# Diversity Embedded Streaming Erasure Codes (DE-SCo): Constructions and Optimality

Ahmed Badr, Ashish Khisti, *Member IEEE*, and Emin Martinian

***Abstract*—Streaming erasure codes encode a source stream to guarantee that each source packet is recovered within a fixed delay at the receiver over a burst-erasure channel. This paper introduces *diversity embedded streaming erasure codes* (DE-SCo), that provide a flexible tradeoff between the channel quality and receiver delay. When the channel conditions are good, the source stream is recovered with a low delay, whereas when the channel conditions are poor the source stream is still recovered, albeit with a larger delay. Information theoretic analysis of the underlying burst-erasure broadcast channel reveals that DE-SCo achieve the minimum possible delay for the weaker user, without sacrificing the performance of the stronger user. Our constructions are explicit, incur polynomial time encoding and decoding complexity and outperform random linear codes over burst-erasure channels.**

***Index Terms*—Low Delay, Streaming Erasure Correction Codes, Burst Erasure Channel, Broadcast Channel, Network Information Theory, Delay Constrained Coding, Application Layer Error Correction**

## I. INTRODUCTION

Forward error correction codes designed for streaming sources require that (a) the channel input stream be produced sequentially from the source stream (b) the decoder sequentially reconstructs the source stream as it observes the channel output. In contrast, traditional error correction codes such as maximum distance separable (MDS) codes map blocks of data to a codeword and the decoder waits until the entire codeword is received before the source data can be reproduced. Rateless codes such as the digital fountain codes also do not form ideal streaming codes. First they require that the entire source data be available before the output stream is reproduced. Secondly they provide no guarantees on the sequential reconstruction of the source stream. Non-block codes such as convolutional codes in conjunction with sequential decoding can be designed for low delay applications [4]. However to our best knowledge, these constructions need to be optimized through a numerical search for finite constraint lengths. Low-delay codes with feedback are recently studied in [5], [6] while compression of streaming sources is studied in [7].

In [1, Chapter 8] a new class of codes, *streaming erasure codes* (SCo) are proposed. The encoder observes a semi-infinite source stream — one packet is revealed in each time slot — and maps it to a coded output stream of rate $R$. The channel is modelled as a burst-erasure channel. Starting at an arbitrary time, it introduces an erasure-burst of maximum length $B$. The decoder is required to reconstruct each source packet with a maximum delay $T$. A fundamental relationship between $R$, $B$ and $T$ is established and SCo codes are constructed that achieve this tradeoff. We emphasize that the parity check symbols in these constructions involve a careful combination of source symbols. In particular, random linear combinations, popularly used in e.g., network coding, do not attain the optimal performance.

The SCo framework however requires that the value of $B$ and $T$ be known apriori. In practice this forces a conservative design i.e., we design the code for the worst case $B$ thereby incurring a higher overhead (or a larger delay) even when the channel is relatively good. Moreover there is often a flexibility in the allowable delay. Techniques such as adaptive media playback [8] have been designed to tune the play-out rate as a function of the received buffer size to deal with a temporary increase in delay. Hence it is not desirable to have to fix $T$ during the design stage either.

The streaming codes introduced in this work do not commit apriori to a specific delay. Instead they realize a delay that depends on the channel conditions. At an information theoretic level, our setup extends the point-to-point link in [1] to a multicast model — there is one source stream and two receivers. The channel for each receiver introduces an erasure-burst of length $B_i$ and each receiver can tolerate a delay of $T_i$. We investigate diversity embedded streaming erasure codes (DE-SCo). These codes modify a single user SCo such that the resulting code can support a second user, whose channel introduces a larger erasure-burst, without sacrificing the performance of the first user. Our construction embeds new parity checks in an SCo code in a manner such that (a) no interference is caused to the stronger (and low delay) user and (b) the weaker user can use some of the parity checks of the stronger user as side information to recover part of the source symbols. DE-SCo constructions outperform baseline schemes that simply concatenate the single user SCo for the two users. An information theoretic converse establishes that DE-SCo achieves the minimum possible delay for the weaker receiver without sacrificing the performance of the stronger user. Finally all our code constructions can be encoded and decoded with a polynomial time complexity in $T$ and $B$.

## II. SYSTEM MODEL

The transmitter encodes a stream of source packets $\{s[t]\}_{t\geq 0}$ intended to be received at two receivers as shown in Fig. 1. The channel packets $\{x[t]\}_{t\geq 0}$ are produced causally from the source stream, .

Ahmed Badr and Ashish Khisti are with the University of Toronto, Toronto, ON, Canada email: {akhisti, abadr}@comm.utoronto.ca. Emin Martinian is with the Massachusetts Institute of Technology (MIT), Cambridge, MA, 02139, USA. Email: emin@alum.mit.edu. Financial support for this work is provided in part by a Natural Science Engineering and Research Council (NSERC) discovery research grant. Part of this paper will appear in IEEE-GLOBECOM Conference, 2010, Miami, FL.

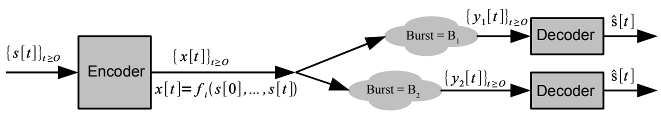


Fig. 1: The source stream $\{\mathsf{s}[i]\}$ is causally mapped into an output stream $\{\mathsf{x}[i]\}$. Both the receivers observe these packets via their channels. The channel introduces an erasure-burst of length $B_i$, and each receiver tolerates a delay of $T_i$, for $i = 1, 2$.

$$x[t] = f_t(s[0], \ldots, s[t]) \tag{1}$$

The channel of receiver $i$ introduces an erasure-burst of length $B_i$ i.e., the channel output at receiver $i$ at time $t$ is given by

$$y_i[t] = \begin{cases} \star & t \in [j_i, j_i + B_i - 1] \\ x[t] & \text{otherwise} \end{cases} \tag{2}$$

for $i = 1, 2$ and for some $j_i \geq 0$. Furthermore, user $i$ tolerates a delay of $T_i$, i.e., there exists a sequence of decoding functions $\gamma_{1t}(.)$ and $\gamma_{2t}(.)$ such that

$$\hat{s}_i[t] = \gamma_{it}(y_1[0], y_1[1], \ldots, y_1[i + T_i]), \qquad i = 1, 2, \tag{3}$$

and $\Pr(s_i[t] \neq \hat{s}_i[t]) = 0, \quad \forall t \geq 0,$ .

The source stream is an i.i.d. process and for convenience we assume that each symbol is an element of $\mathbb{F}_Q^T$; each source symbol is sampled from a distribution $p_{\mathsf{s}}(\cdot)$. The rate of the multicast code is defined as ratio of the (marginal) entropy of the source symbol to the (marginal) entropy of each channel symbol i.e., $R = H(\mathsf{s})/H(\mathsf{x})$. An *optimal multicast streaming erasure code (MU-SCo)* achieves the maximum rate for a given choice of $(B_i, T_i)$. Of particular interest is the following subclass.

**Definition 1** (Diversity Embedded Streaming Erasure Codes (DE-SCo))**.** *Consider the multicast model in Fig. 1 where the channels of the two receivers introduce an erasure burst of lengths $B_1$ and $B_2$ respectively with $B_1 < B_2$. A DE-SCo is a rate $R = \frac{T_1}{T_1+B_1}$ MU-SCo construction that achieves a delay $T_1$ at receiver 1 and supports receiver 2 with delay $T_2$. An optimal DE-SCo minimizes the delay $T_2$ at receiver 2 for given values of $B_1$, $T_1$ and $B_2$.*

Our setup generalizes the point-to-point case in [1, Chapter 8] where single user SCo codes for parameters $(B_1, T_1)$ achieve the streaming capacity $\frac{T_1}{T_1+B_1}$. An optimal MU-SCo construction, attains the maximum rate for fixed parameters $(B_i, T_i)$. An optimal DE-SCo fixes the rate to the capacity of user 1, and supports user 2 with the minimum possible delay $T_2$. In this paper we focus on DE-SCo constructions. The more general MU-SCo constructions will be treated in a subsequent paper.

Note that our model only considers a single erasure burst on each channel. As is the case with (single user) SCo, our constructions correct multiple erasure-bursts separated sufficiently apart. Also we only consider the erasure channel model. It naturally arises when these codes are implemented in application layer multimedia encoding. More general channel models can be transformed into an erasure model by applying an appropriate inner code [1, Chapter 7].

## III. Background: Streaming Codes (SCo)

Streaming burst-erasure codes developed in [1] and [2] are single user codes for the model in the previous section. They correct an erasure burst of length $B$ with a delay of $T$ symbols and achieve the largest possible rate

$$C = \begin{cases} \frac{T}{T+B} & T \geq B \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

### A. Construction

The construction in [1] is described three steps.

1) Create $(T, T-B)$ Burst Erasure Block Code (BEBC)
The construction begins with a systematic generator matrix $\mathbf{G}$ for a $(T, T-B)$ Burst Erasure Block Code (BEBC) over a finite field $\mathbb{F}_Q$, without regard to decoding delay. The code must also correct "end-around" bursts. Recall that any $(n', k')$ cyclic code corrects burst erasures of length $n' - k'$. Since the matrix $\mathbf{G}$ is systematic we can express it in the form

$$\mathbf{G} = \begin{array}{cc} & \begin{array}{cc} T-B & B \end{array} \\ (T-B) & \begin{bmatrix} \mathbf{I} & \mathbf{H} \end{bmatrix} \end{array} \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix whereas $\mathbf{H}$ is a $(T-B) \times B$ matrix.

2) Create $(B+T, T)$ Low-Delay Burst Erasure Block Code (LD-BEBC)
The LD-BEBC code maps a vector of $T$ information symbols $\mathbf{b} \in \mathbb{F}_Q^T$ to a systematic codeword $\mathbf{c} \in \mathbb{F}_Q^{T+B}$ as follows. We first split $\mathbf{b}$ into two sub-vectors of lengths $B$ and $T-B$

$$\mathbf{b} = \begin{array}{c} \begin{array}{cc} B & T-B \end{array} \\ \begin{bmatrix} \mathbf{u} & \mathbf{n} \end{bmatrix} \end{array}, \tag{6}$$

and the resulting codeword is[1]

$$\mathbf{c} = \begin{bmatrix} \mathbf{u} & \mathbf{n} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{B\times B} & \mathbf{0}_{B\times(T-B)} & \mathbf{I}_{B\times B} \\ \mathbf{0}_{(T-B)\times B} & \mathbf{I}_{(T-B)\times(T-B)} & \mathbf{H} \end{bmatrix} \tag{7}$$

$$= \begin{bmatrix} \mathbf{u} & \mathbf{n} & \mathbf{u} + \mathbf{n}\cdot\mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{b} & \mathbf{r} \end{bmatrix} \tag{8}$$

where we have used (6) and introduced $\mathbf{r} = \mathbf{u} + \mathbf{n} \cdot \mathbf{H}$ to denote the parity check symbols in $\mathbf{c}$ in the last step. The codeword $\mathbf{c}$ has the property that it is able to correct any erasure burst of length $B$ with a delay of at-most $T$ symbols. If we express $\mathbf{b} = (b_0, \ldots, b_{T-1})$ then, for any erasure-burst of length $B$, $b_0$ is recovered at time $T$, $b_1$ at time $(T+1)$ and $b_{B-1}$ at time $(T+B-1)$. The remaining symbols $b_B \ldots, b_{T-1}$ are all recovered at the end of the block.
The information symbols in vector $\mathbf{u} = (b_0, \ldots, b_{B-1})$ are referred to as *urgent* symbols whereas the symbols in vector $\mathbf{n} = (b_B, \ldots, b_{T-1})$ are referred to as non-urgent symbols.

3) Diagonal Interleaving
The final step is to construct a streaming code (SCo) from the LD-BEBC code in step 2. Recall that the SCo specified a mapping between the symbols $s[t]$ of

[1]All addition in this paper is defined over $\mathbb{F}_Q$ or its extension field.

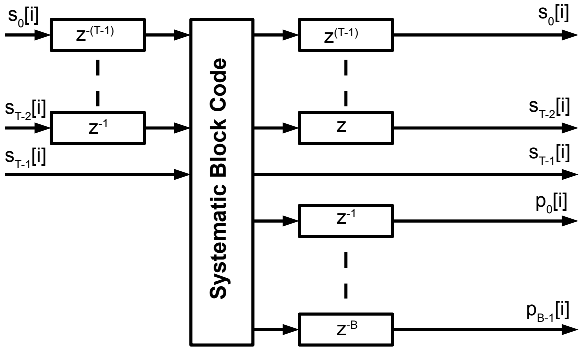


Fig. 2: Streaming Code (SCo) structure based on diagonal interleaving. The $z^{-\lambda}$ and $z^{\lambda}$ elements denote delay or advance by $\lambda$.

the incoming source stream to the symbols $x[t]$ of the channel input stream. This mapping is of the form

$$s[t] = \left[s_0[t], \ldots, s_{T-1}[t]\right]^\dagger \in \mathbb{F}_Q^T,$$
$$x[t] = \left[s_0[t], \ldots, s_{T-1}[t], p_0[t], \ldots, p_{B-1}[t]\right]^\dagger \in \mathbb{F}_Q^{T+B} \quad (9)$$

i.e., we split each source symbol $s[t] \in \mathbb{F}_Q^T$ into $T$ equal sized sub-symbols over $\mathbb{F}_Q$ and then append $B$ parity check sub-symbols over $\mathbb{F}_Q$. Thus we have that $x[t] \in \mathbb{F}_Q^{T+B}$. The parity check sub-symbols $p_0[t], \ldots, p_{B-1}[t]$ constructed through a diagonal interleaving technique described below.

An information vector $\mathbf{b}_t$ in (6) is constructed by collecting symbols along the diagonal of the sub-streams i.e.,

$$\mathbf{b}_t = (s_0[t], s_1[t+1], \ldots, s_{T-1}[t+T-1]). \quad (10)$$

The corresponding codeword $\mathbf{c}_t = (\mathbf{b}_t, \mathbf{r}_t)$ is then constructed according to (7). The resulting parity check sub-symbols in $\mathbf{r}[t]$ are then appended diagonally to the source stream to produce the channel input stream i.e.,

$$(p_0[t+T], \ldots, p_{B-1}[t+T+B-1]) = (r_0[t], \ldots, r_{B-1}[t]) \quad (11)$$

Notice that the operations in (9), (10) and (11) construct a codeword diagonally across the incoming source sub-streams as illustrated in Table. I. A *diagonal codeword* is of the form

$$\mathbf{d}_t = \quad (s_0[t], \ldots, s_{T-1}[t+T-1], p_0[t+T], \ldots, p_{B-1}[t+T+B-1]). \quad (12)$$

The SCo code is a time-invariant convolutional code [9]. The inputs to the convolutional code are source packets $\mathbf{s} \in \mathbb{F}_Q^T$, while the outputs are channel packets $\mathbf{x} \in \mathbb{F}_Q^{T+B}$. The relationship between the properties of the LD-BEBC and the convolutional code obtained by the diagonal interleaving structure is clarified in Fig. 2.

### *B. Decoding of SCo Codes*

The structure of the diagonal codeword (12) is also important in decoding. Suppose that symbols $x[t], \ldots, x[t+B-1]$ are erased. It can be readily verified that there are no more than $B$ erasures in each diagonal codeword $\mathbf{d}_t, \ldots, \mathbf{d}_{t+B-1}$. Since each codeword is a is a $(T+B, T)$ LD-BEBC, it recovers each erased symbol with a delay of no more than $T$ symbols. This in turn implies that all erased symbols are recovered.

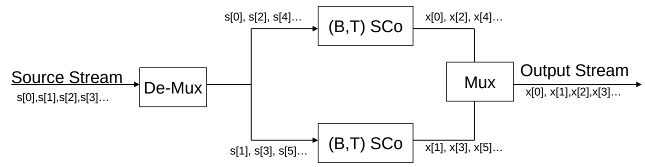


Fig. 3: A vertical interleaving approach to construct a $(2B, 2T)$ SCo code from a $(B, T)$ SCo code.

### *C. Example: (2,3) SCo Code*

Suppose we wish to construct a code capable of correcting any packet burst erasure of length $B = 2$ with delay $T = 3$. A LD-BEBC (7) for these parameters is

$$\mathbf{c} = (b_0, b_1, b_2, b_0 + b_2, b_1 + b_2). \quad (13)$$

To construct the SCo code, we divide the source symbols into $T = 3$ sub-symbols. The diagonal codeword (12) is of the form

$$\mathbf{d}_t = (s_0[t], s_1[t+1], s_2[t+2], s_0[t]+s_2[t+2], s_1[t+1]+s_2[t+2]) \quad (14)$$

and the channel input $x(t)$ is given by

$$x[t] = \left[s_0[t], s_1[t], s_2[t], s_0[t-3] + s_2[t-1], s_1[t-3] + s_2[t-2]\right]^\dagger. \quad (15)$$

The resulting channel input stream is illustrated in Table. I. Note that the rate of this code is $3/5$ as it introduces two parity check sub-symbols for each three source sub-symbols. It can be easily verified that this code corrects a burst erasure of length 2 with a worst-case time delay 3.

## IV. SCo Properties

In this section we describe some additional properties of SCo codes that will be useful in the DE-SCo construction.

### *A. Vertical Interleaving for $(\alpha B, \alpha T)$ SCo*

Suppose $\alpha \geq 2$ is an integer and we need to construct a SCo code with parameters $(\alpha B, \alpha T)$. The scheme described in section III-A requires us to split each source symbol into $\alpha T$ sub-symbols. However we can take advantage of the multiplicity factor $\alpha$ and simply construct the $(\alpha B, \alpha T)$ SCo code from the $(B, T)$ SCo code via vertical interleaving of step $\alpha$.

Fig. 3 illustrates this approach for constructing a $(2B, 2T)$ SCo from a $(B, T)$ SCo. We split the incoming source stream into two disjoint sub-streams; one consisting of source symbols at even time slots and the other consisting of symbols at odd time slots. We apply a $(B, T)$ SCo on the first sub stream to produce channel packets at even time slots. Likewise

| | | | | | |
|---|---|---|---|---|---|
| $\boxed{s_0[i-1]}$ | $s_0[i]$ | $s_0[i+1]$ | $s_0[i+2]$ | $s_0[i+3]$ | $s_0[i+4]$ |
| $s_1[i-1]$ | $\boxed{s_1[i]}$ | $s_1[i+1]$ | $s_1[i+2]$ | $s_1[i+3]$ | $s_1[i+4]$ |
| $s_2[i-1]$ | $s_2[i]$ | $\boxed{s_2[i+1]}$ | $s_2[i+2]$ | $s_2[i+3]$ | $s_2[i+4]$ |
| $s_0[i-4]+s_2[i-2]$ | $s_0[i-3]+s_2[i-1]$ | $s_0[i-2]+s_2[i]$ | $\boxed{s_0[i-1]+s_2[i+1]}$ | $s_0[i]+s_2[i+2]$ | $s_0[i+1]+s_2[i+3]$ |
| $s_1[i-4]+s_2[i-3]$ | $s_1[i-3]+s_2[i-2]$ | $s_1[i-2]+s_2[i-1]$ | $s_1[i-1]+s_2[i]$ | $\boxed{s_1[i]+s_2[i+1]}$ | $s_1[i+1]+s_2[i+2]$ |

TABLE I: (2,3) SCo Code Construction

we apply a $(B,T)$ SCo on the second sub stream to produce channel packets at odd time slots. Since a burst of length $2B$ introduces $B$ erasures on either sub streams each of the SCo, each of the $(B,T)$ code suffices to recover from these erasures. Further each erased symbol is recovered with a delay of $T$ symbols on its individual sub stream, which corresponds to an overall delay of $2T$ symbols.

More generally we split each source symbol into $T$ sub-symbols. The information vector $\mathbf{b}_t$ is modified from (10) as

$$\mathbf{b}_t = (s_0[t], s_1[t+\alpha], \ldots, s_{T-1}[t+(T-1)\alpha]). \qquad (16)$$

The resulting codeword $\mathbf{c}[t]$ of the LD-BEBC is then mapped to a *diagonal codeword* by introducing a step-size of $\alpha$ in (12) i.e.,

$$\begin{aligned}\mathbf{d}_t = (&s_0[t], s_1[t+\alpha], \ldots, s_{T-1}[t+(T-1)\alpha], \\ &p_0[t+T\alpha], \ldots, p_{B-1}[t+(T+B-1)\alpha]). \qquad (17)\end{aligned}$$

As in the case of $\alpha = 2$, the decoding proceeds by splitting the source stream into $\alpha$ sub-streams and applying the decoder for $(B,T)$ SCo on each of the sub-streams. This guarantees that each symbol is recovered with a delay of $\alpha T$ on the original stream.

### B. Memory in Channel Input Stream $\{x[t]\}$

While the definition of SCo allows the channel input symbol $x[t]$ to depend on an arbitrary number of source symbols, the construction limits the memory of symbol $x[t]$ to previous $T$ symbols i.e.,

$$x[t] = f(s[t], s[t-1], \ldots, s[t-T]). \qquad (18)$$

Furthermore a closer look at the parity check symbols (9) of $x[t]$ reveals that the parity checks $p_0[t], \ldots, p_{B-1}[t]$ have the form

$$\begin{aligned}p_j[t] = s_j[t-T] + h_j(s_B[t-j-T+B], \ldots, s_{T-1}[t-j-1]), \\ j = 0, \ldots, B-1, \quad (19)\end{aligned}$$

where $h_j(\cdot)$ denotes a linear combination arising from the LD-BEBC code (8) when applied along the main diagonal.

### C. Urgent and Non-Urgent Sub-Symbols

In the construction of LD-BEBC codes we split the information vector $\mathbf{b}$ into urgent and non-urgent symbols (6). The mapping of source sub-symbols to information vector (10) then implies that the sub-symbols $s_0, \ldots, s_{B-1}$ are the urgent symbols in the source stream whereas the sub-symbols $s_B, \ldots, s_{T-1}$ are non-urgent sub-symbols. We will denote these by

$$\begin{aligned}\mathbf{s}^U[t] &= (s_0[t], \ldots, s_{B-1}[t]), \\ \mathbf{s}^N[t] &= (s_B[t], \ldots, s_{T-1}[t]). \qquad (20)\end{aligned}$$

The urgent and non-urgent symbols are combined into a parity check sub symbol as illustrated in (19). The following observation is useful in the construction of DE-SCo.

**Proposition 1.** *Suppose that the sequence of channel symbols $x[i-B], \ldots, x[i-1]$ are erased by the burst-erasure channel. Then*

1) *All sub-symbols in $\mathbf{s}^N[i-B], \ldots, \mathbf{s}^N[i-1]$ are obtained from the parity checks $\mathbf{p}[i], \ldots, \mathbf{p}[i+T-B-1]$.*
2) *The sub-symbols in $\mathbf{s}^U[j]$ for $i-B \le j < i$ are recovered at time $j+T$ from parity check $\mathbf{p}[j+T]$ and the previously recovered non-urgent sub-symbols.*

The proof follows via (19) and will be omitted due to space constraints.

### D. Off-Diagonal Interleaving

The constructions in section III-A involve interleaving along the main diagonal of the source stream (c.f. (12),(10)). An analogous construction of the $(B,T)$ code along the off diagonal results in

$$\begin{aligned}\bar{\mathbf{b}}_t &= (s_0[t], s_1[t-1], \ldots, s_{T-1}[t-(T-1)]]) \qquad (21) \\ \bar{\mathbf{d}}_t &= (s_{T-1}[t-(T-1), \ldots, s_1[t-1], s_0[t], \bar{p}_0[t+1], \\ &\quad \ldots, \bar{p}_{B-1}[t+B]]) \qquad (22)\end{aligned}$$

and the parity checks $\bar{p}_j$ are given by

$$\begin{aligned}&\bar{p}_j[t] = s_{T-j-1}[t-T] \\ &+ h_j(s_{T-B-1}[t-j-T+B], \ldots, s_0[t-j-1]), j = 0, \ldots, B-1, \\ &\qquad (23)\end{aligned}$$

when applied along the opposite diagonal. Finally off-diagonal interleaving also satisfies Prop. 1 provided with appropriate modifications in the definitions of urgent and non-urgent symbols

$$\begin{aligned}\bar{\mathbf{s}}^U[t] &= (s_{T-1}[t], \ldots, s_{T-B}[t]), \\ \bar{\mathbf{s}}^N[t] &= (s_{T-B-1}[t], \ldots, s_0[t]). \qquad (24)\end{aligned}$$

(a) SCo Construction for $(B,T)=(1,2)$

| | | | | | |
|---|---|---|---|---|---|
| $\boxed{s_0[i-1]}$ | $s_0[i]$ | $s_0[i+1]$ | $s_0[i+2]$ | $s_0[i+3]$ | $s_0[i+4]$ |
| $s_1[i-1]$ | $\boxed{s_1[i]}$ | $s_1[i+1]$ | $s_1[i+2]$ | $s_1[i+3]$ | $s_1[i+4]$ |
| $s_0[i-3]+s_1[i-2]$ | $s_0[i-2]+s_1[i-1]$ | $\boxed{s_0[i-1]+s_1[i]}$ | $s_0[i]+s_1[i+1]$ | $s_0[i+1]+s_1[i+2]$ | $s_0[i+2]+s_1[i+3]$ |

(b) SCo Construction for $(B,T)=(2,4)$

| | | | | | |
|---|---|---|---|---|---|
| $\boxed{s_0[i-1]}$ | $s_0[i]$ | $s_0[i+1]$ | $s_0[i+2]$ | $s_0[i+3]$ | $s_0[i+4]$ |
| $s_1[i-1]$ | $s_1[i]$ | $\boxed{s_1[i+1]}$ | $s_1[i+2]$ | $s_1[i+3]$ | $s_1[i+4]$ |
| $s_0[i-5]+s_1[i-3]$ | $s_0[i-4]+s_1[i-2]$ | $s_0[i-3]+s_1[i-1]$ | $s_0[i-2]+s_1[i]$ | $\boxed{s_0[i-1]+s_1[i+1]}$ | $s_0[i]+s_1[i+2]$ |

TABLE II: Single user SCo constructions are shown in the upper two figures. Note that the $(1,2)$ SCo code recovers a single erasure with a delay $T=2$ but cannot recover from $B=2$. The $(2,4)$ SCo code recovers from $B=2$ with a delay of $T=4$ but does not incur a smaller delay when $B=1$.

## V. Example

We first highlight our results via a numerical example: $(B_1,T_1)=(1,2)$ and $(B_2,T_2)=(2,4)$. Single user SCo constructions from [1], [2] for both users are illustrated in Table II(a) and II(b) respectively. In each case, the source symbol $\mathsf{s}[i]$ is split into two sub-symbols $(\mathsf{s}_0[i],\mathsf{s}_1[i])$ and the channel symbol $\mathsf{x}[i]$ is obtained by concatenating the source symbol $\mathsf{s}[i]$ with a parity check symbol $p[i]$. In the $(1,2)$ SCo construction, parity check symbol $p^{\mathrm{I}}[i]=s_1[i-1]+s_0[i-2]$ is generated by combining the source sub-symbols diagonally across the source stream as illustrated with the rectangular boxes. For the $(B,T)=(2,4)$, the choice $p^{\mathrm{II}}[i]=s_1[i-2]+s_0[i-4]$ is similar, except that an interleaving of step of size 2 is applied before the parity checks are produced. Note that both these codes are single user codes and do not adapt to channel conditions.

In Table III(a) we illustrate a construction that achieves a rate $2/3$ and $(B_1,T_1)=(1,2)$ and still enables user 2 to recover the entire stream with a delay of $T_2=6$. It is obtained by shifting the parity checks of the SCo code in Table II(b) to the right by two symbols and combining with the parity checks of the SCo code in Table II(a) i.e., $q[i]=p^{\mathrm{I}}[i]+p^{\mathrm{II}}[i-2]$. Note that parity check symbols $p^{\mathrm{II}}[\cdot]$ do not interfere with the parity checks of user 1 i.e., when $s[i]$ is erased, receiver 1 can recover $p^{\mathrm{I}}[i+1]$ and $p^{\mathrm{I}}[i+2]$ from $q[i+1]$ and $q[i+2]$ respectively by canceling $p^{\mathrm{II}}[\cdot]$ that combine with these symbols. It then recovers $s[i]$. Likewise if $s[i]$ and $s[i-1]$ are erased, then receiver 2 recovers $p^{\mathrm{II}}[i+1],\ldots,p^{\mathrm{II}}[i+4]$ from $q[i+3],\ldots,q[i+6]$ respectively by canceling out the interfering $p^{\mathrm{I}}[\cdot]$, thus yielding $T_2=6$.

The interference avoidance strategy illustrated above is sub-optimal. Table. III(b) shows the DE-SCo construction that achieves the minimum possible delay of $T_2=5$. In this construction we first construct the parity checks $\check{p}^{\mathrm{II}}[i]=s_1[i-2]+s_0[i-1]$ by combining the source symbols along the opposite diagonal of the $(1,2)$ SCo code in Table II(a). Note that $x(i)=(s[i],\check{p}^{\mathrm{II}}[i])$ is also a single user $(1,2)$ SCo code. We then shift the parity check stream to the right by $T+B=3$ symbols and combine with $p^{\mathrm{I}}[i]$ i.e., $q[i]=p^{\mathrm{I}}[i]+\check{p}^{\mathrm{II}}[i-3]$. In the resulting code, receiver 1 is still able to cancel the effect of $\check{p}^{\mathrm{II}}[\cdot]$ as before and achieve $T_1=2$. Furthermore at receiver 2 if $s[i]$ and $s[i-1]$ are erased, then observe that receiver 2 obtains $s_0[i]$ and $s_0[i-1]$ from $q[i+2]$ and $q[i+3]$ respectively and $s_1[i-1]$ and $s_1[i]$ from $q[i+4]$ and $q[i+5]$ respectively, thus yielding $T_2=5$ symbols.

In the remainder of the paper we generalize this example to arbitrary values of $(B_i,T_i)$.

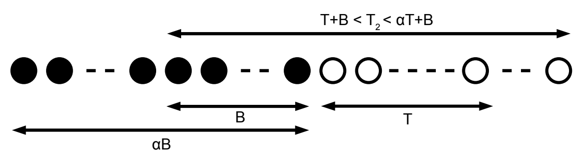


Fig. 4: One period illustration of the Periodic Erasure Channel for $T+B<T_2\leq \alpha T+B$. Black and white circles resemble erased and unerased symbols respectively.

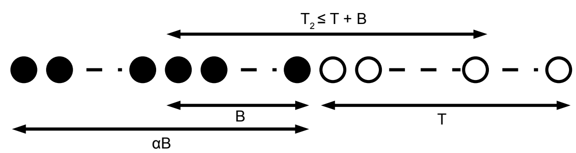


Fig. 5: One period illustration of the Periodic Erasure Channel for $T_2<T+B$. Black and white circles resemble erased and unerased symbols respectively.

## VI. Construction of DE-SCo

In this section we describe the DE-SCo construction. We rely on several properties of the single user SCo explained in section III.

**Theorem 1.** *Let $(B_1,T_1)=(B,T)$ and suppose $B_2=\alpha B$ where $\alpha$ is any integer that exceeds 1. The minimum possible delay for any code of rate $R=\frac{T}{T+B}$ is*

$$T_2^\star=\alpha T+B, \tag{25}$$

*and is achieved by the optimal DE-SCo construction.*

### A. Converse

We first establish converse to theorem 1. Consider any code that achieves $\{(B,T),(B_2,T_2)\}$ with $T_2<T_2^\star$. The rate of this code is strictly below $R=\frac{T}{T+B}$.

(a) IA-SCo Code Construction for $(B_1, T_1) = (1, 2)$ and $(B_2, T_2) = (2, 6)$

| | | | | | |
|---|---|---|---|---|---|
| $s_0[i-1]$ <br> $s_1[i-1]$ | $s_0[i]$ <br> $s_1[i]$ | $s_0[i+1]$ <br> $s_1[i+1]$ | $s_0[i+2]$ <br> $s_1[i+2]$ | $s_0[i+3]$ <br> $s_1[i+3]$ | $s_0[i+4]$ <br> $s_1[i+4]$ |
| $s_0[i-3]+s_1[i-2]$ <br> $+$ <br> $s_0[i-7]+s_1[i-5]$ | $s_0[i-2]+s_1[i-1]$ <br> $+$ <br> $s_0[i-6]+s_1[i-4]$ | $s_0[i-1]+s_1[i]$ <br> $+$ <br> $s_0[i-5]+s_1[i-3]$ | $s_0[i]+s_1[i+1]$ <br> $+$ <br> $s_0[i-4]+s_1[i-2]$ | $s_0[i+1]+s_1[i+2]$ <br> $+$ <br> $s_0[i-3]+s_1[i-1]$ | $s_0[i+2]+s_1[i+3]$ <br> $+$ <br> $s_0[i-2]+s_1[i]$ |

(b) DE-SCo Code Construction for $(B_1, T_1) = (1, 2)$ and $(B_2, T_2) = (2, 5)$

| | | | | | |
|---|---|---|---|---|---|
| $\boxed{s_0[i-1]}$ <br> $\boxed{s_1[i-1]}$ | $\boxed{s_0[i]}$ <br> $\boxed{s_1[i]}$ | $s_0[i+1]$ <br> $s_1[i+1]$ | $s_0[i+2]$ <br> $s_1[i+2]$ | $s_0[i+3]$ <br> $s_1[i+3]$ | $s_0[i+4]$ <br> $s_1[i+4]$ |
| $s_0[i-3]+s_1[i-2]$ <br> $+$ <br> $s_1[i-6]+s_0[i-5]$ | $s_0[i-2]+s_1[i-1]$ <br> $+$ <br> $s_1[i-5]+s_0[i-4]$ | $\boxed{s_0[i-1]+s_1[i]}$ <br> $+$ <br> $s_1[i-4]+s_0[i-3]$ | $s_0[i]+s_1[i+1]$ <br> $+$ <br> $s_1[i-3]+s_0[i-2]$ | $s_0[i+1]+s_1[i+2]$ <br> $+$ <br> $s_1[i-2]+s_0[i-1]$ | $s_0[i+2]+s_1[i+3]$ <br> $+$ <br> $\boxed{s_1[i-1]+s_0[i]}$ |

TABLE III: Rate $2/3$ code constructions that satisfy user 1 with $(B_1, T_1) = (1, 2)$ and user 2 with $B_2 = 2$.

To establish this we separately consider the case when $T + B \le T_2 < T + \alpha B$ and the case when $T_2 \le T + B$. Let us assume the first case.

As shown in Fig. 4, construct a periodic burst-erasure channel in which every period of $(\alpha - 1)B + T_2$ symbols consists of a sequence of $\alpha B$ erasures followed by a sequence of non-erased symbols. Consider one period of the proposed periodic erasure channel with a burst erasure of length $\alpha B$ from time $t = 0, 1, \ldots, \alpha B - 1$ followed by a period of non-erasures for $t = \alpha B, \ldots, \mathcal{T}_P = (\alpha - 1)B + T_2 - 1$. For time $t = 0, \ldots, T_P$ the channel behaves identically to a burst-erasure channel with $\alpha B$ erasures. The first $(\alpha-1)B$ erasures at time $t = 0, 1, \ldots, (\alpha - 1)B - 1$ can be recovered using decoder of user 2 with a delay of $T_2$ i.e., by time $T_P$ and hence the channel packets $x[0], \ldots, x[(\alpha - 1)B - 1]$ can also be recovered via (1).

It remains to show that the symbols at time $t = (\alpha - 1)B, \ldots, \alpha B - 1$ are also recovered by time $T_P$. Note that since the channel symbols $x[0], \ldots, x[(\alpha - 1)B - 1]$ have been recovered, the resulting channel between times $t = 0, \ldots, \alpha B - 1$ is identical to a burst erasure channel with $B$ erasures between time $t = (\alpha - 1)B, \ldots, \alpha B - 1$. The decoder of user 1 applied to this channel recovers the source symbols by time $\alpha B - 1 + T \le T_P$, which follows since $T_2 \ge T + B$. Thus all the erased channel symbols in the first period are recovered by time $T_P$. Since the channel introduces periodic bursts, the same argument can be repeated across all periods. Since the length of each period is $(\alpha-1)B+T_2$ and contains of $\alpha B$ erasures,thus the capacity is upper bounded by $R = 1 - \frac{\alpha B}{(\alpha-1)B+T_2}$ which is less than $R$ if $T_2 < T_2^\star$ and the converse follows.

For the other case with $T_2 \le T + B$ shown in Fig. 5, the same argument applies except that the decoder of user 2 recovers the $(\alpha-1)B$ erasures at time $t = 0, 1, \ldots, (\alpha-1)B-1$ by time $T_P \le \alpha B + T - 1$. Furthermore, the decoder of user 1 recovers the $B$ erasures at time $t = (\alpha-1)B, \ldots, \alpha B - 1$ by time $\alpha B + T - 1$. Now, the length of each period is $\alpha B + T - 1$ with $T$ available symbols, the rate is $\frac{T}{T+\alpha B}$ strictly smaller than $R$ as $\alpha > 1$.

### B. Code Construction

For achievability, $T_2^\star$ in (25) we construct the following code:

- Let $\mathcal{C}_1$ be the single user $(B, T)$ SCo obtained by splitting each source symbol $s[i]$ into $T$ sub-symbols $(s_0[i], \ldots, s_{T-1}[i])$ and producing $B$ parity check sub-symbols $\mathbf{p}^{\mathrm{I}} = (p_0^{\mathrm{I}}[i], \ldots, p_{B-1}^{\mathrm{I}}[i])$ at each time by combining the source sub-symbols along the main diagonal c.f. (19).
- Let $\mathcal{C}_2$ be a $((\alpha - 1)B_1, (\alpha - 1)T_1)$ SCo also obtained by splitting the source symbols into $T$ sub-symbols $(s_0[i], \ldots, s_{T-1}[i])$ and then constructing a total of $B$ parity checks $\mathbf{p}^{\mathrm{II}[i]} = (p_0^{\mathrm{II}}[i], \ldots, p_{B-1}^{\mathrm{II}}[i])$ by combining the source sub-symbols along the opposite diagonal and with an interleaving step of size $\ell = (\alpha - 1)$ i.e.,

$$p_k^{\mathrm{II}}[i] = \mathcal{B}_k(s_0[i - \ell - k\ell], \ldots, s_{T-1}[i - \ell T - k\ell]). \quad (26)$$

- Introduce a shift $\Delta = T + B$ in the stream $p^{\mathrm{II}}[\cdot]$ and combine with the parity check stream $p^{\mathrm{I}}[\cdot]$ i.e., $\mathbf{q}[i] = \mathbf{p}^{\mathrm{I}}[i] + \mathbf{p}^{\mathrm{II}}[i - \Delta]$. The output symbol at time $i$ is $x[i] = (s[i], \mathbf{q}[i])$

Since there are $B$ parity check sub-symbols for every $T$ source sub-symbols it follows that the rate of the code is $\frac{T}{T+B}$.

### C. Example

Fig. 6 illustrates the DE-SCo $\{(4, 7), (8, 18)\}$ construction. Each column represents one time-index between $[-8, 17]$ shown in the top row of the table. We assume that a burst-erasure occurs between time $[-8, -1]$ and only show the relevant symbols and parity-checks. Each source symbol is split into seven sub-symbols, each occupying one row. Each source sub-symbol has two labels - one number and one letter. The letter represents the main diagonal that passes through the sub-symbol e.g., sub-symbols in $\mathbf{b}_{-1}^{\mathrm{I}}$ are marked $a$. The number represents the off-diagonal that passes through the sub-symbols e.g., sub-symbols in $\mathbf{b}_{-1}^{\mathrm{II}}$ are marked 8. The next four rows denote the parity check sub-symbols. The parity checks for $\mathcal{C}_1$, generated by diagonal $\mathbf{b}_i^{\mathrm{I}}$, (c.f. (19)) are marked by the same letter. The four top rows, in lighter font, show the parity checks generated by the diagonal $\mathbf{b}_i^{\mathrm{II}}$ for $\mathcal{C}_2$ (c.f. (26)), shifted by $T+B=11$ slots. These parity checks are combined with the corresponding parity checks of $\mathcal{C}_1$ as shown in Fig. 6. For example the cell marked $(c1)$ at time $t=4$ indicates that this sub-symbol of $\mathbf{q}[4]$ results by adding the parity check of $\mathcal{C}_1$, marked $c$, with the parity check of $\mathcal{C}_2$, marked 1, at $t=4$.

We illustrate the decoding steps as follows.

**Step (1):** Recovery of $\{\mathbf{p}^{\mathrm{II}}[t-\Delta]\}_{t\geq T}$:
By construction of $\mathcal{C}_1$ all the parity checks $\mathbf{p}^{\mathrm{I}}[t]$ for $t\geq 7$ do not involve the erased symbols. In particular the parity checks marked by $a$, $b$ and $c$ at $t\geq 7$ do not involve source sub-symbols before $t=0$ and hence these can be canceled to recover parity checks marked by 1, 2, 3. The remaining parity checks $\mathbf{p}^{\mathrm{II}}[t-11]$ for $t\geq 7$ can also be obtained in a similar manner.

**Step (2):** Upper-left triangle:
The parity checks in step (1) enable us to recover the non-urgent erased sub-symbols in $\mathbf{b}^{\mathrm{II}}_{-8},\ldots,\mathbf{b}^{\mathrm{II}}_{-5}$ by $t=9$. See Fig. 6-Step (2). The proof (which will be established for the general case) exploits the fact that the non-urgent symbols in these diagonals all begin before time $t=-4$ and that in each diagonal codeword of a streaming code, the non-urgent symbols are recovered before the urgent symbols.

**Step (3):** Recovery of $\mathbf{p}^{\mathrm{I}}[t]$ for $0\leq t\leq T-1$:
The sub-symbols recovered in step (2) suffice to recover all parity checks $\mathbf{p}^{\mathrm{I}}[t]$ for $0\leq t\leq 6$. Note that the relevant interfering parity checks from $\mathbf{p}^{\mathrm{II}}[\cdot]$ in this interval are marked by 1, 2 and 3 as illustrated in the shaded area in Fig. 6-Step (3). Since all the corresponding source sub-symbols have been recovered, these parity checks can be canceled.

**Step (4):** Upper-right triangle:
Since the diagonals $\mathbf{b}^{\mathrm{I}}_{-1},\ldots,\mathbf{b}^{\mathrm{I}}_{-4}$ involve four or fewer erasures we can now recover these sub-symbols using parity checks recovered in the previous step.

**Step (5):** Recovery of non-urgent sub-symbols recursively:
The remaining non-urgent sub-symbols need to be recovered in a recursive manner. Note that $\mathbf{b}^{\mathrm{I}}_{-5}$, marked by $e$, has five erased symbols. However the first symbol marked by $(4e)$ also belongs to $\mathbf{b}^{\mathrm{II}}_{-5}$ and has already been recovered. The remaining four sub-symbols can be recovered by the four available parity checks of $p^{\mathrm{I}}[\cdot]$ marked by $e$. Similarly $\mathbf{b}^{\mathrm{II}}_{-4}$, marked by 5, also has five erasures, but the first symbol $(5d)$ also belongs to $\mathbf{b}^{\mathrm{I}}_{-4}$ and has been recovered. Hence the remaining parity checks can be recovered using the parity checks of $\mathbf{p}^{\mathrm{II}}[\cdot]$. Of these, by construction of $\mathcal{C}_2$, the non-urgent symbols will be recovered by time $t=9$. The decoder then recovers $\mathbf{b}^{\mathrm{I}}_{-6}$ and $\mathbf{b}^{\mathrm{II}}_{-3}$ in the next step to recover all non-urgent sub-symbols.

**Step (6):** Recovery of urgent sub-symbols:
After recovering all non-urgent sub-symbols in the previous steps, we can directly recover the urgent ones (i.e., the bottom for rows) using parity checks $\mathbf{p}^{II}[t]$ for $9<t\leq 17$.

### *D. Decoding at User 1*

Suppose that the symbols at time $i-B,\ldots,i-1$ are erased by the channel of user 1. User 1 first recovers parity checks $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$ from $\mathbf{q}[i],\ldots,\mathbf{q}[i+T-1]$ by canceling the parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ that combine with $\mathbf{p}^{\mathrm{I}}[\cdot]$ in this period. Indeed at time $i+T-1$ the interfering parity check is $\mathbf{p}^{\mathrm{II}}[i+T-\Delta-1]=\mathbf{p}^{\mathrm{II}}[i-B-1]$, which clearly depends on the (non-erased) source symbols before time $i-B$. All parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ before this time are also non-interfering. The erased source symbols can be recovered from $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$ by virtue of code $\mathcal{C}_1$.

### *E. Decoding at User 2*

Suppose that the symbols at times $i-1,\ldots,i-\alpha B$ are erased for receiver 2. The decoding involves two main steps: recovery of non-urgent symbols followed by recovery of urgent symbols.

Step 1 **Recover Non-Urgent Symbols**: Let $\mathcal{T}\triangleq i-\alpha B+T_2^\star$. Use parity checks at time $i\leq t\leq \mathcal{T}-1$ to recover $\{\mathbf{s}^N[\tau]\}_{\tau=i-\alpha B}^{i-1}$ where $\mathbf{s}^N[\tau]=(s_0[\tau],\ldots,s_{T-B-1}[\tau])$ denote the set of *non-urgent* sub-symbols for $\mathcal{C}_2$.

Step 2 **Recover Urgent Symbols**: Recover symbols $\mathbf{s}^U[\tau]=(s_{T-B}[\tau],\ldots,s_{T-1}[\tau])$ for $i-\alpha B\leq\tau<i$ at time $t=\tau+T_2^\star$ using the parity check symbols $\mathbf{p}^{\mathrm{II}}[t]$ and the previously decoded non-urgent symbols. The sub-symbols $\mathbf{s}^U[\cdot]$ are the urgent sub-symbols of $\mathcal{C}_2$

We now establish step 1 below. Our decoding steps will exploit diagonal codewords $\mathbf{b}^{\mathrm{I}}_i=(s_0[i],\ldots,s_{T-1}[i+T-1])$, $\mathbf{b}^{\mathrm{II}}_i=(s_0[i],\ldots,s_{T-1}[i-(T-1)\ell])$ that are embedded in the source stream as described earlier.

**Lemma 1.** *The decoder for user 2 recovers the non-urgent symbols $\mathbf{s}^N[\cdot]$ in the following order*

1) *For $t\geq i+T$, recover parity check $\mathbf{p}^{\mathrm{II}}[t-\Delta]$ from $\mathbf{q}[t]$ by canceling the parity checks $\mathbf{p}^{\mathrm{I}}[t]$ which depend only on (non-erased) source symbols at time $i$ or later.*
2) *Recover the non-urgent symbols in $\mathbf{b}^{\mathrm{II}}_{i-\alpha B},\ldots,\mathbf{b}^{\mathrm{II}}_{i-B-1}$ using the parity check symbols $\{\mathbf{p}^{\mathrm{II}}[t-\Delta]\}_{t=i+T}^{\mathcal{T}-1}$;*
3) *Recover the parity checks $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$ from $\mathbf{q}[i],\ldots,\mathbf{q}[i+T-1]$.*
4) *Recover the non-urgent symbols in $\mathbf{b}^{\mathrm{I}}_{i-1},\ldots,\mathbf{b}^{\mathrm{I}}_{i-B}$ using the parity checks $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$.*
5) *For each $k\in\{1,\ldots,T-B-1\}$ recursively recover the remaining non-urgent symbols as follows:*

(Ind. 1) *Recover the non-urgent sub-symbols in $\mathbf{b}^{\mathrm{I}}_{i-B-k}$ using the non-urgent sub-symbols in $\{\mathbf{b}^{\mathrm{II}}_j\}_{j\leq i+(k-1)(\alpha-1)-B-1}$ and parity checks $\mathbf{p}^{\mathrm{I}}[\cdot]$ between $i\leq t<i+T$.*

(Ind. 2) *Recover the non-urgent sub-symbols in $\mathbf{b}^{\mathrm{II}}_{i-B+(k-1)(\alpha-1)},\ldots,\mathbf{b}^{\mathrm{II}}_{i-B+k(\alpha-1)-1}$ using $\{\mathbf{b}^{\mathrm{I}}_j\}_{j\geq i-B-(k-1)}$ and the parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ between $i+T\leq t<\mathcal{T}$.*

*Once this recursion terminates, all the non-urgent sub-symbols $\{\mathbf{s}^N[\tau]\}_{\tau=i-\alpha B}^{i-1}$ are recovered by time $\mathcal{T}-1$.*

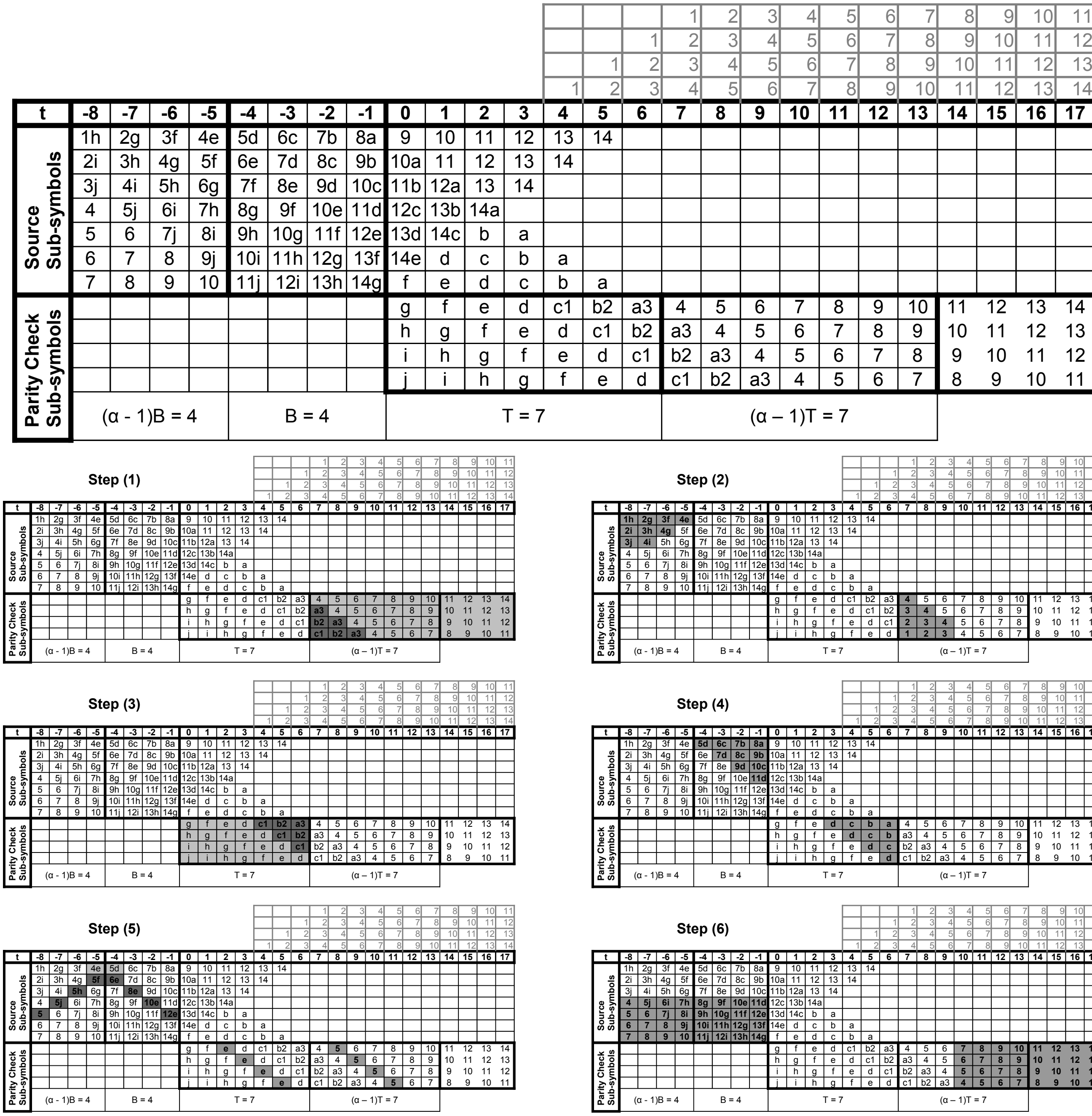


Fig. 6: Decoding for DE-SCo $\{(4,7),(8,18)\}$.

To show claim (1), note that via (19) the memory in channel input stream is limited to previous $T$ symbols. Consequently the parity check symbols $\{\mathbf{p}^{\mathrm{I}}[t]\}_{t\geq i+T}$ depend only on source symbols after time $i$. Hence these parity checks can be canceled and the claim follows.

To show claim (2) consider the set of diagonal vectors $\mathbf{b}^{\mathrm{II}}_{i-\alpha B},\ldots,\mathbf{b}^{\mathrm{II}}_{i-B-1}$ spanning the upper left triangle. Clearly these vectors are affected only by erasures between times $i-\alpha B,\ldots,i-B-1$. Furthermore, the corresponding parity checks $\{\mathbf{p}^{\mathrm{II}}[t-\Delta]\}_{t\geq i+T} \equiv \{\mathbf{p}^{\mathrm{II}}[t]\}_{t\geq i-B}$ have been recovered. Therefore, code $\mathcal{C}_2$ is capable of recovering the erased source sub-symbols in the stated diagonal vectors. By Proposition. 1 the non-urgent symbols are recovered from the first $(\alpha-1)(T-B)$ parity check columns which ends at $i+T+(\alpha-1)(T-B)-1=\mathcal{T}-1$ and the claim follows.

To establish claim (3), consider the column at time $t=i+T-1$. The interfering parity check column $\mathbf{p}^{\mathrm{II}}[i+T-1-\Delta]=\mathbf{p}^{\mathrm{II}}[i-B-1]$ only consists of source symbols at time $i-B-1$ or before. More specifically we have from (26) that for $k=0,1,\ldots,B-1$,

$$\begin{aligned}p^{\mathrm{II}}_k[i-B-1] &= \mathcal{B}_k(s_0[i-B-(\alpha-1)(k+1)-1],\\ &\qquad \ldots,s_{T-1}[i-B-(\alpha-1)(k+T)-1])\\ &= \mathcal{B}_k(\mathbf{b}^{\mathrm{II}}_{i-B-(\alpha-1)(k+1)-1}),\end{aligned}$$

which clearly is a function of the information vector $\mathbf{b}_t^{\mathrm{II}}$ before time $i-B-1$. Furthermore applying (23) and taking into consideration the interleaving step of $(\alpha-1)$,

$$\begin{aligned}p_k^{\mathrm{II}}[i-B-1] &= s_{T-k-1}[i-B-1-(\alpha-1)T] \\ &+h_k(s_{T-B-1}[i-B-1-j-(\alpha-1)(T-B)], \\ &\ldots, s_0[i-B-1-j-(\alpha-1)]).\end{aligned}$$

Thus the only urgent symbols involved are at $i-B-1-(\alpha-1)T$. Since these are unerased and since the non-urgent symbols in $\mathbf{b}^{\mathrm{II}}_{i-B-(\alpha-1)(k+1)-1}$have been already recovered by claim (2), it follows that we can reconstruct $\mathbf{p}^{\mathrm{II}}[i-B-1]$. A similar argument can be used to show that we can recover the all columns $\mathbf{p}^{\mathrm{II}}[i-B-T],\ldots,\mathbf{p}^{\mathrm{II}}[i-B-1]$, cancel their effect on $\mathbf{q}[i],\ldots,\mathbf{q}[i+T-1]$ and recover $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$.

Claim (4) follows in a similar way to claim (2). The diagonal vectors $\mathbf{b}^{\mathrm{I}}_{i-B},\ldots,\mathbf{b}^{\mathrm{I}}_{i-1}$ spanning the upper-right triangle of the erased source sub-symbols are affected by a burst erasure of length $B$ between times $i-B,\ldots,i-1$. Furthermore, the corresponding parity checks $\{\mathbf{p}^{\mathrm{I}}[t]\}_{i\le t<i+T}$ recovered earlier are capable of recovering the erased source sub-symbols in these diagonal vectors by at most time $i+T-1<\mathcal{T}$ and the claim follows.

Finally we consider the recursion in the last part of Lemma 1. Consider the case when $k=1$. According to Ind. 1 the non-urgent symbols $\{\mathbf{b}_j^{\mathrm{II}}\}_{j\le i-B-1}$ are available (from step 1). To recover $\mathbf{b}^{\mathrm{I}}_{i-B-1}$, note that the only erased symbol in this vector before time $i-B$ is $s_0[i-B-1]$ which has already been recovered in $\mathbf{b}^{\mathrm{II}}_{i-B-1}$. Hence the parity checks of $\mathcal{C}_1$ at the times $i,\ldots,i+T-1$ suffice to recover the remaining symbols. According to Ind. 2 the non-urgent sub-symbols in $\{\mathbf{b}_j^{\mathrm{I}}\}_{j\ge i-B}$ have been recovered in claim (4). Furthermore in vectors $\mathbf{b}^{\mathrm{II}}_{i-B},\ldots,\mathbf{b}^{\mathrm{II}}_{i-B+\alpha-2}$ the only erased symbols after time $i-B-1$ are $s_0[i-B],\ldots,s_0[i-B+\alpha-2]$, which are available from $\{\mathbf{b}_j^{\mathrm{I}}\}_{j\ge i-B}$. Thus the parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ can be used to recover the remaining non-urgent sub-symbols in these vectors.

Next suppose the statement holds for some $t=k$. We establish that the statement holds for $t=k+1$. In Ind. 1 the vector of interest is,

$$\begin{aligned}\mathbf{b}^{\mathrm{I}}_{i-B-(k+1)} &= (s_0[i-B-(k+1)], \\ &..., s_k[i-B-1], ..., s_{T-1}[i-B-k+(T-2)]).\end{aligned}$$

The erased elements in the interval $i-\alpha B,\ldots,i-B-1$ are $s_j[i-B-k+j-1]$ (for $j=0,\ldots,k$) which are also elements of $\mathbf{b}^{\mathrm{II}}_{i-B-k+\alpha j-1}$ (i.e., $\mathbf{b}^{\mathrm{II}}_{i-B-k-1},\ldots,\mathbf{b}^{\mathrm{II}}_{i-B+(\alpha-1)k-1}$), already recovered in Ind. 2 in the $k$th recursion. Hence the remaining symbols are recovered using the parity checks of $\mathcal{C}_1$. The first vector of interest in Ind. 2 is

$$\begin{aligned}\mathbf{b}^{\mathrm{II}}_{i-B+k(\alpha-1)} &= (s_0[i-B+k(\alpha-1)], \\ &..., s_k[i-B], s_{k+1}[i-B-(\alpha-1)], ...),\end{aligned}$$

and its erased elements that belong to the interval $i-B,\ldots,i-1$ are $s_j[i-B+(k-j)(\alpha-1)]$ (for $j=0,\ldots,k$) are also elements of the vectors $\mathbf{b}^{\mathrm{I}}_{i-B+(k-j)(\alpha-1)-j}$ (i.e.,$\mathbf{b}^{\mathrm{I}}_{i-B-k},\ldots,\mathbf{b}^{\mathrm{I}}_{i-B+(\alpha-1)k}$). These are recovered in Ind. 1 by the $k$th step. Likewise, the latest vector of interest,

$$\begin{aligned}\mathbf{b}^{\mathrm{II}}_{i-B+(k+1)(\alpha-1)-1} &= (s_0[i-B+(k+1)(\alpha-1)-1], \\ &..., s_k[i-B+(\alpha-1)-1], s_{k+1}[i-B-1], ...),\end{aligned}$$

has the erased elements in the interval $i-B,\ldots,i-1$ are $s_j[i-B+(k-j+1)(\alpha-1)-1]$ (for $j=0,\ldots,k$) belonging to vectors $\mathbf{b}^{\mathrm{I}}_{i-B+(k-j+1)(\alpha-1)-j-1}$ (i.e.,$\mathbf{b}^{\mathrm{I}}_{i-B+(\alpha-1)-k-1},\ldots,\mathbf{b}^{\mathrm{I}}_{i-B+(k+1)(\alpha-1)-1}$). These are recovered in Ind. 1 in step number $k+1-(\alpha-1)<k+1$. Hence the remaining symbols in these diagonals can be recovered using the parity checks of $\mathcal{C}_2$ and furthermore the non-urgent symbols of interest are recovered by time $\mathcal{T}-1$. This completes the claim of the recursion.

It finally remains to show that all the non-urgent symbols are recovered at $k=T-B-1$, it suffices to show that the lower left most non-urgent sub-symbol in the region $i-B,\ldots,i-1$ i.e., $s_{T-B-1}[i-B]$ is an element of $\mathbf{b}^{\mathrm{I}}_{i-B-k}=\mathbf{b}^{\mathrm{I}}_{i-T+1}$ which is clear when applying the definition of $\mathbf{b}^{\mathrm{I}}_i$ at $i-T+1$ as,

$$\mathbf{b}^{\mathrm{I}}_{i-T+1}=(s_0[i-T+1],\ldots,s_{T-B-1}[i-B],\ldots,s_{T-1}[i]).$$

Similarly, we need to show that $\mathbf{b}^{\mathrm{II}}_{i-B+k(\alpha-1)-1}=\mathbf{b}^{\mathrm{II}}_{i-B+(T-B-1)(\alpha-1)-1}$ contains the lower right most non-urgent sub-symbol in the region $i-\alpha B,\ldots,i-B-1$ i.e., $s_{T-B-1}[i-B-1]$. By applying the definition of $\mathbf{b}^{\mathrm{II}}_i$ at time $i-B+(T-B-1)(\alpha-1)-1$ as,

$$\begin{aligned}\mathbf{b}^{\mathrm{II}}{}_{i-B+(T-B-1)(\alpha-1)-1} &= (s_0[i-B+(T-B-1)(\alpha-1)-1], \\ &\ldots, s_{T-B-1}[i-B-1],\ldots,s_{T-1}[i-\alpha B-1]).\end{aligned}$$

Furthermore, we use Proposition. 1 to show that these non-urgent sub-symbols are recovered using the first $(\alpha-1)(T-B)$ columns of code $\mathcal{C}_2$ parity checks which falls in the time range $i+T,\ldots,i+T+(\alpha-1)(T-B)-1=\mathcal{T}-1$ and the claim in Step 2 follows.

The urgent symbols in step 2 is obtained as follows. After recovering all the non-urgent source sub-symbols $\{\mathbf{s}^N[\tau]\}_{\tau=i-\alpha B}^{i-1}$ i we can directly apply the construction of $\mathcal{C}_2$ to recover the urgent symbols $\{\mathbf{s}^U[\tau]\}_{\tau=i-\alpha B}^{i-1}$ using parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ within a delay of $T^\star$.

## VII. General Values of $\alpha$

In this section, we show that DE-SCo codes $\{(B,T),(\alpha B,\alpha T+B)\}$ can be constructed for any non-integer value of $\alpha$ such that $B_2=\alpha B$ is an integer. For any $\alpha=\frac{B_2}{B}>1$, let $\alpha=\frac{a}{b}$ where $a$ and $b$ are integers and $\frac{a}{b}$ is in the simplest form.

### A. DE-SCo Construction

We introduce suitable modifications to the construction given in the previous section. Clearly since $\frac{a}{b}$ is in simplest form $B$ must be an integer multiple of $b$ i.e., $B_0=\frac{B}{b}\in\mathbb{N}$. We first consider the case when $T$ is also an integer multiple of $b$ i.e., $T_0=\frac{T}{b}\in\mathbb{N}$. The case when $T$ is not an integer multiple, can be dealt with by a suitable source expansion, as outlined at the end of the section.

- Let $\mathcal{C}_1$ be the single user $(B,T)=(bB_0,bT_0)$ SCo obtained by splitting each source symbol $s[i]$ into $T_0$ sub-symbols $(s_0[i],\ldots,s_{T_0-1}[i])$ and producing $B_0$ parity check sub-symbols $\mathbf{p}^{\mathrm{I}} = (p_0^{\mathrm{I}}[i], \ldots, p_{B_0-1}^{\mathrm{I}}[i])$ at each time by combining the source sub-symbols along the main diagonal with an interleaving step of size $b$ i.e.,

$$p_k^{\mathrm{I}}[i] = \mathcal{A}_k(s_0[i - bT_0 - kb], \ldots, s_{T_0-1}[i - b - kb]) \quad (27)$$

- Let $\mathcal{C}_2$ be a $((\alpha-1)B_1, (\alpha-1)T_1) = ((a-b)B_0, (a-b)T_0)$ SCo also obtained by splitting the source symbols into $T_0$ sub-symbols $(s_0[i], \ldots, s_{T_0-1}[i])$ and then constructing a total of $B_0$ parity checks $\mathbf{p}^{\mathrm{II}} = (p_0^{\mathrm{II}}[i], \ldots, p_{B_0-1}^{\mathrm{II}}[i])$ by combining the source sub-symbols along the opposite diagonal and with an interleaving step of size $\ell_n = (a-b)$ i.e.,

$$p_k^{\mathrm{II}}[i] = \mathcal{B}_k(s_0[i - \ell_n - k\ell_n], \ldots, s_{T_0-1}[i - \ell_n T_0 - k\ell_n]). \quad (28)$$

- Introduce a shift $\Delta = T + B = b(T_0 + B_0)$ in the stream $p^{\mathrm{II}}[\cdot]$ and combine with the parity check stream $p^{\mathrm{I}}[\cdot]$ i.e., $\mathbf{q}[i] = \mathbf{p}^{\mathrm{I}}[i] + \mathbf{p}^{\mathrm{II}}[i - \Delta]$. The output symbol at time $i$ is $x[i] = (s[i], \mathbf{q}[i])$

### *B. Decoding*

The decoding follows steps analogous to the case when $\alpha$ is integer. We sketch the main steps. As before the decoding is done along the diagonal vectors $\mathbf{b}_i^{\mathrm{I}} = (s_0[i], \ldots, s_{T_0-1}[i + (T_0-1)b])$, $\mathbf{b}_i^{\mathrm{II}} = (s_0[i], \ldots, s_{T_0-1}[i - (T_0-1)\ell_n])$.

*Decoding at User* 1*:* For the first user, the same argument applies as in previous section i.e., a shift of $\Delta = b(T_0 + B_0)$ in $\mathbf{p}^{\mathrm{II}}[\cdot]$ guarantees that user 1 can cancel the interfering parity checks to recover the $\mathbf{p}^{\mathrm{I}}[\cdot]$ stream of interest.

*Decoding at User* 2*:* We verify that steps in section VI-E continue to apply. A little examination shows that the claims (1)—(4) as well as the proofs in the previous case follow immediately as they hold for an arbitrary interleaving step for $\mathcal{C}_2$ and do not rely on the interleaving step of $\mathcal{C}_1$ being 1. The induction step needs to be modified to reflect that the interleaving step size of $\mathcal{C}_1$ is $b > 1$.

For each $k \in \{1, \ldots, T - B - 1\}$ recursively recover the remaining non-urgent symbols as follows:

- **Ind. 1** Recover the non-urgent sub-symbols in $\mathbf{b}_{i-B-(k-1)b-1}^{\mathrm{I}}, \ldots, \mathbf{b}_{i-B-kb}^{\mathrm{I}}$ using the non-urgent sub-symbols in $\{\mathbf{b}_j^{\mathrm{II}}\}_{j \le i+(k-1)(a-b)-B-1}$ and parity checks $\mathbf{p}^{\mathrm{I}}[\cdot]$ between $i \le t < i + T$.
- **Ind. 2** Recover the non-urgent sub-symbols in $\mathbf{b}_{i-B+(k-1)(a-b)}^{\mathrm{II}}, \ldots, \mathbf{b}_{i-B+k(a-b)-1}^{\mathrm{II}}$ using $\{\mathbf{b}_j^{\mathrm{I}}\}_{j \ge i-B-(k-1)b}$ and the parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ between $i + T \le t < \mathcal{T}$.

Once this recursion terminates, all the non-urgent sub-symbols $\{\mathbf{s}^N[\tau]\}_{\tau=i-\alpha B}^{i-1}$ are recovered by time $\mathcal{T} - 1$. The proof of this recursion is also similar to the previous section and will be omitted.

Finally the assumption that $T_1$ is a multiple of $b$ (i.e. $\alpha T_1$ is an integer) can be relaxed through a source pseudo-expansion approach as follows:

- Split each source symbol into $nT_1$ sub-symbols $s_0[i], \ldots, s_{nT_1-1}[i]$ where $n$ is the smallest integer such that $n\alpha T_1$ is an integer.
- Construct an expanded source sequence $\tilde{s}[.]$ such that $\tilde{s}[ni + r] = (s_{rT_1}[i], \ldots, s_{(r+1)T_1-1}[i])$ where $r \in \{0, \ldots, n-1\}$.
- We apply a DESCo code with parameters $\{(nB_1, nT_1) - (n\alpha B_1, n(\alpha T_1 + B_1))\}$ to $\tilde{s}[.]$ using the earlier construction.

Notice that since the channel introduces a total of $B$ erasures on the original input there will be at-most $nB_i$ erasures on the expanded stream. These will be decoded with a delay of $nT_i$ on the expanded stream, which can be easily verified to incur a delay of $\lceil T_i \rceil$ on the original stream.

## VIII. Numerical Results

We compare the performance of proposed DE-SCo codes to that of sequential random network codes numerically and discuss advantages and disadvantages of the proposed codes. In our experiments we divide the coded data stream into segments of 2000 packets each and generate one burst erasure in each segment. Each packet occupies one millisecond. The burst erasure length is uniformly distributed between $[0, B_{max}]$ packets and a packet is declared to be lost if it is not recovered by its deadline. We plot the average loss probability for a stream of $10^5$ segments for both; (1) DE-SCo code with burst-delay parameters $\{(B, T), (\alpha B, \alpha T + B)\}$ for $\alpha = 2$ and (2) sequential random network code of the same rate for the two users in Fig. 7 and Fig. 8 respectively as a function of the maximum erasure burst length.

We make the following observations based on the results.

- We see that if the maximum size of erasure burst is less than a critical threshold for each scheme then the loss probability is zero. For the DES-Co construction this threshold equals $B_i$. For random codes at rate $R$ this threshold equals $\lceil (1-R)T \rceil$ where $T$ is the allowable reconstruction delay.
- The loss probability for each code increases beyond its threshold. For DE-SCo we assume that whenever an erasure burst of length $B_i$ occurs, the entire burst is lost. For random codes, partial bursts are lost starting from $\lceil (1-R)T_i \rceil$ but ultimately when the erasure burst is sufficiently large the entire burst is lost. For example, consider a DE-SCo $(4,4)-(8,12)$ with rate $R = 1/2$ and the delay at user 1 $T_1 = 4$. Random codes starts losing packets when burst length exceeds $\lceil (1-R)T_1 \rceil = 2$ for the first user while the threshold for DE-SCo is $B_1 = 4$. When the burst erasure is 3, the random code is capable of recovering the lost packets with delays $3, 4, 5$ and only the packet with delay of 5 is marked erased. if the burst length reaches $5 > 4$ then all the packets are declared lost.
- DE-SCo outperforms random network coding for user 1 for the erasure bursts of length between $B \in \{\lceil (1-R)T_2 \rceil, B\}$. For burst lengths larger than $B_i$ both schemes do not recover the erased symbols by the deadline. This explains why DE-SCo dominates random network coding in Fig. 7.
- For user 2, unlike DES-Co, random codes fail to correct all erasures for burst lengths between $\{\lceil (1-R)T_i \rceil, B_2\}$.

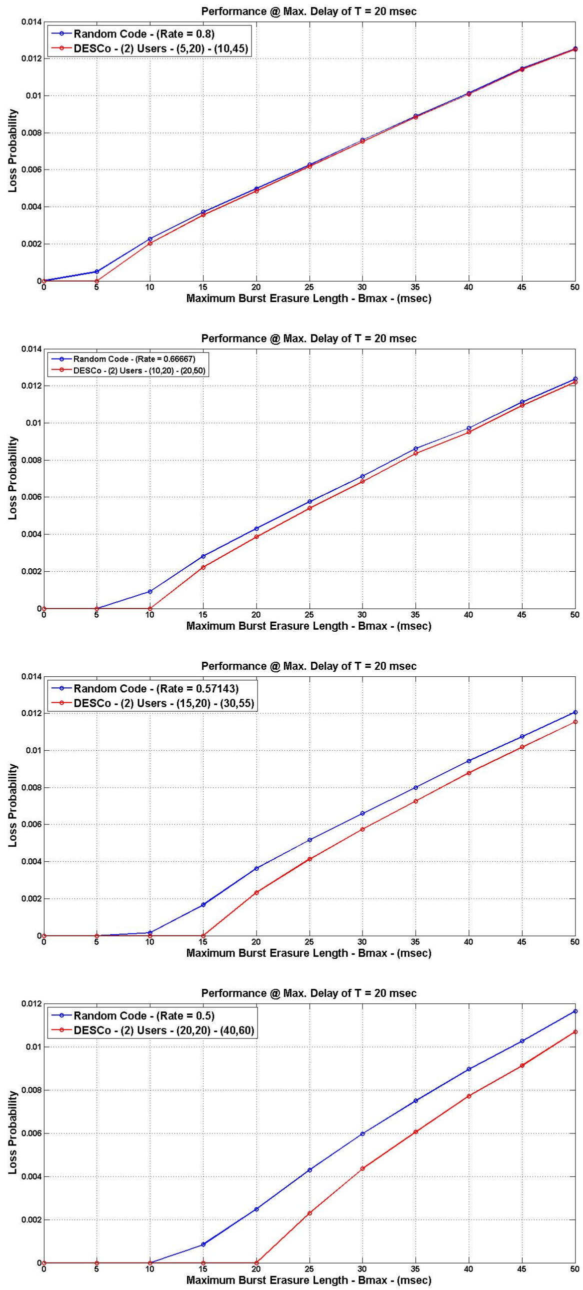


Fig. 7: Loss Probability @ First Receiver.

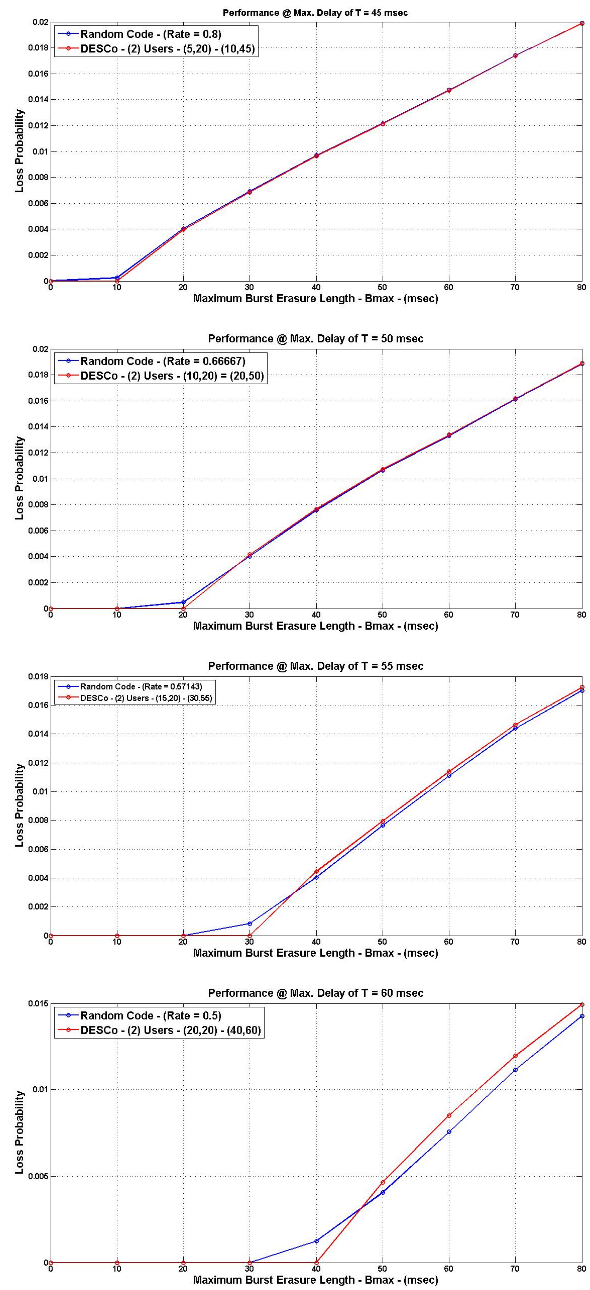


Fig. 8: Loss Probability @ Second Receiver.

Beyond this threshold however DE-SCo experiences entire erasure bursts whereas random network coding still incurs partial erasures until a larger threshold of $B_2 + \frac{B_1^2}{T}$ is reached. This explains why DE-SCo does not outperform random network coding in the high loss regime for user 2.

## IX. Conclusion

This paper constructs a new class of streaming erasure codes that do not commit apriori to a given delay, but rather achieve a delay based on the channel conditions. We model this setup as a multicast problem to two receivers whose channels introduce different erasure-burst lengths and require different delays. The DE-SCo construction embeds new parity checks into the single-user code, in a such a way that we do not compromise the single user performance of the stronger user while the supporting the weaker receiver with an information theoretically optimum delay. We provide an explicit construction of these codes as well as the associated decoding algorithm. Numerical simulations suggest that these codes outperform simple random linear coding techniques that do not exploit the burst-erasure nature of the channel.

A number of interesting future directions remain to be explored. The general problem of designing codes that are optimal for any feasible pair $\{(B_1, T_1), (B_2, T_2)\}$ remains open. We expect to report some recent progress along this lines in the near future. While our construction can be naturally extended to more than two users the optimality remains to be seen. Our initial simulation results indicate that the performance gains of the proposed code constructions are limited to burst-erasure channels. Designing codes with similar properties for more general channels remains an interesting future direction.

## REFERENCES


[1] E. Martinian, “Dynamic Information and Constraints in Source and Channel Coding,” *Ph.D. Thesis,* Massachusetts Inst. of Technology, September 2004.

[2] E. Martinian and M. Trott, “Delay-optimal Burst Erasure Code Construction,” *International Symposium on Information Theory,* (Nice, France) July 2007.

[3] A. Khisti and J.P. Singh, “On multicasting with streaming burst-erasure codes,” *International Symposium on Information Theory,* (Seoul, Korea) July 2009.

[4] F. Jelinek, “Upper bounds on sequential decoding performance parameters, IEEE Trans. Inform. Theory, vol. 20, no. 2, pp. 227—239, Mar.1974

[5] A. Sahai, “Why Do Block Length and Delay Behave Differently if Feedback Is Present?,” IEEE Transactions on Information Theory, May 2008, Volume: 54, Issue: 5, pp. 1860 - 1886

[6] J.K.Sundararajan, D. Shah, and M. Medard, “ARQ for network coding,” in IEEE ISIT 2008, Toronto, Canada, Jul. 2008.

[7] C. Chang, S. C. Draper and A. Sahai,“Lossless coding for distributed streaming sources,” submitted IEEE Trans. Inform. Theory, 2007

[8] M.Kalman, E.Steinbach,and B.Girod, “Adaptive media playout for low-delay video streaming over error-prone channels,” IEEE Trans. on Circuits and Systems for Video Technology, 14(6):841851, 2004.

[9] G. D. Forney, Jr., “Convolutional Codes I: Algebraic Structure,” IEEE Transactions on Information Theory, vol. IT-16, pp. 720738, November 1970.