Streaming-Codes for Multicast over Burst Erasure Channels

Ahmed Badr, Devin Lui and Ashish Khisti

School of Electrical and Computer Engineering

University of Toronto

Toronto, ON, M5S 3G4, Canada

Email: {abadr, dlui, akhisti}@comm.utoronto.ca

Abstract

We study the capacity limits of real-time streaming over burst-erasure channels. A stream of source packets must be sequentially encoded and the resulting channel packets must be transmitted over a two-receiver burst-erasure broadcast channel. The source packets must be sequentially reconstructed at each receiver with a possibly different reconstruction deadline. We study the associated capacity as a function of burst-lengths and delays at the two receivers.

We establish that the operation of the system can be divided into two main regimes: a *low-delay regime* and a *large-delay regime*. We fully characterize the capacity in the large delay regime. The key to this characterization is an inherent *slackness* in the delay of one of the receivers. At every point in this regime we can reduce the delay of at-least one of the users until a certain critical value and thus it suffices to obtain code constructions for certain critical delays. We partially characterize the capacity in the low-delay regime. Our capacity results involve code constructions and converse techniques that appear to be novel. We also provide a rigorous information theoretic converse theorem in the point-to-point setting which was studied by Martinian in an earlier work.

Index Terms

Streaming Communication Systems, Broadcast Channels with Common Message, Delay Constrained Communication, Application Layer Error Correction, Burst Erasure Channels.

I. INTRODUCTION

Delay is often ignored in the analysis of classical communication systems. Traditional error correction codes are designed to operate on message blocks, and can incur arbitrarily long encoding and decoding delays. In contrast several emerging applications are highly delay-sensitive. Both the fundamental limits and error correction techniques in such systems can be very different, see e.g., [1]–[9], and references therein.

An information theoretic framework for the study of low-delay streaming codes has been introduced in [10]–[12]. The encoder observes a stream of source packets, and sequentially encodes it into a a stream of channel packets. The decoder is required to reconstruct each source packet with a maximum delay of T units. The proposed channel is a burst erasure channel. It can erase up to B packets in a single burst, but otherwise transmits each packet instantaneously. The maximum possible rate $C_0(B, T)$ is characterized by proposing a coding scheme and proving a converse. We refer to this class of codes as streaming codes (SCo) throughout this paper.

From a practical point of view, the (B, T) SCo code should be used over a burst-erasure channel, where the maximum length of any single burst is B and the guard interval separating multiple bursts is at-least T. Extensions of SCo codes thats correct both burst and isolated erasures have been recently developed [13], [14]. Such codes were demonstrated to exhibit significant performance gains over the Gilbert-Elliott

$\{s[t]\}_{t\geq o}$ → Encoder → $\{x[t]\}_{t\geq o}$ $x[t]=f_i(s[0],\ldots,s[t])$

Burst = B_1 $\{y_1[t]\}_{t\geq o}$ **Delay = T_1** Decoder 1 $\hat{s}[t]$

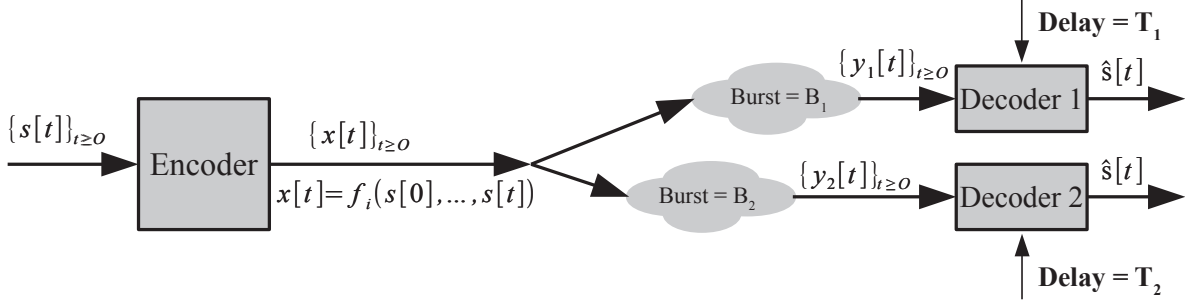Burst = B_2 $\{y_2[t]\}_{t\geq o}$ Decoder 2 $\hat{s}[t]$ **Delay = T_2**

Fig. 1. The source stream $\{s[i]\}$ is causally mapped into an output stream $\{x[i]\}$. 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 $i = 1, 2$.

channel and Fritchman channel models, thus opening up the exciting possibility of developing structured codes for delay-constrained streaming communication in practical wireless networks.

In this paper we are interested in a different extension of the SCo constructions. Instead of committing to a particular burst length B and delay T, our constructions adapt to the burst-length introduced by the channel. When the channel introduces an erasure-burst of length up to B_1, the reconstruction-delay must be no greater than T_1, whereas if the burst-length is larger, say B_2, the reconstruction delay can be increased to T_2. Such constructions can be relevant for error concealment techniques such as adaptive media playback [15]. Such methods adjust the play-out rate as a function of the receiver buffer size, so that a temporary increase in delay can be naturally accommodated. A natural way to study such constructions is to consider a multicast setup involving one sender and two receivers. The two receivers are connected to the sender over a burst-erasure broadcast channel and both the receivers are interested in reconstructing the same source stream, but with different delays. One receiver's channel introduces a burst of length B_1 and the required reconstruction delay is T_1. The second receiver's channel introduces a burst of length B_2 and the associated reconstruction delay is T_2. We seek to characterize the multicast streaming capacity $C(B_1, T_1, B_2, T_2)$ in this paper.

In an earlier work [16], we investigate the necessary and sufficient conditions under which $C(B_1, T_1, B_2, T_2) = C_0(B_1, T_1)$ (with $B_2 > B_1$). In particular, we show that if the delay T_2 of the weaker user is larger than a certain threshold, then the multicast capacity reduces to the single-user capacity of the stronger receiver. A particular code construction — *diversity embedded streaming erasure codes* (DE-SCo) — is proposed to achieve this capacity. In the present paper we obtain several new results. First, we observe that system performance can be divided into two operating regimes. When both the delays T_1 and T_2 are smaller than certain thresholds the system operates in a low-delay regime. Otherwise it operates in a large-delay regime. In the latter case we identify a surprising *slackness* property and use it in our code constructions. The slackness property enables us to reduce the delay of either receiver 1 or receiver 2 to a certain minimum threshold without reducing the capacity. In the low-delay regime the characterization of the capacity is more challenging. We characterize the capacity for a subset of this region by proposing a new coding scheme and a matching converse. For the remainder of this region we propose an upper bound on the capacity, but leave open whether this bound is the true capacity. Preliminary results of our work appeared in [17]. For related work see e.g., [18]–[27] and references therein.

II. System Model

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

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

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

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

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

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

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

The source stream is an i.i.d. process; each source symbol is sampled from a distribution $p_{\mathbf{s}}(\cdot)$ over some finite alphabet $\mathcal{S}$. The channel symbols $\mathbf{x}[t]$ belong to some alphabet $\mathcal{X}$. The rate of the multicast code is defined as the ratio of the (marginal) entropy of the source symbol to the alphabet size i.e., $R = H(\mathbf{s})/\log_2 |\mathcal{X}|$ and the multicast streaming capacity, $C(B_1, T_1, B_2, T_2)$ is the maximum achievable rate. An *optimal multicast streaming erasure code (Mu-SCo)* achieves such capacity for a given choice of (B_1, T_1, B_2, T_2). Without loss of generality, we assume throughout the paper that $B_2 \geq B_1$.

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. More general channel models can be transformed into an erasure model by applying an appropriate inner code [10, Chapter 7].

III. MAIN RESULTS

To keep the paper self contained, we first briefly review the single user scenario [10]–[12]. We point the reader to these references as well as a summary in [16] for a more exhaustive treatment.

A. Single-User Capacity

Theorem 1 (Point-to-Point Capacity: [10]): The capacity of a point-to-point system described by (1), (2) and (3) (with $i = 1$) is

$$C = \begin{cases} \frac{T}{T+B} & T \geq B \\ 0 & T < B, \end{cases} \tag{4}$$

where T_1 and B_1 are replaced by T and B for simplicity.
The associated code construction involves a two step approach.

- Construct a low-delay burst-erasure block code (LD-BEBC) that takes T source symbols, say $(s_0, \ldots, s_{T-1})$ and generates B parity checks, say $(p_0, \ldots, p_{B-1})$. The resulting codeword $\mathbf{x} = (s_0, \ldots, s_{T-1}, p_0, \ldots, p_{B-1})$ has the property that it can fully recover all erased symbols from any erasure burst of length B. Furthermore each of the erased source symbols s_i for $i \in \{0, \ldots, T-1\}$ is recovered by time $\min\{i + T, T + B\}$. An explicit construction of such a code is proposed in [10]–[12].
- Apply diagonal-interleaving to the LD-BEBC code to construct the streaming code.

The resulting streaming code is a time-invariant, systematic convolutional code of memory T, that takes in T source symbols at any given time and outputs $T + B$ symbols[1]. The converse is based on a periodic erasure channel argument, similar to the upper bounding technique used in classical burst-noise channels [29, Section 6.10]. The basic idea is to amplify the effect of a single isolated erasure burst into a periodic erasure channel and use the capacity of such a channel as an upper bound. We compliment this argument with a rigorous information theoretic proof for (4) in Section IV. The information theoretic proof is more general and provides a tighter upper bound when we consider the multicast setup.

---

[1]In this work will not be using any special properties of convolutional codes [28] and the reader is not assumed to have familiarity with this topic. Some properties of SCo codes from the context of convolutional codes are discussed in [11], [13].
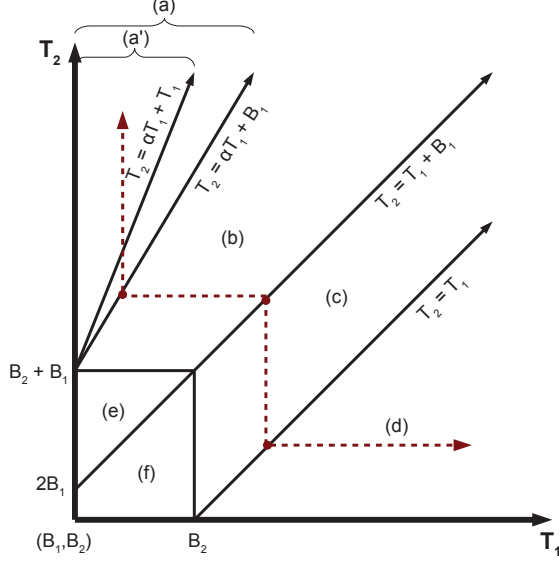
Fig. 2. Capacity behavior in the (T_1, T_2) plane. We hold B_1 and B_2 as constants with $(B_2 > B_1)$, so the regions depend on the relation between T_1 and T_2 only. The red dashed line shows the contour of constant capacity in regions (a), (b), (c) and (d).
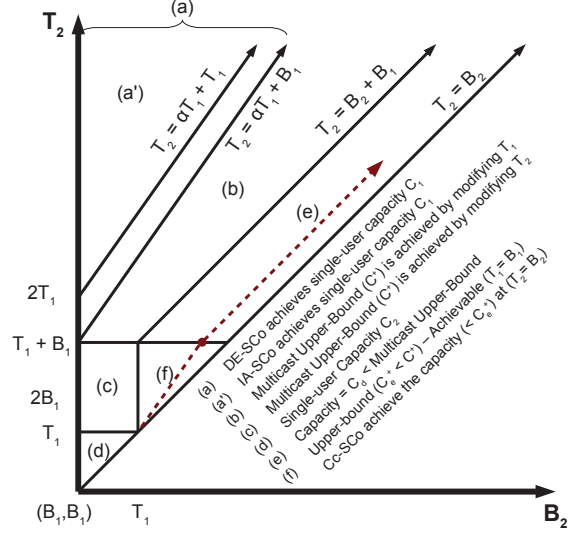
Fig. 3. Capacity behaviour in the (B_2, T_2) plane. We hold B_1 and T_1 as constants, so the regions depend on the relation between T_2 and B_2 only. The dashed line gives the contour of constant capacity in region (e) as well as in the special case of $T_1 = B_1$ in region (f).

B. DE-SCo Construction

In earlier work [16] Badr et. al consider the proposed multicast setup when the delay of the weaker user i.e., user 2 is sufficiently large.

Theorem 2 (Badr et. al [16]): The multicast streaming capacity $C(B_1, T_1, B_2, T_2)$ in the regime where $B_2 > B_1$ and $T_2 \geq \alpha T_1 + B_1$ (with $\alpha = \frac{B_2}{B_1}$) is given by:

$$C_1 = \frac{T_1}{T_1 + B_1}. \tag{5}$$

The associated code construction — Diversity Embedded Streaming Codes (DE-SCo) — involves constructing two groups of parity checks: one along the main diagonal and the other along the off-diagonal and then combining these parity checks in a suitable manner. We refer the reader to [16] for the detailed construction. A converse argument is also provided in [16] to establish that T_2 is indeed the smallest possible threshold to achieve the rate of C_1.

C. Large Delay Regime

The parameters of the DE-SCo construction in Theorem 2 fall within a larger class which we refer to as the *large-delay regime*. In particular if at-least one of T_1 and T_2 is larger than a certain threshold:

$$T_1 \geq B_2, \quad (or) \quad T_2 \geq B_1 + B_2. \tag{6}$$

we have been able to determine the multicast capacity. In Fig. 2 this regime consists of all pairs (T_1, T_2) outside the rectangular box $[B_1, B_2] \times [B_2, B_1 + B_2]$.

Theorem 3: When the delays T_1 and T_2 satisfy (6) and $B_2 > B_1$ the multicast capacity is given by

$$C = \begin{cases} C_1, & T_2 \geq \alpha T_1 + B_1, \\ \frac{T_2 - B_1}{T_2 - B_1 + B_2}, & T_1 + B_1 \leq T_2 \leq \alpha T_1 + B_1, \\ \frac{T_1}{T_1 + B_2}, & T_1 \leq T_2 \leq T_1 + B_1, \\ C_2, & T_2 \leq T_1. \end{cases} \tag{7}$$

where $C_i = \frac{T_i}{T_i + B_i}$ is the single user capacity of user $i = 1, 2$ and we have defined $\alpha = \frac{B_2}{B_1}$. $\quad\square$

The proof of Theorem 3 appears in section V.

Remark 1 (Delay-Slackness Property): A closer look at (7) shows that in each of the four cases the capacity only depends on either T_1 or T_2, but not on both of them simultaneously. In particular as shown in Fig. 2 the contour of constant capacity is a piecewise constant line. On the horizontal portions, the delay T_1 can be reduced without reducing the capacity whereas on the vertical portions the delay T_2 can be reduced without reducing the capacity. This *slackness* in the delay of the receivers is rather unexpected and one of the surprises in this work.

We next comment of the key ingredients in the proof of Theorem 3. The converse is obtained by combining the following upper bound with single user capacity bounds.

Theorem 4: **(Upper-Bounds via Periodic Erasure Channel)** For any two receivers with burst-delay parameters of (B_1, T_1) and (B_2, T_2), the multicast streaming capacity is upper-bounded by $C \leq C^+$, where

$$C^+ = \begin{cases} \frac{T_2 - B_1}{T_2 - B_1 + B_2} & T_2 > T_1 + B_1, \\ \frac{T_1}{T_1 + B_2} & T_2 \leq T_1 + B_1, \end{cases} \tag{8}$$

The proof of Theorem 4 is provided in section VII. It involves simultaneously using the decoding constraints of both the receivers to obtain a tighter upper bound than a simple point-to-point bound.

We next discuss the achievability part of Theorem 3. The first case in (7), i.e., when $T_2 \geq \alpha T_1 + B_1$ coincides with the condition of DE-SCo codes in Theorem 2 and thus (5) applies. The code construction associated with this region appears in [16]. The construction of the remainder of the cases in (7) exploits the delay-slackness property. The second case corresponds to region (b) in Fig. 2, where user 1 experiences slackness in its delay. We can reduce T_1 so that we just hit the boundary of region (a) and then use the DE-SCo code construction. In contrast, region (c) in Fig. 2 which corresponds to the third case in (7) involves slackness in the delay of user 2. We can reduce T_2 till we just hit the boundary of region (d). For region (d) it can be easily seen that a single-user code for user 2 is optimal. The details of the above reductions are presented in section V.

For a subset of region (a), where DE-SCo codes are optimal, we also propose a simpler construction — Interference Avoidance Streaming Codes (IA-SCo) that only requires us to construct two single user codes and combine the associated parity checks to avoid mutual interference.

Proposition 1: **(Interference-Avoidance SCo)** An IA-SCo construction achieves a rate of $C_1 = \frac{T_1}{T_1 + B_1}$ when $B_2 = \alpha B_1$ (with $\alpha > 1$ an integer) and

$$T_2 \geq \alpha T_1 + T_1. \tag{9}$$

The region associated with (9) is marked by (a') in Fig. 2.

The proposed scheme involves starting with single user streaming codes $\mathcal{C}_1 : (B_1, T_1)$ and $\mathcal{C}_2 : (B_2, T_2)$, delaying the parity checks of $\mathcal{C}_2$ by T_1 units and then directly combining them with the parity checks of $\mathcal{C}_1$ such that they do not interfere with one another. The complete IA-SCo construction is provided in Section. VI.

D. Low Delay Regime

We next consider the case when the delay pair (T_1, T_2) falls in the box $[B_1, B_2] \times [B_2, B_1 + B_2]$ i.e.,

$$B_1 \leq T_1 \leq B_2, \quad (and) \quad B_2 \leq T_2 \leq B_1 + B_2. \tag{10}$$

This regime appears to be more challenging and the capacity has only been established in some special cases.

Theorem 5: **(Capacity in Region (e))** The multicast streaming capacity in region (e) defined by $T_1 + B_1 \leq T_2 \leq B_2 + B_1$ and $B_1 \leq T_1 < B_2$ is given by,

$$C_e = \frac{T_1}{2T_1 + B_1 + B_2 - T_2}. \tag{11}$$

Note that the capacity expression C_e only depends on B_2 and T_2 via the difference $B_2 - T_2$. To identify the contour of constant capacity in the (e) region it is natural to fix B_1 and T_1 and classify the various regions as shown in Fig. 3. Observe that the streaming capacity for any point in region (e) is constant across the 45-degrees line and is equal to the multicast upper-bound at the lowest point on the line separating regions (e) and (f) in Fig. 3.

The complete proof for Theorem 5 is divided into two main parts. The achievability scheme is provided in Section VIII while the converse is given in Section IX. The achievability involves first constructing a single user (B_1, T_1) SCo code for the first user and then carefully embedding additional parity checks to satisfy the decoding constraint of user 2. The converse too involves a new insight of revealing some of the source symbols to a virtual decoder to obtain a tighter bound than a periodic erasure channel argument.

The remainder of the low delay regime is called region (f). The capacity remains open except in the special cases of either $T_1 = B_1$ or $T_2 = B_2$.

Theorem 6: **(Upper-bound in Region (f))** An upper-bound on the multicast streaming capacity in region (f) defined by $T_2 < T_1 + B_1$ and $T_1 \in [B_1, B_2]$ is given by,

$$C_f \leq C_f^+ = \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)}. \tag{12}$$

The above expression equals the streaming capacity if $T_1 = B_1$.

The proof of the upper bound is given in Section XI. The code construction for $T_1 = B_1$ case appears in Section. X.

The capacity has also been obtained when $T_2 = B_2$ for any $T_1 \in [B_1, B_2]$.

Theorem 7: **(Capacity in Region (f) at $(T_2 = B_2)$)** The multicast streaming capacity in region (f) defined by $T_2 < T_1 + B_1$ and $T_1 \in [B_1, B_2]$ at the minimum delay case for user 2 $(T_2 = B_2)$ is given by,

$$C_{f(T_2=B_2)} = \frac{T_1}{2T_1 + B_1}. \tag{13}$$

The achievability scheme is based on concatenation of the parity checks of suitably constructed single-user codes and appears in Section XII. The proof of the converse part for Theorem 7 is provided in Section XIII. The technique is significantly different than earlier converses and involves carefully double-counting the redundancy arising from the recovery of certain source symbols.

This concludes the main results of the paper.

IV. CONVERSE PROOF OF THEOREM 1

In this section we provide an information theoretic converse to Theorem 1. While the capacity of the point-to-point case was established in [10]–[12], the converse argument was based on a somewhat informal use a periodic erasure channel (PEC). Our information theoretic approach is not only more rigorous but also generalizes to the multicast setting in subsequent sections. Furthermore it has the following advantages over the PEC approach which might also be of interest.

1) The PEC approach requires the channel packet $\mathbf{x}[t]$ is a deterministic function of the past source packets i.e., $\mathbf{x}[t]$ must be exactly computed given $\mathbf{s}[0], \ldots, \mathbf{s}[t]$. The information theoretic converse does not impose this restriction and allows for e.g., stochastic encoders.

2) The PEC approach as presented in [10] requires the code to be systematic. The information theoretic approach does not impose this restriction.

3) The PEC approach requires zero error in the recovery of each source symbol. The information theoretic approach can remove this restriction by suitably invoking Fano's inequality.

Let us use the following notation:

$$
\mathbf{s}\begin{bmatrix} b \\ a \end{bmatrix} = \begin{cases} \mathbf{s}[a], \mathbf{s}[a+1], \ldots, \mathbf{s}[b-1], \mathbf{s}[b], & a \leq b \\ \emptyset, & \text{otherwise} \end{cases} \tag{14}
$$

$$
W_a^b = \begin{cases} W_a, W_{a+1}, \ldots, W_{b-1}, W_b, & a \leq b \\ \emptyset, & \text{otherwise} \end{cases} \tag{15}
$$

To aid us in our proof, let us introduce the terms

$$
V_i = \mathbf{s}\begin{bmatrix} (i+1)(T+B)-1 \\ i(T+B) \end{bmatrix}, \qquad W_i = \mathbf{x}\begin{bmatrix} (i+1)(T+B)-1 \\ i(T+B)+B \end{bmatrix} \tag{16}
$$

where $i = 0, 1, 2, \ldots$. Note that V_i refers to a group of source packets, whereas W_i is a group of channel packets. Fig. 4 shows the time slots that the packets come from while Fig. 5 shows the size of V_i and W_i.
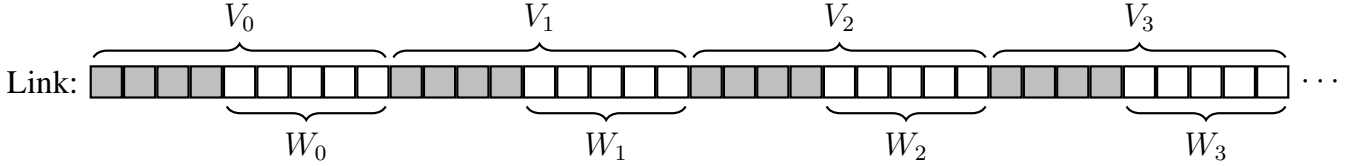

Fig. 4. The periodic erasure channel used in proving the upper-bound of the single user scenario in Theorem 1, but with indication of which packets are in the groups V_i and W_i. Grey and white squares resemble erased and unerased symbols respectively.
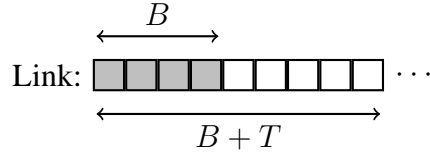

Fig. 5. One period of the periodic erasure channel in Fig. 4, with labels.

We start with the following equations, which are a result of the (B, T) code. If the first B channel packets are erased, and then the next T channel packets are received perfectly, the (B, T) code can be used to recover the source packets $\mathbf{s}[0], \ldots, \mathbf{s}[B-1]$. Using the conditional entropy notation, this can be written as:

$$
H\left(\mathbf{s}\begin{bmatrix} B-1 \\ 0 \end{bmatrix} \middle| W_0\right) = 0. \tag{17}
$$

Although the next T channel packets $\mathbf{x}\begin{bmatrix} B+T-1 \\ B \end{bmatrix}$ are received, we cannot assume that the corresponding source packets $\mathbf{s}\begin{bmatrix} B+T-1 \\ B \end{bmatrix}$ are able to be decoded because the code may not be systematic. To recover those source packets, we can use the next group of T unerased packets in $\mathbf{x}\begin{bmatrix} 2(B+T)-1 \\ (B+T)+B \end{bmatrix}$. In general, we may not need all of these channel packets, but the proof is simpler if we have it all available. We can then write the relation using Fano's Inequality[2]

$$
H\left(\mathbf{s}\begin{bmatrix} B+T-1 \\ B \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} B-1 \\ 0 \end{bmatrix} W_0^1\right) = 0. \tag{18}
$$

---

[2]The conditional entropy is zero because the receiver needs to perfectly recover each of the source packets with zero error. While we do not allow vanishingly small error probabilities at the decoder, the setup can be easily generalized in this case.

The equations (17) and (18) can be generalized to

$$H\left(\mathbf{s}\begin{bmatrix}i(B+T)+B-1\\i(B+T)\end{bmatrix}\Big|\mathbf{x}\begin{bmatrix}i(B+T)-1\\0\end{bmatrix}W_i\right) = 0 \tag{19}$$

$$H\left(\mathbf{s}\begin{bmatrix}(i+1)(B+T)-1\\i(B+T)+B\end{bmatrix}\Big|\mathbf{x}\begin{bmatrix}i(B+T)+B-1\\0\end{bmatrix}W_i^{i+1}\right) = 0 \tag{20}$$

Note that the above expressions still only assume that there was one burst erasure of length B. For instance, in (19), we assume that the packets $\mathbf{x}\begin{bmatrix}i(B+T)+B-1\\i(B+T)\end{bmatrix}$ were erased so they are not used in the expression.

Next, we will prove the following relation for $n \geq 0$:

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n\Big|V_0^{n-1}\mathbf{x}\begin{bmatrix}n(B+T)-1\\0\end{bmatrix}\right). \tag{21}$$

For the base case, substitute $n = 0$ into (21). This gives

$$H(W_0) \geq H(V_0^{-1}) + H\left(W_0\Big|V_0^{-1}\mathbf{x}\begin{bmatrix}-1\\0\end{bmatrix}\right) = H(W_0) \tag{22}$$

which is clearly true. We assume that (21) is true for $n = k$ in the induction step,

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k\Big|V_0^{k-1}\mathbf{x}\begin{bmatrix}k(B+T)-1\\0\end{bmatrix}\right). \tag{23}$$

With the availability of W_k, one can use (19) to recover $\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}$ and (23) can be re-written as follows,

$$H(W_0^k) \geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\mathbf{x}\begin{bmatrix}k(B+T)+B-1\\0\end{bmatrix}\right). \tag{24}$$

The detailed steps from (23) to (24) is shown in Appendix. A.

Next, we add $H(W_{k+1}|W_0^k)$ to both sides and then use (20) to recover the source symbols corresponding to W_k, $\mathbf{s}\begin{bmatrix}(k+1)(B+T)-1\\k(B+T)+B\end{bmatrix}$ and the following can be written (c.f. Appendix. A),

$$H(W_0^{k+1}) \geq H(V_0^k) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}(k+1)(B+T)-1\\0\end{bmatrix}\right), \tag{25}$$

From (23) to (25) and passing by (24), we have shown that if (21) is true for $n = k \geq 0$, then it is also true for $n = k + 1$. Thus, by induction (21) is true for $n \geq 0$. We take (21) and finalize it as

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n\Big|V_0^{n-1}\mathbf{x}\begin{bmatrix}n(B+T)-1\\0\end{bmatrix}\right) \geq H(V_0^{n-1}). \tag{26}$$

Next, we expand the groups of channel packets

$$H(W_0^n) = H\left(\mathbf{x}\begin{bmatrix}T+B-1\\B\end{bmatrix}\mathbf{x}\begin{bmatrix}2(T+B)-1\\(T+B)+B\end{bmatrix}\cdots\mathbf{x}\begin{bmatrix}(n+1)(T+B)-1\\n(T+B)+B\end{bmatrix}\right)$$

$$\leq \sum_{i=0}^{n}\sum_{j=B}^{T+B-1} H(\mathbf{x}[i(T+B)+j]) = (n+1)\cdot T\cdot H(\mathbf{x}) \tag{27}$$

and also expand the groups of source packets

$$H(V_0^{n-1}) = H\left(\mathbf{s}\begin{bmatrix}n(T+B)-1\\0\end{bmatrix}\right) = \sum_{i=0}^{n(T+B)-1} H\left(\mathbf{s}[i]\Big|\mathbf{s}\begin{bmatrix}i-1\\0\end{bmatrix}\right)$$

$$\overset{(a)}{=} \sum_{i=0}^{n(T+B)-1} H(\mathbf{s}[i]) = n\cdot(T+B)\cdot H(\mathbf{s}) \tag{28}$$

where step (a) is because the source packets are independent. Then we can take (26) and write it as

$$H(W_0^n) \geq H(V_0^{n-1})$$
$$(n+1) \cdot T \cdot H(\mathbf{x}) \geq n \cdot (T + B) \cdot H(\mathbf{s})$$
$$\frac{(n+1)}{n} \cdot \frac{T}{T+B} \geq \frac{H(\mathbf{s})}{H(\mathbf{x})}.$$

Finally, we conclude that any (B, T) streaming erasure code must satisfy

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T}{T+B} \quad (\text{as } n \to \infty) \tag{29}$$

which gives our upper bound of the rate.

V. PROOF OF THEOREM 3

For the converse we start with the upper bound in (8) in Theorem 4 which we reproduce below for convenience. The proof of Theorem 4 appears section VII.

$$C^+ = \begin{cases} \frac{T_2 - B_1}{T_2 - B_1 + B_2} & T_2 > T_1 + B_1, \\ \frac{T_1}{T_1 + B_2} & T_2 \leq T_1 + B_1. \end{cases} \tag{30}$$

We further tighten the upper-bound in (30) as follows,

$$C^U = \min\{C^+, C_1, C_2\} = \begin{cases} \min\left\{\frac{T_2 - B_1}{T_2 - B_1 + B_2}, C_1, C_2\right\}, & T_2 > T_1 + B_1 \\ \min\left\{\frac{T_1}{T_1 + B_2}, C_1, C_2\right\}, & T_2 \leq T_1 + B_1. \end{cases} \tag{31}$$

Through straightforward calculations one can further simplify:

$$C^U = \begin{cases} \min\left\{\frac{T_2 - B_1}{T_2 - B_1 + B_2}, C_1\right\}, & T_2 > T_1 + B_1 \\ \min\left\{\frac{T_1}{T_1 + B_2}, C_2\right\}, & T_2 \leq T_1 + B_1 \end{cases}$$
$$= \begin{cases} C_1 \triangleq C_a, & T_2 \geq \alpha T_1 + B_1 \\ \frac{T_2 - B_1}{T_2 - B_1 + B_2} \triangleq C_b, & T_1 + B_1 < T_2 < \alpha T_1 + B_1 \\ \frac{T_1}{T_1 + B_2} \triangleq C_c, & T_1 < T_2 \leq T_1 + B_1 \\ C_2 \triangleq C_d, & T_2 \leq T_1. \end{cases} \tag{32}$$

where recall that $\alpha = \frac{B_2}{B_1}$. This completes the proof of the converse.
We discuss the code constructions for each of these regions below.

A. Region (a)

The code-construction achieving C_1 in region (a) appeared in [16]. We summarize the key-steps for completeness and provide an example with $\{(B_1, T_1) - (B_2, T_2)\} = \{(2, 3) - (4, 8)\}$ in Table I which we will require in a subsequent example. We will assume for simplicity that $B_2 = \alpha B_1$ where α is an integer.

- Generate a (B_1, T_1) SCo code $(\mathbf{s}[i], \mathbf{p}[i])$ consisting of T_1 source sub-symbols and B_1 parity check sub-symbols. Recall that the parity check sub-symbols are generated by applying a low-delay burst-erasure block codes (LD-BEBC) across the main diagonal of the stream of source sub-symbols.
- Generate a $(\alpha B_1, \alpha T_1)$ SCo code $(\mathbf{s}[i], \mathbf{q}[i])$ consisting of $T_1 + B_1$ sub-symbols where the parity check symbols $\mathbf{q}[i]$ are generated by applying a LD-BEBC across the opposite diagonal of the stream of source sub-symbols and with a interleaving factor of $(\alpha - 1)$.

$[i-1]$	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$
$\boxed{s_0[i-1]}$	$s_0[i]$	$\boxed{s_0[i+1]}$	$s_0[i+2]$	$s_0[i+3]$	$s_0[i+4]$
$s_1[i-1]$	$\boxed{s_1[i]}$	$s_1[i+1]$	$s_1[i+2]$	$s_1[i+3]$	$s_1[i+4]$
$\boxed{s_2[i-1]}$	$s_2[i]$	$\boxed{s_2[i+1]}$	$s_2[i+2]$	$s_2[i+3]$	$s_2[i+4]$
$s_0[i-4]\oplus s_2[i-2]$	$s_0[i-3]\oplus s_2[i-1]$	$s_0[i-2]\oplus s_2[i]$	$\boxed{s_0[i-1]\oplus s_2[i+1]}$	$s_0[i]\oplus s_2[i+2]$	$s_0[i+1]\oplus s_2[i+3]$
$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$s_2[i-9]\oplus s_0[i-7]$	$s_2[i-8]\oplus s_0[i-6]$	$s_2[i-7]\oplus s_0[i-5]$	$s_2[i-6]\oplus s_0[i-4]$	$s_2[i-5]\oplus s_0[i-3]$	$s_2[i-4]\oplus s_0[i-2]$
$s_1[i-4]\oplus s_2[i-3]$	$s_1[i-3]\oplus s_2[i-2]$	$s_1[i-2]\oplus s_2[i-1]$	$s_1[i-1]\oplus s_2[i]$	$\boxed{s_1[i]\oplus s_2[i+1]}$	$s_1[i+1]\oplus s_2[i+2]$
$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$s_1[i-9]\oplus s_0[i-8]$	$s_1[i-8]\oplus s_0[i-7]$	$s_1[i-7]\oplus s_0[i-6]$	$s_1[i-6]\oplus s_0[i-5]$	$s_1[i-5]\oplus s_0[i-4]$	$s_1[i-4]\oplus s_0[i-3]$

$[i+5]$	$[i+6]$	$[i+7]$	$[i+8]$	$[i+9]$	$[i+10]$
$s_0[i+5]$	$s_0[i+6]$	$s_0[i+7]$	$s_0[i+8]$	$s_0[i+9]$	$s_0[i+10]$
$s_1[i+5]$	$s_1[i+6]$	$s_1[i+7]$	$s_1[i+8]$	$s_1[i+9]$	$s_1[i+10]$
$s_2[i+5]$	$s_2[i+6]$	$s_2[i+7]$	$s_2[i+8]$	$s_1[i+9]$	$s_1[i+10]$
$s_0[i+2]\oplus s_2[i+4]$	$s_0[i+3]\oplus s_2[i+5]$	$s_0[i+4]\oplus s_2[i+6]$	$s_0[i+5]\oplus s_2[i+7]$	$s_0[i+6]\oplus s_2[i+8]$	$s_0[i+7]\oplus s_2[i+9]$
$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$s_2[i-3]\oplus s_0[i-1]$	$s_2[i-2]\oplus s_0[i]$	$\boxed{s_2[i-1]\oplus s_0[i+1]}$	$s_2[i]\oplus s_0[i+2]$	$s_2[i+1]\oplus s_0[i+3]$	$s_2[i+2]\oplus s_0[i+4]$
$s_1[i+2]\oplus s_2[i+3]$	$s_1[i+3]\oplus s_2[i+4]$	$s_1[i+4]\oplus s_2[i+5]$	$s_1[i+5]\oplus s_2[i+6]$	$s_1[i+6]\oplus s_2[i+7]$	$s_1[i+7]\oplus s_2[i+8]$
$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$s_1[i-3]\oplus s_0[i-2]$	$s_1[i-2]\oplus s_0[i-1]$	$s_1[i-1]\oplus s_0[i]$	$\boxed{s_1[i]\oplus s_0[i+1]}$	$s_1[i+1]\oplus s_0[i+2]$	$s_1[i+2]\oplus s_0[i+3]$

TABLE I

RATE 3/5 DE-SCo CONSTRUCTION THAT SATISFY THE REGION (A) POINT DESCRIBED BY USER 1 WITH $(B_1, T_1) = (2, 3)$ AND USER 2 WITH $(B_2, T_2) = (2B_1, 2T_1 + B_1) = (4, 8)$.

- The transmitted packet at time i is given by $\mathbf{x}[i] = (\mathbf{s}[i], \mathbf{p}[i] + \mathbf{q}[i - T_1])$.

We omit the steps in decoding as they are rather involved and refer to [16].

B. Region (b)

In region (b) in Fig. 2 we show that the rate

$$C_b = \frac{T_2 - B_1}{T_2 - B_1 + B_2}, \qquad T_1 + B_1 \le T_2 \le \alpha T_1 + B_1,$$

is achievable.

Since the capacity does not depend on T_1, we can reduce the value of T_1 to $\tilde{T}_1$ such that we meet the left hand side with equality i.e., we select

$$T_2 = \alpha \tilde{T}_1 + B_1,$$

which in turn implies that

$$\tilde{T}_1 = \frac{B_1}{B_2}(T_2 - B_1). \tag{33}$$

Provided that $\tilde{T}_1 \ge B_1$ and furthermore $\tilde{T}_1$ is an integer we can use a $\{(B_1, \tilde{T}_1) - (B_2, T_2)\}$ DE-SCo code [16] to achieve $\frac{\tilde{T}_1}{\tilde{T}_1 + B_1} = C_b$ and hence for the original point in region (b). The former condition is equivalent to $T_2 \ge B_2 + B_1$ which naturally holds in region (b). If $\tilde{T}_1$ it is not an integer a suitable expansion of every source symbol is needed as discussed below.

[i−1]		[i]		[i+1]	
$s_0[i-1]$	$s_3[i-1]$	$s_0[i]$	$s_3[i]$	$s_0[i+1]$	$s_3[i+1]$
$s_1[i-1]$	$s_4[i-1]$	$s_1[i]$	$s_4[i]$	$s_1[i+1]$	$s_4[i+1]$
$s_2[i-1]$	$s_5[i-1]$	$s_2[i]$	$s_5[i]$	$s_2[i+1]$	$s_5[i+1]$
$s_3[i-3]\oplus s_5[i-2]\ \oplus\ s_2[i-5]\oplus s_0[i-4]$	$s_0[i-2]\oplus s_2[i-1]\ \oplus\ s_5[i-5]\oplus s_3[i-4]$	$s_3[i-2]\oplus s_5[i-1]\ \oplus\ s_2[i-4]\oplus s_0[i-3]$	$s_0[i-1]\oplus s_2[i]\ \oplus\ s_5[i-4]\oplus s_3[i-3]$	$s_3[i-1]\oplus s_5[i]\ \oplus\ s_2[i-3]\oplus s_0[i-2]$	$s_0[i]\oplus s_2[i+1]\ \oplus\ s_5[i-3]\oplus s_3[i-2]$
$s_4[i-3]\oplus s_2[i-2]\ \oplus\ s_1[i-5]\oplus s_3[i-5]$	$s_1[i-2]\oplus s_5[i-2]\ \oplus\ s_4[i-5]\oplus s_0[i-4]$	$s_4[i-2]\oplus s_2[i-1]\ \oplus\ s_1[i-4]\oplus s_3[i-4]$	$s_1[i-1]\oplus s_5[i-1]\ \oplus\ s_4[i-4]\oplus s_0[i-3]$	$s_4[i-1]\oplus s_2[i]\ \oplus\ s_1[i-3]\oplus s_3[i-3]$	$s_1[i]\oplus s_5[i]\ \oplus\ s_4[i-3]\oplus s_0[i-2]$

[i+2]		[i+3]		[i+4]	
$s_0[i+2]$	$s_3[i+2]$	$s_0[i+3]$	$s_3[i+3]$	$s_0[i+4]$	$s_3[i+4]$
$s_1[i+2]$	$s_4[i+2]$	$s_1[i+3]$	$s_4[i+3]$	$s_1[i+4]$	$s_4[i+4]$
$s_2[i+2]$	$s_5[i+2]$	$s_2[i+3]$	$s_5[i+3]$	$s_1[i+4]$	$s_5[i+4]$
$s_3[i]\oplus s_5[i+1]\ \oplus\ s_2[i-2]\oplus s_0[i-1]$	$s_0[i+1]\oplus s_2[i+2]\ \oplus\ s_5[i-2]\oplus s_3[i-1]$	$s_3[i+1]\oplus s_5[i+2]\ \oplus\ s_2[i-1]\oplus s_0[i]$	$s_0[i+2]\oplus s_2[i+3]\ \oplus\ s_5[i-1]\oplus s_3[i]$	$s_3[i+2]\oplus s_5[i+3]\ \oplus\ s_2[i]\oplus s_0[i+1]$	$s_0[i+3]\oplus s_2[i+4]\ \oplus\ s_5[i]\oplus s_3[i+1]$
$s_4[i]\oplus s_2[i+1]\ \oplus\ s_1[i-2]\oplus s_3[i-2]$	$s_1[i+1]\oplus s_5[i+1]\ \oplus\ s_4[i-2]\oplus s_0[i-1]$	$s_4[i+1]\oplus s_2[i+2]\ \oplus\ s_1[i-1]\oplus s_3[i-1]$	$s_1[i+2]\oplus s_5[i+2]\ \oplus\ s_4[i-1]\oplus s_0[i]$	$s_4[i+2]\oplus s_2[i+3]\ \oplus\ s_1[i]\oplus s_3[i]$	$s_1[i+3]\oplus s_5[i+3]\ \oplus\ s_4[i]\oplus s_0[i+1]$

TABLE II

RATE 3/5 MU-SCO CONSTRUCTION THAT SATISFY THE REGION (B) POINT DESCRIBED BY USER 1 WITH $(B_1, T_1) = (1, 2)$ AND USER 2 WITH $(B_2, T_2) = (2, 4)$.

- Split each source symbol into $n^2\tilde{T}_1$ sub-symbols $s_0[i], \ldots, s_{n^2\tilde{T}_1-1}[i]$ where n is the smallest integer such that $n\tilde{T}_1$ is an integer.
- Construct an expanded source sequence $\tilde{\mathbf{s}}[\cdot]$ such that $\tilde{\mathbf{s}}[ni+r] = (s_{rn\tilde{T}_1}[i], \ldots, s_{(r+1)n\tilde{T}_1-1}[i])$ where $r \in \{0, \ldots, n-1\}$.
- We apply a DESCo code with parameters $\{(nB, n\tilde{T}_1) - (n\alpha B, n(\alpha\tilde{T}_1 + B))\}$ to $\tilde{\mathbf{s}}[\cdot]$ using the earlier construction.

With the channel of user 2 introducing B_2 erasures on the original input stream, there will be nB_2 erasures on the expanded stream. These will be decoded with a delay of $n(\alpha\tilde{T}_1 + B) = nT_2$ on the expanded stream, which can be easily verified to incur a delay of T_2 on the original stream.

For user 1 the expanded source stream incurs a delay of $n\tilde{T}_1$, which reduces to a delay of $\lceil\tilde{T}_1\rceil$ on the original stream. This suffices the requirements of user 1 as by construction $T_1 \geq \lceil\tilde{T}_1\rceil$.

We provide a numerical example below.

1) Example — Source Expansion: Consider a Mu-SCo with parameters $\{(1, 2), (2, 4)\}$ which falls in the (b) region. The capacity is given by $R = 3/5$. The construction is provided in Table II. Through direct calculation note that $\tilde{T}_1 = 1.5$. Hence we implement a source expansion technique with $n = 2$ as follows.

We split each source symbol $\mathbf{s}[i]$ into six sub-symbols $s_0[i], \ldots, s_5[i]$ and construct an expanded source sequence $\tilde{\mathbf{s}}[\cdot]$ such that $\tilde{\mathbf{s}}[2i] = (s_0[i], s_1[i], s_2[i])$ and $\tilde{\mathbf{s}}[2i+1] = (s_3[i], s_4[i], s_5[i])$. We use the $\{(2, 3), (4, 8)\}$ DE-SCo code (see Table I) that we apply to $\tilde{\mathbf{s}}[\cdot]$ to produce the parity checks $\tilde{\mathbf{p}}[\cdot]$ and transmit $\mathbf{p}[i] = (\tilde{\mathbf{p}}[2i], \tilde{\mathbf{p}}[2i+1])$ along with $\mathbf{s}[i]$ at time i. It can be verified directly that the resulting code corrects a single erasure with a delay of 2 symbols and an erasure-burst of length 2 with a delay of 4.

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

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

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

$[i-1]$	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$
$s_0[i-1]$	$s_0[i]$	$s_0[i+1]$	$s_0[i+2]$	$s_0[i+3]$	$s_0[i+4]$
$s_1[i-1]$	$s_1[i]$	$s_1[i+1]$	$s_1[i+2]$	$s_1[i+3]$	$s_1[i+4]$
$s_0[i-3]\oplus s_1[i-2]$	$s_0[i-2]\oplus s_1[i-1]$	$s_0[i-1]\oplus s_1[i]$	$s_0[i]\oplus s_1[i+1]$	$s_0[i+1]\oplus s_1[i+2]$	$s_0[i+2]\oplus s_1[i+3]$
$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$s_1[i-6]\oplus s_0[i-5]$	$s_1[i-5]\oplus s_0[i-4]$	$s_1[i-4]\oplus s_0[i-3]$	$s_1[i-3]\oplus s_0[i-2]$	$s_1[i-2]\oplus s_0[i-1]$	$s_1[i-1]\oplus s_0[i]$

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 TWO POINTS $\{(1,2)-(2,6)\}$ AND $\{(1,2)-(2,5)\}$ LIES IN REGION (A).

C. Region (c)

Region (c) is sandwiched between $T_1 \leq T_2 \leq T_1+B_1$ and also satisfies $T_1 \geq B_2$ in Fig. 2. The capacity is given by

$$C_c = \frac{T_1}{T_1 + B_2}. \tag{34}$$

For the achievability scheme, we use an approach similar to region (b). We can reduce the delay T_2 of user 2 in region (c) so that it meets the $T_1 = T_2$ line without changing the capacity C_c. We can then apply a single user (B_2, T_1) code that simultaneously satisfies both the users. Clearly this code is feasible since $T_1 \geq B_2$. The rate of this SCo code meets the capacity. Note also that we do not require *source expansion* in this step.

D. Region (d)

In this region $T_2 \leq T_1$ and $B_2 \geq B_1$. It suffices to serve user 2 and the upper bound shows that the capacity $C_d = C_2$ is also achieved using a single user SCo of parameters (B_2, T_2).

VI. IA-SCo CONSTRUCTION (PROPOSITION 1)

We first provide a simple example to illustrate the main idea behind IA-SCo and then provide the general construction. We note that the IA-SCo codes achieve the capacity in a subset of region (a) in Fig. 2. While IA-SCo codes do not provide any new capacity results, their construction is much simpler than DE-SCo and perhaps easier to generalize when there are more than two receivers.

A. Example

Consider an example with the first and second users experiencing burst erasures of length $B_1 = 1$ and $B_2 = 2$ symbols respectively (i.e., $\alpha = 2$) and the corresponding delay for the first user is $T_1 = 2$. From Prop. 1 we have that $T_2 = 6$. Table III(a) illustrates the IA-SCo construction. For comparison the optimal DE-SCo construction achieving $T_2 = 5$, proposed in [16] is provided in Table III(b).

The construction of the IA-SCo code is as follows. We split each source symbol $\mathbf{s}[i]$ into two sub-symbols $s_0[i]$ and $s_1[i]$ of equal size. Let $p_1[i] = s_0[i-2] \oplus s_1[i-1]$ be the parity check associated with the $(1,2)$ SCo-code [10] and let $p_2[i] = s_0[i-4] \oplus s_1[i-2]$ be the parity check for the $(2,4)$ SCo-code [10]. The parity check row is obtained by combining $q[i] = p_1[i] \oplus p_2[i-T_1]$ i.e., by shifting $p_2[i]$ by T_1 units and then combining. The parity check stream $q[\cdot]$ are then concatenated with the source symbols as shown in Table III(a).

When an erasure of one symbol occurs say at $t = i - 1$ for user 1, it needs to recover $\mathbf{s}[i-1]$ at time $t = i + 1$. Note that user 1 can cancel the second row of parity checks $p_2[\cdot]$, which combines unerased sub-symbols. For user 2 suppose that a burst erasure of length $B_2 = 2$ symbols occurs at times $t = i - 2, i - 1$. User 2 simply ignores the parity checks $q[i]$ and $q[i+1]$. Starting from $t = i + 2$, the parity checks $p_1[\cdot]$ are functions of symbols $\mathbf{s}[i], \mathbf{s}[i+1], \ldots$ and do not involve the erased symbols $\mathbf{s}[i-1]$ and $\mathbf{s}[i-2]$. Therefore we can subtract $p_1[\cdot]$ from $q[i+2], \ldots, q[i+6]$ and recover $p_2[i], \ldots, p_2[i+4]$, which suffice to recover the missing symbols.

B. General Construction

The main idea behind the general construction is to start with two single-user codes for the two users, (B_1, T_1) and $(\alpha B_1, \alpha T_1)$ and delay the parity checks of the second by T_1 so that they can be combined with the parity checks of user 1 without causing any interference to the two users.

Throughout our discussion we let $T_1 = T$ and $B_1 = B$ and $B_2 = \alpha B$ and $T_2 = \alpha T + T$.

1) Code Construction:

- Let $\mathcal{C}_1$ be the single-user code of user 1 [10], [16]. Assume that the source symbols $\mathbf{s}[i]$ are divided into T sub-symbols $(s_0[i], \ldots, s_{T-1}[i])$ and combined to produce B parity check sub-symbols $\mathbf{p}^{\mathrm{I}}[i] = (p_0^{\mathrm{I}}[i], \ldots, p_{B-1}^{\mathrm{I}}[i])$ according to

$$p_j^{\mathrm{I}}[i] = s_j[i - T] + h_j(s_B[i - (j + T - B)], \ldots, s_{T-1}[i - (j + 1)]), \quad j = 0, \ldots, B - 1. \quad (35)$$

- Let $\mathcal{C}_2$ be a rate C_1 SCo with parameters $(\alpha B, \alpha T)$ also obtained by splitting the source symbols into T sub-symbols $(s_0[i], \ldots, s_{T-1}[i])$ combined to produce B parity checks $\mathbf{p}^{\mathrm{II}}[i] = (p_0^{\mathrm{II}}[i], \ldots, p_{B-1}^{\mathrm{II}}[i])$ according to the vertical interleaving property in [16], i.e.,

$$p_j^{\mathrm{II}}[i] = s_j[i - \alpha T] + h_j(s_B[i - \alpha(j + T - B)], \ldots, s_{T-1}[i - (j + 1)\alpha]), j = 0, \ldots, B - 1. \quad (36)$$

- Construct $\mathcal{C}_M$ whose symbols have the form $(\mathbf{s}[i], \mathbf{q}[i])$, where $\mathbf{q}[i] = \mathbf{p}^{\mathrm{I}}[i] + \mathbf{p}^{\mathrm{II}}[i - T]$. Intuitively the stream of parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ is delayed by T symbols and then the resulting non-interfering streams are combined.

Clearly the rate of $\mathcal{C}_M$ equals C_1. We need to show that user 1 and user 2 can recover from erasure bursts of B and $B_2 = \alpha B$ within delays of T and $T_2 = \alpha T + T$ respectively.

2) User 1 Decoding: Assume that the symbols at time $i, \ldots, i + B - 1$ are erased on user 1's channel. By virtue of $\mathcal{C}_1$, symbol $\mathbf{s}[i+k]$ (for $k = 0, 1, \ldots, B - 1$) can be recovered by time $i + k + T$ using parity checks $\mathbf{p}^{\mathrm{I}}[i + B], \ldots, \mathbf{p}^{\mathrm{I}}[i + k + T]$. Thus it suffices to show that we can recover $\mathbf{p}^{\mathrm{I}}[i + k]$ from $\mathbf{q}[i + k]$ for $k = B, \ldots, B + T - 1$.

First note that the $\mathbf{p}^{\mathrm{I}}[i+k]$ for $k = B, \ldots, T$ can be directly recovered from $\mathbf{q}[i+k]$ since the interfering parity checks $\mathbf{p}^{\mathrm{II}}[\cdot]$ only consist of source symbols before time i. Indeed the parity check at $k = T$ is

$$\mathbf{q}[i + T] = \mathbf{p}^{\mathrm{I}}[i + T] + \mathbf{p}^{\mathrm{II}}[i]$$

and from (36) the sub-symbols in $\mathbf{p}^{\mathrm{II}}[i]$ only depend on the source symbols before time i. Thus upon receiving $\mathbf{q}[i+T]$ user 1 can recover the erased symbol $\mathbf{s}[i]$. Furthermore, we can also compute $\mathbf{p}^{\mathrm{II}}[i+1]$ which only consists of source symbols up to time i and upon receiving $\mathbf{q}[i+T+1]$ can compute $\mathbf{p}^{\mathrm{I}}[i+T+1]$ from

$$\mathbf{p}^{\mathrm{I}}[i + T + 1] = \mathbf{q}[i + T + 1] - \mathbf{p}^{\mathrm{II}}[i + 1].$$

(a) Step (1) (b) Step (2) (c) Step (3)
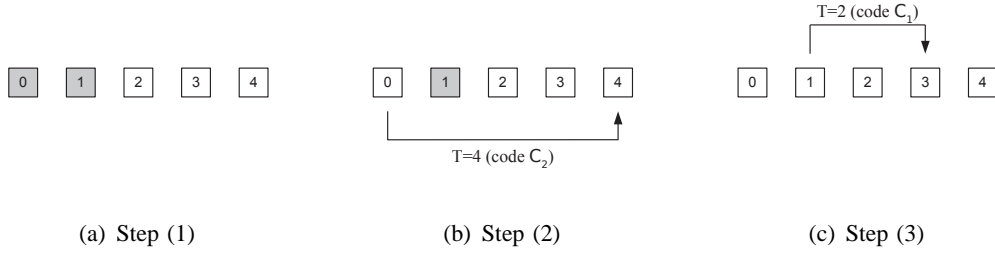
Fig. 6. Main steps of finding the upper-bound for the $\{(1,2)-(2,4)\}$ point through one period illustration of the Periodic Erasure Channel. Grey and white squares resemble erased and unerased symbols respectively.

In turn it recovers $\mathbf{s}[i+1]$. Continuing this process it can recover all the erased symbols $\mathbf{s}[i+k]$ by time $i+T+k$.

3) User 2 Decoding: Suppose that the symbols at time $i,\ldots,i+B_2-1$ are erased on user 2's channel. By virtue of $\mathcal{C}_2$, symbol $\mathbf{s}[i+k]$ (for $k=0,1,\ldots,B_2-1$) can be recovered by time $i+k+\alpha T$ using parity checks $\mathbf{p}^{\mathrm{II}}[i+B_2],\ldots,\mathbf{p}^{\mathrm{II}}[i+k+\alpha T]$. To establish (9) it suffices to show that symbols $\mathbf{p}^{\mathrm{II}}[i+B_2],\ldots,\mathbf{p}^{\mathrm{II}}[i+k+\alpha T]$ can be recovered from symbols $\mathbf{q}[i+T+B_2],\ldots,\mathbf{q}[i+k+\alpha T+T]$. Indeed since

$$\mathbf{q}[i+B_2+T] = \mathbf{p}^{\mathrm{I}}[i+B_2+T] + \mathbf{p}^{\mathrm{II}}[i+B_2],$$

it suffices to observe that user 2 can cancel $\mathbf{p}^{\mathrm{I}}[i+B_2+T]$ upon receiving $\mathbf{q}[i+B_2+T]$. It however immediately follows from (35) that $\mathbf{p}^{\mathrm{I}}[i+B_2+T]$ involves source symbols at time $i+B_2$ or later (the construction limits the memory in the channel input stream to previous T symbols). The symbols $\mathbf{p}^{\mathrm{I}}[\cdot]$ after this time also depend on $\mathbf{s}[\cdot]$ at time $i+B_2$ or later.

VII. PROOF OF THEOREM 4

We first provide an example to illustrate the upper bound using periodic erasure channel. Then we outline the general periodic erasure channel (PEC) based argument. Finally we provide a rigorous information theoretic converse.

A. Example

The main steps of this proof can be illustrated by first considering a specific example, $\{(1,2)-(2,4)\}$ which is shown in Fig. 6. We consider a periodic erasure channel with each period having two consecutive erasures followed by three unerased symbols. Thus, one can start by using code $\mathcal{C}_2 = (2,4)$ to recover the erasure at time 0, wtih a delay of 4, leaving only one erasure at time 1 (c.f. Fig. 6(b)). Now, code $\mathcal{C}_1 = (1,2)$ can be used as it is capable of recovering this one erasure within a delay of 2 (i.e., by time 3) (c.f. Fig. 6(c)). Let us assume that the code is systematic and thus one can recover the symbols at time 2, 3 and 4 from their corresponding unerased channel symbols. Thus, one can recover a total of 5 source symbols from 3 unerased channel symbols which implies that $3/5$ is an upper-bound of this channel.

B. PEC based Converse

For the general case of Theorem 4, we start by the case, $T_2 > T_1 + B_1$ and then consider $T_2 \leq T_1 + B_1$.

Lemma 1: When $T_2 > T_1 + B_1$, suppose there exists a sequence of feasible encoding functions $\{f_t(\cdot)\}$ and decoding functions $\{\gamma_{1t}(\cdot)\}$ and $\{\gamma_{2t}(\cdot)\}$. Then there also exist decoding functions $\gamma_t(\cdot)$ that can reproduce the source symbols $\mathbf{s}[t]$, over a channel with periodic bursts as stated below

$$\mathbf{y}[t] = \begin{cases} \star, & t \in \left[T^k, T^k+B_2-1\right] \\ \mathbf{x}[t], & t \in \left[T^k+B_2, T^{k+1}-1\right] \end{cases} \tag{37}$$

where $T^k = kT_2 + k(B_2-B_1)$, $k = 0,1,\ldots$

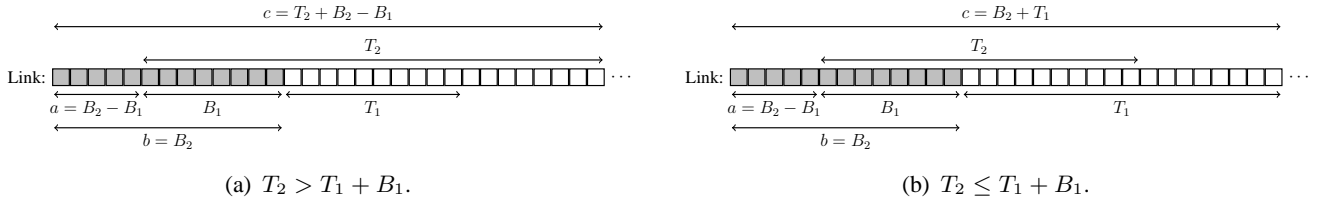(a) $T_2 > T_1 + B_1$.   (b) $T_2 \leq T_1 + B_1$.

Fig. 7. One period illustration of the Periodic Erasure Channel in Fig. 4 to be used for proving the multicast upper-bound provided in Theorem 4.

An illustration of one period of the proposed periodic-erasure channel (from T^0 to T^1) in the case $T_2 > T_1 + B_1$ is shown in Fig. 7(a). The capacity of the periodic erasure channel in Lemma 1 is

$$C \leq \frac{T_2 - B_1}{T_2 - B_1 + B_2}. \tag{38}$$

To establish Lemma 1, it suffices to show that by time $T^k - 1$, the receiver is able to recover symbols $\mathbf{x}[0], \ldots, \mathbf{x}[T^k - 1]$. We first show that by time $T^1 - 1$, the receiver is able to recover symbols $\mathbf{x}[0], \ldots, \mathbf{x}[T^1 - 1]$. Since only symbols $\mathbf{x}[0], \ldots, \mathbf{x}[B_2 - 1]$ are erased by time $T^1 - 1$ we focus on these symbols.

Consider a single-burst channel that introduces a burst of length B_2 from times $t = 0, 1, \ldots, B_2 - 1$. Note that this channel behaves identically to the periodic burst channel up to time $T^1 - 1$. Applying the decoder $\gamma_{2t}(\cdot)$ for $t = 0, 1, \ldots, (B_2 - B_1) - 1$, the receiver recovers symbols $\mathbf{s}[0], \ldots, \mathbf{s}[t]$ with a delay of T_2 i.e., by time $T^1 - 1$ and hence it also recovers the channel packets $\mathbf{x}[0], \ldots, \mathbf{x}[t]$ via (1). It remains to show that the symbols at time $t = (B_2 - B_1), \ldots, B_2 - 1$ are also recovered by time $T^1 - 1$. One cannot apply the decoder $\gamma_{2t}(\cdot)$ to recover these symbols since the decoding will require symbols beyond time T^1, which are available on the single-burst channel but not on the periodic burst channel. However, to recover these symbols we use the multicast property of the code as follows. Consider a channel that introduces a single erasure burst of length B_1 between times $t = (B_2 - B_1), \ldots, B_2 - 1$. Note that up to time $T^1 - 1$, this channel is identical to our periodic burst-erasure channel (which has recovered $\mathbf{x}[0], \ldots, \mathbf{x}[B_2 - B_1 - 1]$). For this channel, and hence the periodic erasure channel, using the decoder $\gamma_{1t}(\cdot)$ the source symbols are recovered by time $B_2 + T_1 - 1 \leq T^1 - 1$. Furthermore via (1), the erased channel symbols $\mathbf{x}[B_2 - B_1], \ldots, \mathbf{x}[B_2 - 1]$ are also recovered by time $T^1 - 1$. Since the channel introduces periodic bursts, the same argument can be repeated to recover all symbols up to time $T^k - 1$ for each k.

The same argument applies in the case when $T_2 \leq T_1 + B_1$ (in Fig. 7.(b)) except that the periodic bursts are stated as,

$$\mathbf{y}[t] = \begin{cases} \star, & t \in \left[T^k, T^k + B_2 - 1\right] \\ \mathbf{x}[t], & t \in \left[T^k + B_2, T^{k+1} - 1\right] \end{cases} \tag{39}$$

where $T^k = k(T_1 + B_2)$ and the theorem follows.

C. Information Theoretic Converse

Recall that our PEC argument assumed that (1) the channel packets $\mathbf{x}[t]$ are deterministic functions of the source packets up to time t, (2) the channel code is systematic and (3) the recovery must happen with zero error. All of these assumptions can be removed by resorting to the information theoretic converse discussed next.

We start by proving the first case $T_2 > T_1 + B_1$. We use the periodic erasure channel shown in Fig. 7.(a),

where each period has B_2 erasures followed by $T_2 - B_1$ non-erasures. We can assign

$$a = B_2 - B_1, \quad b = B_2, \quad c = B_2 + T_2 - B_1 \quad \text{(period length)},$$
$$W_i = \mathbf{x}\left[\begin{smallmatrix}(i+1)c-1\\ic+b\end{smallmatrix}\right], \quad V_i = \mathbf{s}\left[\begin{smallmatrix}(i+1)c-1\\ic\end{smallmatrix}\right].$$

We define the capability of the $\mathcal{C}_1 = (B_1, T_1)$ and $\mathcal{C}_2 = (B_2, T_2)$ codes by,

$$H\left(\mathbf{s}[i]\Big|\mathbf{x}\left[\begin{smallmatrix}i+T_1\\i+B_1\end{smallmatrix}\right]\mathbf{x}\left[\begin{smallmatrix}i-1\\0\end{smallmatrix}\right]\right) = 0 \tag{40}$$

$$H\left(\mathbf{s}[i]\Big|\mathbf{x}\left[\begin{smallmatrix}i+T_2\\i+B_2\end{smallmatrix}\right]\mathbf{x}\left[\begin{smallmatrix}i-1\\0\end{smallmatrix}\right]\right) = 0, \tag{41}$$

We use mathematical induction to prove that for $n \geq 0$

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n\Big|V_0^{n-1}\mathbf{x}\left[\begin{smallmatrix}nc-1\\0\end{smallmatrix}\right]\right). \tag{42}$$

The base case for (42) is given by substituting $n = 0$ into it:

$$H(W_0) \geq H(V_0^{-1}) + H\left(W_0\Big|V_0^{-1}\mathbf{x}\left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right]\right) \geq H(W_0) \tag{43}$$

which is obviously true. For the induction step, let us start by assuming that (42) is true for $n = k$,

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k\Big|V_0^{k-1}\mathbf{x}\left[\begin{smallmatrix}kc-1\\0\end{smallmatrix}\right]\right). \tag{44}$$

In the first part of the induction step, some entropy manipulations are applied (c.f. Appendix. B), to show that:

$$H(W_0^k) \geq H\left(V_0^{k-1}\mathbf{s}\left[\begin{smallmatrix}kc+b-1\\kc\end{smallmatrix}\right]\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\left[\begin{smallmatrix}kc+b-1\\kc\end{smallmatrix}\right]\mathbf{x}\left[\begin{smallmatrix}kc+b-1\\0\end{smallmatrix}\right]\right) \tag{45}$$

These entropy manipulations can be summarized in two main steps, the first of which is recovering the first $a = B_2 - B_1$ source symbols, $\mathbf{s}\left[\begin{smallmatrix}kc+a-1\\kc\end{smallmatrix}\right]$ using code $\mathcal{C}_2$ defined in (41) due to the availability of W_k, while the second step is recovering the next $b - a = B_1$ source symbols, $\mathbf{s}\left[\begin{smallmatrix}kc+b-1\\kc+a\end{smallmatrix}\right]$ using $\mathcal{C}_1$ defined in (40).

In the second part, we add $H(W_{k+1}|W_0^k)$ to both sides of the inequality. Because the channel code is not necessarily systematic, we will use the additional channel packets in W_{k+1} to help decode the source packets $\mathbf{s}\left[\begin{smallmatrix}(k+1)c-1\\kc+b\end{smallmatrix}\right]$ (detailed steps are shown in Appendix. B).

$$H(W_0^{k+1}) \geq H(V_0^k) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\left[\begin{smallmatrix}(k+1)c-1\\0\end{smallmatrix}\right]\right). \tag{46}$$

The working in (46) shows that if (42) is true for $n = k$, then it is also true for $n = k + 1$. By induction, (42) is true for $n \geq 0$. Finally,

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n\Big|V_0^{n-1}\mathbf{x}\left[\begin{smallmatrix}nc-1\\0\end{smallmatrix}\right]\right) \geq H(V_0^{n-1}).$$

Using the fact that all of the channel packets have the same entropy, and all of the source packets have the same entropy, we can continue to get

$$H(W_0^n) \geq H(V_0^{n-1})$$
$$(n+1) \cdot (T_2 - B_1) \cdot H(\mathbf{x}) \geq n \cdot (T_2 + B_2 - B_1) \cdot H(\mathbf{s})$$
$$\frac{n+1}{n} \cdot \frac{T_2 - B_1}{T_2 + B_2 - B_1} \geq \frac{H(\mathbf{s})}{H(\mathbf{x})}. \tag{47}$$

Finally, we get

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T_2 - B_1}{T_2 - B_1 + B_2}. \quad (\text{as } n \to \infty) \tag{48}$$

Therefore, any $\{(B_1, T_1), (B_2, T_2)\}$ code with $T_2 > T_1 + B_1$ must satisfy (48).

For the case with $T_2 \leq T_1 + B_1$, the same proof applies except that the values of a, b and c are updated as follows,

$$a = B_2 - B_1, \quad b = B_2, \quad c = B_2 + T_1 \quad (\text{period length}),$$

and again we end up having,

$$H(W_0^n) \geq H(V_0^{n-1})$$
$$(n+1) \cdot T_1 \cdot H(\mathbf{x}) \geq n \cdot (T_1 + B_2) \cdot H(\mathbf{s})$$
$$\frac{n+1}{n} \cdot \frac{T_1}{T_1 + B_2} \geq \frac{H(\mathbf{s})}{H(\mathbf{x})}. \tag{49}$$

In other words,

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T_1}{T_1 + B_2}. \quad (\text{as } n \to \infty) \tag{50}$$

Therefore, any $(B_1, T_1), (B_2, T_2)$ code with $T_2 \leq T_1 + B_1$ must satisfy (50).

VIII. CODE CONSTRUCTION IN REGION (E) (THEOREM 5)

Recall that region (e) in Fig. 2 and Fig. 3 is contained within $T_2 \geq B_1 + T_1$, $T_2 \geq B_2$ and $T_2 \leq B_1 + B_2$. Since the capacity C_e given by

$$C_e = \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \tag{51}$$

is constant along each 45-degree line starting from the line $T_2 = B_1 + T_1$ in the (B_2, T_2) plane in Fig. 3, we can parameterize the T_2 and B_2 as

$$T_2 = T_1 + B_1 + m, \qquad B_2 = T_1 + k + m,$$

where $m \geq 0$ is the number of steps upwards on the 45 line in Fig. 3 starting from $T_2 = T_1 + B_1$ line dividing regions (e) and (f), and where k is an integer taking values from 0 to B_1, which horizontally spans region (e) from $T_2 = B_1 + B_2$ to $T_2 = B_2$. Substituting into (51) we have

$$C_e = \frac{T_1}{2T_1 + k}, \tag{52}$$

which we will show is achievable.

In Appendix C we provide two examples of the code constructions with parameters $\{(4, 5), (7, 10)\}$ and $\{(3, 5), (7, 9)\}$. These examples compliment the general description below and might be worthwhile reading in parallel with this section.

The construction generates three layers of parity checks and carefully combines them to satisfy the decoding constraints of both the receivers. The main construction steps are described below.

- Split each source symbols $\mathbf{s}[i]$ in T_1 sub-symbols

$$\mathbf{s}[i] = (s_0[i], \ldots, s_{T_1-1}[i])$$

- Apply a $\mathcal{C}_1 = (B_1, T_1)$ single user SCo code to the source symbols $\mathbf{s}[i]$ producing B_1 parity check sub-symbols

$$\mathbf{p}^{\mathrm{I}}[i] = (p_0^{\mathrm{I}}[i], \ldots, p_{B_1-1}^{\mathrm{I}}[i])$$

at each time by combining the source sub-symbols along the main diagonal,

$$p_j^{\mathrm{I}}[i] = s_j[i - T_1] + h_j^{\mathrm{II}}(s_{B_1}[i - j - T_1 + B_1], \ldots, s_{T_1 - 1}[i - j - 1]), \quad j = \{0, 1, \ldots, B_1 - 1\}.$$

- Apply a $\mathcal{C}_2$ repetition code to the source symbols $\mathbf{s}[i]$ with a delay of T_2, i.e., the corresponding parity check symbols are,

$$\mathbf{p}^{\mathrm{II}}[i] = (p_0^{\mathrm{II}}[i], \ldots, p_{T_1 - 1}^{\mathrm{II}}[i]) = (s_0[i - T_2], \ldots, s_{T_1 - 1}[i - T_2]) = \mathbf{s}[i - T_2]. \tag{53}$$

- Concatenate the two streams $\mathbf{p}^{\mathrm{I}}[\cdot]$ and $\mathbf{p}^{\mathrm{II}}[\cdot]$ with a *partial* overlap as illustrated in (54). In particular, the two streams of parity checks $\mathbf{p}^{\mathrm{I}}[\cdot]$ and $\mathbf{p}^{\mathrm{II}}[\cdot]$ are concatenated with the last $B_1 - k$ rows of the first added to upper most $B_1 - k$ rows of the second.

$$\tilde{\mathbf{x}}[i] = \begin{bmatrix} s_0[i] \\ \vdots \\ s_{T-1}[i] \\ p_0^{\mathrm{I}}[i] \\ \vdots \\ p_{k-1}^{\mathrm{I}}[i] \\ p_k^{\mathrm{I}}[i] \quad +s_0[i - T_2] \\ \vdots \\ p_{B_1 - 1}^{\mathrm{I}}[i] \quad +s_{B_1 - k - 1}[i - T_2] \\ s_{B_1 - k}[i - T_2] \\ \vdots \\ s_{T_1 - 1}[i - T_2] \end{bmatrix} \tag{54}$$

We further combine the last $T_1 - (B_1 - k)$ parity checks of $\mathbf{p}^{\mathrm{II}}[\cdot]$ with additional parity checks of code $\mathcal{C}_3$ as explained below.

- Consider the two cases:

(A) $T_1 \leq 2(B_1 - k)$

- Apply a $\mathcal{C}_3 = (B_3, T_3) = (T_1 - (B_1 - k), B_1 - k)$ single user SCo code on the last $B_1 - k$ *parity check* sub-symbols of $\mathcal{C}_1$, $(p_k^{\mathrm{I}}[.], \ldots, p_{B_1 - 1}^{\mathrm{I}}[.])$ constructing $T_1 - (B_1 - k)$ parity checks

$$\mathbf{p}^3[i] = (p_0^3[i], \ldots, p_{T_1 - (B_1 - k) - 1}^3[i])$$

at each time by combining the last $B_1 - k$ *parity check* sub-symbols, $(p_k^{\mathrm{I}}[.], \ldots, p_{B_1 - 1}^{\mathrm{I}}[.])$, along the main diagonal, i.e.,

$$p_j^3[i] = p_{k+j}^{\mathrm{I}}[i - T_3] + h_j^3(p_{k+B_3}^{\mathrm{I}}[i - j - T_3 + B_3], \ldots, p_{k+T_3 - 1}^{\mathrm{I}}[i - j - 1]),$$

for $j = \{0, 1, \ldots, T_1 - (B_1 - k) - 1\}$.

- Combine a $\Delta_3 = -T_1$ shifted version of the produced stream of parity checks $\mathbf{p}^3[.]$ to the last $T_1 - (B_1 - k)$ rows of $\mathbf{x}[.]$, thus,

$$\mathbf{x}[i] = \begin{bmatrix} \mathbf{s}[i] \\ p_0^{\mathrm{I}}[i] \\ \vdots \\ p_{k-1}^{\mathrm{I}}[i] \\ p_k^{\mathrm{I}}[i] \quad +s_0[i-T_2] \\ \vdots \\ p_{B_1-1}^{\mathrm{I}}[i] \quad +s_{B_1-k-1}[i-T_2] \\ s_{B_1-k}[i-T_2] \quad +p_0^3[i+T_1]\big|_i \\ \vdots \\ s_{T_1-1}[i-T_2] \quad +p_{T_1-(B_1-k)-1}^3[i+T_1]\big|_i \end{bmatrix} \tag{55}$$

where

$$\mathbf{p}^3[t_1]\big|_{t_2} = \left(p_0^3[t_1]\big|_{t_2}, \ldots, p_{T_1-(B_1-k)-1}^3[t_1]\big|_{t_2} \right)$$

is the parity-check $\mathbf{p}^3[t_1]$ shifted to time t_2.

We note that the construction of $\mathbf{x}[i]$ in (55) requires us to have access to source symbols after time i as the parity checks $\mathbf{p}^3[i+T_1]$ may include source symbols after time i. Since our encoder is causal we cannot have access to these source symbols. Instead we transmit only the causal part of the underlying parity checks. In particular, we decompose each parity check into two parts as follows. For any $t_2 \le t_1$ we have,

$$p_j^3[t_1]\big|_{t_2} = \tilde{p}_j^3[t_1]\big|_{t_2} + \hat{p}_j^3[t_1]\big|_{t_2} \tag{56}$$

where $\tilde{p}_j^3[t_1]\big|_{t_2}$ denotes the causal part of the parity check with respect to t_2 whereas $\hat{p}_j^3[t_1]\big|_{t_2}$ denotes the non-causal part of the parity check with respect to t_2 i.e.,

$$\tilde{p}_j^3[t_1]\big|_{t_2} = f_j(\mathbf{s}[t_2], \mathbf{s}[t_2-1], \mathbf{s}[t_2-2]\ldots) \tag{57}$$

$$\hat{p}_j^3[t_1]\big|_{t_2} = g_j(\mathbf{s}[t_2+1], \mathbf{s}[t_2+2]\ldots). \tag{58}$$

The resulting input symbol at time i is given by

$$\mathbf{x}[i] = \begin{bmatrix} \mathbf{s}[i] \\ p_0^{\mathrm{I}}[i] \\ \vdots \\ p_{k-1}^{\mathrm{I}}[i] \\ p_k^{\mathrm{I}}[i] \quad +s_0[i-T_2] \\ \vdots \\ p_{B_1-1}^{\mathrm{I}}[i] \quad +s_{B_1-k-1}[i-T_2] \\ s_{B_1-k}[i-T_2] \quad +\tilde{p}_0^3[i+T_1]\big|_i \\ \vdots \\ s_{T_1-1}[i-T_2] \quad +\tilde{p}_{T_1-(B_1-k)-1}^3[i+T_1]\big|_i \end{bmatrix} \tag{59}$$

The symbol $\mathbf{x}[i]$ in (59) is the transmitted symbol at time i.

(B) $T_1 > 2(B_1 - k)$

Since, $B_1 - k > T_1 - (B_1 - k)$, a SCo of parameters $(T_1 - (B_1 - k), B_1 - k)$ constructed in case (A) is not feasible and is thus replaced by a set of SCo codes. For the associated values of T_1, B_1

and k we let

$$T_1 - (B_1 - k) = r(B_1 - k) + q, \quad q < (B_1 - k). \tag{60}$$

– Let

$$\mathcal{C}_{3,n} = (B_{3,n}, T_{3,n}) = (B_1 - k, B_1 - k), \ n = 1, \dots, r,$$

be a set of r SCo *repetition* codes applied on the last $B_1 - k$ *parity check* sub-symbols $(p_k^{\mathrm{I}}[i], \dots, p_{B_1-1}^{\mathrm{I}}[i])$ and repeating them to construct r sets of parity check vectors each of size $B_1 - k$ as follows,

$$\mathbf{p}^{3,n}[i] = (p_0^{3,n}[i], \dots, p_{B_1-k-1}^{3,n}[i]) = (p_k^{\mathrm{I}}[i + n(B_1 - k)], \dots, p_{B_1-1}^{\mathrm{I}}[i + n(B_1 - k)]), \tag{61}$$

at each time, i.e., a $\mathcal{C}_{3,n}$ code is a $(B_1 - k, B_1 - k)$ SCo repetition code shifted back by $(n+1)(B_1 - k)$.

– Let $\mathcal{C}_{3,r+1}$ be a $(B_{3,r+1}, T_{3,r+1}) = (q, B_1 - k)$ SCo again applied on the last $B_1 - k$ *parity check* sub-symbols $(p_k^{\mathrm{I}}[i], \dots, p_{B_1-1}^{\mathrm{I}}[i])$ and then constructing q parity checks $\mathbf{p}^{3,r+1}[i] = (p_0^{3,r+1}[i], \dots, p_{q-1}^{3,r+1}[i])$ at each time by combining the last $B_1 - k$ *parity check* sub-symbols, $(p_k^{\mathrm{I}}[.], \dots, p_{B_1-1}^{\mathrm{I}}[.])$, along the main diagonal.

– Concatenate the set of streams $\mathbf{p}^{3,n}[.]$ for $n = 1, \dots, r$ and $\mathbf{p}^{3,r+1}[.]$ after introducing a shift of $\Delta_{3,r+1} = -T_1$ in the later. The output symbol at time i is,

$$\mathbf{x}[i] = \begin{bmatrix} \mathbf{s}[i] \\ p_0^{\mathrm{I}}[i] \\ \vdots \\ p_{k-1}^{\mathrm{I}}[i] \\ p_k^{\mathrm{I}}[i] \quad +s_0[i - T_2] \\ \vdots \\ p_{B_1-1}^{\mathrm{I}}[i] \quad +s_{B_1-k-1}[i - T_2] \\ s_{B_1-k}[i - T_2] \quad +\tilde{p}_0^3[i] \\ \vdots \\ s_{T_1-1}[i - T_2] \quad +\tilde{p}_{T_1-(B_1-k)-1}^3[i] \end{bmatrix} \tag{62}$$

where

$$(\tilde{p}_0^3[i], \dots, \tilde{p}_{T_1-(B_1-k)-1}^3[i]) = (\tilde{\mathbf{p}}^{3,1}[i]\big|_i, \dots, \tilde{\mathbf{p}}^{3,r}[i]\big|_i, \tilde{\mathbf{p}}^{3,r+1}[i + T_1]\big|_i) \tag{63}$$

is the concatenation of the $r + 1$ parity check sub-streams for the codes $\mathcal{C}_{3,n}$ for $n = 1, \dots, r+1$, respectively. Since each of the first r of these sub-streams is composed of $B_1 - k$ parity check sub-symbols while the last of which is composed of q parity check sub-symbols, then the $\mathbf{p}^3[i]$ has a sum of $r(B_1 - k) + q = T_1 - (B_1 - k)$ parity check sub-symbols which will be denoted by the parity check sub-symbols of code $\mathcal{C}_3$ (the set of codes $\{\mathcal{C}_{3,1}, \mathcal{C}_{3,2}, \dots, \mathcal{C}_{3,r+1}\}$), and hence can be combined with the last $T_1 - (B_1 - k)$ parity check sub-symbols of code $\mathcal{C}_2$, $(p_{B_1-k}^{\mathrm{II}}[i], \dots, p_{T_1-1}^{\mathrm{II}}[i])$.

Since there are T_1 source sub-symbols and two streams of parity checks one with B_1 and the other with T_1 parity check sub-symbols for every T_1 source sub-symbols but partially overlapping in $B_1 - k$ rows, it follows that the rate of the code is $\frac{T_1}{2T_1 + B_1 - (B_1 - k)} = \frac{T_1}{2T_1 + k} = C_e$ (c.f. Fig. 8).

A graphical representation of such coding scheme is illustrated in Fig. 8. The horizontal axis represents time while the vertical axis represents the index of sub-symbols in the channel symbol at each time instant.
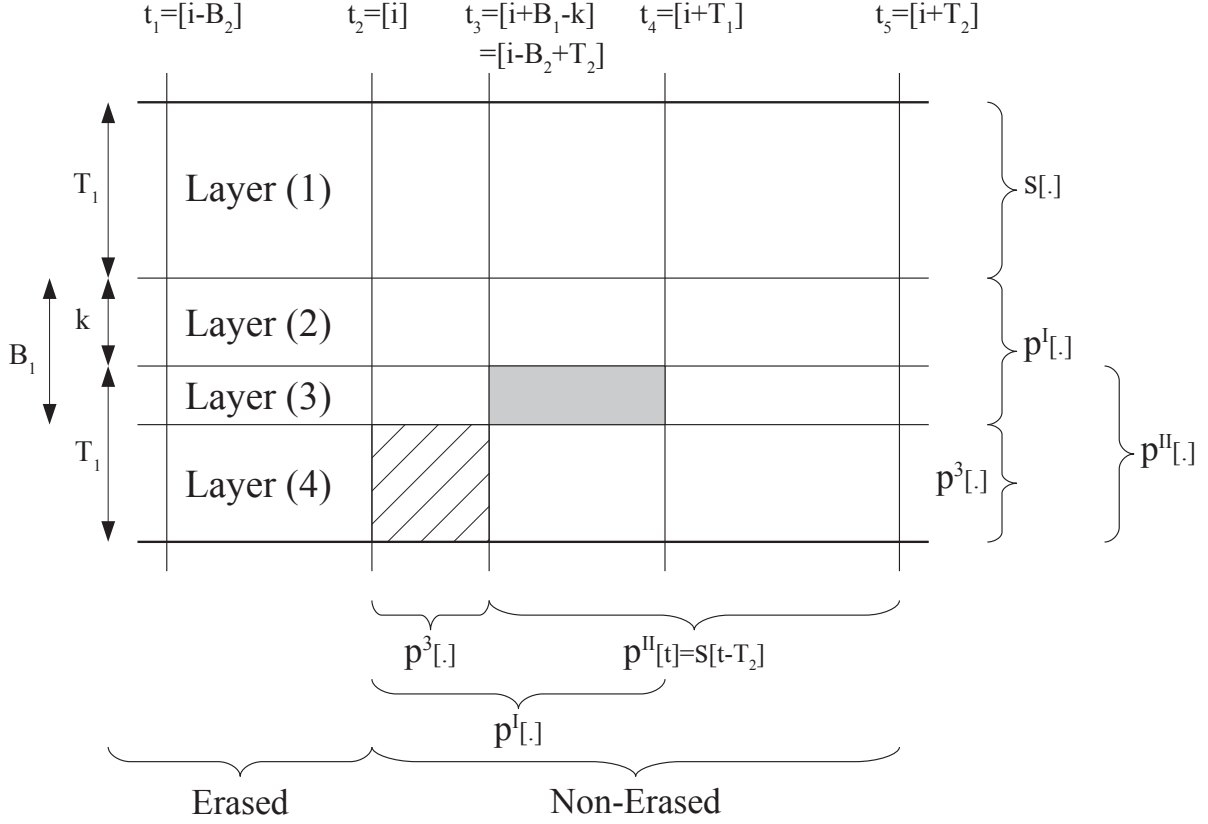
Fig. 8. A graphical illustration of both the Encoding and Decoding Steps for a general point lying in the (e) Region. The labels on the right show the layers spanned by each set of parity check sub-symbols. The labels at the bottom show the intervals in which each set of parity check sub-symbols combine erased source sub-symbols.

We divide each channel symbol into four layers,

- Layer (1) contains the T_1 source sub-symbols
- Layer (2) contains the first k of the B_1 parity check sub-symbols, $\mathbf{p}^{\mathrm{I}}[\cdot]$ produced by code $\mathcal{C}_1$
- Layer (3) has the remaining $B_1 - k$ parity check sub-symbols $\mathbf{p}^{\mathrm{I}}[\cdot]$ of $\mathcal{C}_1$ combined with the first $B_1 - k$ of the the parity check sub-symbols, $\mathbf{p}^{\mathrm{II}}[\cdot]$ produced by the repetition code, $\mathcal{C}_2$.
- Layer (4) has the remaining $T_1 - (B_1 - k)$ parity check sub-symbols of $\mathcal{C}_2$ combined with the parity checks of $\mathcal{C}_3$.

Note that two overlaps between codes exist in this coding scheme. The first is between codes $\mathcal{C}_1$ and $\mathcal{C}_2$ and takes place in layer (3), while the second is between $\mathcal{C}_2$ and $\mathcal{C}_3$ and takes place in layer (4).

A. Decoding of User 1

A burst erasure of length B_1 in the interval $\mathcal{I}_1 = [i - B_1, i - 1]$ can be directly recovered using the stream of parity checks $\mathbf{p}^{\mathrm{I}}[\cdot]$ in the interval $[i, i + T_1 - 1] = [t_2, t_4)$ (c.f. Fig. 8) produced by code $\mathcal{C}_1$ within a delay of T_1. The overlapping parity checks $\mathbf{p}^{\mathrm{II}}[t] = \mathbf{s}[t - T_2]$ in this interval consist of source symbols from the interval $\mathcal{I}_2 = [i - T_2, i + T_1 - T_2 - 1] = [i - T_1 - B_1 - m, i - B_1 - m - 1]$ which are unerased (i.e., $\mathcal{I}_2 \cap \mathcal{I}_1 = \Phi$ since $m \geq 0$).

B. Decoding of User 2

Suppose that the symbols in the interval $\mathcal{I}_2 = [i - B_2, i - 1]$ are erased by the channel of user 2. We start by summarizing the main decoding steps. Thereafter we describe each step in detail.

- **Step (1) (Recovery of $\mathbf{p}^{\mathrm{I}}[\cdot]$):** The parity checks of code $\mathcal{C}_3$, $\mathbf{p}^3[\cdot]$ in the interval $[i, \ldots, i+T_2-B_2-1]$ (in layer (4)) are capable of recovering the last $B_1 - k$ sub-symbols of $\mathbf{p}^{\mathrm{I}}[t]$ for $t \in \{i+T_2-B_2, \ldots, i+T_1-1\}$ by time t.
- **Step (2) (Removal of $\mathbf{p}^{\mathrm{I}}[\cdot]$):** Subtract $\mathbf{p}^{\mathrm{I}}[\cdot]$ in layer (3) starting at $i - B_2 + T_2$.
- **Step (3) (Removal of $\mathbf{p}^3[\cdot]$):** Compute and subtract $\mathbf{p}^3[\cdot]$ in layer (4) starting at $i - B_2 + T_2$.
- **Step (4) (Recovery using $\mathbf{p}^{\mathrm{II}}[\cdot]$):** Use $\mathbf{p}^{\mathrm{II}}[t]$ for $t \in \{i + T_2 - B_2, \ldots, i + T_2 - 1\}$ to recover the erased source symbols, $(\mathbf{s}[i - B_2], \ldots, \mathbf{s}[i - 1])$.

Step (1) (Recovery of $\mathbf{p}^{\mathrm{I}}[\cdot]$): Step (1) involves applying code $\mathcal{C}_3$ in computing some missing parity checks $\mathbf{p}^{\mathrm{I}}[t]$. This is the most elaborate step and is established in the following lemma.

Lemma 2: The parity check sub-symbols $p_j^{\mathrm{I}}[t]$ for $t \in \{i+T_2-B_2, \ldots, i+T_1-1\}$ and $j \in \{k, \ldots, B_1 - 1\}$ can be recovered using the parity check symbols $\mathbf{p}^3[\cdot]$ in the interval $[i, \ldots, i+T_2 - B_2 - 1]$ (in layer (4)) by time t, i.e., with a zero delay.

Since the proof of Lemma 2 is rather long it is deferred to Appendix. D.

Step (2) (Removal of $\mathbf{p}^{\mathrm{I}}[\cdot]$): Next we show that the parity check sub-symbols of $\mathcal{C}_2$ in layer (3) are free of interference starting at $t_3 = i + B_1 - k$. This is because the parity check sub-symbols of $\mathcal{C}_1$ in the interval $[i + B_1 - k, i + T_1 - 1] = [t_3, t_4 - 1]$ are recovered in Step (1) and those appearing at time $i + T_1$ and later are functions of unerased source symbols at times i and later (this follows from the fact that a (B, T) SCo code has a memory of T).

Step (3) (Removal of $\mathbf{p}^3[\cdot]$): We next claim that the rest of parity check sub-symbols of $\mathcal{C}_2$ in layer (4) are also free of interference. In case (A) considered before, this follows immediately from the memory of the SCo code. The parity check sub-symbols of $\mathcal{C}_3$ has a memory of $T_3 = B_1 - k$ and thus these parity checks at time $i + B_1 - k$ combine parity check sub-symbols of $\mathcal{C}_1$ of time $i + B_1 - k - T_3 + T_1 = i + T_1$ and later (where the addition of T_1 is due to the shift back applied on these parity checks). Moreover, we have shown in Step (2) that the parity check sub-symbols of $\mathcal{C}_1$ at time $i + T_1$ and later combines only unerased source symbols from time i and the claim follows. While for case (B), the same argument follows in the last q rows. But for the first $r(B_1 - k)$ rows of layer (4), the parity checks of $\mathcal{C}_3$ are repetition codes. These are either recovered in Step (2) or contain only unerased source symbols.

Step (4) (Recovery using $\mathbf{p}^{\mathrm{II}}[\cdot]$): Step (4) uses the previous two steps to recover the parity check sub-symbols of $\mathcal{C}_2$ in layers (3) and (4) starting at $\mathbf{p}^{\mathrm{II}}[i + B_1 - k] = \mathbf{s}[i + B_1 - k - T_2] = \mathbf{s}[i - B_2]$ and thus the erased source-symbols can be recovered.

IX. The Converse For Region (e) (Theorem 5)

We want to prove that the capacity is at most $\frac{T_1}{2T_1 + B_1 + B_2 - T_2}$ in the (e)-region defined by the inequalities $B_2 \leq T_2 < B_2 + B_1$ and $T_2 \geq T_1 + B_1$.

We start by considering the example $\{(4, 5) - (7, 10)\}$ illustrating the steps of the converse proof. We again use the periodic erasure channel strategy with a period of length 12 and the first 7 of which are erased. With 7 erasures, code $\mathcal{C}_2 = (7, 10)$ can recover the first two symbols at time 0 and 1 by time 10 and 11, respectively (c.f. Fig. 9(b)). Since code $\mathcal{C}_1 = (4, 5)$ is not capable of recovering the remaining 5 erasures, we reveal the first of which to the decoder. Now, $\mathcal{C}_1$ can recover the source symbols at times 3 to 6 by times 8 to 11, respectively (i.e., incurring a delay of 5 symbols). Again with the assumption of systematic encoding, one can see that a rate of $5/11$ upper-bounds the capacity of this channel as 5 channel symbols where able to decode 6 of the erased source symbols.

For the general case, the periodic erasure channel to be used is shown in Fig. 10, where each period
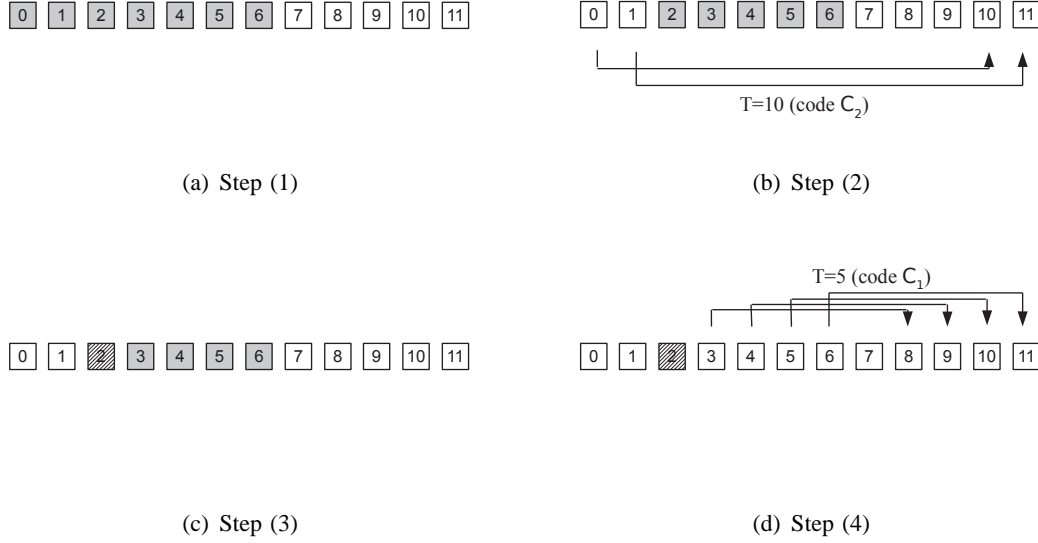
(a) Step (1)

(b) Step (2)

(c) Step (3)

(d) Step (4)

Fig. 9. Main steps of finding the upper-bound for the $\{(4,5) - (7,10)\}$ point lying in Region (e) through one period illustration of the Periodic Erasure Channel. Grey and white squares resemble erased and unerased symbols respectively while hatched squares resemble symbols revealed to the receiver.

has B_2 erasures followed by T_1 non-erasures. We can assign

$$a = T_1 + B_2 - T_2, \quad b = B_2 - B_1, \quad c = B_2, \quad d = B_2 + T_1 \quad \text{(period length)},$$
$$W_i = \mathbf{x}\begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix}, \quad V_i = \mathbf{s}\begin{bmatrix} id+a-1 \\ id \end{bmatrix} \mathbf{s}\begin{bmatrix} (i+1)d-1 \\ id+b \end{bmatrix}.$$

The idea behind the converse proof is similar to before, but instead we have two decoding functions to use.

We use the decoder of receiver 2 to recover $\mathbf{s}\begin{bmatrix} a-1 \\ 0 \end{bmatrix}$ within a delay of T_2 using the channel packets $\mathbf{x}\begin{bmatrix} d-1 \\ c \end{bmatrix}$. We then reveal the channel symbols $\mathbf{x}\begin{bmatrix} b-1 \\ a \end{bmatrix}$. The decoder of receiver 1 can now be used to recover the next B_1 source packets, which are the packets $\mathbf{s}\begin{bmatrix} c-1 \\ b \end{bmatrix}$, using $\mathbf{x}\begin{bmatrix} d-1 \\ c \end{bmatrix}$ again. In general, we may not have a systematic code, so even if $\mathbf{x}\begin{bmatrix} d-1 \\ c \end{bmatrix}$ is received, we may not be able to recover the corresponding source packet $\mathbf{s}\begin{bmatrix} d-1 \\ c \end{bmatrix}$. Instead, $\mathbf{s}\begin{bmatrix} d-1 \\ c \end{bmatrix}$ can be recovered using the second decoder and the first and second sets of channel packets that are not erased, i.e. $\mathbf{x}\begin{bmatrix} d-1 \\ c \end{bmatrix}$ and $\mathbf{x}\begin{bmatrix} 2d-1 \\ d+c \end{bmatrix}$.

So far, we have recovered $(T_1 + B_2 - T_2) + B_1 + T_1 = 2T_1 + B_1 + B_2 - T_2$ source packets, using $2T_1$ channel packets. We do not include the source packets $\mathbf{s}\begin{bmatrix} b-1 \\ a \end{bmatrix}$, because it cannot be decoded from the information in the unerased channel packets. The channel has a period of $B_2 + T_1$ packets, and if we had n periods, then we would be able to recover $n(2T_1 + B_1 + B_2 - T_2)$ source packets using $(n+1)T_1$ channel packets. Therefore, we can suppose that the upper bound on the multicast streaming capacity is

given by

$$n \cdot (2T_1 + B_1 + B_2 - T_2) \cdot H(\mathbf{s}) \leq (n+1) \cdot T_1 \cdot H(\mathbf{x})$$

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{n+1}{n} \cdot \frac{T_1}{2T_1 + B_1 + B_2 - T_2}$$

$$\xrightarrow{n \to \infty} \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \tag{64}$$
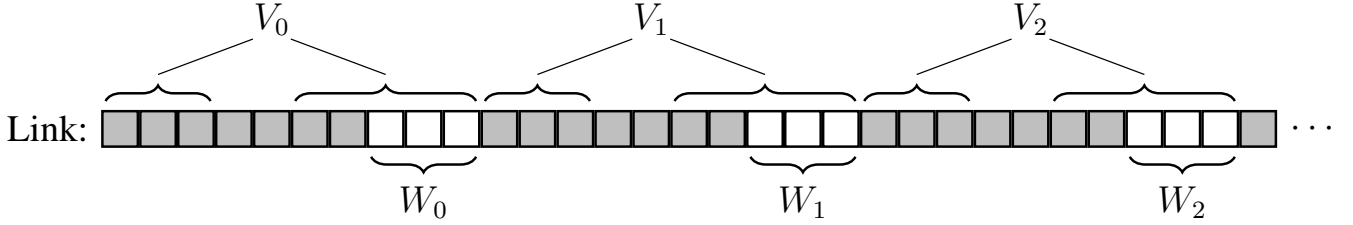
The more formal proof is given below.

Fig. 10. The periodic erasure channel used to prove an upper bound on capacity in region (e) indicating which symbols are in groups W_i and V_i. Grey and white squares resemble erased and unerased symbols respectively.
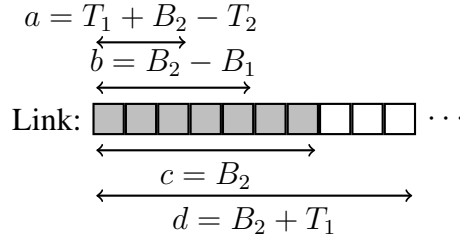
Fig. 11. One period of the periodic erasure channel in Fig. 10, with labels.

Proof: From the (B_1, T_1) decoder, we have for $i \geq 0$:

$$H\left(\mathbf{s}\begin{bmatrix} id+c-1 \\ id+b \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} id+b-1 \\ 0 \end{bmatrix} W_i\right) = 0 \tag{65}$$

From the (B_2, T_2) decoder, we have for $i \geq 0$:

$$H\left(\mathbf{s}\begin{bmatrix} id+a-1 \\ id \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} id-1 \\ 0 \end{bmatrix} W_i\right) = 0 \tag{66}$$

$$H\left(\mathbf{s}\begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} id+c-1 \\ 0 \end{bmatrix} W_i^{i+1}\right) = 0. \tag{67}$$

We want to use mathematical induction to prove that for $n \geq 0$

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n \middle| V_0^{n-1}\mathbf{x}\begin{bmatrix} nd-1 \\ 0 \end{bmatrix}\right). \tag{68}$$

The base case for (68) is given by substituting $n = 0$ into it:

$$H(W_0) \geq H(V_0^{-1}) + H\left(W_0 \middle| V_0^{-1}\mathbf{x}\begin{bmatrix} -1 \\ 0 \end{bmatrix}\right) \geq H(W_0) \tag{69}$$

which is obviously true. Let us assume that (68) is true for $n = k$. This gives:

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k \middle| V_0^{k-1}\mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right). \tag{70}$$

We can manipulate the expression in two parts. In the first part, we use W_k to recover the source packets

$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
$s_0[i]$	$s_0[i+1]$	$s_0[i+2]$	$s_0[i+3]$	$s_0[i+4]$	$s_0[i+5]$
$s_1[i]$	$s_1[i+1]$	$s_1[i+2]$	$s_1[i+3]$	$s_1[i+4]$	$s_1[i+5]$
$s_0[i-4]$	$s_0[i-3]$	$s_0[i-2]$	$s_0[i-1]$	$s_0[i]$	$s_0[i+1]$
$s_1[i-4]$	$s_1[i-3]$	$s_1[i-2]$	$s_1[i-1]$	$s_1[i]$	$s_1[i+1]$
$s_0[i-6]+s_1[i-5]$	$s_0[i-5]+s_1[i-4]$	$s_0[i-4]+s_1[i-3]$	$s_0[i-3]+s_1[i-2]$	$s_0[i-2]+s_1[i-1]$	$s_0[i-1]+s_1[i]$

TABLE IV

MU-SCO CODE CONSTRUCTION FOR $(B_1, T_1) = (4, 4)$ AND $(B_2, T_2) = (5, 6)$. THIS POINT ACHIEVES THE UPPER-BOUND GIVEN IN THEOREM 6 AS $T_1 = B_1 = 4$.

$\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}$ and $\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}$ and one can write,

$$H(W_0^k) \geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(W_k \Big| V_0^{k-1}\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right), \qquad (71)$$

where the first term on the R.H.S. gives the entropy of the source symbols recovered in previous periods V_0^{k-1} as well as the source symbols recovered in this step due to the availability of W_k. The second term gives the remaining ambiguity in W_k to be used in the next step. The detailed steps from (70) to (71) is shown in Appendix. E.

In the second part, we add $H(W_{k+1}|W_0^k)$ to both sides of the inequality. Because the channel code is not necessarily systematic, we will use the additional channel packets in W_{k+1} to help decode the source packets $\mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+c \end{bmatrix}$. The corresponding steps provided in Appendix. E shows that,

$$H(W_0^{k+1}) \geq H(V_0^k) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix} (k+1)d-1 \\ 0 \end{bmatrix}\right) \qquad (72)$$

The working in (113) shows that if (68) is true for $n = k$, then it is also true for $n = k + 1$. By induction, (68) is true for $n \geq 0$. Finally,

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n\Big|V_0^{n-1}\mathbf{x}\begin{bmatrix} nd-1 \\ 0 \end{bmatrix}\right) \geq H(V_0^{n-1}).$$

Using the fact that all of the channel packets have the same entropy, and all of the source packets have the same entropy, we can continue to get

$$H(W_0^n) \geq H(V_0^{n-1})$$
$$(n + 1) \cdot T_1 \cdot H(\mathbf{x}) \geq n \cdot (2T_1 + B_1 + B_2 - T_2) \cdot H(\mathbf{s})$$
$$\frac{n+1}{n} \cdot \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \geq \frac{H(\mathbf{s})}{H(\mathbf{x})}. \qquad (73)$$

Finally, we get

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T_1}{2T_1 + B_1 + B_2 - T_2}. \text{ (as } n \to \infty) \qquad (74)$$

Therefore, any $(B_1, T_1), (B_2, T_2)$ code in the (e)-region must satisfy (74). ■

X. ACHIEVABILITY SCHEME IN REGION (F) AT $T_1 = B_1$ (THEOREM 6)

We begin with an example of $\{(4, 4) - (5, 6)\}$ Mu-SCo construction of rate $2/5$, as shown in Table IV. A $(4, 4)$ SCo repetition code is then applied resulting in the first two rows of parity checks and then a $(B_2 - B_1, T_2 - T_1) = (1, 2)$ SCo is applied and the resulting parity checks are shifted by $T_1 = 4$ forming the last row. Note that the first user can recover from any burst erasure of length 4 within a delay of 4 symbols using the first two rows of parity check sub-symbols. For the second user, assume a burst

erasure of length 5 takes place from time $i-5$ to $i-1$. Notice that user 2 recovers $s_1[i-5]$ and $s_0[i-5]$ respectively from the last two parity checks at time $t = i+1$ i.e., with a delay of $T_2 = 6$. The rest of the erased source symbols are recovered with a delay of $T_1 = 4$ using the repetition code.

A. Code Construction

Our proposed code construction, which achieves the minimum delay for user 1 i.e., $T_1 = B_1$ is as folows

- Let $\mathcal{C}_1$ be the single user $(B_1, T_1) = (T_1, T_1)$ SCo obtained by splitting each source symbol $\mathbf{s}[i]$ into $(T_2 - B_1) = (T_2 - T_1)$ sub-symbols

$$\mathbf{s}[i] = (s_0[i], \dots, s_{T_2-T_1-1}[i])$$

and repeating them to produce $(T_2 - T_1)$ parity check sub-symbols.

$$\mathbf{p}^{\mathrm{I}}[i] = (p_0^{\mathrm{I}}[i], \dots, p_{T_2-T_1-1}^{\mathrm{I}}[i]) = (s_0[i - T_1], \dots, s_{T_2-T_1-1}[i - T_1]) = \mathbf{s}[i - T_1]. \tag{75}$$

- Let $\mathcal{C}_2$ be a $(B_2 - B_1, T_2 - T_1)$ SCo also obtained by splitting each source symbol $\mathbf{s}[i]$ into $(T_2 - B_1) = (T_2 - T_1)$ sub-symbols $(s_0[i], \dots, s_{T_2-T_1-1}[i])$ and then constructing $(B_2 - B_1)$ parity checks $\mathbf{p}^{\mathrm{II}}[i] = (p_0^{\mathrm{II}}[i], \dots, p_{B_2-B_1-1}^{\mathrm{II}}[i])$ at each time by combining the source sub-symbols along the main diagonal.
- Concatenate the two streams $p^{\mathrm{I}}[\cdot]$ and $p^{\mathrm{II}}[\cdot]$ after introducing a shift of T_1 in the second stream. The output symbol at time i is $\mathbf{x}[i] = (\mathbf{s}[i], \mathbf{p}^{\mathrm{I}}[i], \mathbf{p}^{\mathrm{II}}[i - T_1])$.

Since there are $T_2 - T_1$ and $B_2 - B_1$ parity check sub-symbols for every $T_2 - T_1$ source sub-symbols, it follows that the rate of the code is $\frac{T_2-T_1}{2(T_2-T_1)+(B_2-B_1)} = C_f^+$.

B. Decoding at User 1

A burst erasure of length B_1 can be directly recovered using the stream of parity checks $p^{\mathrm{I}}[\cdot]$ produced by code $\mathcal{C}_1$ within a delay of T_1. Recall that this immediately follows since the parity checks of the two codes are concatenated and not added.

C. Decoding at User 2

Suppose that the symbols at time $i - B_2, \dots, i - 1$ are erased by the channel of user 2. We first show how the receiver can recover $\mathbf{s}[t]$ for $t \in [i - B_2, i - B_1 - 1]$ at time $t + T_2$. To recover $\mathbf{s}[t]$, the code $\mathcal{C}_2$ which is a $(T_2 - T_1, B_2 - B_1)$ code, can be used provided that the corresponding parity checks starting at time $i - B_1$ are available. Due to the forward shift of $T_1 = B_1$ applied in our construction, these parity checks appear starting at time $t = i$ and are clearly not erased. Secondly for the recovery of $\mathbf{s}[t]$ we also need the source symbols in the interval $[i - B_1, t + T_2 - T_1]$. The $\mathcal{C}_1$ repetition code guarantees that these are in fact available by time $t + T_2$. This shows that all the erased symbols in the interval $[i - B_2, i - B_1 - 1]$ can be recovered. The remaining symbols in the interval $[i - B_1, i - 1]$ are recovered using the $\mathcal{C}_1$ repetition code.

XI. Upper-Bound For Region (f) (Theorem 6)

The converse proof for region (f) is similar to the proof for region (e). We shall use Fig. 12 and 13 to illustrate the periodic erasure channel used in this proof. Each period, in this case, contains B_2 erasures followed by $T_2 - B_1$ non-erasures, for a total of $B_2 + T_2 - B_1$ symbols.

The first $B_2 - B_1$ source symbols can be recovered with code $\mathcal{C}_2$, from $\mathbf{x}\left[\begin{smallmatrix} B_2+T_2-B_1 \\ B_2 \end{smallmatrix}\right]$, which are the $T_2 - B_1$ unerased channel symbols. We can see that s_0 is recovered at time T_2, while $s_{B_2-B_1-1}$ is recovered at time $B_2 + T_2 - B_1 - 1$. Code $\mathcal{C}_1$ recovers the next $T_2 - T_1$ source symbols, which is $\mathbf{s}\left[\begin{smallmatrix} B_2-B_1+T_2-T_1-1 \\ B_2-B_1 \end{smallmatrix}\right]$.
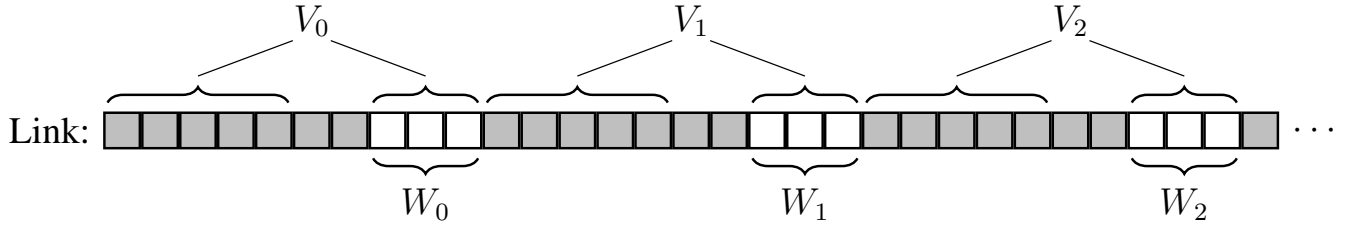
Fig. 12.   The periodic erasure channel used to prove the first upper bound in region (f) showing the locations of the symbols in groups V_i and W_i. Grey and white squares resemble erased and unerased symbols respectively.
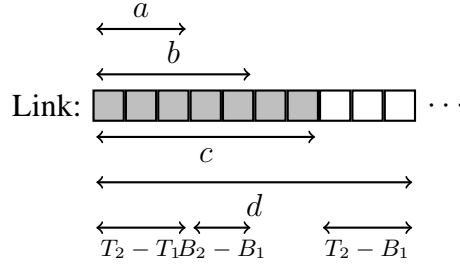

Fig. 13.   One period of the periodic erasure channel in Fig. 12, with labels

We then reveal the remaining channel symbols in the block of B_2 erased symbols, which are the symbols $\mathbf{x}\begin{bmatrix} B_2-1 \\ B_2-B_1+T_2-T_1 \end{bmatrix}$. Finally, code $\mathcal{C}_2$ is used to recover $\mathbf{s}\begin{bmatrix} B_2+T_2-B_1-1 \\ B_2 \end{bmatrix}$, using two sets of $T_2 - B_1$ unerased channel symbols, which are $\mathbf{x}\begin{bmatrix} B_2+T_2-B_1-1 \\ B_2 \end{bmatrix}$ and $\mathbf{x}\begin{bmatrix} 2B_2+2T_2-2B_1-1 \\ 2B_2+T_2-B_1 \end{bmatrix}$.

In this one period of $B_2 + T_2 - B_1$ symbols, we have recovered $\mathbf{s}\begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix}$, $\mathbf{s}\begin{bmatrix} B_2-B_1+T_2-T_1-1 \\ B_2-B_1 \end{bmatrix}$ and $\mathbf{s}\begin{bmatrix} B_2+T_2-B_1 \\ B_2 \end{bmatrix}$. This is a total of $2(T_2 - B_1) + (B_2 - T_1)$ source symbols recovered by $2(T_2 - B_1)$ channel symbols. We can extrapolate that $n(2(T_2 - B_1) + (B_2 - T_1))$ source symbols can be recovered by $(n + 1)(T_2 - B_1)$ channel symbols. As in region (d) proof, we can suppose that the upper bound on the capacity is:

$$n(2(T_2 - B_1) + (B_2 - T_1)H(\mathbf{s}) \le (n+1)(T_2 - B_1)H(\mathbf{x})$$
$$\frac{H(\mathbf{s})}{H(\mathbf{x})} \le \frac{(n+1)(T_2 - B_1)}{n(2(T_2 - B_1) + (B_2 - T_1))}$$
$$\xrightarrow{n \to \infty} \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)}$$
$$\therefore C_f^+ = \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)} \tag{76}$$

For the formal proof, we assign the following:

$$a = B_2 - B_1, \quad b = B_2 - B_1 + T_2 - T_1, \quad c = B_2, \quad d = B_2 + T_2 - B_1 \quad \text{(period length)},$$
$$W_i = \mathbf{x}\begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix}, \quad V_i = \mathbf{s}\begin{bmatrix} id+b-1 \\ id \end{bmatrix}\mathbf{s}\begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix}.$$

From code $\mathcal{C}_1$, we have for $i \ge 0$:

$$H\left(\mathbf{s}\begin{bmatrix} id+b-1 \\ id+a \end{bmatrix}\middle|\mathbf{x}\begin{bmatrix} id+a-1 \\ 0 \end{bmatrix}W_i\right) = 0 \tag{77}$$

From code $\mathcal{C}_2$, we have for $i \geq 0$:

$$H\left(\mathbf{s}\begin{bmatrix} id+a-1 \\ id \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} id-1 \\ 0 \end{bmatrix} W_i\right) = 0 \tag{78}$$

$$H\left(\mathbf{s}\begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} id+c-1 \\ 0 \end{bmatrix} W_i^{i+1}\right) = 0. \tag{79}$$

We want to show, using mathematical induction, that for $n \geq 0$

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n \middle| V_0^{n-1}\mathbf{x}\begin{bmatrix} nd-1 \\ 0 \end{bmatrix}\right). \tag{80}$$

The base case for (80) is given by:

$$H(W_0) = H(V_0^{-1}) + H\left(W_0 \middle| V_0^{-1}\mathbf{x}\begin{bmatrix} -1 \\ 0 \end{bmatrix}\right) \geq H(W_0) \tag{81}$$

which is true. For the induction step, we assume (80) is true for $n = k$,

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n \middle| V_0^{n-1}\mathbf{x}\begin{bmatrix} nd-1 \\ 0 \end{bmatrix}\right). \tag{82}$$

The second term of the R.H.S. can be used to recover $\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}$ and $\mathbf{s}\begin{bmatrix} kd+b-1 \\ kd+a \end{bmatrix}$ through codes $\mathcal{C}_2$ and $\mathcal{C}_1$, respectively. The corresponding entropy manipulations are provided in Appendix. F and the following is deduced,

$$H(W_0^k) \geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix} kd+b-1 \\ kd \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1}\mathbf{s}\begin{bmatrix} kd+b-1 \\ kd \end{bmatrix}\mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right) \tag{83}$$

Next, we add $H(W_{k+1}|W_0^k)$ to both sides and show that the newly added W_{k+1} is capable of recovering the source symbols $\mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+c \end{bmatrix}$ corresponding to W_k,

$$H(W_0^{k+1}) \geq H(V_0^k) + H\left(W_{k+1} \middle| V_0^k \mathbf{x}\begin{bmatrix} (k+1)d-1 \\ 0 \end{bmatrix}\right) \tag{84}$$

The working out of (84) is provided in Appendix. F.

The working in (83) and (84) shows that if (80) is true for $n = k$, then it is true for $n = k + 1$. By induction, (80) is true for $n \geq 0$. Therefore,

$$H(W_0^n) \geq H(V_0^{n-1}) + H\left(W_n \middle| V_0^{n-1}\mathbf{x}\begin{bmatrix} nd-1 \\ 0 \end{bmatrix}\right) \geq H(V_0^{n-1}).$$

We can use the fact that the source symbols have the same entropy and the same for channel symbols to obtain:

$$H(W_0^n) \geq H(V_0^{n-1})$$
$$(n+1) \cdot (T_2 - B_1) \cdot H(\mathbf{x}) \geq n \cdot (2(T_2 - B_1) + (B_2 - T_1)) \cdot H(\mathbf{s}) \tag{85}$$

In other words,

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)}. \text{ (as } n \to \infty) \tag{86}$$

Therefore, (86) governs any $\{(B_1, T_1), (B_2, T_2)\}$ code in the (f)-region.

XII. Code Construction For Region (f) At $T_2 = B_2$ (Theorem 7)

We simply use a concatenation of two codes — one for user 1 and one for user 2. In particular, we divide each source symbol into T_1 sub-symbols, apply a (B_1, T_1) SCo to get B_1 parity check sub-symbols, apply the (T_2, T_2) SCo which is just a repetition code resulting in T_1 parity check sub-symbols and finally

$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
$s_0[i]$	$s_0[i+1]$	$s_0[i+2]$	$s_0[i+3]$	$s_0[i+4]$	$s_0[i+5]$
$s_1[i]$	$s_1[i+1]$	$s_1[i+2]$	$s_1[i+3]$	$s_1[i+4]$	$s_1[i+5]$
$s_2[i]$	$s_2[i+1]$	$s_2[i+2]$	$s_2[i+3]$	$s_2[i+4]$	$s_2[i+5]$
$s_0[i-3]+s_2[i-1]$	$s_0[i-2]+s_2[i]$	$s_0[i-1]+s_2[i+1]$	$s_0[i]+s_2[i+2]$	$s_0[i+1]+s_2[i+3]$	$s_0[i+2]+s_2[i+4]$
$s_1[i-3]+s_2[i-2]$	$s_1[i-2]+s_2[i-1]$	$s_1[i-1]+s_2[i]$	$s_1[i]+s_2[i+1]$	$s_1[i+1]+s_2[i+2]$	$s_1[i+2]+s_2[i+3]$
$s_0[i-4]$	$s_0[i-3]$	$s_0[i-2]$	$s_0[i-1]$	$s_0[i]$	$s_0[i+1]$
$s_1[i-4]$	$s_1[i-3]$	$s_1[i-2]$	$s_1[i-1]$	$s_1[i]$	$s_1[i+1]$
$s_2[i-4]$	$s_2[i-3]$	$s_2[i-2]$	$s_2[i-1]$	$s_2[i]$	$s_2[i+1]$

TABLE V

MU-SCO CODE CONSTRUCTION FOR $(B_1, T_1) = (2, 3)$ AND $(B_2, T_2) = (4, 4)$. THE RATE OF $3/8$ OF SUCH MU-SCO IS THE CAPACITY GIVEN IN THEOREM 7 FOR $T_2 = B_2$ CASE IN REGION (F).

concatenate them to have $B_1 + T_1$ parity check sub-symbols for each T_1 source sub-symbol (i.e., a rate of $\frac{T_1}{2T_1 + B_1} = C_{f(T_2 = B_2)}$).

Consider the example of $\{(2, 3) - (4, 4)\}$ code in Table V. Each source symbols is divided into $T_1 = 3$ sub-symbols. A $(B_1, T_1) = (2, 3)$ SCo is applied to generate the first two rows of parity check sub-symbols which are concatenated to three more rows of parity check sub-symbols generated by the $(B_2, T_2) = (4, 4)$ repetition code. User 1 and 2 can recover from bursts of length 2 and 4 within delays of 3 and 4 respectively by considering the corresponding rows of parity checks while neglecting the other rows.

XIII. THE CONVERSE FOR REGION (F) AT $T_2 = B_2$ (THEOREM 7)

The converse for Theorem 7 depends on double recovery of some source symbols, once using code $\mathcal{C}_1$ and another using $\mathcal{C}_2$. We illustrate the main idea of such converse through considering the specific point $\{(2, 3) - (4, 4)\}$ shown in Fig. 14. We start by considering a periodic erasure channel with period length 7. The first 4 symbols are erased while the rest are unerased. With 4 erasures, code $\mathcal{C}_2 = (4, 4)$ can recover the first two symbols at time 0 and 1 by time 4 and 5, respectively. We note that the channel symbol at time i is sufficient to recover the source symbol at time $i-4$ (i.e., no more channel symbols are required). In step (3) in Fig. 14 gives the main idea of this converse. Since, there are two remaining erasures, the source symbol at time 2 can be recovered using $\mathcal{C}_1 = (2, 3)$ within a delay of 3 (i.e., by time 5). Also, the same source symbol can be decoded using $\mathcal{C}_2$ by time 6 (double recovery). The remaining erasure can be recovered using $\mathcal{C}_1$ by time 6. Moreover, the repetition code $\mathcal{C}_2 = (4, 4)$ can recover the source symbols at time 4, 5 and 6 from their corresponding channel symbols. Therefore, the three channel symbols are capable of recovering a total of 8 source symbols (symbol at time 2 is recovered twice) which implies that a rate of $3/8$ is an upper-bound.

For the general case, the corresponding periodic erasure channel to be used for proving the upper-bound is given in Figure 15. Each period has B_2 erasures followed by T_1 non-erasures.

It so happens that the $B_2 = T_2$ restriction means that we can prove the converse by only analyzing one period. The reason will be made clear later. But this simplifies the proof and allows us to study the technique of double counting source packets more easily.

In Figure 16, we have the first period of the erasure channel. The key is to show that the received channel packets $\mathbf{x}\begin{bmatrix} B_2 + T_1 - 1 \\ B_2 \end{bmatrix}$ alone can recover all of the source packets in the period, but there is enough information in the channel packets to recover some of the source packets twice. The fact that we have two decoders allows some of the source packets to be decoded by mutually exclusive groups of channel packets, but when we put all of the channel packets together, the redundant information in the channel packets does affect the maximum achievable rate of the code.

The source packets that can be recovered by $\mathbf{x}\begin{bmatrix} B_2 + T_1 - 1 \\ B_2 \end{bmatrix}$ are $\mathbf{s}\begin{bmatrix} T_1 - 1 \\ 0 \end{bmatrix}$, $\mathbf{s}\begin{bmatrix} B_2 - 1 \\ B_2 - B_1 \end{bmatrix}$ and $\mathbf{s}\begin{bmatrix} B_2 + T_1 - 1 \\ B_2 \end{bmatrix}$. As Figure 16 shows, the first two groups of source packets overlap. The overlap consists of the packets

(a) Step (1)

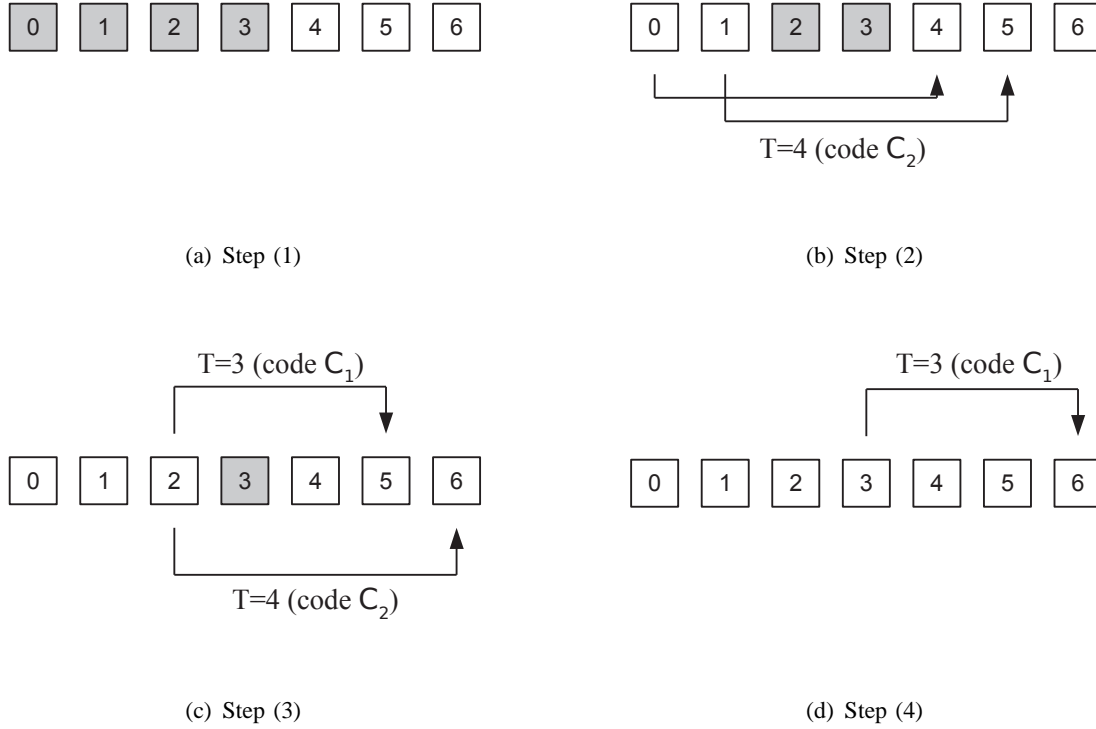(b) Step (2)

(c) Step (3)

(d) Step (4)

Fig. 14. Main steps of finding the upper-bound for the $\{(2,3) - (4,4)\}$ point lying in Region (f) through one period illustration of the Periodic Erasure Channel. Grey and white squares resemble erased and unerased symbols respectively.
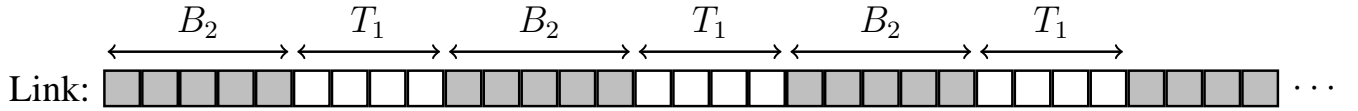

Fig. 15. The periodic erasure channel used to prove an upper bound on capacity in region (f) for the special case $T_2 = B_2$.

$\mathbf{s}\begin{bmatrix} T_1-1 \\ B_2-B_1 \end{bmatrix}$. The reason why we can use a single period in the proof is because the $B_2 = T_2$ constraint allows us to decode the final group of source packets $\mathbf{s}\begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ using only the packets $\mathbf{x}\begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ and does not require any future channel packets.

Assuming that what we have just described is possible, then we have T_1 channel packets recovered
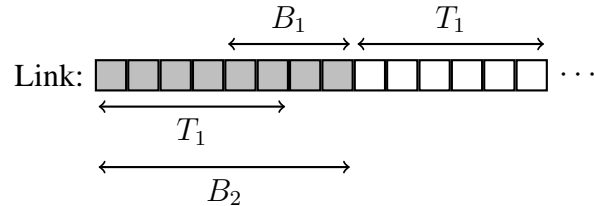
Fig. 16. One period of the periodic erasure channel in Fig. 15, with labels.

$2T_1 + B_1$ source packets. We should be able to write the relation:

$$(2T_1 + B_1) \cdot H(\mathbf{s}) \leq T_1 \cdot H(\mathbf{x})$$
$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T_1}{2T_1 + B_1} \tag{87}$$

The formal proof shows that this is indeed possible.

Proof: We can split the proof into three major parts.

1. The source packets $\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}$ can be recovered from the channel packets $\mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix}$ using the (B_2, B_2) decoder, so we can write

$$H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix}\right) = 0. \tag{88}$$

Next, we can write

$$H\left(\mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix}\right) = H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix}\right) - H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix}\right)$$
$$\overset{(a)}{=} H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix}\right)$$
$$= H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right)$$
$$\geq H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \mathbf{x}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right). \tag{89}$$

We used (88) to remove the negative term before step (a).

2. In this step, we want to prove the following inequality for $m \geq B_2 + T_1 - B_1 - 1$:

$$\sum_{i=B_2}^{m} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} m-B_2 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} m-T_1 \\ B_2-B_1 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} m \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} m-B_2 \\ 0 \end{bmatrix} \mathbf{s}\begin{bmatrix} m-T_1 \\ B_2-B_1 \end{bmatrix} \mathbf{x}\begin{bmatrix} m-B_2 \\ 0 \end{bmatrix}\right) \tag{90}$$

Using the first decoder with a (B_1, T_1) property, we can write the following relation:

$$H\left(\mathbf{s}[i-T_1] \middle| \mathbf{x}\begin{bmatrix} i \\ i-T_1+B_1 \end{bmatrix} \mathbf{x}\begin{bmatrix} i-T_1-1 \\ 0 \end{bmatrix}\right) = 0. \tag{91}$$

Using the (B_2, B_2) decoder, we can write the following relation:

$$H\left(\mathbf{s}[i-B_2] \middle| \mathbf{x}[i]\mathbf{x}\begin{bmatrix} i-B_2-1 \\ 0 \end{bmatrix}\right) = 0 \tag{92}$$

which can be used in the following steps

$$H\left(\mathbf{x}[i] \middle| \mathbf{x}\begin{bmatrix} i-B_2-1 \\ 0 \end{bmatrix}\right) = H\left(\mathbf{s}[i-B_2]\mathbf{x}[i] \middle| \mathbf{x}\begin{bmatrix} i-B_2-1 \\ 0 \end{bmatrix}\right) - H\left(\mathbf{s}[i-B_2] \middle| \mathbf{x}[i]\mathbf{x}\begin{bmatrix} i-B_2-1 \\ 0 \end{bmatrix}\right)$$
$$\overset{(a)}{=} H\left(\mathbf{s}[i-B_2]\mathbf{x}[i] \middle| \mathbf{x}\begin{bmatrix} i-B_2-1 \\ 0 \end{bmatrix}\right)$$
$$= H(\mathbf{s}[i-B_2]) + H\left(\mathbf{x}[i] \middle| \mathbf{s}[i-B_2]\mathbf{x}\begin{bmatrix} i-B_2-1 \\ 0 \end{bmatrix}\right).$$

Therefore,

$$H(\mathbf{x}[i]) \geq H(\mathbf{s}[i-B_2]) + H\left(\mathbf{x}[i] \middle| \mathbf{s}[i-B_2]\mathbf{x}\begin{bmatrix} i-B_2 \\ 0 \end{bmatrix}\right). \tag{93}$$

The second decoder (92) was used to remove the negative term before step (a).

Now we can use mathematical induction to prove (90). For the base case, we substitute $m = B_2 + T_1 -$

$B_1 - 1$

$$\sum_{i=B_2}^{B_2+T_1-B_1-1} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} B_2-B_1-1 \\ B_2-B_1 \end{bmatrix}\right)$$

$$+ H\left(\mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \mathbf{s}\begin{bmatrix} B_2-B_1-1 \\ B_2-B_1 \end{bmatrix} \mathbf{x}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right)$$

$$= H\left(\mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} B_2+T_1-B_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix} \mathbf{x}\begin{bmatrix} T_1-B_1-1 \\ 0 \end{bmatrix}\right). \quad (94)$$

This is proved by the result of (89).

Assume that (90) is true for $m = j$, which gives us

$$\sum_{i=B_2}^{j} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} j-B_2 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} j-T_1 \\ B_2-B_1 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} j \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} j-B_2 \\ 0 \end{bmatrix} \mathbf{s}\begin{bmatrix} j-T_1 \\ B_2-B_1 \end{bmatrix} \mathbf{x}\begin{bmatrix} j-B_2 \\ 0 \end{bmatrix}\right). \quad (95)$$

We add $H(\mathbf{x}[j+1])$ to both sides, and use (91) and (92) to recover the source symbols $\mathbf{s}[j+1-B_2]$ and $\mathbf{s}[j+1-T_1]$ respectively giving:

$$\sum_{i=B_2}^{j+1} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} j+1-B_2 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} j+1-T_1 \\ B_2-B_1 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} j+1 \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} j+1-B_2 \\ 0 \end{bmatrix} \mathbf{s}\begin{bmatrix} j+1-T_1 \\ B_2-B_1 \end{bmatrix} \mathbf{x}\begin{bmatrix} j+1-B_2 \\ 0 \end{bmatrix}\right). \quad (96)$$

The corresponding detailed steps are provided in Appendix. G. By induction, we have proved (90) for $m \geq B_2 + T_1 - B_1 - 1$.

3. We substitute $m = B_2 + T_1 - 1$ into (90)

$$\sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} T_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} B_2-1 \\ B_2-B_1 \end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s}\begin{bmatrix} T_1-1 \\ 0 \end{bmatrix} \mathbf{s}\begin{bmatrix} B_2-1 \\ B_2-B_1 \end{bmatrix} \mathbf{x}\begin{bmatrix} T_1-1 \\ 0 \end{bmatrix}\right). \quad (97)$$

We can recover $\mathbf{s}\begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ from $\mathbf{x}\begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ given the previous channel symbols $\mathbf{x}\begin{bmatrix} B_2-1 \\ 0 \end{bmatrix}$ using decoder 2, so we can write

$$H\left(\mathbf{s}\begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{x}\begin{bmatrix} B_2+T_1-1 \\ 0 \end{bmatrix}\right) = 0. \quad (98)$$

Using (98), we continue with (97) to get (c.f. Appendix. G):

$$\sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} T_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} B_2+T_1-1 \\ B_2-B_1 \end{bmatrix}\right) \quad (99)$$

Finally, we use the fact that all source symbols have the same entropy and all channel symbols have the same entropy to write,

$$\sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\begin{bmatrix} T_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} B_2+T_1-1 \\ B_2-B_1 \end{bmatrix}\right)$$

$$T_1 \cdot H(\mathbf{x}) \geq (2T_1 + B_1) \cdot H(\mathbf{s})$$

$$R = \frac{H(\mathbf{s})}{H(\mathbf{x})} \leq \frac{T_1}{2T_1 + B_1} \quad (100)$$

which is the proper upper bound. ∎

XIV. CONCLUSION

We study a multicast extension of the low-delay codes for streaming over burst erasure channels. The proposed setup has several interesting implications. From a capacity point of view, we observe an

interesting interplay between the delay of the two receivers. In particular, in the large delay regime we characterize the capacity and observe a surprising delay-slackness property i.e., for most parameters, the delay of one of the receivers can be reduced up to a certain critical value without reducing the capacity. In the low-delay regime the capacity has only been partially characterized. New code constructions are developed for various regimes. Our constructions generate parity checks in multiple layers and carefully combine them to meet the require decoding constraints.

Our ongoing work involves further investigating the capacity in the low-delay regime. Furthermore the results in this work are a step towards developing robust streaming code constructions, which can be used in time-varying channel conditions where the burst-length cannot be determined apriori.

APPENDIX A
PROOF OF (24) AND (25)

The steps to get the result in (24) is as follows: Using (24) we have that

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k \Big| V_0^{k-1} \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right). \tag{101}$$

This can be further simplified as follows.

$$
\begin{aligned}
H(W_0^k) &\geq H(V_0^{k-1}) + H\left(W_k \Big| V_0^{k-1} \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right) \\
&\overset{(a)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] W_k \Big| V_0^{k-1} \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right) \\
&\quad - H\left(\mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \Big| V_0^{k-1} \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right] W_k\right) \\
&\overset{(b)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] W_k \Big| V_0^{k-1} \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right) \\
&\overset{(c)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \Big| V_0^{k-1} \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right) \\
&\quad + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right) \\
&\overset{(d)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \Big| V_0^{k-1}\right) + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]\right) \\
&\overset{(e)}{\geq} H\left(V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right]\right) + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \mathbf{x}\left[\begin{matrix} k(B+T)+B-1 \\ 0 \end{matrix}\right]\right). \tag{102}
\end{aligned}
$$

Step (a) uses the joint entropy expansion formula, step (b) uses (19) to remove the negative term and step (c) is a joint entropy expansion. Step (d) uses the fact that source packets are independent of each other, so therefore the source packets $\mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right]$ must be independent of the past channel packets $\mathbf{x}\left[\begin{matrix} k(B+T)-1 \\ 0 \end{matrix}\right]$. Step (e) joins the first two terms from (d), and also uses the fact that conditioning reduces entropy in the last term and the result in (24) follows.

To get the result in (25), we start by adding $H(W_{k+1}|W_0^k)$ to both sides of (24) to get,

$$
\begin{aligned}
H(W_0^{k+1}) \\
&\geq H\left(V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right]\right) + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right] \mathbf{x}\left[\begin{matrix} k(B+T)+B-1 \\ 0 \end{matrix}\right]\right) + H(W_{k+1}|W_0^k) \\
&\overset{(a)}{\geq} H\left(V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right]\right) + H\left(W_k \Big| R\right) + H\left(W_{k+1} \Big| R \, W_k\right) \\
&\overset{(b)}{\geq} H\left(V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right]\right) + H\left(W_k^{k+1} \Big| R\right) \\
&\overset{(c)}{=} H\left(V_0^{k-1} \mathbf{s}\left[\begin{matrix} k(B+T)+B-1 \\ k(B+T) \end{matrix}\right]\right) + H\left(\mathbf{s}\left[\begin{matrix} (k+1)(B+T)-1 \\ k(B+T)+B \end{matrix}\right] W_k^{k+1} \Big| R\right) - H\left(\mathbf{s}\left[\begin{matrix} (k+1)(B+T)-1 \\ k(B+T)+B \end{matrix}\right] \Big| R \, W_k^{k+1}\right)
\end{aligned}
$$

$$\stackrel{(d)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)(B+T)-1\\k(B+T)+B\end{bmatrix}W_k^{k+1}\Big|R\right)$$

$$\stackrel{(e)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)(B+T)-1\\k(B+T)+B\end{bmatrix}\Big|R\right)$$
$$+ H\left(W_k^{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}(k+1)(B+T)-1\\k(B+T)\end{bmatrix}\mathbf{x}\begin{bmatrix}k(B+T)+B-1\\0\end{bmatrix}\right)$$

$$\stackrel{(f)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)(B+T)-1\\k(B+T)+B\end{bmatrix}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\right)$$
$$+ H\left(W_k^{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}k(B+T)+B-1\\0\end{bmatrix}\right)$$

$$\stackrel{(g)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}(k+1)(B+T)-1\\k(B+T)\end{bmatrix}\right) + H\left(W_k\Big|V_0^k\mathbf{x}\begin{bmatrix}k(B+T)+B-1\\0\end{bmatrix}\right) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}k(B+T)+B-1\\0\end{bmatrix}W_k\right)$$

$$\geq H(V_0^k) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}(k+1)(B+T)-1\\0\end{bmatrix}\right), \tag{103}$$

where $R = V_0^{k-1}\mathbf{s}\begin{bmatrix}k(B+T)+B-1\\k(B+T)\end{bmatrix}\mathbf{x}\begin{bmatrix}k(B+T)+B-1\\0\end{bmatrix}$. Step (a) introduces extra conditions in the final term, so entropy is reduced, step (b) uses the joint entropy formula, step (c) uses the joint entropy expansion formula and step (d) uses (20) to remove the negative term in (c). Step (e) uses the joint entropy formula again to expand the second term of (d) and step (f) uses the fact that source packets are independent of previous channel packets. Step (g) once again uses the joint entropy formula and (25) follows.

APPENDIX B
PROOF OF (45) AND (46)

The working out of (45) is as follows:

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k\Big|V_0^{k-1}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}\right)$$

$$= H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}W_k\Big|V_0^{k-1}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}\right) - H\left(\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\Big|V_0^{k-1}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}W_k\right)$$

$$\stackrel{(a)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}W_k\Big|V_0^{k-1}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}\right)$$

$$= H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\Big|V_0^{k-1}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}\right)$$

$$\stackrel{(b)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc-1\\0\end{bmatrix}\right)$$

$$\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+a-1\\0\end{bmatrix}\right)$$

$$= H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}kc+b-1\\kc+a\end{bmatrix}W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{s}\begin{bmatrix}kc+a-1\\0\end{bmatrix}\right)$$
$$- H\left(\mathbf{s}\begin{bmatrix}kc+b-1\\kc+a\end{bmatrix}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+a-1\\0\end{bmatrix}W_k\right)$$

$$\stackrel{(c)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}kc+b-1\\kc+a\end{bmatrix}W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+a-1\\0\end{bmatrix}\right)$$

$$= H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}kc+b-1\\kc+a\end{bmatrix}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+a-1\\0\end{bmatrix}\right)$$
$$+ H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+a-1\\kc\end{bmatrix}\mathbf{s}\begin{bmatrix}kc+b-1\\kc+a\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+a-1\\0\end{bmatrix}\right)$$

$$\stackrel{(d)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+a-1\\0\end{bmatrix}\right)$$

$$\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \tag{104}$$

We use (40) to remove the negative term before step (a). Similarly, we remove the negative term before step (c) using (41). Steps (b) and (d) use the fact that source packets are independent of each other and of previous channel packets.

While for (46), we start by adding $H(W_{k+1}|W_0^k)$ to both sides of (45) to get,

$$
\begin{aligned}
H(W_0^{k+1}) &\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(W_k\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) + H(W_{k+1}|W_0^k) \\
&\overset{(e)}{\geq} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(W_k^{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \\
&= H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc+b\end{bmatrix}W_k^{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \\
&\quad - H\left(\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc+b\end{bmatrix}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}W_k^{k+1}\right) \\
&\overset{(f)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc+b\end{bmatrix}W_k^{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \\
&= H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc+b\end{bmatrix}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \\
&\quad + H\left(W_k^{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \\
&\overset{(g)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc\end{bmatrix}\right) + H\left(W_k^{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}(k+1)c-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \\
&\geq H(V_0^k) + H\left(W_k^{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}(k+1)c-1\\0\end{bmatrix}\right) \\
&= H(V_0^k) + H\left(W_k\Big|V_0^k\mathbf{x}\begin{bmatrix}(k+1)c-1\\0\end{bmatrix}\right) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}(k+1)c-1\\0\end{bmatrix}W_k\right) \\
&\geq H(V_0^k) + H\left(W_{k+1}\Big|V_0^k\mathbf{x}\begin{bmatrix}(k+1)c-1\\0\end{bmatrix}\right)
\end{aligned}
\tag{105}
$$

Step (e) follows by the fact that conditioning reduces entropy knowing that $W_0^{k-1} \subset \mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}$ and thus $H\left(W_{k+1}\Big|V_0^{k-1}\mathbf{s}\begin{bmatrix}kc+b-1\\kc\end{bmatrix}\mathbf{x}\begin{bmatrix}kc+b-1\\0\end{bmatrix}\right) \leq H(W_{k+1}|W_k)$, and again we remove the negative term before step (f) using (41). Step (g) uses the fact that source packets are independent of each other and (46) follows.

APPENDIX C
EXAMPLES OF CODE CONSTRUCTION IN THE (E) REGION

We give the construction for two specific points in this region, Table VI shows the code construction for the point $\{(4,5)-(7,10)\}$ whereas Table VII shows the code construction for the point $\{(3,5)-(7,9)\}|$. In both cases $k=1$ and $m=1$. The former satisfies $T_1 < 2(B_1 - k)$ whereas the latter satisfies $T_1 > 2(B_1 - k)$.

A. Example (1): $\{(4,5)-(7,10)\} \Rightarrow k=1, m=1$

The code construction achieving the optimal rate of $5/11$ is illustrated in Table VI. In this example, we walk through the steps of both the encoder and the decoder. We note that this point resembles case (A) defined by $T_1 \leq 2(B_1 - k)$ in the general code construction given in Section. VIII.

- Encoder
  - Each source symbol is divided into $T_1 = 5$ sub-symbols $(s_0[.], \ldots, s_4[.])$. A $\mathcal{C}_1 = (4,5)$ is applied along the diagonal of such source sub-symbols producing $B_1 = 4$ parity check sub-symbols $(p_0[.], \ldots, p_3[.])$ defined as follows,

$$
\begin{aligned}
p_0[i] &= s_0[i-5] + s_4[i-1] \\
p_1[i] &= s_1[i-5] + s_4[i-2] \\
p_2[i] &= s_2[i-5] + s_4[i-3] \\
p_3[i] &= s_3[i-5] + s_4[i-4]
\end{aligned}
\tag{106}
$$

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
(1)	$s_0[i]$	$s_0[i+1]$	$s_0[i+2]$	$s_0[i+3]$	$s_0[i+4]$	$s_0[i+5]$
	$s_1[i]$	$s_1[i+1]$	$s_1[i+2]$	$s_1[i+3]$	$s_1[i+4]$	$s_1[i+5]$
	$s_2[i]$	$s_2[i+1]$	$s_2[i+2]$	$s_2[i+3]$	$s_2[i+4]$	$s_2[i+5]$
	$s_3[i]$	$s_3[i+1]$	$s_3[i+2]$	$s_3[i+3]$	$s_3[i+4]$	$s_3[i+5]$
	$s_4[i]$	$s_4[i+1]$	$s_4[i+2]$	$s_4[i+3]$	$s_4[i+4]$	$s_4[i+5]$
(2)	$p_0[i]$	$p_0[i+1]$	$p_0[i+2]$	$p_0[i+3]$	$p_0[i+4]$	$p_0[i+5]$
(3)	$s_0[i-10]+p_1[i]$	$s_0[i-9]+p_1[i+1]$	$s_0[i-8]+p_1[i+2]$	$s_0[i-7]+p_1[i+3]$	$s_0[i-6]+p_1[i+4]$	$s_0[i-5]+p_1[i+5]$
	$s_1[i-10]+p_2[i]$	$s_1[i-9]+p_2[i+1]$	$s_1[i-8]+p_2[i+2]$	$s_1[i-7]+p_2[i+3]$	$s_1[i-6]+p_2[i+4]$	$s_1[i-5]+p_2[i+5]$
	$s_2[i-10]+p_3[i]$	$s_2[i-9]+p_3[i+1]$	$s_2[i-8]+p_3[i+2]$	$s_2[i-7]+p_3[i+3]$	$s_2[i-6]+p_3[i+4]$	$s_2[i-5]+p_3[i+5]$
(4)	$s_3[i-10]$	$s_3[i-9]$	$s_3[i-8]$	$s_3[i-7]$	$s_3[i-6]$	$s_3[i-5]$
	$+$	$+$	$+$	$+$	$+$	$+$
	$\tilde{p}_1[i+2]+\tilde{p}_3[i+4]$	$\tilde{p}_1[i+3]+\tilde{p}_3[i+5]$	$\tilde{p}_1[i+4]+\tilde{p}_3[i+6]$	$\tilde{p}_1[i+5]+\tilde{p}_3[i+7]$	$\tilde{p}_1[i+6]+\tilde{p}_3[i+8]$	$\tilde{p}_1[i+7]+\tilde{p}_3[i+9]$
	$s_4[i-10]$	$s_4[i-9]$	$s_4[i-8]$	$s_4[i-7]$	$s_4[i-6]$	$s_4[i-5]$
	$+$	$+$	$+$	$+$	$+$	$+$
	$\tilde{p}_2[i+2]+\tilde{p}_3[i+3]$	$\tilde{p}_2[i+3]+\tilde{p}_3[i+4]$	$\tilde{p}_2[i+4]+\tilde{p}_3[i+5]$	$\tilde{p}_2[i+5]+\tilde{p}_3[i+6]$	$\tilde{p}_2[i+6]+\tilde{p}_3[i+7]$	$\tilde{p}_2[i+7]+\tilde{p}_3[i+8]$

TABLE VI

RATE 5/11 MU-SCO CODE CONSTRUCTION FOR THE POINT, $(B_1, T_1) = (4,5)$ AND $(B_2, T_2) = (7,10)$ LYING IN REGION (E). THIS POINT IS ALSO ILLUSTRATING CASE (A) DEFINED BY $T_1 \leq 2(B_1 - k)$. FOR THE CAUSAL PART OF PARITY CHECK SUB-SYMBOLS OF $\mathcal{C}_1$ SHIFTED BACK TO TIME $i - t$, WE WRITE $\tilde{p}_j[i]$ INSTEAD OF $\tilde{p}_j[i]\big|_{i-t}$ FOR SIMPLICITY.

– Then, the $T_1 = 5$ parity check-symbols of code $\mathcal{C}_2 = (10,10)$ which are repetitions of the source sub-symbols such that $p_j^{\mathrm{II}}[i] = s_j[i-10]$ for $j \in \{0, \ldots, 4\}$ are concatenated to the parity checks of $\mathcal{C}_1$ with partial overlap of $B_1 - k = 3$ rows as shown in Table VI.

– A $\mathcal{C}_3 = (T_1 - (B_1 - k), B_1 - k) = (2,3)$ SCo code is applied on the last $B_1 - k = 3$ rows of parity check sub-symbols of $\mathcal{C}_1$, $(p_1[.], p_2[.], p_3[.])$ producing $T_1 - (B_1 - k) = 2$ parity check sub-symbols, $(p_0^3[.], p_1^3[.])$. The produced parity checks is shifted back by $T_1 = 5$ and combined with the last two rows of parity check sub-symbols of $\mathcal{C}_2$.

We note that applying a shift back of $T_1 = 5$ on the parity check sub-symbols of $\mathcal{C}_3$ explains why $p_0^3[i] = p_1[i+2]+p_3[i+4]$ appears at time i and not $i+5$. Moreover, since $p_1[i+2]+p_3[i+4]$ in general combines source sub-symbols at time $i+3$ and earlier, they can not appear at time i as this violates the causality of the code construction. Thus, the causal part of such parity checks shifted to any time instant t (denoted by $\tilde{p}_j[.]\big|_t$) is to be sent instead. For example, the first parity check sub-symbol of $\mathcal{C}_3$ at time i is $p_0^3[i+5] = p_1[i+2]+p_3[i+4] = s_1[i-3]+s_4[i+1]+s_3[i-1]+s_4[i]$. The causal part of this parity check is sent instead, i.e., $\tilde{p}_0^3[i+5]\big|_i = \tilde{p}_1[i+2]\big|_i + \tilde{p}_3[i+4]\big|_i = s_1[i-3]+s_3[i-1]$. According to Fig. 8, we divide each channel packet into four layers,

– Layer (1) contains the first five rows which are the source sub-symbols.

– Layer (2) contains the next row.

– Layer (3) contains the next three rows where overlap between the parity checks of codes $\mathcal{C}_1$ and $\mathcal{C}_2$ takes place.

– Layer (4) contains the last two rows. The overlap between the parity checks of codes $\mathcal{C}_2$ and $\mathcal{C}_3$ takes place.

• Decoder

With a burst erasure of length $B_1 = 4$ taking place at times $[i-4, i-1]$, the decoder at user 1 simply uses the first four rows of parity checks at times $[i, i+4]$ after subtracting the unerased source sub-symbols $s_0[t], s_1[t], s_2[t]$ for $t \in \{i-10, \ldots, i-6\}$. For user 2, we assume a burst erasure of length $B_2 = 7$ at times $[i-7, i-1]$. The decoding steps are as follows.

– **Step (1):** Recover $p_j[i+3]$ and $p_j[i+4]$ for $j = \{1,2,3\}$.

(a) In layer (3), spanning the second, third and fourth rows of parity checks, one can see that the parity check sub-symbols of C_2 in the interval $[i, i+2]$ are unerased source sub-symbols. Thus, the corresponding combined parity check sub-symbols of C_1 can be computed in this interval.

(b) In the same layer but in the interval $[i + 5, \infty)$, the parity check sub-symbols of C_1 are of indices $i+5$ and later. Using the fact that (B_1, T_1) SCo code has a memory of T_1 symbols, it can be easily shown that these parity check sub-symbols combine only source sub-symbols of time i and later which are not erased and thus can be computed as well (c.f. (106)).

(c) Steps (a) and (b) show that all the parity check sub-symbols of C_1 in layer (3) can be computed except for the interval $[i + 3, i + 4]$.

(d) The parity check sub-symbols of C_2 in layer (4) spanning the last two rows of parity check sub-symbols in the interval $[i, i + 2]$ are again unerased source sub-symbols and thus can be cancelled and the corresponding parity check sub-symbols of C_3 can be computed in this interval.

(e) The parity-check sub-symbols of C_3 in the interval $[i, i + 2]$,

$$\begin{pmatrix} p_0^3[i + 5]\big|_i & p_0^3[i + 6]\big|_{i+1} & p_0^3[i + 7]\big|_{i+2} \\ p_1^3[i + 5]\big|_i & p_1^3[i + 6]\big|_{i+1} & p_1^3[i + 7]\big|_{i+2} \end{pmatrix}, \tag{107}$$

can recover the remaining two columns of parity-check sub-symbols of C_1 in the interval $[i + 3, i + 4]$ lying in layer (3),

$$\begin{pmatrix} p_1[i + 3] & p_1[i + 4] \\ p_2[i + 3] & p_2[i + 4] \\ p_3[i + 3] & p_3[i + 4] \end{pmatrix},$$

since C_3 is a $(2, 3)$ SCo code whose parity-check sub-symbols are shifted back by $T_1 = 5$. However, only the causal part of the parity checks of C_3 are available. Thus, the non-causal part is to be computed and added to the causal-part to recover the original parity checks of the SCo code. Using (106), it can be seen that the recovery of the non-causal part does not require the availability of source sub-symbols after time[3] $i + 3$. For example, $p_0^3[i + 5]\big|_i = p_1[i+2]+p_3[i+4] = s_1[i-3]+s_4[i]+s_3[i-1]+s_4[i]$, while $\tilde{p}_0^3[i+5]\big|_i = \tilde{p}_1[i+2]\big|_i+\tilde{p}_3[i+4]\big|_i = s_1[i - 3] + s_3[i - 1]$, i.e., the non-causal part of $p_0^3[i + 5]\big|_i$ is $\bar{p}_0^3[i + 5]\big|_i = 2s_4[i]$ which is clearly available before time $i + 3$. Thus the non-causal portions of all the parity checks are computed and then (107) is applied.

– **Step (2):** After recovering these parity check sub-symbols, the decoder can cancel their effect in the second, third and fourth rows of parity checks (layer (3)) at times $i + 3$ and $i + 4$. Moreover, in the same rows and starting at time $i + 5$ all parity checks of code C_1 combine only unerased source symbols (c.f. (106)) and thus can be cancelled as well.

– **Step (3):** Furthermore, one can see that the parity check sub-symbols of C_3 interfering in the last two rows (layer (4)) starting at time $i+3$ combine parity check sub-symbols of C_1 of indices $i + 5$ and later which was shown before to combine unerased source sub-symbols (c.f. (106)).

According to Step (2) and (3), the parity checks of $C_2 = (10, 10)$ repetition code in layers (3) and (4) are now free of any interference from $i + 3$ and later. Thus, the decoder of user 2 is capable of recovering the erased source sub-symbols in the interval $[i - 7, i - 1]$.

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
1	$s_0[i]$	$s_0[i+1]$	$s_0[i+2]$	$s_0[i+3]$	$s_0[i+4]$	$s_0[i+5]$
	$s_1[i]$	$s_1[i+1]$	$s_1[i+2]$	$s_1[i+3]$	$s_1[i+4]$	$s_1[i+5]$
	$s_2[i]$	$s_2[i+1]$	$s_2[i+2]$	$s_2[i+3]$	$s_2[i+4]$	$s_2[i+5]$
	$s_3[i]$	$s_3[i+1]$	$s_3[i+2]$	$s_3[i+3]$	$s_3[i+4]$	$s_3[i+5]$
	$s_4[i]$	$s_4[i+1]$	$s_4[i+2]$	$s_4[i+3]$	$s_4[i+4]$	$s_4[i+5]$
2	$p_0[i]$	$p_0[i+1]$	$p_0[i+2]$	$p_0[i+3]$	$p_0[i+4]$	$p_0[i+5]$
3	$s_0[i-9]+p_1[i]$	$s_0[i-8]+p_1[i+1]$	$s_0[i-7]+p_1[i+2]$	$s_0[i-6]+p_1[i+3]$	$s_0[i-5]+p_1[i+4]$	$s_0[i-4]+p_1[i+5]$
	$s_1[i-9]+p_2[i]$	$s_1[i-8]+p_2[i+1]$	$s_1[i-7]+p_2[i+2]$	$s_1[i-6]+p_2[i+3]$	$s_1[i-5]+p_2[i+4]$	$s_1[i-4]+p_2[i+5]$
4	$s_2[i-9]+\tilde{p}_1[i+2]$	$s_2[i-8]+\tilde{p}_1[i+3]$	$s_2[i-7]+\tilde{p}_1[i+4]$	$s_2[i-6]+\tilde{p}_1[i+5]$	$s_2[i-5]+\tilde{p}_1[i+6]$	$s_2[i-4]+\tilde{p}_1[i+7]$
	$s_3[i-9]+\tilde{p}_2[i+2]$	$s_3[i-8]+\tilde{p}_2[i+3]$	$s_3[i-7]+\tilde{p}_2[i+4]$	$s_3[i-6]+\tilde{p}_2[i+5]$	$s_3[i-5]+\tilde{p}_2[i+6]$	$s_3[i-4]+\tilde{p}_2[i+7]$
	$s_4[i-9]$	$s_4[i-8]$	$s_4[i-7]$	$s_4[i-6]$	$s_4[i-5]$	$s_4[i-4]$
	$+$	$+$	$+$	$+$	$+$	$+$
	$\tilde{p}_1[i+3]+\tilde{p}_2[i+4]$	$\tilde{p}_1[i+4]+\tilde{p}_2[i+5]$	$\tilde{p}_1[i+5]+\tilde{p}_2[i+6]$	$\tilde{p}_1[i+6]+\tilde{p}_2[i+7]$	$\tilde{p}_1[i+7]+\tilde{p}_2[i+8]$	$\tilde{p}_1[i+8]+\tilde{p}_2[i+9]$

TABLE VII

RATE 5/11 MU-SCO CODE CONSTRUCTION FOR THE POINT, $(B_1, T_1) = (3, 5)$ AND $(B_2, T_2) = (7, 9)$ LYING IN REGION (E). THIS POINT IS ALSO ILLUSTRATING CASE (B) DEFINED BY $T_1 > 2(B_1 - k)$. FOR THE CAUSAL PART OF PARITY CHECK SUB-SYMBOLS OF $\mathcal{C}_1$ SHIFTED BACK TO TIME $i - t$, WE WRITE $\tilde{p}_j[i]$ INSTEAD OF $\tilde{p}_j[i]\big|_{i-t}$ FOR SIMPLICITY.

B. Example (2): $\{(3, 5) - (7, 9)\} \Rightarrow k = 1, m = 1$

Again the capacity equals $5/11$. The code construction achieving such rate is illustrated in Table VII. The reason we give the detailed encoding and decoding steps for one more example is to show the main differences between case (A): $T_1 \leq 2(B_1 - k)$ illustrated by the previous example $\{(4, 5) - (7, 10)\}$ and case (B): $T_1 > 2(B_1 - k)$ illustrated by this example, $\{(3, 5) - (7, 9)\}$.

- Encoder
  - Each source symbol is divided into $T_1 = 5$ sub-symbols $(s_0[.], \ldots, s_4[.])$ (layer (1)). A $\mathcal{C}_1 = (3, 5)$ is applied along the diagonal of such source sub-symbols producing $B_1 = 3$ parity check sub-symbols $(p_0[.], p_1[.], p_2[.])$ defined as follows,

$$p_0[i] = s_0[i - 5] + s_3[i - 2]$$
$$p_1[i] = s_1[i - 5] + s_4[i - 2]$$
$$p_2[i] = s_2[i - 5] + s_3[i - 4] + s_4[i - 3] \tag{108}$$

  - Then, the $T_1 = 5$ parity check-symbols of code $\mathcal{C}_2 = (9, 9)$ which are repetitions of the corresponding source sub-symbols are concatenated to the parity checks of $\mathcal{C}_1$ with partial overlap of $B_1 - k = 2$ rows as shown in Table VII.
  - Since $T_1 = 5 > 4 = 2(B_1 - k)$, this point falls in case (B), one can write $T_1 - (B_1 - k) = r(B_1 - k) + q$ as $3 = 1(2) + 1$, i.e., $r = 1$ and $q = 1$. Thus, $r + 1 = 2$ SCo codes are to be constructed. The first is a repetition code of parameters $\mathcal{C}_{3,1} = (B_1 - k, B_1 - k) = (2, 2)$ is applied on the last $B_1 - k = 2$ rows of parity check sub-symbols of $\mathcal{C}_1$, $(p_1[.], p_2[.])$ producing $(B_1 - k) = 2$ parity check sub-symbols, $(p_0^3[\cdot], p_1^3[\cdot])$ which are then shifted back by $2(B_1 - k) = 4$ symbols, while the second is a $\mathcal{C}_{3,2} = (q, B_1 - k) = (1, 2)$ SCo code applied again on the last two rows of parity check sub-symbols of $\mathcal{C}_1$ along the main diagonal producing one row of parity check sub-symbols, $p_2^3[\cdot]$ which is shifted back by $T_1 = 5$ symbols. The parity check sub-symbols of $\mathcal{C}_{3,1}$ and $\mathcal{C}_{3,2}$ (denoted by $\mathcal{C}_3$) are then concatenated forming $T_1 - (B_1 - k) = 3$ rows of parity check sub-symbols and then combined with the last three rows of parity check sub-symbols of

[3]A proof of this in the general case is provided in the proof of Lemma 2 in Appendix D.

$\mathcal{C}_2$ (layer (3)).

The same causality argument stated in the previous example applies and the causal parts of the corresponding parity check sub-symbols shifted to any time instant t denoted by $\tilde{p}_j[.]\big|_t$ are sent instead (c.f. Table VII).

Similar to the previous example, we divide each channel packet into four layers (c.f. Fig. 8),

- Layer (1) contains the first five rows which are the source sub-symbols.
- Layer (2) contains the next row.
- Layer (3) contains the next two rows where overlap between the parity checks of codes $\mathcal{C}_1$ and $\mathcal{C}_2$ takes place.
- Layer (4) contains the last three rows. The overlap between the parity checks of codes $\mathcal{C}_2$ and $\mathcal{C}_3$ takes place.

- Decoding:

  For user 1, the decoding is similar to the previous example. We assume a burst erasure of length $B_1 = 3$ taking place at times $[i-3, i-1]$. One can recover the parity checks of code $\mathcal{C}_1$ in the first three rows of parity checks at times $[i, i+4]$ after subtracting the unerased combined source sub-symbols $s_0[t], s_1[t], s_2[t]$ for $t \in \{i-9, \ldots, i-5\}$. For user 2, we assume a burst erasure of length $B_2 = 7$ in the interval $[i-7, i-1]$. The decoding steps are as follows.

  - Recover $p_j[i+2]$, $p_j[i+3]$ and $p_j[i+4]$ for $j = \{1, 2\}$.

    (a) In layer (3), spanning the second and third rows of parity checks, one can see that the parity check sub-symbols of $\mathcal{C}_2$ in the interval $[i, i+1]$ are unerased source sub-symbols. Thus, the overlapping parity check sub-symbols of $\mathcal{C}_1$ can be computed in this interval.

    (b) In the same layer but in the interval $[i+5, \infty)$, the parity check sub-symbols of $\mathcal{C}_1$ are of indices $i+5$ and later. Using the fact that (B_1, T_1) SCo code has a memory of T_1 symbols, it can be easily shown that these parity check sub-symbols combine only source sub-symbols of time i and later which are not erased and thus can be computed as well(c.f. (108)).

    (c) In steps (a) and (b), we show that all the parity check sub-symbols of $\mathcal{C}_1$ in layer (3) can be computed except for the interval $[i+2, i+4]$. Let us mark the uncomputed parity check sub-symbols as erased source sub-symbols with two rows and three columns.

    (d) Moreover, the parity check sub-symbols of $\mathcal{C}_2$ in layer (4) spanning the last three rows of parity check sub-symbols in the interval $[i, i+1]$ are again unerased source sub-symbols and thus can be cancelled and the corresponding parity check sub-symbols of $\mathcal{C}_3$ can be computed in this interval.

    (e) $\mathcal{C}_3$ is a concatenation of $\mathcal{C}_{3,1} = (2, 2)$ repetition code producing two parity-check sub-symbols $(p_0^3[.], p_1^3[.])$ and a $\mathcal{C}_{3,2} = (1, 2)$ SCo code producing a single parity-check sub-symbol $p_2^3[.]$. At time i and $i+1$, the parity checks of $\mathcal{C}_{3,1}$,

    $$\left(\begin{array}{c} \tilde{p}_0^3[i]\big|_i \\ \tilde{p}_1^3[i]\big|_i \end{array} \right) = \left(\begin{array}{c} \tilde{p}_1[i+2]\big|_i \\ \tilde{p}_2[i+2]\big|_i \end{array} \right),$$

    thus, $\tilde{p}_1[i+2]\big|_i$ and $\tilde{p}_2[i+2]\big|_i$ can be directly recovered, while their corresponding non-causal parts can be computed before time $i+2$. Similarly, $\tilde{p}_1[i+3]\big|_i$ and $\tilde{p}_2[i+3]\big|_i$ can be recovered at time $i+1$ and their corresponding non-causal parts can be retrieved before $i+3$. The remaining column, $(\tilde{p}_1[i+4]\big|_i, \tilde{p}_2[i+4]\big|_i)^T$ can be recovered using the parity checks of $\mathcal{C}_{3,2} = (1, 2)$ SCo code at time i and $i+1$, $p_2^3[i]$ and $p_2^3[i+1]$ in a similar way used in the previous example.

  After recovering these parity check sub-symbols of $\mathcal{C}_1$, the decoder can cancel their effect in the second and third rows of parity checks (layer (3)) at times $i+2$, $i+3$ and $i+4$. Moreover, in the same rows and starting at time $i+5$ all parity checks of code $\mathcal{C}_1$ combine only unerased source symbols (c.f. (108)) and thus can be cancelled as well.

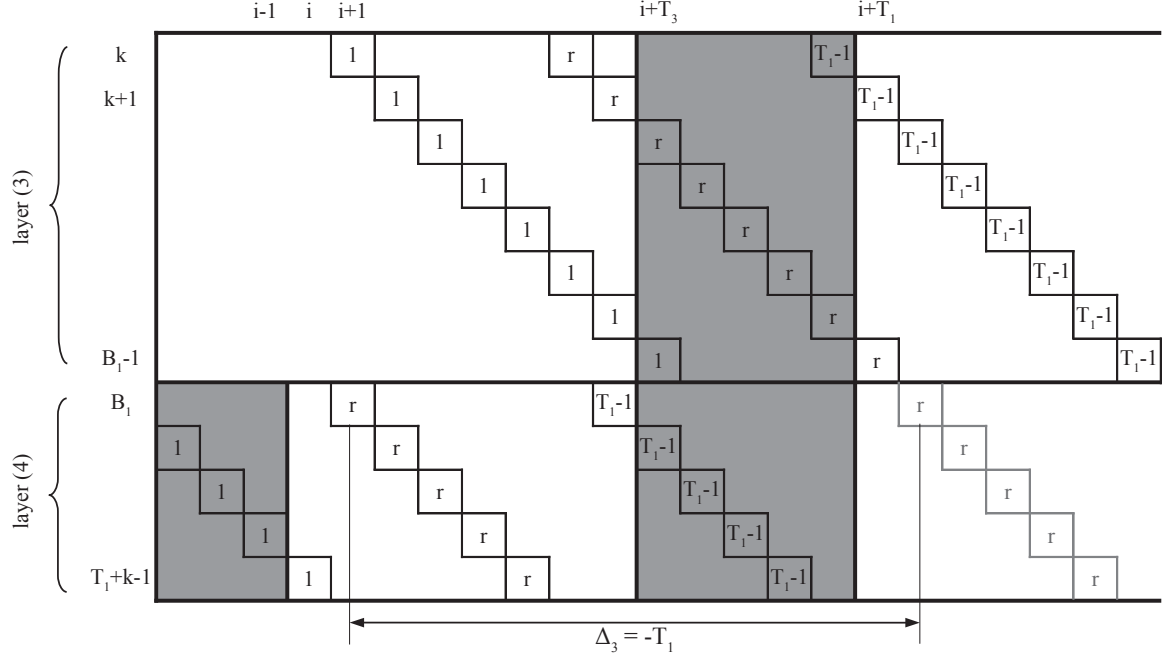 - Remove interference in layer (4) starting at time $i+2$.

Fig. 17. Diagonal Embedding of parity checks for the construction in section VIII. The parity checks $p^3[\cdot]$ are applied using a (T_3, B_3) SCo code onto the last $B_1 - k$ parity checks of $p^I[\cdot]$ in layer 3. The parity checks $p^3[\cdot]$ are shifted back by T_1 units as discussed before.

The parity check sub-symbols of $\mathcal{C}_3$ interfering in the last two rows (layer (4)) starting at time $i + 2$ are of indices $i + 4$ and later which are either recovered in Step (1) or can be calculated as they combine unerased source sub-symbols (c.f. (108)).

– Use the parity-checks in layer (3) and (4) to recover $\mathbf{s}[i - 7], \ldots, \mathbf{s}[i - 1]$.

According to Step (3) and (4), the parity checks of $\mathcal{C}_2$ in layers (3) and (4) are now free of any interference starting at time $i + 2$ and thus, the decoder of user 2 is capable of recovering the erased source sub-symbols.

APPENDIX D
PROOF OF LEMMA. 2

The parity-check sub-symbols of $\mathcal{C}_2$ in the interval $[t_2, t_3 - 1] = [i, i - B_2 + T_2 - 1]$ are source sub-symbols in the interval $[t_2 - T_2, t_3 - T_2 - 1] = [i - T_2, i - B_2 - 1]$ which are not erased. Thus, they can be computed and subtracted to recover the combined parity-check sub-symbols of $\mathcal{C}_1$ and the causal part of that of $\mathcal{C}_3$ in layers (3) and (4), respectively. More specifically, the parity-check sub-symbols $p_{j_1}^I[\cdot]$ for $j_1 \in \{k, \ldots, B_1 - 1\}$ and $\tilde{p}_{j_2}^3[\cdot]$ for $j_2 \in \{0, \ldots, B_3 - 1\}$ are recovered.

Recall that $\mathcal{C}_3$ is a (B_3, T_3) is a SCo code applied by taking the last $B_1 - k$ parity check sub-symbols of $\mathcal{C}_1$ as source sub-symbols.

Let us define the parity-check symbols that need to be recovered as

$$\mathbf{w}[t] = (w_0[t], \ldots, w_{T_3 - 1}[t]) = (p_k^I[t], \ldots, p_{B_1 - 1}^I[t]). \tag{109}$$

We first consider case (A) i.e., when $T_1 \le 2(B_1 - k)$. Since $\mathcal{C}_3$ is an SCo which involves diagonal

interleaving of Low Delay - Burst Erasure Block Codes (LD-BEBC), the diagonals that span the sub-symbols of interest are as follows:

$$\bar{\mathbf{d}}_r = (w_0[i+r], \ldots, w_{T_3-1}[i+r+T_3-1], p_0^3[i+r+T_3], \ldots, p_{B_3-1}^3[i+r+T_3+B_3-1]),$$
$$r \in 1, \ldots, T_3 + B_3 - 1 \quad (110)$$

Since the parity check sub-symbols of $\mathcal{C}_3$ are shifted back by $T_1 = T_3 + B_3$ keeping only their causal part, the corresponding diagonals of interest are

$$\mathbf{d}_r = (w_0[i+r], \ldots, w_{T_3-1}[i+r+T_3-1], \tilde{p}_0^3[i+r+T_3]\big|_{i+r-B_3},$$
$$\ldots, \tilde{p}_{B_3-1}^3[i+r+T_3+B_3-1]\big|_{i+r-1}). \quad (111)$$

where recall that $\tilde{p}_j[t_1]\big|_{t_2}$ denotes the causal part of the parity check $p_j[t_1]$ w.r.t. t_2 (c.f. (57)).

With every parity check sub-symbol projected to a different time instant, one can clearly see that $\mathbf{d}_r$ is no more a code-word of an LD-BEBC code.

The following conditions are sufficient to establish Lemma. 2,

c1 The diagonals d_r for $r \in \{1, \ldots, T_3 + B_3 - 1\}$ span all the parity-check sub-symbols that need to be recovered, i.e., $p_j^I[\cdot]$ for $j \in \{k, \ldots, B_1 - 1\}$ in the interval $[t_3, t_4 - 1] = [i + T_2 - B_2, i + T_1 - 1]$.

c2 The decoder can compute the non-causal part of each parity-check $p_j^3[\cdot]$ in the interval $[t_2, t_3 - 1]$ and reduce (111) to (110). This step should not violate the zero-delay constraint for any erased symbol on the diagonal i.e. the non-causal part of the parity-check sub-symbol $p_{j_1}^3[t_x]$ responsible for the recovery of a given parity check $w_{j_2}[t_y]$ should combine source sub-symbols $\mathbf{s}[.]$ which are both, not erased and from time earlier than t_y.

c3 Each diagonal d_r should have no more than B_3 erased sub-symbols.

For c1, we note that the diagonal $\mathbf{d}_1$ covers $w_{T_3-1}[i+T_3] = p_{B_1-1}^I[i+T_3]$ which is the lower left most sub-symbol that needs to be recovered. At $r = T_3 + B_3 - 1$, one can see that $\mathbf{d}_r$ combines $w_0[i+T_3+B_3-1] = p_k^I[i+T_3+B_3-1]$ which is the upper right most sub-symbol that needs to be recovered. Fig. 17 easily illustrates that the diagonal $\mathbf{d}_r$ for $r \in [1, T_3 + B_3 - 1]$ cover all of the erased sub-symbols.

For c2, we note that all elements of a diagonal $\mathbf{d}_r$ combine source symbols $\mathbf{s}[\cdot]$ from time $i + r - 1$ and earlier according to the diagonal interleaving property of SCo codes. Thus, one can conclude that the non-causal part of any parity-check sub-symbol $p_j^3[i+r+T_3+j]\big|_{i+r-B_3+j}$ for $j \in 0, \ldots, B_3 - 1$ in $\mathbf{d}_r$ is just a combination of source symbols in the interval $[i+r-B_3+j, i+r-1]$. Thus the entire non-causal part of each parity check is available before time $i + r$ and the reduction to (110) is possible for each $\mathbf{d}_r$.

Finally note that the zero delay constraint also requires that the symbols $w_j[t]$ with $t \geq i + T_1$ in $\mathbf{d}_r$ be made available before time $t = i + r$. Since each $w_j[t]$ for $t \geq i + T_1$ only consists of combinations of source symbols in $[i, i+r-1]$ these symbols can be explicitly computed by the decoder by time $i+r-1$ and c2 follows.

For c3, we divide the values of r into three intervals.

- $\mathbf{d}_r$ for $r \in \{1, \ldots, T_1 - T_3\}$
  In this range, one can see that the following symbols are available,

  $$(w_0[i+r], \ldots, w_{T_3-r-1}[i+T_3-1], \tilde{p}_{B_3-r}^3[i+T_3+B_3], \ldots, \tilde{p}_{B_3-1}^3[i+r+T_3+B_3-1]),$$

  which are a total of T_3 sub-symbols in the beginning and the end of the diagonals $\mathbf{d}_r$ which contains $T_3 + B_3$ sub-symbols. In other words, each such diagonal has B_3 erased sub-symbols happening in a burst.

- $\mathbf{d}_r$ for $r \in \{T_1 - T_3 + 1, \ldots, T_3\}$

In these diagonals, the following symbols are available,

$$(w_0[i+r], \ldots, w_{T_3-r-1}[i+T_3-1], w_{T_1-r}[i+T_1], \ldots, w_{T_3-1}[i+r+T_3-1], \tilde{p}_0^3[i+r+T_3]$$
$$, \ldots, \tilde{p}_{B_3-1}^3[i+r+T_3+B_3-1]),$$

The first group is a total of $T_3 - r$ consecutive sub-symbols, while the other two groups are a total of r consecutive sub-symbols. This implies that each such diagonal $\mathbf{d}_r$ has B_3 erased sub-symbols in a burst.

- $\mathbf{d}_r$ for $r \in \{T_3 + 1, \ldots, T_3 + B_3 - 1\}$
  The available sub-symbols in these diagonals are,

$$(w_{T_1-r}[i+T_1], \ldots, w_{T_3-1}[i+r+T_3-1], \tilde{p}_0^3[i+r+T_3], \ldots, \tilde{p}_{T_3+B_3-r-1}^3[i+2T_3+B_3-1]),$$

which are again a total of T_3 consecutive sub-symbols which implies that the considered diagonals $\mathbf{d}_r$ has B_3 erased sub-symbols in a burst and the c3 follows. We note that LD-BEBC codes are capable of recovering wrap-around burst which may start at the end of the block and wrap around to the beginning of that block.

When $T_1 > 2(B_1 - k)$ note that $\mathcal{C}_3$ is a concatenation of $r + 1$ codes, the first r of which are repetition codes with parity check sub-symbols given by (61). These parity-check sub-symbols in the interval $[i, i + (B_1 - k) - 1]$ can be used to recover the causal part of the parity-check sub-symbols $(p_k^I[t_1], \ldots, p_{B_1-1}^I[t_1])$ for $t_1 \in \{i + (B_1 - k), \ldots, i + (r+1)(B_1 - k) - 1\} = \{i + T_2 - B_2, \ldots, i + T_1 - q - 1\}$. The non-causal part of these parity-check sub-symbols combine source sub-symbols in the interval $[i, t_1 - 1]$ which are not erased and thus can be recovered.

The remaining q columns of parity-check sub-symbols $(p_k^I[t_2], \ldots, p_{B_1-1}^I[t_2])$ for $t_2 \in \{i + (r+1)(B_1 - k), \ldots, i + (r+1)(B_1 - k) + q - 1\} = \{i + T_1 - q, \ldots, i + T_1 - 1\}$ can be recovered using the parity-check sub-symbols of $\mathcal{C}_{3,r+1} = (q, B_1 - k)$. This step is similar to that of recovering the $T_1 - (B_1 - k)$ columns of parity-check sub-symbols of $\mathcal{C}_1$ using $\mathcal{C}_3 = (T_1 - (B_1 - k), B_1 - k)$ done above, except that $B_3 = T_1 - (B_1 - k)$ is replaced by $B_{3,r+1} = q$.

APPENDIX E
PROOF OF (71) AND (72)

One can get the result in (71) through the following steps,

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k \Big| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$= H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} W_k \Big| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right) - H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \Big| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix} W_k\right)$$

$$\overset{(a)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} W_k \Big| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$= H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \Big| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right) + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$\overset{(b)}{=} H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$\geq H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(W_k \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix}\right)$$

$$= H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} W_k \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix}\right)$$

$$- H\left(\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix} W_k\right)$$

$$\overset{(c)}{=} H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} W_k \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix}\right)$$

$$= H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \Big| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix}\right)$$

$$+ H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix}\right)$$

$$\overset{(d)}{=} H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+b-1 \\ 0 \end{bmatrix}\right)$$

$$\geq H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right) \tag{112}$$

We use (66) to remove the negative term before step (a). Similarly, we remove the negative term before step (c) using (65). Steps (b) and (d) use the fact that source packets are independent of each other and of previous channel packets.

The following steps help finding the result in (72),

$$H(W_0^{k+1})$$

$$\geq H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right) + H(W_{k+1}|W_0^k)$$

$$\geq H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(W_k^{k+1} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$= H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+c \end{bmatrix} W_k^{k+1} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$- H\left(\mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+c \end{bmatrix} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix} W_k^{k+1}\right)$$

$$\overset{(e)}{=} H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+c \end{bmatrix} W_k^{k+1} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$= H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+c \end{bmatrix} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} kd+c-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$+ H\left(W_k^{k+1} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$\overset{(f)}{=} H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+b \end{bmatrix}\right) + H\left(W_k^{k+1} \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{s}\begin{bmatrix} (k+1)d-1 \\ kd+b \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$= H(V_0^k) + H\left(W_k^{k+1} \middle| V_0^k \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right)$$

$$= H(V_0^k) + H\left(W_k \middle| V_0^k \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix}\right) + H\left(W_{k+1} \middle| V_0^k \mathbf{x}\begin{bmatrix} kd+c-1 \\ 0 \end{bmatrix} W_k\right)$$

$$\geq H(V_0^k) + H\left(W_{k+1} \middle| V_0^k \mathbf{x}\begin{bmatrix} (k+1)d-1 \\ 0 \end{bmatrix}\right) \tag{113}$$

Once again, we remove the negative term before step (e) using and (67). Steps (f) uses the fact that each source packet is independent of each other.

APPENDIX F
PROOF OF (83) AND (84)

For the result in (83), we walk through the following steps,

$$H(W_0^k) \geq H(V_0^{k-1}) + H\left(W_k \middle| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$\overset{(a)}{=} H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} W_k \middle| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right) - H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \middle| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix} W_k\right)$$

$$= H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} W_k \middle| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$= H(V_0^{k-1}) + H\left(\mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \middle| V_0^{k-1} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$= H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd-1 \\ 0 \end{bmatrix}\right)$$

$$\geq H\left(V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix}\right) + H\left(W_k \middle| V_0^{k-1} \mathbf{s}\begin{bmatrix} kd+a-1 \\ kd \end{bmatrix} \mathbf{x}\begin{bmatrix} kd+a-1 \\ 0 \end{bmatrix}\right)$$

$$
\begin{aligned}
&= H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+a-1\\kd\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}kd+b-1\\kd+a\end{bmatrix}W_k\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+a-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+a-1\\0\end{bmatrix}\right)\\
&\quad - H\left(\mathbf{s}\begin{bmatrix}kd+b-1\\kd+a\end{bmatrix}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+a-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+a-1\\0\end{bmatrix}W_k\right)\\
&\stackrel{(b)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+a-1\\kd\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}kd+b-1\\kd+a\end{bmatrix}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+a-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+a-1\\0\end{bmatrix}\right)\\
&\quad + H\left(W_k\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+a-1\\0\end{bmatrix}\right)\\
&\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\right) + H\left(W_k\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)
\end{aligned}
\tag{114}
$$

The negative terms in (a) and (b) are removed using (78) and (77) respectively.

Next, we start by adding $H(W_{k+1}|W_0^k)$ to both sides of (83) to find (84) as follows:

$$
\begin{aligned}
&H(W_0^{k+1})\\
&\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\right) + H\left(W_k\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right) + H(W_{k+1}|W_0^k)\\
&\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\right) + H\left(W_k\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)\\
&\quad + H(W_{k+1}|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}W_0^k)\\
&\geq H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\right) + H\left(W_k^{k+1}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)\\
&= H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)d-1\\kd+c\end{bmatrix}W_k^{k+1}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)\\
&\quad - H\left(\mathbf{s}\begin{bmatrix}(k+1)d-1\\kd+c\end{bmatrix}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}W_k^{k+1}\right)\\
&\stackrel{(c)}{=} H\left(V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}(k+1)d-1\\kd+c\end{bmatrix}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)\\
&\quad + H\left(W_k^{k+1}\middle|V_0^{k-1}\mathbf{s}\begin{bmatrix}kd+b-1\\kd\end{bmatrix}\mathbf{s}\begin{bmatrix}(k+1)d-1\\kd+c\end{bmatrix}\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)\\
&\stackrel{(d)}{=} H(V_0^k) + H\left(W_k^{k+1}\middle|V_0^k\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right)\\
&= H(V_0^k) + H\left(W_k\middle|V_0^k\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}\right) + H\left(W_{k+1}\middle|V_0^k\mathbf{x}\begin{bmatrix}kd+c-1\\0\end{bmatrix}W_k\right)\\
&\geq H(V_0^k) + H\left(W_{k+1}\middle|V_0^k\mathbf{x}\begin{bmatrix}(k+1)d-1\\0\end{bmatrix}\right)
\end{aligned}
\tag{115}
$$

The negative in (c) is removed using (79). Step (d) follows from the fact that source symbols are independent and (84) follows.

APPENDIX G
PROOF OF (96) AND (99)

The steps to get (96) are,

$$
\begin{aligned}
&\sum_{i=B_2}^{j+1} H(\mathbf{x}[i])\\
&\stackrel{(a)}{\geq} H\left(\mathbf{s}\begin{bmatrix}j-B_2\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix}j\\B_2\end{bmatrix}\middle|\mathbf{s}\begin{bmatrix}j-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j-B_2\\0\end{bmatrix}\right)\\
&\quad + H(\mathbf{s}[j+1-B_2]) + H\left(\mathbf{x}[j+1]\middle|\mathbf{s}[j+1-B_2]\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)\\
&\geq H\left(\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix}j+1\\B_2\end{bmatrix}\middle|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)\\
&= H\left(\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{s}[j+1-T_1]\mathbf{x}\begin{bmatrix}j+1\\B_2\end{bmatrix}\middle|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)
\end{aligned}
$$

$$- H\left(\mathbf{s}[j+1-T_1]\Big|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1\\B_2\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)$$

$$\overset{(b)}{=} H\left(\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{s}[j+1-T_1]\mathbf{x}\begin{bmatrix}j+1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)$$

$$= H\left(\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{s}[j+1-T_1]\Big|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)$$

$$+ H\left(\mathbf{x}\begin{bmatrix}j+1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j+1-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right)$$

$$\overset{(c)}{\geq} H\left(\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}j+1-T_1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix}j+1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}j+1-T_1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}j+1-B_2\\0\end{bmatrix}\right) \tag{116}$$

Step (a) is the addition of (93) and (90), step (b) uses (91) to remove the negative term in the previous step, and step (c) uses the fact that the source packets are independent of each other. The result is the form (90) for $m = l + 1$.

While the working out to get (99) is as follows,

$$\sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i])$$

$$\geq H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right)$$

$$\geq H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2-1\\0\end{bmatrix}\right)$$

$$= H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2-1\\0\end{bmatrix}\right)$$

$$- H\left(\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2+T_1-1\\0\end{bmatrix}\right)$$

$$\overset{(d)}{=} H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2-1\\0\end{bmatrix}\right)$$

$$= H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2-1\\0\end{bmatrix}\right)$$

$$+ H\left(\mathbf{x}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2-1\\0\end{bmatrix}\right)$$

$$= H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2-B_1\end{bmatrix}\right) + H\left(\mathbf{x}\begin{bmatrix}B_2+T_1-1\\B_2\end{bmatrix}\Big|\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2-B_1\end{bmatrix}\mathbf{x}\begin{bmatrix}B_2-1\\0\end{bmatrix}\right)$$

$$\geq H\left(\mathbf{s}\begin{bmatrix}T_1-1\\0\end{bmatrix}\right) + H\left(\mathbf{s}\begin{bmatrix}B_2+T_1-1\\B_2-B_1\end{bmatrix}\right) \tag{117}$$

where step (d) makes use of (98).

References

[1] A. Sahai, "Why Do Block Length and Delay Behave Differently if Feedback Is Present?," IEEE Transactions on Information Theory, May 2008, Vol. 54, pp. 1860 — 1886.

[2] R. T. Sukhavasi and B. Hassibi, "Linear error correcting codes with anytime reliability," ISIT 2011, 1748-1752

[3] C Li and S. Yang and R. W. Yeung, "Multi-rate sequential data transmission," ITA Workshop, 2013

[4] M. Effros, "On dependence and delay: Capacity bounds for wireless networks. WCNC 2012: 550-554

[5] D. L. Neuhoff, R. K. Gilbert, "Causal source codes," IEEE Transactions on Information Theory 28(5): 701-713 (1982)

[6] 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.

[7] 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.

[8] D.J. Leith and D. Vasudevan, "Coding packets over reordering channels", ISIT 2010, Austin, TX.

[9] J.K.Sundararajan, D. Shah, and M. Medard, "ARQ for network coding," in IEEE ISIT 2008, (Toronto, Canada), Jul. 2008.

[10] E. Martinian, "Dynamic Information and Constraints in Source and Channel Coding," *Ph.D. Thesis,* MIT, September 2004.

[11] E. Martinian, C. W. Sundberg, "Burst erasure correction codes with low decoding delay," IEEE Transactions on Information Theory vol. 50, No. 10, pp. 2494-2502, 2004.

[12] E. Martinian and M. Trott, "Delay-optimal Burst Erasure Code Construction," *ISIT,* (Nice, France) July 2007.

[13] A. Badr and A. Khisti and W. T. Tan and J. Apostolopoulos, "Streaming Codes for Channels with Burst and Isolated Erasures," INFOCOMM 2013 http://eprintweb.org/S/article/cs/1208.0072/

[14] A. Badr and A. Khisti and W. T. Tan and J. Apostolopoulos, "Robust Streaming Erasure Codes Using Deterministic Channel Approximations" Submitted, ISIT 2013

[15] 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, vol. 14, pp. 841—851, 2004.

[16] A. Badr, A. Khisti, and E. Martinian, "Diversity Embedded Streaming Erasure Codes (DE-SCo): Constructions and Optimality", IEEE JSAC, Special Issue on Trading Rate for Delay at Transport and Application Layers, March 2011.

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

[18] A. Khisti and J.P. Singh, "On multicasting with streaming burst-erasure codes," *ISIT,* (Seoul, Korea) July 2009.

[19] S.S. Vyetrenko, "Network coding for error correction," Ph.D. Thesis, California Institute of Technology, May 2011.

[20] Zhi Li, A. Khisti and B. Girod, "Correcting erasure bursts with minimum decoding delay," Asilomar Conference on Signals, Systems and Computers, November 2011.

[21] H. Deng, M. Kuijper and J.S. Evans, "Burst Erasure Correction Capabilities of (n,n-1) Convolutional Codes", International Conference on Communications (ICC), 2009, Germany.

[22] Zhi Li, A. Khisti and B. Girod, "Forward error correction for low-delay packet video," in Proceedings of Packet Video Workshop, Dec. 2010, Hong Kong.

[23] H. Yao, Y. Kochman, G. W. Wornell, "Delay-Throughput Tradeoff for Streaming Over Blockage Channels with Delayed Feedback, in MILCOM, (San Jose, CA), Nov. 2010.

[24] Y. Li and E. Soljanin, "Rateless Codes for Single-Server Streaming to Diverse Users," Allerton Conference on Communication, Control, and Computing, 2009.

[25] T. K. Dikaliotis, A. Dimakis, T. Ho, and M. Effros, "On the Delay of Advantage of Coding in Packet Erasure Networks," Information Theory Workshop (ITW), 2010.

[26] D. Leong and T. Ho,"Erasure Coding for Real-Time Streaming," IEEE International Symposium on Information Theory (ISIT) 2012.

[27] O. F. Tekin, T. Ho, H. Yao and S. Jaggi "On erasure correction coding for streaming," Information Theory and Applications Workshop (ITA) 2012.

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

[29] R. G. Gallager, "Information Theory and Reliable Communication," New York: Wiley, 1968