengths short, which in turn reduces the error correction capability [2].

Low-delay error correction codes for streaming sources have been recently studied in [3]–[5] and further generalized in [6], [7]. The focus in [3]–[5] was on burst-erasure channels. The transmitter is required to encode a stream of source packets sequentially and the receiver is required to reconstruct each

source packet in the stream with a fixed delay. The channel can introduce an erasure burst of a given maximum length. The maximum achievable rate was characterized in this setup and a new class of codes, Maximally Short Codes (MS), were shown to outperform the classical Maximum Distance Separable (MDS) codes. More recently [6], [7] propose robust extensions of streaming codes that are resilient against both burst and isolated losses. In particular it is shown that in the streaming setup, there exists a fundamental tradeoff between the burst-error correction and isolated-error correction capability of any code. A new family of codes, MiDAS Codes, that achieve near optimal performance has also been developed [7].

The above works however assume that the source and channel transmission rates are identical i.e., one source packet arrives before the transmission of every channel packet. In many practical systems there is a mismatch between the source and channel transmission rates. For example in most video streaming systems, each source frame arrives once approximately every 40 ms, whereas the interval between successive channel packets is typically much smaller. Thus a large number of channel packets may need to be transmitted in-between the arrival of two successive source frames. We refer to this mismatched scenario as *source-channel rate mismatch*. A straightforward way of implementing the streaming codes in this scenario is to split each source frame into multiple packets such that there is one source packet for each transmitted channel packet. We show that such a naive approach is sub-optimal and propose a new class of optimal codes for this mis-matched scenario. Due to page constraints, we only focus on the case of burst erasure channels in this paper. We expect that robust extensions that are resilient against both burst and isolated losses can be obtained using the techniques in [6], [7]. For other related works on low-delay streaming codes we refer to [8]–[16].

## II. SYSTEM MODEL

We study low-delay codes when there is a mismatch between source and channel frame rates. We assume that one source packet arrives at the encoder every  $M$  channel packets. We call the collection of such  $M$  channel-packets as a macro-packet. Each source packet is encoded into the channel stream in a causal fashion and needs to be reconstructed at the



Fig. 1. System under consideration. Each  $\mathbf{X}[i, :]$  denotes a (channel) macro-packet consisting of  $M$  channel packets ( $\mathbf{x}[i, 1], \dots, \mathbf{x}[i, M]$ ). One source packet arrives at the start of each macro packet. The channel erases up to  $B$  consecutive channel packets. Each source packet needs to be reconstructed with a delay of  $T$  macro packets.

destination after a delay of  $T$  macro-packets. In this work we focus on the burst-erasure channels i.e., we assume that up to  $B$  consecutive channel packets are erased in a single burst. Fig. 1 depicts the system under consideration. We discuss the operation of each of the blocks in Fig. 1 in detail below.

### A. Encoder

At each  $i \geq 0$ , the encoder receives a source packet  $\mathbf{s}[i] \in \mathbb{F}_q^k$ , where  $\mathbb{F}_q$  denotes the underlying base-field and  $k$  denotes the number of sub-symbols in  $\mathbf{s}[i]$ . At the start of macro-packet  $i$ , the encoder generates  $M$  channel packets  $\mathbf{x}[i, j] \in \mathbb{F}_q^n$ ,  $j = \{1, \dots, M\}$  which can depend on all the observed source packets up to that time i.e.<sup>1</sup>,

$$\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \quad (1)$$

and transmit them in the  $M$  slots corresponding to the macro-packet  $i$ . It will be convenient to use the notation

$$\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]] \in \mathbb{F}_q^{n \times M} \quad (2)$$

to denote the macro-packet  $i$ . Fig. 1 denotes the operation of our system whereas Fig. 2 denotes the structure of each macro-packet.

### B. Channel

The received packets corresponding to macro-packet  $i$  are denoted by  $\mathbf{y}[i, j]$  for  $j = \{1, \dots, M\}$ . We assume a burst erasure channel. The channel introduce an erasure burst of maximum length  $B$  channel packets starting at arbitrary time slot  $[i_s, j_s]$  during the transmission of  $\mathbf{X}[i_s, :]$  and ending at  $[i_f, j_f]$  during the transmission of  $\mathbf{X}[i_f, :]$ . Note that the number of symbols spanned in the period  $[i_s, j_s] \rightarrow [i_f, j_f]$  is given by:

$$\tau = (M - j_s + 1) + (i_f - i_s - 1)M + j_f, \quad j_s, j_f \in \{1, \dots, M\}, i_f \geq i_s \geq 0. \quad (3)$$

Thus, the output channel packets are given by,

$$\mathbf{y}[i, j] = \begin{cases} \star, & \text{for } [i, j] \in \{[i_s, j_s], [i_f, j_f]\}, \\ \mathbf{x}[i, j], & \text{otherwise.} \end{cases} \quad (4)$$

We note that the erasure burst can occur across multiple channel macro packets as shown in Fig. 2. The erasure burst can also start at any arbitrary position within each macro-packet. We will denote the set of all channel packets corresponding to time index  $i$  by the matrix  $\mathbf{Y}[i, :] = [\mathbf{y}[i, 1] \mid \dots \mid \mathbf{y}[i, M]] \in \mathbb{F}_q^{n \times M}$ , where again  $\mathbf{y}[\cdot]$  denotes a column vector of length  $n$ .

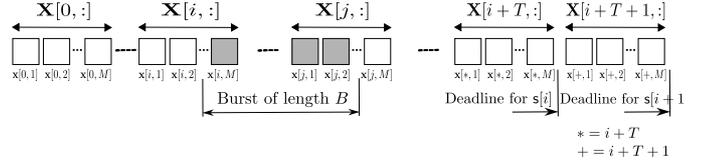


Fig. 2. Channel Model. The erasure burst spans a total of  $B$  channel symbols as shown. Each source packet  $\mathbf{s}[i]$  arrives just before the transmission of  $\mathbf{X}[i, :]$  and needs to be reconstructed by the destination after a delay of  $T$  macro-packets.

### C. Decoder

The decoder is required to decode each source packet with a maximum delay of  $T$  macro packets i.e., the decoder uses a reconstruction function  $g_i(\cdot)$ :

$$\hat{\mathbf{s}}[i] = g_i(\mathbf{Y}[0, :], \mathbf{Y}[1, :], \dots, \mathbf{Y}[i+T, :]). \quad (5)$$

The rate of the streaming code is defined as the ratio of the entropy of the source packet to the size of the channel macro packet i.e.,

$$R = \frac{H(\mathbf{s})}{n \times M}. \quad (6)$$

Note that in (6) we assume that the source sequence  $\{\mathbf{s}[i]\}_{i \geq 0}$  is sampled i.i.d. from a distribution  $p_s(\cdot)$ . We say that a rate  $R$  is achievable if there exists a streaming code of rate  $R$  such that  $\Pr(\hat{\mathbf{s}}[i] \neq \mathbf{s}[i]) = 0$ , for each  $i \geq 0$ . The largest achievable rate is the streaming capacity, which is the quantity of interest.

## III. MAIN RESULT

As our main result, the following Theorem provides a characterization of the streaming capacity defined in the previous section.

*Theorem 1:* For the streaming setup in section II, with any  $M$ ,  $T$  and  $B$ , the *streaming capacity*  $C$  is given by the following expression:

$$C = \begin{cases} \frac{T}{T+b}, & B' \leq \frac{b}{T+b}M, T \geq b, \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & B' > \frac{b}{T+b}M, T > b, \\ \frac{M-B'}{M}, & B' > \frac{M}{2}, T = b, \\ 0, & T < b. \end{cases} \quad (7)$$

where the constants  $b$  and  $B'$  are defined via

$$B = bM + B', \quad B' \in \{1, \dots, M-1\}, b \in \mathbb{N}^0. \quad (8)$$

The proof of Theorem 1 is divided into two main parts. The code construction is illustrated in section V while the converse appears in section VI. In the remainder of this section we elaborate on the different cases associated with (7).

We note that the capacity is zero if  $T < b$ . It can be easily verified that in this case, there exists an erasure burst of length  $B$  that spans all underlying channel packets up to the deadline thus making the recovery impossible. This case will therefore not be discussed further in the paper.

Next consider the case when  $T = b$ , which corresponds to the minimum possible delay for which the capacity is positive.

<sup>1</sup>The vectors  $\mathbf{s}[i]$  and  $\mathbf{x}[i, j]$  denote column vectors. We will later use the notation  $\mathbf{s}^\dagger[i]$  and  $\mathbf{x}^\dagger[i, j]$  to denote the transpose of these vectors.

In this case the capacity in Theorem 1 reduces to the following:

$$C = \begin{cases} \frac{1}{2}, & 0 \leq B' \leq \frac{M}{2}, T = b, \\ \frac{M-B'}{M}, & \frac{M}{2} \leq B' \leq M, T = b. \end{cases} \quad (9)$$

Since a burst of length  $B$  spans at-least  $b$  macro-packets, during the recovery of  $\mathbf{s}[i]$  we can only use the unerased symbols of  $\mathbf{Y}[i, :]$  and  $\mathbf{Y}[i + b, :]$ ; all the intermediate macro-packets are completely erased. It turns out that a simple repetition code that uses  $\min\{M - B', \frac{M}{2}\}$  information packets and an identical number of parity check packets in each macro-packet achieves the capacity when  $T = b$ .

Finally, when  $T > b$  the capacity in Theorem 1 is given by the following

$$C = \begin{cases} \frac{T}{T+b}, & 0 \leq B' \leq \frac{b}{T+b}M, T > b \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & \frac{b}{T+b}M < B' \leq M-1, T > b \end{cases} \quad (10)$$

Examining (10) we note that, quite remarkably, the capacity does not decrease with  $B'$  as it is increased in the interval  $[0, \frac{b}{T+b}M]$ . We refer the reader to Fig. 4 in section VII where this characteristic of the capacity function is illustrated using a numerical example.

#### IV. BACKGROUND

In this section we review previously proposed code constructions — Baseline Erasure Codes and SCo codes — and study their error correction properties in the present setup. We will conclude that the rates achieved by these schemes do not meet the stated capacity in Theorem 1. Nevertheless our proposed codes build upon these ideas, and hence their review is essential before stating the proposed construction.

##### A. Baseline Erasure Codes

Classical erasure codes are designed for maximizing the underlying distance properties. In a streaming setup, roughly speaking, such codes will be able to recover all the missing source symbols simultaneously once sufficiently many parity checks have been collected. Indeed, the motivation behind *random-linear codes* is to guarantee that the underlying system of equations is nearly of a full rank [17]–[19]. Instead of the random linear codes, we study a class of deterministic codes with maximum distance that are relevant for our streaming setup.

Consider a  $(n, k, m)$  convolutional code that maps an input source stream  $\mathbf{s}[i] \in \mathbb{F}_q^k$  to an output  $\mathbf{x}[i] \in \mathbb{F}_q^n$  using a memory  $m$  encoder i.e.,

$$\mathbf{x}[i] = \left( \sum_{t=0}^m \mathbf{s}^\dagger[i-t] \cdot \mathbf{G}_t \right)^\dagger \quad (11)$$

where  $\mathbf{G}_0, \dots, \mathbf{G}_m$  are  $k \times n$  matrices with elements in  $\mathbb{F}_q$ . The first  $j+1$  output symbols can be expressed as,

$$[\mathbf{x}^\dagger[0], \mathbf{x}^\dagger[1], \dots, \mathbf{x}^\dagger[j]] = [\mathbf{s}^\dagger[0], \mathbf{s}^\dagger[1], \dots, \mathbf{s}^\dagger[j]] \cdot \mathbf{G}^s. \quad (12)$$

where

$$\mathbf{G}^s = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_j \\ 0 & \mathbf{G}_0 & & \mathbf{G}_{j-1} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & \mathbf{G}_0 \end{bmatrix} \quad (13)$$

is the truncated generator matrix to the first  $j+1$  columns and we define  $\mathbf{G}_j = 0$  if  $j > m$ .

The column distance of order  $j$  is defined as:

$$d_j = \min_{\substack{\mathbf{s}=(\mathbf{s}[0], \dots, \mathbf{s}[j]) \\ \mathbf{s}[0] \neq 0}} \text{wt}(\mathbf{x}^\dagger[0], \dots, \mathbf{x}^\dagger[j]), \quad (14)$$

where  $\text{wt}(\mathbf{x}^\dagger[0], \dots, \mathbf{x}^\dagger[j])$  counts the number of non-zero *sub-symbols* in  $(\mathbf{x}^\dagger[0], \dots, \mathbf{x}^\dagger[j]) \in \mathbb{F}_q^{(j+1)n}$ . We are particularly interested in systematic convolutional codes where  $\mathbf{G}^s$  is expressed in the systematic form

$$\mathbf{G}_0 = [\mathbf{I}_{k \times k} \ \mathbf{H}_0], \quad \mathbf{G}_i = [\mathbf{0}_{k \times k} \ \mathbf{H}_i], \quad i = 1, \dots, m, \\ \mathbf{H}_0, \mathbf{H}_i \in \mathbb{F}_q^{k \times (n-k)} \quad (15)$$

*Theorem 2 (Gluesing-Luerssen et. al, Gabidulin [20], [21]):* For every  $j \in \mathbb{N}$  we have that

$$d_j \leq (n-k)(j+1) + 1 \quad (16)$$

A systematic, strongly MDS  $(n, k, m)$  code  $\mathcal{C}$  achieves

$$d_j = (n-k)(j+1) + 1, \quad j = 0, 1, \dots, m. \quad (17)$$

□

Thus the systematic Strongly-MDS codes achieve the maximum distance in the window of interest. The following lemma, which follows by an application of Theorem 2 will be used in our analysis. It will not be included due to space constraints.

*Lemma 1:* Consider a systematic Strongly-MDS  $(n, k, m)$  code defined in Theorem 2 that maps source symbols  $\mathbf{s}[i] \in \mathbb{F}_q^k$  to channel symbols  $\mathbf{x}[i] \in \mathbb{F}_q^n$  as illustrated in (11). Let  $\mathbf{x}[i] = \begin{bmatrix} \mathbf{s}[i] \\ \mathbf{p}[i] \end{bmatrix}$ , and suppose that the sub-symbols in  $\mathbf{x}[i] = (s_1[i], \dots, s_k[i], p_1[i], \dots, p_{n-k}[i])^\dagger$  are transmitted sequentially in the interval  $[i \cdot n, (i+1)n - 1]$  over the channel.

- 1) For any  $j \in [0, m]$  where  $d_j$  defined in (17), the following holds: if no more than  $d_j - 1$  sub-symbols are erased in the interval  $[0, (j+1)n - 1]$  the source symbol  $\mathbf{s}[0] = (s_1[0], \dots, s_k[0])$  can be recovered by time  $(j+1)n - 1$ .
- 2) If the channel introduces an erasure-burst of length  $B$  sub-symbols in the interval  $[0, B-1]$ , where  $B \leq d_j - 1$ , then all erased source symbols are recovered by time  $(j+1)n - 1$ .

Intuitively property 2 above states that a strongly MDS code does simultaneous recovery of all the erased source symbols in the burst, once sufficiently many parity checks are available. We refer to codes with such a property as Baseline Erasure Codes (BEC), and use this throughout the rest of the paper.

From Property 2, an  $(n, k, T)$  BEC code, is guaranteed to recover from an erasure burst of length  $B$  channel packets (equivalently up to  $nB$  sub-symbols) with a delay of  $T$  if

$$B \leq \frac{(n-k)}{n}M(T+1). \quad (18)$$

Using  $R = \frac{k}{n}$  we have that an  $(n, k, m)$  BEC code with

$$R^{\text{BEC}} = 1 - \frac{B}{M(T+1)} \quad (19)$$

is feasible.

### B. Streaming Codes (SCo) for $M = 1$

Unlike the erasure codes in the previous section, Maximally Short Codes (MS) introduced in [4] and further generalized in [6, Section IV-B] enable sequential recovery in the presence of burst-erasures. These codes are constructed for the special case when there is no mis-match between the source and channel frame rates i.e.,  $M = 1$ . A  $(B, T)$  SCo code encodes a stream of source packets  $\mathbf{s}[i] \in \mathbb{F}_q^T$  into a stream of channel packets  $\mathbf{x}[i] \in \mathbb{F}_q^{T+B}$  such that every source symbol  $\mathbf{s}[i]$  can be recovered with a delay of  $T$  when the channel introduces an erasure burst of length at-most  $B$ . Note that rate of an SCo code is  $R = \frac{T}{T+B}$ . We briefly review the SCo construction from [6]. The encoding steps are as follows:

- Split each source symbol  $\mathbf{s}[i] \in \mathbb{F}_q^T$  into two groups  $\mathbf{u}[i] \in \mathbb{F}_q^B$  and  $\mathbf{v}[i] \in \mathbb{F}_q^{T-B}$  as follows:

$$\mathbf{s}[i] = \left\{ \underbrace{u_1[i], \dots, u_B[i]}_{=\mathbf{u}[i]}, \underbrace{v_1[i], \dots, v_{T-B}[i]}_{=\mathbf{v}[i]} \right\}. \quad (20)$$

- Apply a BEC code from the previous sub-section on the symbols  $\mathbf{v}[i]$  and generate parity-check symbols

$$\mathbf{p}_v^\dagger[i] = \sum_{j=1}^T \mathbf{v}^\dagger[i-j] \cdot \mathbf{H}_j^v, \quad \mathbf{p}_v[i] \in \mathbb{F}_q^B, \quad (21)$$

where the matrices  $\mathbf{H}_j^v$  are  $(T-B) \times B$  matrices associated with the systematic strongly MDS code (15).

- Super-impose the  $\mathbf{u}[\cdot]$  symbols onto  $\mathbf{p}_v[\cdot]$  and let

$$\mathbf{q}[i] = \mathbf{p}_v[i] + \mathbf{u}[i-T]. \quad (22)$$

The channel input at time  $i$  is given by  $\mathbf{x}^\dagger[i] = (\mathbf{u}[i], \mathbf{v}[i], \mathbf{q}[i])^\dagger \in \mathbb{F}_q^{T+B}$ .

The decoding of SCo codes proceeds as follows. Suppose that an erasure burst happens in the interval  $[i, i+B-1]$ . The decoder first recovers  $\mathbf{p}_v[i+B], \dots, \mathbf{p}_v[i+T-1]$  from  $\mathbf{q}[i+B], \dots, \mathbf{q}[i+T-1]$  by cancelling the interfering  $\mathbf{u}[\cdot]$  symbols (c.f. (22)) which have not been erased. Using property 2 of the BEC code in Lemma 1 it can be verified that the erased symbols  $\mathbf{v}[i], \dots, \mathbf{v}[i+B-1]$  can be recovered by time  $t = i+T-1$ . Once all the  $\mathbf{v}[\cdot]$  symbols have been recovered, the symbols  $\mathbf{p}[i+T], \dots, \mathbf{p}[i+T+B-1]$  can be computed and cancelled from the associated  $\mathbf{q}[\cdot]$  symbols. Thus each  $\mathbf{u}[i], \dots, \mathbf{u}[i+B-1]$  can be recovered at their deadline. Since  $\mathbf{s}[i] = (\mathbf{u}[i], \mathbf{v}[i])$ , this shows that each erased source symbol is recovered with a delay of  $T$  symbols.

*Adapting SCo codes for Mis-Matched Case:* We now discuss how the SCo codes can be adapted to the mis-matched case. We propose to split each symbol  $\mathbf{s}[i]$  into  $M$  sub-symbols, one for each time-slot in the macro-packet and then apply an SCo code to this expanded source stream.

- Assume that each  $\mathbf{s}[i] \in \mathbb{F}_q^{TM}$  and split each  $\mathbf{s}[i] = (\mathbf{w}[i, 1], \dots, \mathbf{w}[i, M])$  where  $\mathbf{w}[i, j] \in \mathbb{F}_q^T$  holds.
- Apply a  $(B, MT)$  SCo code of rate

$$R^{\text{SCo}} = \frac{MT}{MT+B} = \frac{T}{T+b+\frac{B'}{M}} \quad (23)$$

to the source stream  $\{\mathbf{w}[\cdot, j]\}$ , where  $M \cdot T$  denotes the delay in channel-packets. Transmit the associated channel

packet  $\mathbf{x}[i, j]$  in slot  $j$  of the macro-packet  $i$ .

Note that the delay of  $M \cdot T$  channel packets implies that the source packet  $\mathbf{w}[i, j]$  is recovered at time  $[i+T, j]$  for each  $j \in \{1, 2, \dots, M\}$ . Thus the entire source packet  $\mathbf{s}[i]$  is guaranteed to be recovered by at the end of macro-packet  $i+T$ , thus satisfying the delay constraint. We note that (23) only attains the capacity when  $B' = 0$  and  $B < MT$ . Furthermore if  $B > MT$  the above construction is not feasible and the rate attained is zero.

## V. CODE CONSTRUCTION

We first present the encoding and decoding for the case when  $T > b$  in (7) in Theorem 1. The case when  $T = b$  uses a repetition code and will not be treated due to space constraints.

### A. Encoding: $T > b$ and $B' \leq \frac{b}{T+b}M$

Following (7) we need to present a construction that achieves  $R = \frac{T}{T+b}$ . We let

$$n = T + b, \quad k = MT, \quad (24)$$

throughout this case. Note that the rate  $R = \frac{k}{Mn}$  reduces to the desired expression.

The main steps in our encoder are stated below.

- **Source Splitting:** We partition each source vector  $\mathbf{s}[i] \in \mathbb{F}_q^k$  into  $k$  sub-symbols and divide them into two groups  $\mathbf{u}^{\text{vec}}[i] \in \mathbb{F}_q^{k_u}$  and  $\mathbf{v}^{\text{vec}}[i] \in \mathbb{F}_q^{k_v}$  as follows,

$$\begin{aligned} \mathbf{s}[i] &= (s_1[i], \dots, s_k[i]) \\ &= \underbrace{(u_1[i], \dots, u_{k_u}[i])}_{\mathbf{u}^{\text{vec}}[i]}, \underbrace{(v_1[i], \dots, v_{k_v}[i])}_{\mathbf{v}^{\text{vec}}[i]} \end{aligned} \quad (25)$$

where we select

$$k_u = Mb, \quad k_v = M(T-b). \quad (26)$$

- **BEC Parity Checks:** Apply a  $(k_v + k_u, k_v, T)$  BEC code of rate  $\frac{k_v}{k_v+k_u}$  to the sub-stream of  $\mathbf{v}^{\text{vec}}[\cdot]$  symbols generating  $k_u$  parity-check sub-symbols,  $\mathbf{q}^{\text{vec}}[i] = (q_1[i], \dots, q_{k_u}[i]) \in \mathbb{F}_q^{k_u}$  for each macro-packet. In particular we have that

$$\mathbf{q}^{\text{vec}}[i] = \left( \sum_{j=0}^T \mathbf{v}^{\text{vec}, \dagger}[i-j] \cdot \mathbf{H}_j \right)^\dagger \quad (27)$$

where  $\mathbf{H}_j \in \mathbb{F}_q^{k_v \times k_u}$  are the sub-matrices associated with the BEC code.

- **Parity-Check Generation:** Combine the  $\mathbf{u}^{\text{vec}}[\cdot]$  symbols with the  $\mathbf{q}^{\text{vec}}[\cdot]$  parity-checks after applying a shift of  $T$  to the former, i.e.,

$$\mathbf{p}^{\text{vec}}[i] = \mathbf{q}^{\text{vec}}[i] + \mathbf{u}^{\text{vec}}[i-T], \quad (28)$$

where  $\mathbf{p}^{\text{vec}}[i] \in \mathbb{F}_q^{k_u}$ .

- **Re-shaping:** In order to construct the macro-packet, we reshape  $\mathbf{u}^{\text{vec}}[i]$ ,  $\mathbf{v}^{\text{vec}}[i]$  and  $\mathbf{p}^{\text{vec}}[i]$  into groups each of  $n$  sub-symbols generating the matrices  $\mathbf{U}[i]$ ,  $\mathbf{V}[i]$  and  $\mathbf{P}[i]$  as illustrated below.

$$\mathbf{U}[i, :] = \left[ \mathbf{u}[i, 1] \mid \dots \mid \mathbf{u}[i, r] \mid \frac{\mathbf{u}[i, r+1]}{0} \right] \in \mathbb{F}_q^{n \times r+1}$$

$$\mathbf{V}[i, :] = \left[ \begin{array}{c|c|c|c|c} \mathbf{0} & \mathbf{v}[i, 2] & \cdots & \mathbf{v}[i, M-2r-1] & \mathbf{0} \\ \hline \mathbf{v}[i, 1] & & & & \mathbf{v}[i, M-2r] \end{array} \right]$$

$$\mathbf{P}[i, :] = \left[ \begin{array}{c|c|c|c} \mathbf{p}[i, r+1] & \mathbf{p}[i, r] & \cdots & \mathbf{p}[i, 1] \\ \hline \mathbf{0} & & & \end{array} \right] \in \mathbb{F}_q^{n \times r+1}, \quad (29)$$

where the vectors  $\mathbf{u}[i, j]$ ,  $\mathbf{v}[i, j]$  and  $\mathbf{p}[i, j]$  are expressed via

$$\mathbf{u}^{\text{vec}}[i] = \begin{bmatrix} \mathbf{u}[i, 1] \\ \mathbf{u}[i, 2] \\ \vdots \\ \mathbf{u}[i, r] \\ \mathbf{u}[i, r+1] \end{bmatrix} \quad \mathbf{v}^{\text{vec}}[i] = \begin{bmatrix} \mathbf{v}[i, 1] \\ \mathbf{v}[i, 2] \\ \vdots \\ \mathbf{v}[i, M-2r-1] \\ \mathbf{v}[i, M-2r] \end{bmatrix}$$

$$\mathbf{p}^{\text{vec}}[i] = \begin{bmatrix} \mathbf{p}[i, 1] \\ \mathbf{p}[i, 2] \\ \vdots \\ \mathbf{p}[i, r] \\ \mathbf{p}[i, r+1] \end{bmatrix} \quad (30)$$

In (29) we define  $r \in \mathbb{N}^0$  and  $r' \in \{0, 1, \dots, n-1\}$  via

$$k_u = r \cdot n + r'. \quad (31)$$

Note that  $\mathbf{u}[i, j] \in \mathbb{F}_q^n$  for each  $j \in \{1, \dots, r\}$  and  $\mathbf{u}[i, r+1] \in \mathbb{F}_q^{r'}$ . The splitting of  $\mathbf{p}^{\text{vec}}[i]$  into  $\mathbf{p}[i, j]$  in (29) follows in an analogous manner. In particular we can express

$$\mathbf{p}[i, j] = \mathbf{u}[i-T, j] + \mathbf{q}[i, j], \quad j = 1, 2, \dots, r+1 \quad (32)$$

where  $\mathbf{q}[i, j]$  is a sub-sequence of  $\mathbf{q}^{\text{vec}}[i]$  defined in a similar manner. In the splitting of  $\mathbf{v}^{\text{vec}}[i]$  into  $\mathbf{v}[i, j]$  we note that  $\mathbf{v}[i, 1], \mathbf{v}[i, M-2r] \in \mathbb{F}_q^{n-r'}$  whereas  $\mathbf{v}[i, j] \in \mathbb{F}_q^n$  for  $2 \leq j \leq M-2r-1$ .

- **Macro-Packet Generation** Concatenate  $\mathbf{U}[i, :]$ ,  $\mathbf{V}[i, :]$  and  $\mathbf{P}[i, :]$  to construct the channel macro packet  $\mathbf{X}[i, :]$  as follows<sup>2</sup>

$$\mathbf{X}[i, :] = [\mathbf{x}[i, 1], \dots, \mathbf{x}[i, M]] = \left[ \begin{array}{c|c|c|c|c} \mathbf{u}[i, 1] & \cdots & \mathbf{u}[i, r] & \begin{array}{c} \mathbf{u}[i, r+1] \\ \mathbf{v}[i, 1] \end{array} & \mathbf{v}[i, 2] \\ \cdots & \mathbf{v}[i, M-2r-1] & \begin{array}{c} \mathbf{p}[i, r+1] \\ \mathbf{v}[i, M-2r] \end{array} & \mathbf{p}[i, r] & \cdots \\ & & & \mathbf{p}[i, 1] & \end{array} \right] \quad (33)$$

Note that the channel macro packet at time  $i$  is denoted by  $\mathbf{X}[i, :] \in \mathbb{F}_q^{n \times M}$  and the  $j$ th channel packet in  $\mathbf{X}[i, :]$  by  $\mathbf{x}[i, j] \in \mathbb{F}_q^n$  for  $j \in \{1, \dots, M\}$ .

This completes the description of the encoding function for the first case in (7).

### B. Encoding: $T > b$ and $B' > \frac{b}{T+b}M$

We begin by choosing the following values of  $n$  and  $k$ :

$$n = T + b + 1, \quad k = M(T + b + 1) - B \quad (34)$$

and note that the rate  $R = \frac{k}{Mn}$  reduces to the second case in (7).

<sup>2</sup>The expression assume that  $M-2r > 1$ . If  $M-2r = 1$  then the  $\mathbf{v}^{\text{vec}}[i]$  symbols will only occupy one single column and the symbols of  $\mathbf{u}[i, r+1]$  and  $\mathbf{p}[i, r+1]$  may be present in the same column. The analysis also applies in this case. We can easily show that  $M-2r > 0$  in all of our analysis.

We partition each source vector  $\mathbf{s}[i] \in \mathbb{F}_q^k$  into  $k$  sub-symbols and divide them into two groups  $\mathbf{u}^{\text{vec}}[i] \in \mathbb{F}_q^{k_u}$  and  $\mathbf{v}^{\text{vec}}[i] \in \mathbb{F}_q^{k_v}$  as in (25). This time we select

$$k_u = B = Mb + B', \quad k_v = M(T + b + 1) - 2B \quad (35)$$

We generate the parity checks  $\mathbf{p}^{\text{vec}}[i] = \mathbf{q}^{\text{vec}}[i] + \mathbf{u}^{\text{vec}}[i-T]$  as in the previous sub-section. Thereafter we reshape vectors  $\mathbf{u}^{\text{vec}}[i]$ ,  $\mathbf{v}^{\text{vec}}[i]$  and  $\mathbf{p}^{\text{vec}}[i]$  into matrices  $\mathbf{U}[i, :]$ ,  $\mathbf{V}[i, :]$  and  $\mathbf{P}[i, :]$  as in (29) and subsequently generate the macro-packet  $\mathbf{X}[i, :]$  as in (33). These steps will not be repeated due to space constraints.

### C. Decoding

We consider a channel that introduces a burst of length  $B = bM + B'$  starting at the  $i$ th macro packet. The total number of patterns to consider is  $M$  which corresponds to burst erasures of length  $B$  starting at  $\mathbf{x}[i, j]$  for  $j \in \{1, \dots, M\}$ .

We begin by considering the patterns with the erasure burst starting at  $\mathbf{x}[i, 1]$  which erases  $\mathbf{X}[i, \dots, \mathbf{X}[i+b-1], \mathbf{x}[i+b, 1], \dots, \mathbf{x}[i+b, B']$ . We will then discuss the case when the burst-loss begins at  $\mathbf{x}[i, j]$  where  $j > 1$ . The main steps in the decoding are as follows —

- 1) In each macro-packet  $t \in [i+b, i+T-1]$  recover all un-erased  $\mathbf{q}^{\text{vec}}[t]$  subtracting out  $\mathbf{u}^{\text{vec}}[t-T]$  from the associated  $\mathbf{p}^{\text{vec}}[t]$  as the former are not erased (c.f. (28)).
- 2) Recover all erased  $\mathbf{v}^{\text{vec}}[i]$  symbols by macro-packet  $i+T-1$  using the underlying BEC code.
- 3) Compute  $\mathbf{q}^{\text{vec}}[i+T], \dots, \mathbf{q}^{\text{vec}}[i+T+b]$  as they combine  $\mathbf{v}^{\text{vec}}[i]$  symbols which are either not erased or recovered in the previous step.
- 4) Subtract  $\mathbf{q}^{\text{vec}}[i+T], \dots, \mathbf{q}^{\text{vec}}[i+T+b]$  from  $\mathbf{p}^{\text{vec}}[i+T], \dots, \mathbf{p}^{\text{vec}}[i+T+b]$  to recover  $\mathbf{u}^{\text{vec}}[i], \dots, \mathbf{u}^{\text{vec}}[i+b]$  respectively within a delay of  $T$  macro packets. At this point all the source packets have been recovered with a delay of  $T$  macro-packets as required.

It only remains to show the sufficiency of the BEC code in the recovery during the second step. This can be established by showing that no more than  $k_u T$  sub-symbols are lost for the  $(k_u + k_v, T)$  BEC code  $(\mathbf{v}^{\text{vec}}[t], \mathbf{q}^{\text{vec}}[t])$  due to the above erasure burst. The recovery then follows using Property 2 of Lemma 1. We separately consider two cases as below.

1)  $B' \leq \frac{b}{T+b}M$ : In this case, the code parameters are chosen according to (24) and (26). One needs to count the total number of erased sub-symbols among  $(\mathbf{v}^{\text{vec}}[t], \mathbf{q}^{\text{vec}}[t])$  for  $t \in [i, i+b]$  due to the erasure burst of length  $B = bM + B'$  starting at  $\mathbf{x}[i, 1]$ . This burst erases the channel packets,  $\mathbf{X}[i, :], \dots, \mathbf{X}[i+b-1, :], \mathbf{x}[i+b, 1], \dots, \mathbf{x}[i+b, B']$ . Clearly, all sub-symbols of  $(\mathbf{v}^{\text{vec}}[t], \mathbf{q}^{\text{vec}}[t])$  for  $t \in [i, i+b-1]$  are erased which are a total of  $b(k_u + k_v) = bMT$ . For the  $(i+b)$ th macro packet, the first  $B'$  channel packets are erased, which correspond only to sub-symbols in  $\mathbf{u}^{\text{vec}}[i+b]$  as  $B'n \leq k_u$  since  $B' \leq \frac{b}{T+b}M$ . Hence, the total number of erased sub-symbols in  $(\mathbf{v}^{\text{vec}}[t], \mathbf{q}^{\text{vec}}[t])$  are  $bMT = k_u T$  as required.

2)  $B' > \frac{b}{T+b}M$ : In this case the code parameters are chosen as in (34) and (35). Similar to the previous case, we count the number of erased sub-symbols among  $(\mathbf{v}^{\text{vec}}[t], \mathbf{q}^{\text{vec}}[t])$  for

$t \in [i, i + b]$  due to an erasure burst of length  $B = bM + B'$  starting at  $\mathbf{x}[i, 1]$ . The first  $b$  macro packets  $\mathbf{X}[i, :], \dots, \mathbf{X}[i + b - 1, :]$  are all erased. Thus, the total number of erased  $\mathbf{v}^{\text{vec}}[t]$  and  $\mathbf{q}^{\text{vec}}[t]$  sub-symbols in this interval is  $b(k_u + k_v) = b(M(T + b + 1) - B)$ . In the  $(i + b)$ th macro packet, the first  $B'$  channel packets are erased which correspond to all  $\mathbf{u}^{\text{vec}}[i + b]$  sub-symbols and some of the  $(\mathbf{v}^{\text{vec}}[i + b], \mathbf{q}^{\text{vec}}[i + b])$  sub-symbols as  $B'n > k_u$  since  $B'(T + b + 1) > Mb + B'$ . Thus, the number of erased sub-symbols among  $(\mathbf{v}^{\text{vec}}[i + b], \mathbf{q}^{\text{vec}}[i + b])$  are  $B'n - k_u = B'(T + b + 1) - B$ . Hence, the total number of erasures among  $(\mathbf{v}^{\text{vec}}[t], \mathbf{q}^{\text{vec}}[t])$  for  $t \in [i, i + b]$  are

$$b(M(T + b + 1) - B) + B'(T + b + 1) - B = BT = k_u T$$

as required.

In the above decoding steps, we only considered bursts that start at  $\mathbf{x}[i, 1]$ . Here, we extend the decoding steps for erasure bursts that start at any channel packet within the macro packet. Consider an erasure bursts  $\mathcal{B}_j$  of length  $B = bM + B'$  starting at  $\mathbf{x}[i, j]$  for  $j = \{1, \dots, M\}$ . We argue that going from  $\mathcal{B}_j$  to  $\mathcal{B}_{j+1}$  we do not increase the total number of erased sub-symbols in the  $(\mathbf{v}^{\text{vec}}[i], \mathbf{q}^{\text{vec}}[i])$  BEC code. Thus the case when  $j = 1$  is in-fact the worst case.

Note that in going from  $\mathcal{B}_j$  to  $\mathcal{B}_{j+1}$  the channel packet  $\mathbf{x}[i, j]$  is no longer erased and is available to the decoder. From the construction of the macro-packet  $\mathbf{X}[i, :]$  in (33) it takes one of the following forms:

$$\mathbf{x}[i, j] = \begin{cases} \begin{bmatrix} \mathbf{u}[i, j] \\ \mathbf{u}[i, r + 1] \\ \mathbf{v}[i, 1] \end{bmatrix} & j = \{1, \dots, r\} \\ \begin{bmatrix} \mathbf{u}[i, r + 1] \\ \mathbf{v}[i, 1] \end{bmatrix} & j = r + 1 \\ \begin{bmatrix} \mathbf{v}[i, j - r] \\ \mathbf{p}[i, r + 1] \\ \mathbf{v}[i, j - r] \end{bmatrix} & j = \{r + 2, \dots, M - r - 1\} \\ \begin{bmatrix} \mathbf{p}[i, r + 1] \\ \mathbf{v}[i, j - r] \\ \mathbf{p}[i, M - j + 1] \end{bmatrix} & j = M - r \\ \begin{bmatrix} \mathbf{p}[i, M - j + 1] \end{bmatrix} & j = \{M - r + 1, \dots, M\}. \end{cases} \quad (36)$$

We claim that in each case the number of sub-symbols released by  $\mathbf{x}[i, j]$  compensates for any additional erasures introduced by  $\mathcal{B}_{j+1}$ . In particular note that  $\mathcal{B}_{j+1}$  can introduce no more than  $n$  additional erasures via the last channel packet which is not included in  $\mathcal{B}_j$ .

- $j = \{1, \dots, r\}$ : In this case, the revealed  $\mathbf{x}[i, j] = \mathbf{u}[i, j]$  can be subtracted from  $\mathbf{p}[i + T, j]$  to recover  $\mathbf{q}[i + T, j] \in \mathbf{q}^{\text{vec}}[\cdot]$  having  $n$  sub-symbols. Thus, compensates for the  $n$  extra erased sub-symbols.
- $j = r + 1$ : The  $r'$  sub-symbols of  $\mathbf{u}[i, r + 1]$  helps in recovering the  $r'$  sub-symbols of  $\mathbf{q}[i + T, r + 1] \in \mathbf{q}^{\text{vec}}[\cdot]$ . This together with the revealed  $n - r'$  sub-symbols of  $\mathbf{v}[i, 1] \in \mathbf{v}^{\text{vec}}[\cdot]$  compensates for the  $n$  extra erasures and the claim follows.
- $j = \{r + 2, \dots, M - r - 1\}$ : In this case, the revealed channel packet is  $\mathbf{x}[i, j] = \mathbf{v}[i, j - r] \in \mathbf{v}^{\text{vec}}[\cdot]$  and has  $n$  sub-symbols.
- $j = M - r$ : The decoder can subtract  $\mathbf{u}[i - T, r + 1]$  from  $\mathbf{p}[i, r + 1]$  to recover the  $r'$  sub-symbols  $\mathbf{q}[i, r + 1] \in \mathbf{q}^{\text{vec}}[\cdot]$ . This together with the  $n - r'$  sub-symbols of

$\mathbf{v}[i, M - 2r] \in \mathbf{v}^{\text{vec}}[\cdot]$  add up to  $n$  sub-symbols and the claim follows.

- $j = \{M - r + 1, \dots, M\}$ : Finally, in this case there are no erasures in the source symbols  $\mathbf{s}[i]$  as only the parity check symbols in the macro packet  $i$  are erased. Thus one can ignore these erasures, directly move to the erasures in macro packet  $i + 1$ , and repeat the above argument.

## VI. CONVERSE

To establish the converse to Theorem 1 for the case when  $T > b$  it suffices to show that

$$C \leq \min \left( \frac{M(T + b + 1) - (bM + B')}{M(T + b + 1)}, \frac{T}{T + b} \right). \quad (37)$$

For the first expression, we use the technique of periodic erasure channel ([4], [9], [10]) to derive an upper bound on the rate  $R$ . Consider periodic bursts each of length  $B$  with a guard interval of  $M(b + T + 1) - B$  as shown in Fig. 3. One period of this channel is of length  $T_{\text{period}} = T + b + 1$ . We let the first burst start from  $\mathbf{x}[0, 1]$ . We require the erased source symbol  $\mathbf{s}[0]$  to be recovered by the end of macro-packet  $t = T + b$ . Once  $\mathbf{s}[0]$  has been recovered, we reconstruct the underlying  $\mathbf{X}[0, :]$  and consider the burst starting at macro-packet  $\mathbf{X}[1, :]$  and note that  $\mathbf{s}[1]$  is reconstructed at the end of macro packet  $T + 1$ . Repeating this argument the last erased source vector  $\mathbf{s}[b]$  is recovered at the end of macro packet  $b + T$ . Thus all of  $\mathbf{X}[0, :], \dots, \mathbf{X}[b, :]$  are now reconstructed at the end of macro packet  $b + T$ . Thus as shown in Fig. 3, at the start of macro packet  $b + T + 1$  we can have a second erasure burst and repeat the above argument. Thus we can recover all erased source packets from the above periodic erasure channel. Thus the rate of our streaming code must be upper bounded by the capacity of the periodic erasure channel.

Since the above periodic erasure channel consists of  $M(T + b + 1)$  channel packets in each period out of which  $B$  are erased it follows that

$$C \leq \frac{M(T + b + 1) - (bM + B')}{M(T + b + 1)} \quad (38)$$

which establishes the first part of our upper bound.

To establish the second part of the upper bound, note that when  $B' = 0$  i.e.,  $B = bM$  then in the above argument it is sufficient to take  $T_{\text{period}} = T + b$ . Therefore repeating the above argument in this special case we have that

$$C_{B=bM} \leq \frac{T}{T + b}. \quad (39)$$

Furthermore since the capacity is always a decreasing function in  $B$  (as the receiver can always simulate a longer erasure burst) it follows that (39) is also an upper bound for any  $B = bM + B'$  where  $B' > 0$ . This completes the justification for (37).

It only remains to consider last case with  $B' > \frac{M}{2}$  and  $T = b$  in Theorem 1. The periodic erasure channel argument is not tight and therefore we consider the following argument which simultaneously considers the effect of two different erasure bursts. We start by considering a channel that erases the first  $B = bM + B'$  channel packets  $\left\{ \mathbf{x}[i, 1], \dots, \mathbf{x}[i + b, B'] \right\}$ .

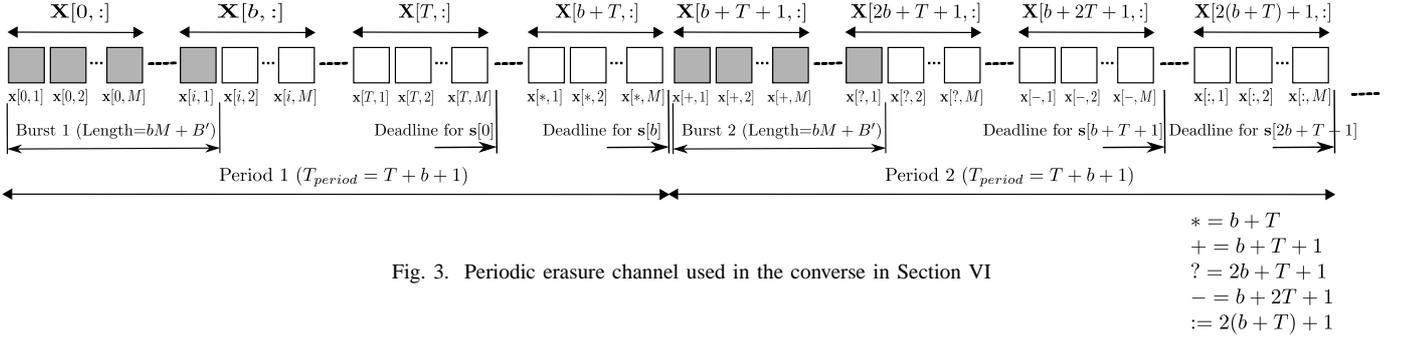


Fig. 3. Periodic erasure channel used in the converse in Section VI

Since the delay constraint for  $s[i]$  is  $i+T = i+b$ , the following equation should be satisfied,

$$H(s[i]|\mathbf{x}[i+b, B'+1], \dots, \mathbf{x}[i+b, M]) = 0. \quad (40)$$

Next we consider the channel erasing the channel packets  $\{\mathbf{x}[i+b, M-B'+1], \dots, \mathbf{x}[i+2b, M]\}$ . To recover  $s[i+b]$  by macro-packet  $i+2b$  the only available symbols are  $\mathbf{x}[i+b, 1], \dots, \mathbf{x}[i+b, M-B']$

$$H(s[i+b]|\mathbf{x}[i+b, 1], \dots, \mathbf{x}[i+b, M-B']) = 0. \quad (41)$$

Combining (40) and (41) and using  $B' < M/2$  we have that

$$H(s[i], s[i+b]|\mathbf{x}[i+b, 1], \dots, \mathbf{x}[i+b, M-B'], \mathbf{x}[i+b, B'+1], \dots, \mathbf{x}[i+b, M]) = 0 \quad (42)$$

Through some standard manipulations it follows that

$$\begin{aligned} H(\mathbf{x}[i+b, 1], \dots, \mathbf{x}[i+b, M-B'], \mathbf{x}[i+b, B'+1], \dots, \mathbf{x}[i+b, M]) \\ \geq H(s[i], s[i+b]) = 2H(s) \end{aligned} \quad (43)$$

Therefore we have that

$$2(M-B')H(\mathbf{x}) \geq 2H(s) \quad (44)$$

and thus

$$C = \frac{H(s)}{MH(\mathbf{x})} \leq \frac{M-B'}{M}. \quad (45)$$

as required.

## VII. NUMERICAL COMPARISONS AND SIMULATIONS

Fig. 4 illustrates a numerical example comparing capacity with some baseline schemes. The achievable rate is shown on the y-axis and the associated erasure burst length is shown on the x-axis. We consider  $M = 20$  and a delay of  $T = 5$  macro packets. We plot  $B \in [40, 80]$  in the left figure and  $B \in [80, 120]$  in the right figure.

The capacity in each plot is shown by the blue-curve marked with squares whereas the red curve marked with circles denotes the rate achieved by a suitable modification of the SCo code [4], [6] which is discussed in Section IV-B. We note that the curves intersect whenever  $B$  is an integer multiple of  $M$ , indicating the optimality of the SCo codes for these special values i.e., at  $B = \{40, 60, 80, 100\}$ . Furthermore for burst lengths  $B > MT = 100$ , SCo codes are not feasible and the associated rate is zero. The capacity function is constant in the intervals  $B \in [40, 45], [60, 67], [80, 88], [100, 110]$ , as indicated in (10) and monotonically decreasing in the rest of the intervals. The third class of codes — Baseline Erasure Codes — discussed in Section IV-A are erasure codes that only

*simultaneously* recover all the erased source symbols after the erasure burst. Since they do not perform sequential recovery, their achievable rates is significantly lower.

In our simulations in Fig. 5, we consider a two-state Gilbert channel model. In the bad state, each channel packet is lost with a probability of 1 whereas in the good state, the loss probability is 0. We let  $\alpha$  and  $\beta$  denote the transition probability from the good state to the bad state and vice versa, respectively for this channel. Note that the average burst length for this channel is  $\frac{1}{\beta}$  whereas the average loss rate is  $\frac{\alpha}{\alpha+\beta}$ .

In Fig. 5(a), we select  $\alpha = 10^{-5}$  and  $\beta$  is varied on the x-axis in the interval  $[0.1, 0.4]$  which in turn changes the burst length distribution. We further select  $M = 10$ , i.e., 10 channel packets are generated for every source packet received at the encoder. We fix the rate  $R = 3/5$  and the delay  $T = 3$  macro packets. Under these conditions, the BEC code can correct burst erasures of length up to  $B_{BEC} = 16$ , whereas a Streaming Code (SCo) achieves  $B_{SCo} = 20$ . The optimal code achieves  $B = 24$ . This gain in the burst-length is reflected in Fig. 5(a) as one can see that the proposed codes achieve a smaller loss probability. While the code parameters in Fig. 5(a) correspond to the first case in (7) the code parameters used in Fig. 5(b) correspond to the second case in (7). In this case we select  $M = 20$ ,  $T = 4$  and  $R = 9/14$ . The achievable burst lengths for the BEC and SCo codes are  $B_{BEC} = 35$ ,  $B_{SCo} = 44$  while the optimal codes achieve  $B = 50$ . We again select  $\alpha = 10^{-5}$  and vary  $\beta$  on the x-axis as illustrated.

## VIII. CONCLUSIONS

Motivated by the application to wireless video, we propose a new family of low-delay streaming codes when there is a mismatch between the source frame rate and channel transmission rate. Our proposed codes are optimal over the burst-erasure channel. We show that a naive extension of previously proposed streaming codes designed when the source-channel rates are matched can be sub-optimal. We also explicitly characterize the associated capacity and show that it remains constant over a certain interval of burst-lengths, as illustrated in Fig. 4. Simulation results over the Gilbert channel are also presented to show the improvements from the proposed codes in achievable packet-loss rate.

In this paper we only focused on the case when the channel is an erasure burst channel. We expect that our constructions can be naturally extended to the case when the channel introduces both burst and isolated erasures. Such an extension

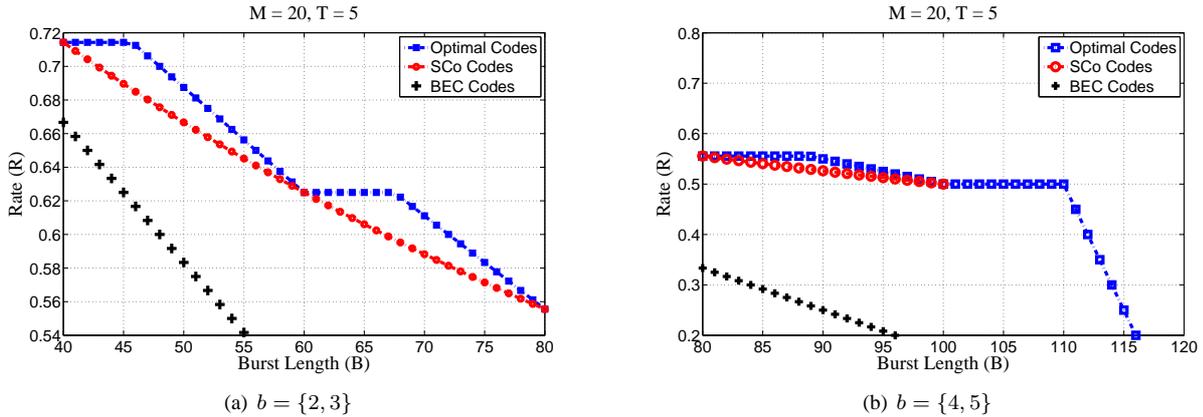


Fig. 4. Achievable rates for different code constructions for a given burst length  $B$  and delay of  $T = 5$  macro packets with each having  $M = 20$  channel packets.

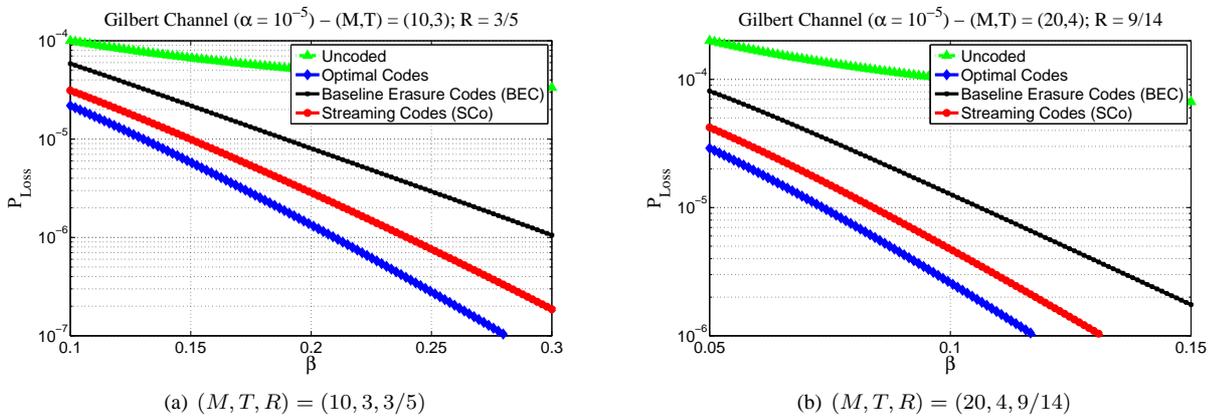


Fig. 5. Gilbert Channel Experiments with different parameters illustrating the loss probabilities of different code constructions.

can be done using a layered approach as was done for the case of matched source-channel rates in [6], [7]. This extension is left for a future investigation.

## REFERENCES

- [1] T. Stockhammer and M. Hannuksela, "H.264/AVC video for wireless transmission," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 613, Aug. 2005.
- [2] Z. Li, A. Khisti and B. Girod, "Forward error correction for low-delay packet video," in *Packet Video Workshop*, Dec. 2010
- [3] E. Martinian, "Dynamic Information and Constraints in Source and Channel Coding," *Ph.D. Thesis*, MIT, September 2004
- [4] Emin Martinian and Carl-Erik W. Sundberg, "Burst Erasure Correction Codes With Low Decoding Delay," *IEEE Transactions on Information Theory*, October 2004.
- [5] Emin Martinian and Mitchell Trott, "Delay-optimal Burst Erasure Code Construction," *International Symposium on Information Theory*, (Nice, France), July 2007.
- [6] A. Badr and A. Khisti and W. Tan and J. Apostolopoulos, "Streaming Codes for Channels with Burst and Isolated Erasures," *INFOCOM*, Turin, Italy, 2013.
- [7] A. Badr, A. Khisti, W. Tan and J. Apostolopoulos, "Robust Streaming Erasure Codes for Channels with Burst and Isolated Erasures," in *ISIT Istanbul*, Turkey, 2013
- [8] A. Khisti and J. Singh, "On multicasting with streaming burst-erasure codes," in *ISIT*, 2009.
- [9] A. Badr, A. Khisti and E. Martinian, "Diversity Embedded Streaming Erasure Codes (DE-SCo): Construction and Optimality," *IEEE Journal on Selected Areas in Communications (JSAC)*, special issue on trading rate for delay at the transport and application layers, vol. 29, no. 5, pp. 1042-1054, May 2011
- [10] A. Badr, D. Lui and A. Khisti, "Multicast Streaming Codes (Mu-SCo) for Burst Erasure Channels," *Allerton Conf.*, Monticello, IL, USA, 2010.
- [11] A. Badr, D. Lui and A. Khisti, "Streaming-Codes for Multicast over Burst Erasure Channels," submitted *IEEE Trans. Inform. Theory*, March 2013, [online] <http://arxiv.org/abs/1303.4370>
- [12] Z. Li, A. Khisti and B. Girod, "Correcting erasure bursts with minimum decoding delay," *Asilomar Conference on Signals, Systems and Computers*, Nov. 2011, Pacific Grove, CA
- [13] D. Lui, A. Badr and A. Khisti, "Streaming codes for a double-link burst erasure channel," *Canadian Workshop on Information Theory (CWIT)*, May 2011
- [14] D. Lui, "Coding Theorems for Delay Sensitive Communication over Burst-Erasure Channels," *MASc Thesis*, University of Toronto, 2011
- [15] O. Tekin, T. Ho, H. Yao, and S. Jaggi, "On erasure correction coding for streaming," in *ITA*, 2012, pp. 221-226.
- [16] D. Leong and T. Ho, "Erasure coding for real-time streaming," in *ISIT*, 2012.
- [17] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, 2006,
- [18] Y. Li, P. Vingelmann, M. V. Pedersen, and E. Soljanin, "Round-robin streaming with generations," *CoRR*, vol. abs/1206.3014, 2012.
- [19] S. Nazir, D. Vukobratovic, and V. Stankovic, "Expanding window random linear codes for data-partitioned H.264 video transmission over DVB-H network," *IEEE ICASSP*, 2011
- [20] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, "Strongly MDS convolutional codes," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 584-598, 2006.

- [21] E. M. Gabidulin, "Convolutional codes over large alphabets," in Proc.Int. Workshop on Algebraic Combinatorial and Coding Theory, Varna,Bulgaria, 1988, pp. 8084.