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 symbol 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 bursty 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 reconstruct 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 are not ideally suited for streaming sources. 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. Nevertheless there has been a significant interest in adapting such constructions for streaming applications see e.g., [13], [14], [15], [16], [17], [18].

In [1, Chapter 8] a class of systematic time-invariant convolutional codes *streaming erasure codes* (SCo) are proposed for the burst erasure channel. The encoder observes a semi-infinite source stream and maps it to a coded output stream of rate R. The channel considered is 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 symbol 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 [11] 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.

We introduce a class of streaming codes that 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 for $i = 1, 2$. 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.

In recent works, [5], [6], [7] study the low-delay codes with feedback, the compression of streaming sources is studied in [10] while a comparison of block and streaming codes for low delay systems is provided in [9].

II. System Model

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

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

Manuscript received July, 2003; revised December, 2010. 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

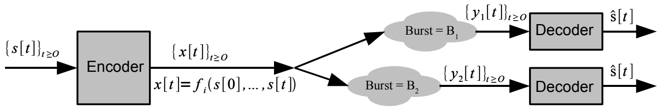


Fig. 1. The source stream $\{\mathsf{s}[i]\}$ is causally mapped into an output stream $\{\mathsf{x}[i]\}$. Both the receivers observe these symbols via their channels. The channel introduces an erasure-burst of length B_i, and each receiver tolerates a delay of T_i, for $i = 1, 2$.

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

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

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

The source stream is an i.i.d. sequence and we assume that each symbol is sampled from a distribution $p_{\mathsf{s}}(\cdot)$ over the finite field $\mathbb{F}_Q^T$. The rate of the multicast code is defined as ratio of the 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.*

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 in 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

$$\mathbf{G} = \begin{array}{c} \\ {\scriptstyle (T-B)} \end{array} \begin{array}{cc} {\scriptstyle T-B} & {\scriptstyle B} \\ \left[\; \mathbf{I} \right. & \left. \mathbf{H} \; \right] \end{array} \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix and $\mathbf{H}$ 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 $T - B$

$$\mathbf{b} = \begin{array}{cc} {\scriptstyle B} & {\scriptstyle T-B} \\ \left[\; \mathbf{u} \right. & \left. \mathbf{n} \; \right] \end{array}, \tag{6}$$

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

$$s[t] = \begin{bmatrix} s_0[t] \\ \vdots \\ s_{T-1}[t] \end{bmatrix} \in \mathbb{F}_Q^T, \qquad x[t] = \begin{bmatrix} s_0[t] \\ \vdots \\ s_{T-1}[t] \\ p_0[t] \\ \vdots \\ p_{B-1}[t] \end{bmatrix} \in \mathbb{F}_Q^{T+B} \tag{9}$$

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

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 $p_0[t], \dots, p_{B-1}[t]$ are constructed through a diagonal interleaving technique described below.

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

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

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.,

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

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

$$\begin{aligned}\mathbf{d}_t = \quad & (s_0[t], \dots, s_{T-1}[t+T-1], p_0[t+T], \qquad (12)\\ & \dots, p_{B-1}[t+T+B-1]).\end{aligned}$$

The SCo code is a time-invariant convolutional code [12]. The inputs to the convolutional code are source symbols $\mathbf{s} \in \mathbb{F}_Q^T$, while the outputs are channel symbols $\mathbf{x} \in \mathbb{F}_Q^{T+B}$. We emphasize that the actual transmitted symbol is given in (9). The diagonal codeword (12) above simply maps the LD-BEBC to a SCo.

B. Decoding of SCo Codes

The structure of the diagonal codeword (12) is also important in decoding. Suppose that symbols $x[t], \dots, x[t+B-1]$ are erased. It can be readily verified that there are no more than B erasures in each diagonal codeword $\{\mathbf{d}_t\}$ (c.f. (12)). Since each codeword 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.

C. Example: (2,3) SCo Code

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

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

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

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

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

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

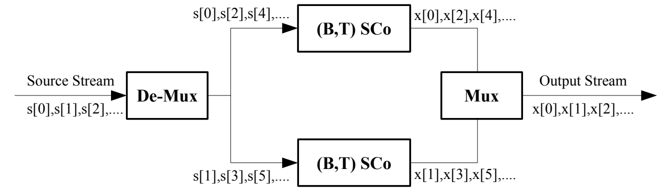


Fig. 2. A vertical interleaving approach to construct a $(2B, 2T)$ SCo code from a (B, T) SCo code.

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 αT sub-symbols. However we can take advantage of the multiplicity factor α and simply construct the $(\alpha B, \alpha T)$ SCo code from the (B, T) SCo code via vertical interleaving of step α.

Fig. 2 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 symbols at even time slots. Likewise we apply a (B, T) SCo on the second sub stream to produce channel symbols at odd time slots. Since a burst of length $2B$ introduces B erasures on either sub-streams, 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

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

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

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

As in the case of $\alpha = 2$, the decoding proceeds by splitting the source stream into α 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 αT on the original stream.

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

TABLE I
A (2,3) SINGLE USER SCO CODE CONSTRUCTION IS GIVEN WHERE EACH SOURCE SYMBOL $\mathbf{s}[.]$ IS DIVIDED INTO THREE SUB-SYMBOLS $s_0[.]$, $s_1[.]$ AND $s_2[.]$ AND A $(5,3)$ LD-BEBC CODE IS THEN APPLIED ACROSS THE DIAGONAL TO GENERATE TWO PARITY CHECK SUB-SYMBOLS GENERATING A RATE $3/5$ CODE. EACH COLUMN CORRESPONDS TO ONE CHANNEL SYMBOL.

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.,

$$x[t] = f(s[t], s[t-1], \dots, s[t-T]). \tag{18}$$

Furthermore a closer look at the parity check sub-symbols (9) of $x[t]$ reveals that the parity checks $p_0[t], \dots, p_{B-1}[t]$ constructed from the LD-BEBC in (8) have the form

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

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 sub-symbols (6). The mapping of source sub-symbols to information vector (10) then implies that the sub-symbols $s_0, \dots, s_{B-1}$ are the urgent sub-symbols in the source stream whereas the sub-symbols $s_B, \dots, s_{T-1}$ are the non-urgent sub-symbols. We will denote these by

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

The urgent and non-urgent sub-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 $x[i-B], \dots, x[i-1]$ are erased by the burst-erasure channel. Then*

1) *All sub-symbols in $\mathbf{s}^N[i-B], \dots, \mathbf{s}^N[i-1]$ are obtained from the parity checks $\mathbf{p}[i], \dots, \mathbf{p}[i+T-B-1]$.*
2) *The sub-symbols in $\mathbf{s}^U[j]$ for $i-B \leq j < i$ 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 (18), (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

$$\bar{\mathbf{b}}_t = (s_0[t], s_1[t-1], \dots, s_{T-1}[t-(T-1)]) \tag{21}$$

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

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

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

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 sub-symbols

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

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 to the $(1, 2)$ SCo, 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

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

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

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

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

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 $B=1$.

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

$s_0[i-1]$ $s_1[i-1]$	$\boxed{s_0[i]}$ $\boxed{s_1[i]}$	$s_0[i+1]$ $\boxed{s_1[i+1]}$	$s_0[i+2]$ $s_1[i+2]$	$s_0[i+3]$ $s_1[i+3]$	$s_0[i+4]$ $s_1[i+4]$
$s_0[i-3]+s_1[i-2]$ $+$ $s_0[i-7]+s_1[i-5]$	$s_0[i-2]+s_1[i-1]$ $+$ $s_0[i-6]+s_1[i-4]$	$s_0[i-1]+s_1[i]$ $+$ $s_0[i-5]+s_1[i-3]$	$\boxed{s_0[i]+s_1[i+1]}$ $+$ $s_0[i-4]+s_1[i-2]$	$s_0[i+1]+s_1[i+2]$ $+$ $s_0[i-3]+s_1[i-1]$	$s_0[i+2]+s_1[i+3]$ $+$ $\boxed{s_0[i-2]+s_1[i]}$

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

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

TABLE III

RATE $2/3$ CODE CONSTRUCTIONS THAT SATISFY USER 1 WITH $(B_1,T_1)=(1,2)$ AND USER 2 WITH $B_2=2$.

the parity checks of user 1 i.e., when $s[i]$ is erased, receiver 1 can recover $p^{\text{I}}[i+1]$ and $p^{\text{I}}[i+2]$ from $q[i+1]$ and $q[i+2]$ respectively by canceling $p^{\text{II}}[\cdot]$ that combine with these symbols. It then recovers $s[i]$. Likewise if $s[i]$ and $s[i-1]$ are erased, then receiver 2 recovers $p^{\text{II}}[i+1],\ldots,p^{\text{II}}[i+4]$ from $q[i+3],\ldots,q[i+6]$ respectively by canceling out the interfering $p^{\text{I}}[\cdot]$, thus yielding $T_2=6$.

While the interference avoidance strategy illustrated above naturally generalizes to arbitrary values of B and T, it 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 $\check{p}^{\text{II}}[i]=s_1[i-2]+s_0[i-1]$ by combining the source sub-symbols along the opposite diagonal of the $(1,2)$ SCo code in Table II(a). Note that $\check{x}(i)=(s[i],\check{p}^{\text{II}}[i])$ 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 $p^{\text{I}}[i]$ i.e., $q[i]=p^{\text{I}}[i]+\check{p}^{\text{II}}[i-3]$. In the resulting code, receiver 1 is still able to cancel the effect of $\check{p}^{\text{II}}[\cdot]$ 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 this paper we generalize the above construction to arbitrary values of (B_i,T_i).

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 α 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 less than $R=\frac{T}{T+B}$.

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

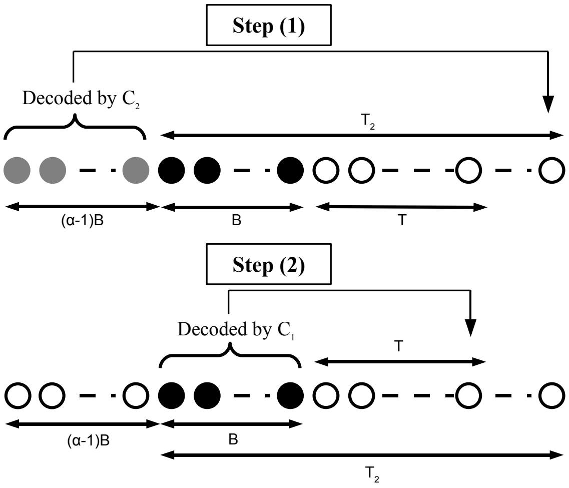


Fig. 3. One period illustration of the Periodic Erasure Channel for $T+B < T_2 \leq \alpha T + B$. White circles resemble unerased symbols. Black and Gray circles resemble erased symbols to be recovered using $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

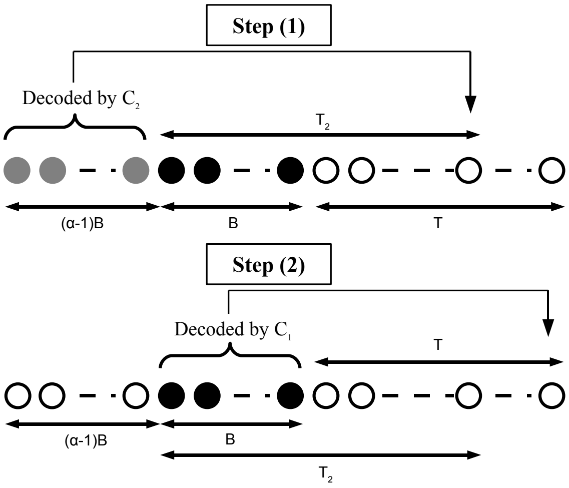


Fig. 4. One period illustration of the Periodic Erasure Channel for $T_2 \leq T + B$. White circles resemble unerased symbols. Black and Gray circles resemble erased symbols to be recovered using $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

As shown in Fig. 3, construct a periodic burst-erasure channel in which every period of $(\alpha-1)B + T_2$ symbols consists of a sequence of αB erasures followed by a sequence of non-erased symbols. Consider one period of the proposed periodic erasure channel with a burst erasure of length αB from time $t = 0, 1, \ldots, \alpha B - 1$ followed by a period of non-erasures for $t = \alpha B, \ldots, \mathcal{T}_{P_1} \triangleq (\alpha-1)B + T_2 - 1$. For time $t = 0, \ldots, \mathcal{T}_{P_1}$ the channel behaves identically to a burst-erasure channel with αB 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 $\mathcal{T}_{P_1}$ and hence the channel symbols $x[0], \ldots, x[(\alpha-1)B-1]$ 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 $\mathcal{T}_{P_1}$. 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 $\alpha B - 1 + T \leq \mathcal{T}_{P_1}$, which follows since $T_2 \geq T + B$. Thus all the erased channel symbols in the first period are recovered by time $\mathcal{T}_{P_1}$. 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 αB erasures,thus the capacity is upper bounded by $1 - \frac{\alpha B}{(\alpha-1)B+T_2}$ which is less than $R = \frac{T}{T+B}$ if $T_2 < T_2^\star$.

For the other case with $T_2 < T + B$ shown in Fig. 4, the same argument applies except that the periodic channel has a period of $T + \alpha B$ symbols. Each period consists of a burst erasure of length αB from time $t = 0, 1, \ldots, \alpha B - 1$ followed by a period of non-erasures for $t = \alpha B, \ldots, \mathcal{T}_{P_2} \triangleq \alpha B + T - 1$. The decoder of user 2 recovers the $(\alpha-1)B$ erasures at time $t = 0, 1, \ldots, (\alpha-1)B - 1$ with a delay of T_2 (i.e., by time $< \mathcal{T}_{P_2}$) as $T_2 < T + B$. Furthermore, the decoder of user 1 recovers the B erasures at time $t = (\alpha-1)B, \ldots, \alpha B - 1$ with a delay of T symbols (i.e., by time $\alpha B + T - 1 = \mathcal{T}_{P_2}$). Now, the length of each period is $\alpha B + T$ with T available symbols, the rate is $\frac{T}{T+\alpha B}$ strictly smaller than R as $\alpha > 1$ and the converse follows.

B. Code Construction

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

- **Construction of** $\mathcal{C}_1$: 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.

  In other words, a $(T+B, T)$ LD-BEBC code is applied along the diagonal $\mathbf{b}_i^{\mathrm{I}} = (s_0[i], s_1[i], \ldots, s_{T-1}[i + T - 1])$ constructing the diagonal codeword $\mathbf{d}_i^{\mathrm{I}} = (s_0[i], \ldots, s_{T-1}[i+T-1], p_0^{\mathrm{I}}[i+T], \ldots, p_{B-1}^{\mathrm{I}}[i + T + B - 1])$ where, from (19),

$$\begin{aligned} p_k^{\mathrm{I}}[i] &= \mathcal{A}_k(s_0[i-T-k], \ldots, s_{T-1}[i-1-k]) \\ &= \mathcal{A}_k(\mathbf{b}_{i-T-k}^{\mathrm{I}}) \\ &= s_k[i-T] + h_k(s_B[i-k-T+B], \ldots, \\ &\quad s_{T-1}[i-k-1]), \qquad k = 0, \ldots, B-1. \end{aligned} \tag{26}$$

- **Construction of** $\mathcal{C}_2$: Let $\mathcal{C}_2$ be a $((\alpha-1)B, (\alpha-1)T)$ SCo also obtained by splitting each source symbol 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 $\ell = (\alpha - 1)$.

  In other words, a $(T+B, T)$ LD-BEBC code is applied along the diagonal $\mathbf{b}_i^{\mathrm{II}} = (s_{T-1}[i - \ell(T-1)], s_{T-2}[i - \ell(T-2)], \ldots, s_0[i])$ to construct a diagonal codeword $\mathbf{d}_i^{\mathrm{II}} = (s_{T-1}[i - \ell(T-1)], \ldots, s_0[i], p_0^{\mathrm{II}}[i +$

$\ell], \dots, p^{\mathrm{II}}_{B-1}[i+\ell B])$ where

$$\begin{aligned} p^{\mathrm{II}}_k[i] &= \mathcal{B}_k(s_0[i-\ell-k\ell], \dots, s_{T-1}[i-\ell T-k\ell]) \\ &= \mathcal{B}_k(\mathbf{b}^{\mathrm{II}}_{i-\ell-k\ell}) \\ &= s_{T-k-1}[i-\ell T] + h_k(s_{T-B-1}[i-\ell(k+T-B)], \\ &\dots, s_0[i-\ell(k+1)]), \quad k=0,\dots,B-1, \qquad (27) \end{aligned}$$

- **Combination of Parity Checks of $\mathcal{C}_1$ and $\mathcal{C}_2$**: 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 $x[i] = (s[i], \mathbf{q}[i])$

Throughout our discussion we refer to the non-urgent and urgent symbols of code $\mathcal{C}_2$. The set of urgent symbols and non-urgent symbols are as stated in (24). Also note that 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. 5 illustrates the DE-SCo $\{(2,5),(4,12)\}$ construction. Each column represents one time-index between $[-4,9]$ shown in the top row of the table. We assume that a burst-erasure occurs between time $[-4,-1]$ for user 2. Each source symbol is split into five sub-symbols $(a[.], b[.], c[.], d[.], e[.])$, each occupying one row. The first $T-B=3$ of which, $a[.], b[.], c[.]$ are non-urgent sub-symbols and the rest $(d[.], e[.])$ are urgent. The next two rows denote the parity check sub-symbols. The parity checks for $\mathcal{C}_1$, generated by diagonal $\mathbf{b}^{\mathrm{I}}_i$, (c.f. (19)) are marked $p[.]$ and $q[.]$ and are given by,

$$\begin{aligned} p[i] &= a[i-5] + c[i-3] + e[i-1] \\ q[i] &= b[i-5] + d[i-3] + e[i-2]. \qquad (28) \end{aligned}$$

The shaded two top rows show the parity checks $y[.]$ and $z[.]$, generated by the diagonal $\mathbf{b}^{\mathrm{II}}_i$ for $\mathcal{C}_2$ (c.f. (27)),

$$\begin{aligned} y[i] &= e[i-5] + c[i-3] + a[i-1] \\ z[i] &= d[i-5] + b[i-3] + a[i-2]. \qquad (29) \end{aligned}$$

These parity checks are then shifted by $T+B=7$ slots and combined with the corresponding parity checks of $\mathcal{C}_1$ as shown in Fig. 5.

We illustrate the decoding steps for user 2 as follows.

(1) Recover $\{\mathbf{p}^{\mathrm{II}}[t-\Delta]\}_{t \geq T}$:
By construction of $\mathcal{C}_1$ all the parity checks $\mathbf{p}^{\mathrm{I}}[t]$ for $t \geq 5$ do not involve the erased sub-symbols. In particular the parity checks marked by $p[.]$ and $q[.]$ at $t \geq 5$ do not involve source sub-symbols before $t=0$ (c.f. (28)) and hence these can be canceled to recover the parity checks $y[.]$ and $z[.]$ for $t \geq 5$.

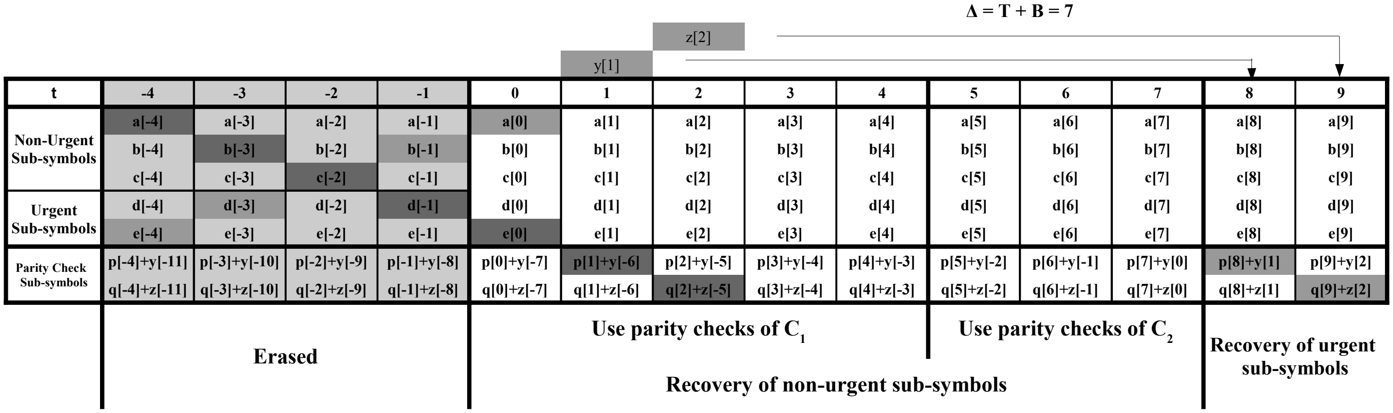
(2) Upper-left triangle:
The parity checks in step (1) enable us to recover the non-urgent erased sub-symbols in $\mathbf{b}^{\mathrm{II}}_{-3} = (e[-7], d[-6], c[-5], b[-4], a[-3])$ and $\mathbf{b}^{\mathrm{II}}_{-4} = (e[-8], d[-7], c[-6], b[-5], a[-4])$ which are $a[-4]$, $a[-3]$ and $b[-4]$ i.e., the upper-left triangle sub-symbols. We use the corresponding diagonal codewords, $\mathbf{d}^{\mathrm{II}}_{-3} = (e[-7], d[-6], c[-5], b[-4], a[-3], y[-2], z[-1])$ to recover $a[-3]$ and $b[-4]$ from the parity checks $y[-2]$ and $z[-1]$ and $\mathbf{d}_{-4} = (e[-8], d[-7], c[-6], b[-5], a[-4], y[-3], z[-2])$ to recover $a[-4]$ from the parity check $z[-2]$. We note that $a[-4]$ is recovered from $z[-2]$ at $t=5$ and not from $y[-3]$ which appears at $t=4$ and is not recovered in step (1). More generally, as we note later, the parity checks at $i+T$ and later suffice to recover symbols in this step.

(3) Recover $\mathbf{p}^{\mathrm{I}}[t]$ 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 $0 \leq t \leq 4$. Note that the relevant interfering parity checks from $\mathbf{p}^{\mathrm{II}}[\cdot]$ in this interval is $y[-3] = e[-8] + c[-6] + a[-4]$. Since the only erased sub-symbol $a[-4]$ is already recovered in step (2), these parity checks can be canceled. More generally as we show later, for the general case, our construction guarantees that the interfering parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ in the interval $0 \leq t \leq T-1$ only involve erased symbols from the upper left triangle, which are decoded in step (2).

(4) Upper-right triangle:
Since the diagonals $\mathbf{b}^{\mathrm{I}}_{-2} = (a[-2], b[-1], c[0], d[1], e[2])$ and $\mathbf{b}^{\mathrm{I}}_{-1} = (a[-1], b[0], c[1], d[2], e[3])$ involve two or fewer erasures, we can now recover these sub-symbols using parity checks of code $\mathcal{C}_1$ recovered in the previous step. In particular, the upper-right triangle source sub-symbols $a[-2]$, $b[-1]$ and $a[-1]$ can be recovered from $p[3]$, $q[4]$ and $p[4]$ respectively.

(5) Recover non-urgent sub-symbols recursively:
The remaining non-urgent sub-symbols need to be recovered in a recursive manner. Note that $\mathbf{b}^{\mathrm{I}}_{-3} = (a[-3], b[-2], c[-1], d[0], e[1])$ has three erased sub-symbols. However, the first sub-symbol $a[-3]$ also belongs to $\mathbf{b}^{\mathrm{II}}_{-3}$ and has already been recovered in step (2). The remaining two sub-symbols, $b[-2]$ and $c[-1]$, can be recovered by the two available parity checks of code $\mathcal{C}_1$ in $\mathbf{d}^{\mathrm{I}}_{-3} = (a[-3], b[-2], c[-1], d[0], e[1], p[2], q[3])$, i.e., from $p[2]$ and $q[3]$. Similarly $\mathbf{b}^{\mathrm{II}}_{-2} = (e[-6], d[-5], c[-4], b[-3], a[-2])$ also has three erasures, but the upper-most sub-symbol $a[-2]$ also belongs to $\mathbf{b}^{\mathrm{I}}_{-2}$ which has been recovered in step (4). Hence the remaining erased sub-symbols in $\mathbf{b}^{\mathrm{II}}_{-2}$, $c[-4]$ and $b[-3]$, can be recovered using the parity checks $y[-1]$ and $z[0]$ in $\mathbf{d}^{\mathrm{II}}_{-2} = (e[-6], d[-5], c[-4], b[-3], a[-2], y[-1], z[0])$.
At this stage it only remains to recover the two remaining non-urgent sub-symbols $c[-3]$ and $c[-4]$ by time $t=7$. These are recovered in the next step of the recursion. Note that the symbols $c[-2]$ and $d[-1]$ are the only remaining erased symbols on the diagonal $\mathbf{b}^{\mathrm{I}}_{-5}$ and are recovered from parity checks $p[1]$ and $q[2]$. Likewise, $c[-3]$ and $d[-4]$ are the only remaining erase symbols on the diagonal $\mathbf{b}^{\mathrm{II}}_{-1}$ and can be recovered using the parity checks $y[0]$ and $z[1]$. Since $c[-3]$ is the non-urgent symbol, from Prop. 1 it is recovered before $d[-4]$ using only $y[0]$. Thus both $c[-3]$ and $c[-4]$ are recovered by $t=7$.

(6) Recover urgent sub-symbols:



t	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
Non-Urgent Sub-symbols	a[-4]	a[-3]	a[-2]	a[-1]	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
	b[-4]	b[-3]	b[-2]	b[-1]	b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]
	c[-4]	c[-3]	c[-2]	c[-1]	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
Urgent Sub-symbols	d[-4]	d[-3]	d[-2]	d[-1]	d[0]	d[1]	d[2]	d[3]	d[4]	d[5]	d[6]	d[7]	d[8]	d[9]
	e[-4]	e[-3]	e[-2]	e[-1]	e[0]	e[1]	e[2]	e[3]	e[4]	e[5]	e[6]	e[7]	e[8]	e[9]
Parity Check Sub-symbols	p[-4]+y[-11]	p[-3]+y[-10]	p[-2]+y[-9]	p[-1]+y[-8]	p[0]+y[-7]	p[1]+y[-6]	p[2]+y[-5]	p[3]+y[-4]	p[4]+y[-3]	p[5]+y[-2]	p[6]+y[-1]	p[7]+y[0]	p[8]+y[1]	p[9]+y[2]
	q[-4]+z[-11]	q[-3]+z[-10]	q[-2]+z[-9]	q[-1]+z[-8]	q[0]+z[-7]	q[1]+z[-6]	q[2]+z[-5]	q[3]+z[-4]	q[4]+z[-3]	q[5]+z[-2]	q[6]+z[-1]	q[7]+z[0]	q[8]+z[1]	q[9]+z[2]

Fig. 5. A $\{(2,5)-(4,12)\}$ DE-SCo code construction is given in the above figure. The parity check sub-symbols $p[t]$ and $q[t]$ of a $(2,5)$ SCo across the main diagonal is added to another $(2,5)$ SCo parity check sub-symbols $x[t]$ and $y[t]$ but applied across the opposite diagonal and shifted by $T+B=7$ (i.e., the two parity check checks at time instant t are $p[t]+x[t-7]$ and $q[t]+y[t-7]$).

After recovering all non-urgent sub-symbols in the previous steps, we can directly recover the urgent ones (i.e., the bottom two rows) using parity checks $\mathbf{p}^{\mathrm{II}}[t]$ for $8 < t \leq 11$.

We now study the general case.

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 sub-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 $i-\alpha B,\ldots,i-1$ are erased for receiver 2. Let $\mathcal{T}\stackrel{\Delta}{=}i-\alpha B+T_2^\star$. We use parity checks at time $i\leq t\leq\mathcal{T}-1$ to recover $\{\mathbf{s}^N[\tau]\}_{\tau=i-\alpha B}^{i-1}$ in the first five steps where $\mathbf{s}^N[\tau]=(s_0[\tau],\ldots,s_{T-B-1}[\tau])$ denote the set of *non-urgent* sub-symbols for $\mathcal{C}_2$. In the last step, we use parity checks at time $t\geq\mathcal{T}$ to recover the set of *non-urgent* sub-symbols for $\mathcal{C}_2$, $\mathbf{s}^U[\tau]=(s_{T-B}[\tau],\ldots,s_{T-1}[\tau])$.

(1) Recover $\{\mathbf{p}^{\mathrm{II}}[t-\Delta]\}_{t\geq i+T}$:
For $t\geq i+T$, the decoder recovers 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 as via (19) the memory in $\mathcal{C}_1$ is limited to previous T symbols. Consequently the parity check symbols $\{\mathbf{p}^{\mathrm{I}}[t]\}_{t\geq i+T}$ depend only on source sub-symbols after time i. Hence these parity checks can be canceled.

(2) Upper-left triangle:
In this step, the decoder recovers the non-urgent sub-symbols in $\mathbf{b}_{i-\alpha B}^{\mathrm{II}},\ldots,\mathbf{b}_{i-B-1}^{\mathrm{II}}$ using the parity check symbols $\{\mathbf{p}^{\mathrm{II}}[t-\Delta]\}_{t=i+T}^{\mathcal{T}-1}$. Clearly these vectors are affected by at most $(\alpha-1)B$ 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 in step (1). By construction $\mathcal{C}_2$ can recover the erased source sub-symbols in the stated diagonal vectors. Furthermore by applying Prop. 1, the non-urgent sub-symbols are recovered from the first $(\alpha-1)(T-B)$ parity check columns. Taking into account the shift of $\Delta=T+B$, it follows that all the non-urgent source sub-symbols are recovered by time $i+T+(\alpha-1)(T-B)-1=\mathcal{T}-1$.

(3) Recover $\mathbf{p}^{\mathrm{I}}[t]$ for $i\leq t\leq i+T-1$:
We consider the last column of parity checks, $\mathbf{q}[i+T-1]=\mathbf{p}^{\mathrm{I}}[i+T-1]+\mathbf{p}^{\mathrm{II}}[i-B-1]$. From (27), for $k=0,1,\ldots,B-1$ we have,

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

Thus the only urgent sub-symbols involved in $\mathbf{p}^{\mathrm{II}}[i-B-1]$ are at time $t=i-B-1-(\alpha-1)T$, which are unerased. Moreover, the non-urgent sub-symbols involved are those of $\mathbf{b}_{i-B-(\alpha-1)(k+1)-1}^{\mathrm{II}}$ which have already been recovered in step (2). Thus, 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 all the 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 $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$.

(4) Upper-right triangle:
In this step, the decoder recovers the non-urgent sub-symbols in $\mathbf{b}_{i-1}^{\mathrm{I}},\ldots,\mathbf{b}_{i-B}^{\mathrm{I}}$ using the parity checks $\mathbf{p}^{\mathrm{I}}[i],\ldots,\mathbf{p}^{\mathrm{I}}[i+T-1]$. Step (4) follows in a similar way to step (2). The diagonal vectors $\mathbf{b}_{i-B}^{\mathrm{I}},\ldots,\mathbf{b}_{i-1}^{\mathrm{I}}$ 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 $\{\mathbf{p}^{\mathrm{I}}[t]\}_{i\leq t<i+T}$ recovered earlier are capable of recovering the erased source sub-symbols in these diagonal vectors by at most time $i+T-1<\mathcal{T}$.

(5) Recover non-urgent sub-symbols recursively:
For each $k \in \{1, \ldots, T-B-1\}$ recursively recover the remaining non-urgent sub-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 $\{\mathbf{b}^{\mathrm{II}}_j\}_{j \le i+(k-1)(\alpha-1)-B-1}$ and parity checks $\mathbf{p}^{\mathrm{I}}[\cdot]$ between $i \le t < i+T$.

(Ind. 2) Recover the non-urgent sub-symbols in $\mathbf{b}^{\mathrm{II}}_{i-B+(k-1)(\alpha-1)}, \ldots, \mathbf{b}^{\mathrm{II}}_{i-B+k(\alpha-1)-1}$ using $\{\mathbf{b}^{\mathrm{I}}_j\}_{j \ge i-B-(k-1)}$ 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 $\mathcal{T}-1$.

We establish the claim of the recursion using induction. Consider the case when $k=1$. According to Ind. 1 the non-urgent sub-symbols $\{\mathbf{b}^{\mathrm{II}}_j\}_{j \le i-B-1}$ are available (from step 1). To recover $\mathbf{b}^{\mathrm{I}}_{i-B-1}$, note that the only erased sub-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 sub-symbols. According to Ind. 2 the non-urgent sub-symbols in $\{\mathbf{b}^{\mathrm{I}}_j\}_{j \ge i-B}$ have been recovered in step (4). Furthermore in vectors $\mathbf{b}^{\mathrm{II}}_{i-B}, \ldots, \mathbf{b}^{\mathrm{II}}_{i-B+\alpha-2}$ the only erased sub-symbols after time $i-B-1$ are $s_0[i-B], \ldots, s_0[i-B+\alpha-2]$, which are available from $\{\mathbf{b}^{\mathrm{I}}_j\}_{j \ge i-B}$. 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 $j=0,\ldots,k$. Note that $s_j[i-B-k+j-1]$ is precisely the $j-$th sub-symbol in the diagonal vector $\mathbf{b}^{\mathrm{II}}_{i-B-k+\alpha j-1}$. Furthermore the diagonals of interest $\mathbf{b}^{\mathrm{II}}_{i-B-k-1}, \ldots, \mathbf{b}^{\mathrm{II}}_{i-B+(\alpha-1)k-1}$, already visited in Ind. 2 in the k-th recursion. Hence the remaining sub-symbols are recovered using the parity checks of $\mathcal{C}_1$.

For Ind. 2, the first vector of interest at step $k+1$ is

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

Note that the sub-symbols $s_0[.], \ldots, s_k[.]$ above, also belong to vectors $\mathbf{b}^{\mathrm{I}}_{i-B+(\alpha-1)k}, \ldots, \mathbf{b}^{\mathrm{I}}_{i-B-k}$, and are recovered in Ind. 1 by the $k-$th step. Since the remaining erased symbols span the interval $[i-\alpha B, i-B)$ the parity checks $\{\mathbf{p}^{\mathrm{II}}[\cdot]\}_{t \ge i-B}$ recovered in step (3) can be used to recover these erased symbols.

Likewise, the last vector of interest at step $k+1$ is

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

Note that the sub-symbols $s_0[.], \ldots, s_k[.]$ above, also belong to vectors $\mathbf{b}^{\mathrm{I}}_{i-B+(k+1)(\alpha-1)-1}$, $\ldots$ $\mathbf{b}^{\mathrm{I}}_{i-B+(\alpha-1)-k-1}$ which are recovered in Ind. 1 by step number $k+1-(\alpha-1) < k+1$. Since the remaining erased symbols span the interval $[i-\alpha B, i-B)$ the parity checks $\{\mathbf{p}^{\mathrm{II}}[\cdot]\}_{t \ge i-B}$ recovered in step (3) can be used to recover these erased symbols.

It only remains to show that the non-urgent symbols in the diagonal $\mathbf{b}^{\mathrm{II}}$ are all recovered before time $\mathcal{T}$. From Proposition. 1 all the non-urgent sub-symbols are recovered using the first $(\alpha-1)(T-B)$ columns of the parity checks $\{\mathbf{p}^{\mathrm{II}}[\cdot]\}_{t \ge i-B}$. Since these parity checks are shifted by $T+B$, the fall in the interval $i+T, \ldots, i+T+(\alpha-1)(T-B)-1 = \mathcal{T}-1$. Thus only the parity checks before time $\mathcal{T}$ are required to recover the non-urgent source sub-symbols.

This completes the claim in the Ind. 1 and Ind. 2. We finally show that all the non-urgent erased source sub-symbols are recovered at $k = T-B-1$. Because of the recovery along the diagonals, 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 from 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., $s_{T-B-1}[i-B-1]$. This too immediately follows by applying the definition of $\mathbf{b}^{\mathrm{II}}_i$ at time $i-B+(T-B-1)(\alpha-1)-1$ as,

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

(6) Recover urgent sub-symbols:
Finally, the decoder recovers urgent sub-symbols $\mathbf{s}^U[\tau] = (s_{T-B}[\tau], \ldots, s_{T-1}[\tau])$ for $i-\alpha B \le \tau < i$ at time $t = \tau + T_2^\star$ using the parity check symbols $\mathbf{p}^{\mathrm{II}}[t]$ and the previously decoded non-urgent sub-symbols. We establish this claim as follows. After recovering all the non-urgent source sub-symbols $\{\mathbf{s}^N[\tau]\}_{\tau=i-\alpha B}^{i-1}$, we can directly apply the construction of $\mathcal{C}_2$ to recover the urgent sub-symbols $\{\mathbf{s}^U[\tau]\}_{\tau=i-\alpha B}^{i-1}$ using parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ within a delay of $T_2^\star$.

Note on Computational Complexity: We note that a DE-SCo encoder and decoder are of a polynomial complexity as the DE-SCo constructions are built upon a linear convolutional code with finite memory. Specifically, going through the steps (1)-(4) of the DE-SCo decoder, we can conclude that since every erased sub-symbol is processed at-most once, the complexity of any step is no more than αBT. In step (5) we use a recursive decoder that terminates in $T+B-1$ recursions. Also each step has at-most αBT and thus the complexity is polynomial in α, B and T.

VII. General Values of α

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 α 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 $(s_0[i],\dots,s_{T_0-1}[i])$ and producing B_0 parity check sub-symbols $\mathbf{p}^{\mathrm{I}}=(p_0^{\mathrm{I}}[i],\dots,p_{B_0-1}^{\mathrm{I}}[i])$ at each time by combining the source sub-symbols along the main diagonal with an interleaving step of size b i.e.,
$$p_k^{\mathrm{I}}[i]=\mathcal{A}_k(s_0[i-bT_0-kb],\dots,s_{T_0-1}[i-b-kb]) \quad (30)$$
- Let $\mathcal{C}_2$ be a $((\alpha-1)B,(\alpha-1)T)=((a-b)B_0,(a-b)T_0)$ SCo also obtained by splitting the source symbols into T_0 sub-symbols $(s_0[i],\dots,s_{T_0-1}[i])$ and then constructing a total of B_0 parity checks $\mathbf{p}^{\mathrm{II}}=(p_0^{\mathrm{II}}[i],\dots,p_{B_0-1}^{\mathrm{II}}[i])$ by combining the source sub-symbols along the opposite diagonal and with an interleaving step of size $\ell=(a-b)$ i.e.,
$$p_k^{\mathrm{II}}[i]=\mathcal{B}_k(s_0[i-\ell-k\ell],\dots,s_{T_0-1}[i-\ell T_0-k\ell]). \quad (31)$$
- 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 steps is analogous to the case when α is integer. We sketch the main steps. As before the decoding is done along the diagonal vectors $\mathbf{b}_i^{\mathrm{I}}=(s_0[i],\dots,s_{T_0-1}[i+(T_0-1)b])$, $\mathbf{b}_i^{\mathrm{II}}=(s_0[i],\dots,s_{T_0-1}[i-(T_0-1)\ell])$.

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

- **Ind. 1** Recover the non-urgent sub-symbols in $\mathbf{b}_{i-B-(k-1)b-1}^{\mathrm{I}},\dots,\mathbf{b}_{i-B-kb}^{\mathrm{I}}$ 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 $\mathbf{b}_{i-B+(k-1)(a-b)}^{\mathrm{II}},\dots,\mathbf{b}_{i-B+k(a-b)-1}^{\mathrm{II}}$ 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 is a multiple of b (i.e. αT is an integer) can be relaxed through a source pseudo-expansion approach as follows:

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

Notice that since the channel introduces a total of B_i erasures on the original input there will be 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 T_1 and $\lceil T_2\rceil$ on the original stream for user 1 and 2 respectively.

VIII. Numerical Results

To examine fundamental performance, we compare between the proposed DE-SCo codes and sequential random linear codes (RLC) numerically and discuss advantages and disadvantages of the proposed codes. The encoder for DE-SCo codes is the one discussed in section VI-B. For RLC, at each time step t a new source symbol $s[t]$ over an alphabet $\mathcal{S}$ is revealed to the transmitter and encoded into a channel symbol $x[t]$ through a random mapping $f_t(.)$ as follows,
$$x[t]=f_t(s[0],\dots,s[t]), \quad (32)$$
i.e., $f_t:\mathcal{S}^t\to\mathcal{X}$.

In our simulations, we do not construct an explicit function $f_t(\cdot)$ but instead assume that the decoder succeeds with high probability whenever the *instantaneous information debt* becomes non-positive. Intuitively, the information debt is a running sum of the gap between information transmitted over the channel and the information acquired by the receiver. We refer the reader to [1, Chapter 9], [19] for details. Our simulated decoder keeps track of the erasure pattern and retrieves the current segment of source symbols as soon as the information debt is non-positive. While every symbol in this setup is ultimately decoded, any symbols that incur a delay that exceeds the maximum delay, are declared to be lost.

In our simulations we divide the coded data stream into segments of 2000 symbols each and generate one burst erasure in each segment. Each symbol occupies one millisecond. The

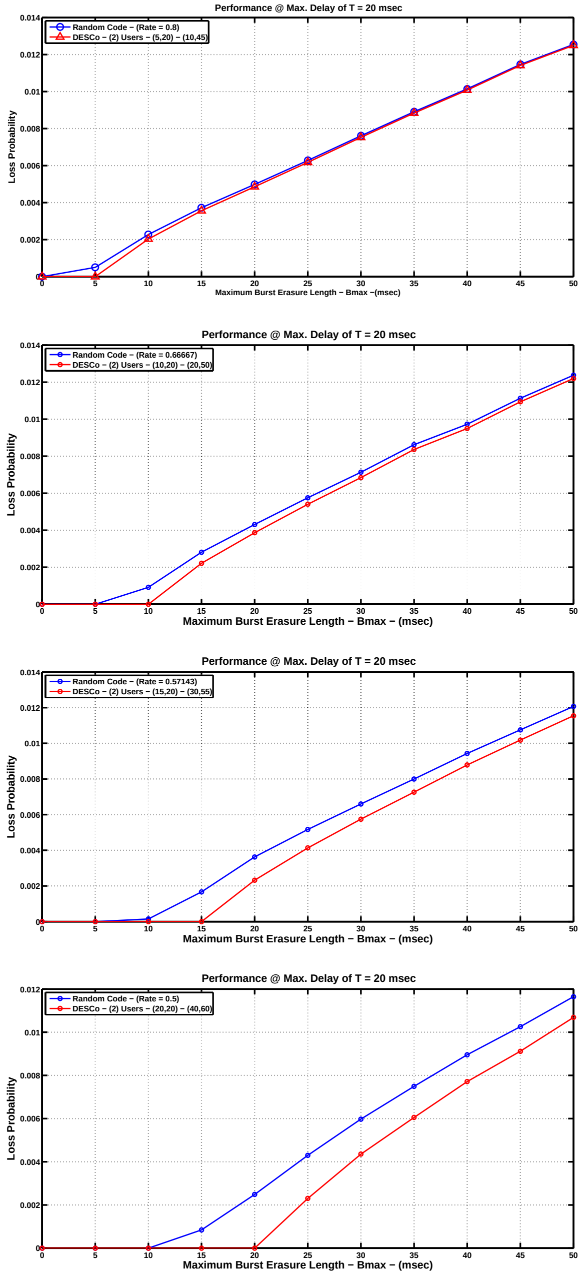


Fig. 6. Loss Probability at the first receiver.

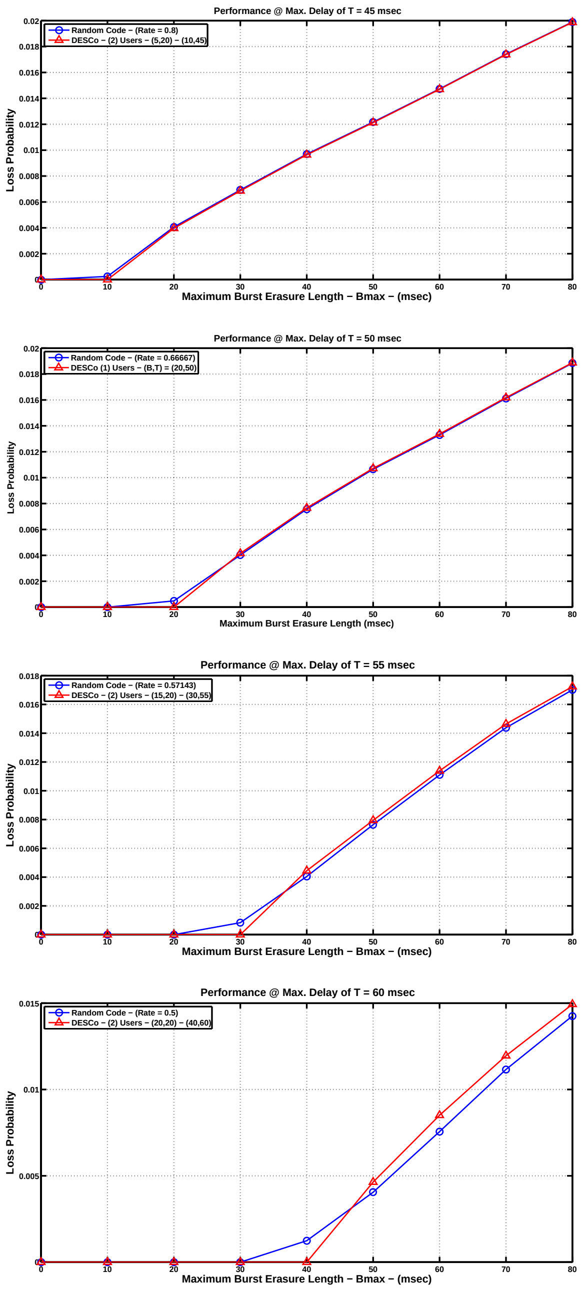


Fig. 7. Loss Probability at the second receiver.

burst erasure length is uniformly distributed between $[0, B_{max}]$ symbols and a symbol 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 RLC of the same rate for the two users in Fig. 6 and Fig. 7 respectively as a function of the maximum erasure burst length.

We make a few remarks on the numerical 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 RLC at rate R it can be easily verified that if the burst-length exceeds $\lceil (1-R)T \rceil$, the first symbol will not be decoded with a delay of T.

Next we see that DE-SCo always outperforms RLC for user 1. This can be explained as follows. A rate R DE-SCo can recover completely from an erasure burst of length B_1 or smaller for user 1. It fails to recover the erased symbols if the burst length exceeds B_1. The RLC only recovers completely from an erasure burst of length $\lceil (1-R)T_i \rceil$. It provides partial recovery for burst erasures up to length B_1 and fails to recover any source symbols when the erasure length exceeds B_1. Thus the performance of DE-SCo always dominates RLC for user 1 as illustrated in Fig. 6.

For user 2, the delay is given by $T_2 = \frac{B_2}{B_1} T_1 + B_1$. DE-SCo can correct all erasures up to length B_2 and fail to recover any symbols if the erasure length is beyond B_2. While threshold for perfect recovery for RLC is $\lceil (1-R)T_i \rceil \leq B_2$, interestingly it allows for partial recovery for burst lengths up to $\beta = B_2 + \frac{B_1^2}{T_1}$. This threshold is obtained as follows. Suppose an erasure of length β occurs at time $t = 0, 1, \ldots, \beta - 1$. The total information debt at this point is βR. If the information debt becomes non-positive after ν subsequent non-erased channel symbols then we must have that $\beta R \leq \nu(1-R)$. Substituting $R = \frac{T_1}{T_1+B_1}$ and $\nu = T_2 = \frac{B_2}{B_1}T_1 + B_1$, which is the maximum allowable delay for user 2, we recover the desired threshold. Since $\beta > B_2$, there is a range of erasure burst lengths where the RLC code can recover a partial subset of source sub-symbols whereas DE-SCo fails to recover any source sub-symbols. 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 way such 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] H. Yao, Y. Kochman and G. W. Wornell, “On Delay in Real-Time Streaming Communication Systems,” Allerton 2010

[8] A. ParandehGheibi, M. Mdard, S. Shakkottai, A. Ozdaglar, “Avoiding Interruptions - QoE Trade-offs in Block-coded Streaming Media Applications, submitted to ISIT, 2010.

[9] Z. Li, A. Khisti and B. Girod, “Forward error correction for low-delay packet video”, to appear in Packet Video Workshop, Dec. 2010.

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

[11] 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):841 - 851, 2004.

[12] G. D. Forney, Jr., “Convolutional Codes I: Algebraic Structure,” IEEE Transactions on Information Theory, vol. IT-16, pp. 720 - 738, November 1970.

[13] S. K. Filipovic, P. Spasojevic, E. Soljanin, “Doped Fountain Coding for Minimum Delay Data Collection in Circular Networks,” IEEE Journal on Selected Areas in Communications, June 2009, Volume: 27, Issue: 5, pp. 673 - 684.

[14] S. Filipovic, P. Spasojevic, R. D. Yates, E. Soljanin, “Decentralized fountain codes for minimum-delay data collection,” in Conference on Information Sciences and Systems, Princeton, NJ, July 2008.

[15] M. Bogino, P. Cataldi, M. Grangetto, E. Magli, E and G. Olmo “Sliding-window digital fountain codes for streaming of multimedia contents,” In IEEE International Symposium on Circuits and Systems, 2007

[16] T. Tirronen, and J. Virtamo, J. “Finding fountain codes for real-time data by fixed point method,” In Proceedings of International Symposium on Information Theory and its applications, ISITA 2008.

[17] D. Wu, Y. Hoy, W. Zhu, Y. Zhang, and J. Peha, “Streaming video over the internet: approaches and directions,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, pp. 282300, 2001

[18] T. Tirronen and J. Virtamo, “Fountain-inspired erasure coding for real-time traffic Performance analysis and simulation,” Telecommunication Systems, Springer Netherlands, pp. 1-14, 2010, url = http://dx.doi.org/10.1007/s11235-010-9330-2,

[19] E. Martinian and G. W. Wornell, “Universal Codes for Minimizing Per-User Delay on Streaming Broadcast Channels,” In Proc. Allerton Conf. Commun., Contr., and Computing, 2003.

Ahmed Badr Ahmed Badr received a B.Sc. degree in Electrical Engineering from Cairo University, Egypt in 2007 and M.Sc. degree in Electrical Engineering from Nile University, Egypt in 2009. From September 2007 to August 2009, he was a Research Assistant in the Wireless Intelligent Networks Center (WINC), Nile University. In September 2009, he assumed his current position as a Research Assistant at Signals Multimedia and Security Laboratory in University of Toronto while pursuing his Ph.D. degree in Electrical Engineering. His research interests include information theory, coding theory and wireless communications.

Ashish Khisti Ashish Khisti is an assistant professor in the Electrical and Computer Engineering (ECE) department at the University of Toronto, Toronto, Ontario Canada. He received his BASc degree in Engineering Sciences from University of Toronto and his S.M and Ph.D. Degrees from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. His research interests span the areas of information theory, wireless physical layer security and streaming in multimedia communication systems. At the University of Toronto, he heads the signals, multimedia and security laboratory. For his graduate studies he was a recipient of the NSERC postgraduate fellowship, HP/MIT alliance fellowship, Harold H. Hazen Teaching award and the Morris Joseph Levin Masterworks award.

Emin Martinian Emin Martinian earned a B.S. from UC Berkeley in 1997, and an S.M and Ph.D. from MIT in 2000 and 2004 (all in electrical engineering and computer science). Emin has been awarded a National Science Foundation Fellowship, the Capocelli Award for Best Paper, second place for best Computer Science Ph.D. at MIT, and over 10 patents. Emin worked at various technology startups (OPC Technologies, PinPoint) and research labs (Bell Labs, Mitsubishi Electric Research Labs) before joining Bain Capital in 2006. He currently works on research and strategy development at Bain’s global macro hedge fund.