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 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 their 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.

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. Robust extensions that are resilient against both burst and isolated losses are reported in [8]. For other related works on low-delay streaming codes we refer to [9]–[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 destination after a delay of T macro-packets (or equivalently T source 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.

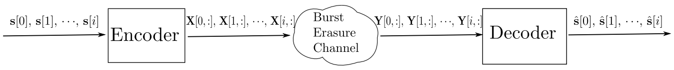


Fig. 1. System under consideration. Each $\mathbf{X}[i,:]$ denotes a (channel) macro-packet consisting of M channel packets $(\mathbf{x}[i,1],\ldots,\mathbf{x}[i,M])$. One source packet arrives at the start of each macro packet. The channel erases up to B consecutive channel packets. Each source packet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

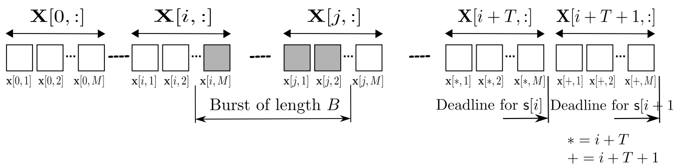


Fig. 2. Channel Model. The erasure burst spans a total of B channel symbols as shown. Each source packet $\mathbf{s}[i]$ arrives just before the transmission of $\mathbf{X}[i,:]$ and needs to be reconstructed by the destination after a delay of T macro-packets.

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 $\mathbf{x}[i,j] \in \mathbb{F}_q^n$, $j = \{1,\ldots,M\}$ which can depend on all the observed source packets up to that time i.e.[1],

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

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

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

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

Channel: The received packets corresponding to macro-packet i are denoted by $\mathbf{y}[i,j]$ for $j = \{1,\ldots,M\}$. We assume a burst erasure channel. The channel can 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 $\mathbf{X}[i_f,:]$. Thus, the output channel packets are given by,

$$\mathbf{y}[i,j] = \begin{cases} \star, & \text{for } [i,j] \in \{[i_s,j_s],[i_f,j_f]\}, \\ \mathbf{x}[i,j], & \text{otherwise.} \end{cases} \tag{3}$$

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 $\mathbf{Y}[i,:] = [\mathbf{y}[i,1] \mid \ldots \mid \mathbf{y}[i,M]] \in \mathbb{F}_q^{n\times M}$, where again $\mathbf{y}[\cdot]$ denotes a column vector of length n.

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 $g_i(.)$:

$$\hat{\mathbf{s}}[i] = g_i(\mathbf{Y}[0,:],\mathbf{Y}[1,:],\cdots,\mathbf{Y}[i+T,:]). \tag{4}$$

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.,

$$R = \frac{\mathsf{H}(\mathbf{s})}{n \times M}. \tag{5}$$

Note that in (5) we assume that the source sequence $\{\mathbf{s}[i]\}_{i\geq 0}$ is sampled i.i.d. from a distribution $p_{\mathbf{s}}(\cdot)$. 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.

[1]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.

III. 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:

$$C = \begin{cases} \frac{T}{T+b}, & B' \leq \frac{b}{T+b}M,\ T \geq b, \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & B' > \frac{b}{T+b}M,\ T > b, \\ \frac{M-B'}{M}, & B' > \frac{M}{2},\ T = b, \\ 0, & T < b. \end{cases} \tag{6}$$

where the constants b and B' are defined via

$$B = bM + B', \quad B' \in \{0,\ldots,M-1\}, b \in \mathbb{N}^0. \tag{7}$$

$\square$

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 (6).

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

$$C = \begin{cases} \frac{1}{2}, & 0 \leq B' \leq \frac{M}{2},\ T = b, \\ \frac{M-B'}{M}, & \frac{M}{2} \leq B' \leq M,\ T = b. \end{cases} \tag{8}$$

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 $\min\left\{M-B', \frac{M}{2}\right\}$ 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

$$C = \begin{cases} \frac{T}{T+b}, & 0 \leq B' \leq \frac{b}{T+b}M, T > b \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & \frac{b}{T+B}M < B' \leq M-1, T > b \end{cases} \tag{9}$$

Examining (9) 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 — Strongly-MDS 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 Therem 1. Nevertheless our proposed codes build upon these ideas, and hence their review is essential before stating the proposed construction.

A. Strognly-MDS 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. In this section, we review error correction properties of a class of deterministic codes - Strongly-MDS codes [17], [18] that are relevant for our streaming setup.

Consider a systematic Strongly-MDS (n, k, m) 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.,

$$\mathbf{x}[i] = \left(\sum_{t=0}^{m} \mathbf{s}^\dagger[i-t] \cdot \mathbf{G}_t \right)^\dagger \tag{10}$$

where $\mathbf{G}_0, \ldots, \mathbf{G}_m$ are $k \times n$ matrices with elements in $\mathbb{F}_q$. Let $\mathbf{x}[i] = \begin{bmatrix} \mathbf{s}[i] \\ \mathbf{p}[i] \end{bmatrix}$, and suppose that the sub-symbols in $\mathbf{x}[i] = (s_1[i], \ldots, s_k[i], p_1[i], \ldots, p_{n-k}[i])^\dagger$ are transmitted sequentially in the interval $[i \cdot n, (i+1)n - 1]$ over the channel. Then the code results in following error correction properties in the streaming setup. For proof, see [8, Appendix B].

Lemma 1:

1) For any $j \in [0, m]$, the following holds: if no more than $(n-k)(j+1)$ sub-symbols are erased in the interval $[0, (j+1)n - 1]$ the source symbol $\mathbf{s}[0] = (s_1[0], \ldots, s_k[0])$ 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 $B \leq (n-k)(j+1)$, 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

$$B \leq \frac{(n-k)}{n} M(T+1). \tag{11}$$

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

$$R^{\text{BEC}} = 1 - \frac{B}{M(T+1)} \tag{12}$$

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:
1. Split each source symbol $\mathbf{s}[i] \in \mathbb{F}_q^T$ into two groups $\mathbf{u}[i] \in \mathbb{F}_q^B$ and $\mathbf{v}[i] \in \mathbb{F}_q^{T-B}$.
2. Apply a BEC code from the previous sub-section on the symbols $\mathbf{v}[i]$ and generate parity-check symbols

$$\mathbf{p}_v^\dagger[i] = \sum_{j=1}^{T} \mathbf{v}^\dagger[i-j] \cdot \mathbf{H}_j^v, \quad \mathbf{p}_v[i] \in \mathbb{F}_q^B, \tag{13}$$

where the matrices $\mathbf{H}_j^v$ are $(T-B) \times B$ matrices associated with the systematic Strongly-MDS code.
3. Super-impose the $\mathbf{u}[\cdot]$ symbols onto $\mathbf{p}_v[\cdot]$ and let

$$\mathbf{q}[i] = \mathbf{p}_v[i] + \mathbf{u}[i-T]. \tag{14}$$

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

For decoding of the SCo codes from an erasure burst starting at time i, the interfering $\mathbf{u}[\cdot]$ symbols (c.f. (14)) until time $t = i + T - 1$ which have not been erased are canceled from parity checks $\mathbf{q}[\cdot]$. All the lost $\mathbf{v}[\cdot]$ symbols are then recovered by time $t = i + T - 1$. Once all the $\mathbf{v}[\cdot]$ symbols have been recovered, each $\mathbf{u}[i], \ldots, \mathbf{u}[i+B-1]$ can be recovered at their deadline by canceling $\mathbf{p}[\cdot]$ from the associated $\mathbf{q}[\cdot]$ 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 $\mathbf{s}[i] = (\mathbf{w}[i, 1], \ldots, \mathbf{w}[i, M])$ where $\mathbf{w}[i, j] \in \mathbb{F}_q^T$ holds.
- Apply a (B, MT) SCo code of rate

$$R^{\text{SCO}} = \frac{MT}{MT+B} = \frac{T}{T + b + \frac{B'}{M}} \tag{15}$$

  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 $j \in \{1, 2, \ldots, M\}$. 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 (15) 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 present the encoding steps and the decoding analysis for $T > b$ in (6) in Theorem 1. The case when $T = b$ uses a repetition code and will not be treated due to space constraints.

A. Encoding

The main encoding steps are as described below:

1) **Source Splitting:** 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:
$$\begin{aligned}\mathbf{s}[i] &= (s_1[i], \ldots, s_k[i]) \\ &= (\underbrace{u_1[i], \ldots, u_{k_u}[i]}_{\mathbf{u}_{\text{vec}}[i]}, \underbrace{v_1[i], \ldots, v_{k_v}[i]}_{\mathbf{v}_{\text{vec}}[i]})\end{aligned} \tag{16}$$
Note that $k_u + k_v = k$.
2) **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, $\mathbf{q}_{\text{vec}}[i] = (q_1[i], \ldots, q_{k_u}[i]) \in \mathbb{F}_q^{k_u}$ for each macro-packet. In particular we have that
$$\mathbf{q}_{\text{vec}}[i] = \left(\sum_{j=0}^{T} \mathbf{v}_{\text{vec}}^{\dagger}[i-j] \cdot \mathbf{H}_j \right)^{\dagger} \tag{17}$$
where $\mathbf{H}_j \in \mathbb{F}_q^{k_v \times k_u}$ are the sub-matrices associated with the BEC code.
3) **Parity-Check Generation:** Combine the $\mathbf{q}_{\text{vec}}[\cdot]$ parity-checks with the $\mathbf{u}_{\text{vec}}[\cdot]$ symbols after applying a shift of T to generate final parity-checks $\mathbf{p}_{\text{vec}}[i] \in \mathbf{F}_q^{k_u}$ i.e.,
$$\mathbf{p}_{\text{vec}}[i] = \mathbf{q}_{\text{vec}}[i] + \mathbf{u}_{\text{vec}}[i-T]. \tag{18}$$
4) **Re-shaping:** In order the construct the macro-packet $\mathbf{X}[i,:]$, 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 following matrices:
$$\mathbf{U}[i,:] = \left[\begin{array}{c|c|c|c} \mathbf{u}[i,1] & \cdots & \mathbf{u}[i,r] & \frac{\mathbf{u}[i,r+1]}{\mathbf{0}} \end{array} \right] \in \mathbb{F}_q^{n \times r+1}$$
$$\mathbf{V}[i,:] =$$
$$\left[\begin{array}{c|c|c|c|c} \frac{\mathbf{0}}{\mathbf{v}[i,1]} & \mathbf{v}[i,2] & \cdots & \mathbf{v}[i,M-2r-1] & \frac{\mathbf{0}}{\mathbf{v}[i,M-2r]} \end{array} \right] \in \mathbb{F}_q^{n \times M-2r}$$
$$\mathbf{P}[i,:] = \left[\begin{array}{c|c|c|c} \frac{\mathbf{p}[i,r+1]}{\mathbf{0}} & \mathbf{p}[i,r] & \cdots & \mathbf{p}[i,1] \end{array} \right] \in \mathbb{F}_q^{n \times r+1}$$
where
$$\mathbf{u}^{\text{vec}}[i] = \begin{bmatrix} \mathbf{u}[i,1] \\ \mathbf{u}[i,2] \\ \vdots \\ \mathbf{u}[i,r+1] \end{bmatrix}, \mathbf{v}^{\text{vec}}[i] = \begin{bmatrix} \mathbf{v}[i,1] \\ \mathbf{v}[i,2] \\ \vdots \\ \mathbf{v}[i,M-2r] \end{bmatrix}, \mathbf{p}^{\text{vec}}[i] = \begin{bmatrix} \mathbf{p}[i,1] \\ \mathbf{p}[i,2] \\ \vdots \\ \mathbf{p}[i,r+1] \end{bmatrix},$$
$r \in \mathbb{N}^0$ is defined via $k_u = r \cdot n + r'$ for $r' \in \{0, 1, \ldots, n-1\}$.
Note that $\mathbf{u}[i,j] \in \mathbb{F}_q^n$ for each $j \in \{1, \ldots, r\}$ and $\mathbf{u}[i, r+1] \in \mathbb{F}_q^{r'}$. The splitting of $\mathbf{p}^{\text{vec}}[i]$ into $\mathbf{p}[i,j]$ follows in an analogous manner. In particular we can write
$$\mathbf{p}[i,j] = \mathbf{u}[i-T,j] + \mathbf{q}[i,j], \quad j = 1, \ldots, r+1 \tag{19}$$
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 $2 \leq j \leq M-2r-1$.
5) **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[2]
$$\mathbf{X}[i,:] = [\mathbf{x}[i,1], \ldots, \mathbf{x}[i,M]] =$$
$$\left[\begin{array}{c|c|c|c|c|c} \mathbf{u}[i,1] & \cdots & \mathbf{u}[i,r] & \begin{matrix}\mathbf{u}[i,r+1] \\ \mathbf{v}[i,1]\end{matrix} & \mathbf{v}[i,2] & \cdots \end{array} \right.$$
$$\left. \begin{array}{c|c|c|c|c} \cdots \quad \mathbf{v}[i,M-2r-1] & \begin{matrix}\mathbf{p}[i,r+1] \\ \mathbf{v}[i,M-2r]\end{matrix} & \mathbf{p}[i,r] & \cdots & \mathbf{p}[i,1] \end{array} \right]. \tag{20}$$
Note that the channel macro-packet at time i is denoted by $\mathbf{X}[i,:] \in \mathbb{F}_q^{n \times M}$ and the jth channel packet in $\mathbf{X}[i,:]$ by $\mathbf{x}[i,j] \in \mathbb{F}_q^n$ for $j \in \{1, \ldots, M\}$. Since each macro-packet has $k_u + k_v$ source sub-symbols and k_u parity-check sub-symbols, we have that $2k_u + k_v = nM$.

Rate of the code described above is $R = \frac{k}{nM} = \frac{k_u + k_v}{2k_u + k_v}$. We choose following parameters for the two cases in Theorem 1.
1. $B' \leq \frac{b}{T+b}M$: $k_u = Mb$, $k_v = M(T-b)$
2. $B' > \frac{b}{T+b}M$: $k_u = B'$, $k_v = M(T+b+1) - 2B$

B. Decoding

We show that above code construction can completely recover from any arbitrary burst of length B within the deadline. We consider a channel that introduces such a burst of length $B = bM + B'$ starting from $\mathbf{x}[i,j]$ for $j \in \{1, \ldots, M\}$. The total number of patterns to consider is M.

We begin by considering the burst pattern starting at $\mathbf{x}[i,1]$ which erases $\mathbf{X}[i], \ldots, \mathbf{X}[i+b-1], \mathbf{x}[i+b,1], \ldots, \mathbf{x}[i+b,B']$. We will then discuss the cases when the burst 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. (18)).
2) Recover all erased $\mathbf{v}_{\text{vec}}[\cdot]$ symbols by macro-packet $i + T - 1$ using the underlying BEC code.
3) Compute $\mathbf{q}_{\text{vec}}[i+T], \ldots, \mathbf{q}_{\text{vec}}[i+T+b]$ as they combine $\mathbf{v}_{\text{vec}}[\cdot]$ symbols which are either not erased or recovered in the previous step.
4) Subtract $\mathbf{q}_{\text{vec}}[i+T], \ldots, \mathbf{q}_{\text{vec}}[i+T+b]$ from $\mathbf{p}_{\text{vec}}[i+T], \ldots, \mathbf{p}_{\text{vec}}[i+T+b]$ to recover $\mathbf{u}_{\text{vec}}[i], \ldots, \mathbf{u}_{\text{vec}}[i+b]$ 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 $(k_u + k_v, k_v, T)$ 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. For exact details, refer [8, Appendix D]

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 $j = \{1, \ldots, M\}$. The main decoding steps are

[2] 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.

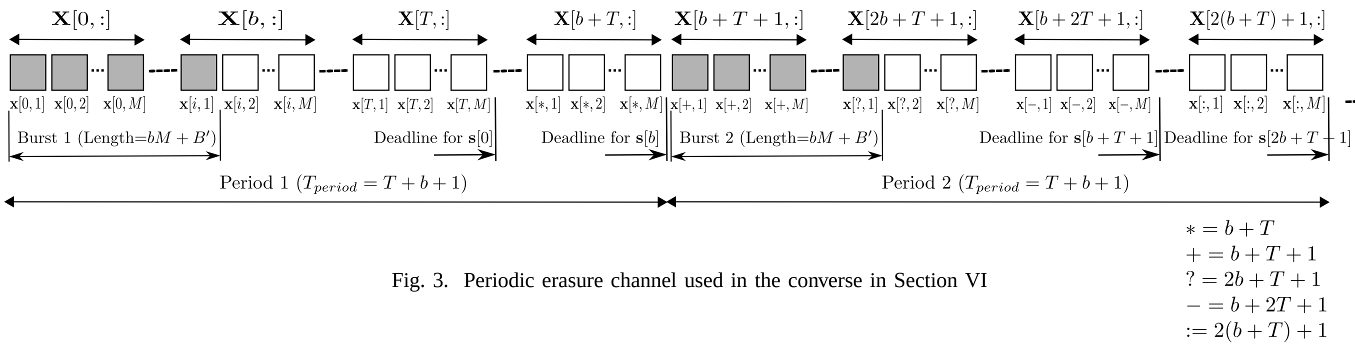


Fig. 3. Periodic erasure channel used in the converse in Section VI

similar to those described above where we first recover *all* the erased $\mathbf{v}_{\text{vec}}[\cdot]$ simultaneously and then sequentially recover $\mathbf{u}_{\text{vec}}[\cdot]$ at their respective deadlines. To show the sufficiency of BEC code for the recovery of $\mathbf{v}^{\text{vec}}[i]$, we can 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. Due to space constraints, we skip the details of the argument here. Readers are referred to [8, Section V-C] for in depth decoding analysis.

VI. Converse

To establish the converse to Theorem 1, we first consider the case $T > b$. We show that any achievable rate R must satisfy

$$R \leq \min\left(\frac{M(T+b+1)-(bM+B')}{M(T+b+1)}, \frac{T}{T+b}\right). \tag{21}$$

Consider a periodic erasure channel with periodic bursts of length B and guard intervals of length $M(b+T+1)-B$ macro-packets as shown in Fig. 3. Each period contains $T_{period} = T+b+1$ macro-packets. By definition, we require $\mathbf{s}[0]$ to be recovered by the end of macro-packet $t = T$, $\mathbf{s}[1]$ by macro-packet $T+1$ and so on. The last erased source packet $\mathbf{s}[b]$ in the first period is to be recovered at the end of macro packet $b+T$. Thus all of $\mathbf{X}[0,:], \ldots, \mathbf{X}[b,:]$ can reconstructed at the end of macro-packet $b+T$ and we can treat these erasures as having never happened and repeat the argument for the next period. Thus any streaming code must be a feasible code for such a channel. Since the capacity of the periodic erasure channel is just the fraction of non-erased symbols, it follows that

$$R \leq \frac{M(T+b+1)-(bM+B')}{M(T+b+1)} \tag{22}$$

which establishes the first inequality in (21). To establish the second inequality, we consider a periodic erasure channel with burst lengths $\hat{B} = bM \leq B$. We can see that it is sufficient to take $T_{period} = T+b$. Therefore repeating the above argument we have that

$$R \leq \frac{T}{T+b}. \tag{23}$$

For the case $T = b$, we can easily show that

$$R \leq \min\left(\frac{M-B'}{M}, \frac{1}{2}\right). \tag{24}$$

Combining (21) and (24), the converse follows.

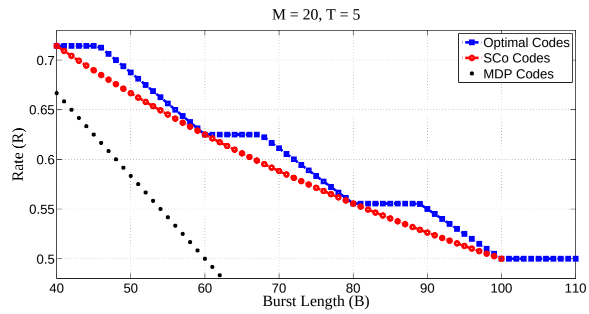
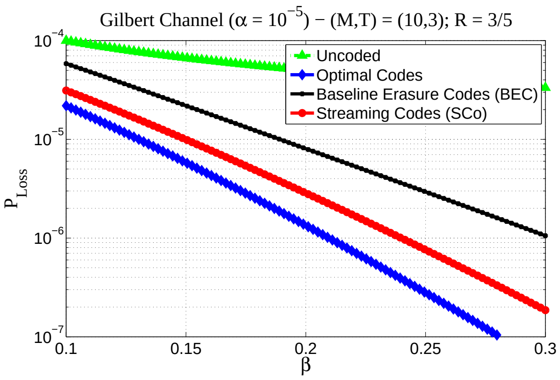


Fig. 4. Achievable rates for different code constructions for a given burst length B and delay of $T = 5$ macro packets with each having $M = 20$ channel packets.

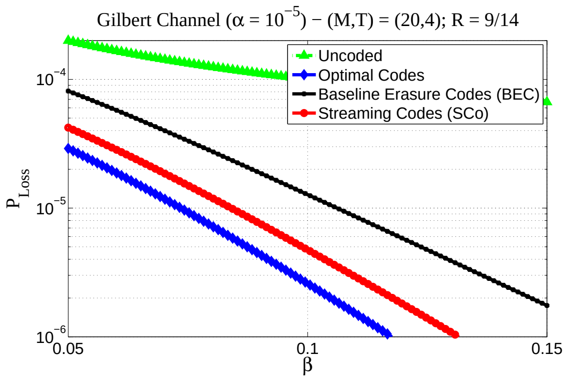
VII. Numerical Comparisons and Simulations

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 and burst length B is varied from 40 to 110. The capacity 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 (9) 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 are 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 α and β 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 $\frac{\alpha}{\alpha+\beta}$.



(a) $(M, T, R) = (10, 3, 3/5)$



(b) $(M, T, R) = (20, 4, 9/14)$

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

In Fig. 5(a), we select $\alpha = 10^{-5}$ and β 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 (6) the code parameters used in Fig. 5(b) correspond to the second case in (6). 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 β 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 the 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. Our constructions can be naturally extended to the case when the channel introduces both burst and isolated erasures. Such an extension can be done using a layered approach as was done for the case of matched source-channel rates in [6], [7] and is discussed in [8].

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. Apostolopolous, "Robust Streaming Erasure Codes for Channels with Burst and Isolated Erasurs," in ISIT Istanbul, Turkey, 2013

[8] A. Badr, P. Patil, A. Khisti, W. Tan and J. Apostolopoulos, "Layered Construction for Low-Delay Streaming Codes," submitted IEEE Trans. Inform. Theory, August 2013, [online] http://arxiv.org/abs/1308.3827

[9] A. Khisti and J. Singh, "On multicasting with streaming burst-erasure codes," in *ISIT*, 2009.

[10] 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

[11] A. Badr, D. Lui and A. Khisti, "Multicast Streaming Codes (Mu-SCo) for Burst Erasure Channels," *Allerton Conf.*, Monticello, IL, USA, 2010.

[12] 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

[13] 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

[14] D. Lui, A. Badr and A. Khisti, "Streaming codes for a double-link burst erasure channel, Canadian Workshop on Information Theory (CWIT), May 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] 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.

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