

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 low-delay error correction codes in a real-time streaming setup. The encoder observes a stream of source packets and outputs the channel packets in a causal fashion, which are broadcast to two receivers over burst-erasure channels. Each receiver must decode the source packets sequentially with a deadline of T_i , while its channel can introduce an erasure burst of maximum length B_i , where $i \in \{1, 2\}$ and w.l.o.g. $B_2 > B_1$. We study the associated capacity as a function of the burst lengths and decoding deadlines.

We observe that the operation of the system can be divided into two main regimes. The so-called *large-delay regime* corresponds to the case when either $T_1 \geq B_2$ or $T_2 \geq B_1 + B_2$. We show that for these parameters, the optimal code is obtained through simple modifications of previously proposed single-user codes by Martinian et al. and the diversity embedded streaming codes proposed by Badr, Khisti and Martinian. When both $T_1 < B_2$ and $T_2 < B_1 + B_2$, the system is said to be in the *low-delay regime*. We propose a new code construction and establish its optimality when $T_2 \geq T_1 + B_1$. In the case when $T_2 < T_1 + B_1$, we establish upper and lower bounds on the capacity and characterize the exact capacity when either $T_1 = B_1$ or $T_2 = B_2$. Our upper bounds in the low-delay regime are based on novel information theoretic arguments that capture the tension between the decoding constraints at the two receivers.

Index Terms

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

I. INTRODUCTION

A growing number of multimedia applications including video conferencing, cloud computing, and mobile gaming operate in real-time and under strict delay constraints. Recent studies [1] indicate that voice over IP applications such as Skype use a significant amount of forward error correction to mitigate packet losses over networks. Error correction codes are also proactively used in many video conferencing systems [2]. Such systems are highly vulnerable to sporadic burst packet losses and long packet delays in wireless networks. Thus the study of low-delay error correction codes over burst-erasure channels is naturally motivated by these applications. Both the fundamental capacity limits and the error correction techniques for communication systems that operate under strict delay constraints can be very different from classical capacity results. It is well known that the (Shannon) capacity of an erasure channel only depends on the fraction of erasures — the actual location of the erasures is not relevant. However this is not the case in streaming applications. For example, the decoding delay over channels with burst erasures can be very different than the delay over memoryless channels.

As a first step towards understanding properties of optimal low-delay error correction codes in the presence of correlated erasures, a new communication model has been introduced in [3]–[5]. A stream of source packets arrives sequentially at the encoder, and is mapped to a stream of channel packets. The channel considered is a burst erasure channel which introduces an erasure burst of maximum length B starting at an arbitrary time. Each source packet must be recovered within a maximum (peak) delay of T packets. Such a streaming setup is relevant to many multimedia applications. For example in audio/video streaming, the value of T is governed by the play-back delay of each source frame. While the burst erasure model considered in [3]–[5] is somewhat simplistic, its analysis provides useful insights into constructing codes when channel losses are correlated. A novel class of codes, Maximally Short (MS) codes, that satisfy the decoding constraints and achieve the maximum possible rate over the burst erasure channel model is proposed.

In this work we extend the point to point model in [3]–[5] to a two user multicast setup, where the channel of the first user introduces an erasure burst of length up to B_1 , while that of the second user introduces an erasure burst of length up to B_2 , greater than B_1 . The decoding delay at the first user is T_1 whereas that at the second user is T_2 . Both users are interested in decoding the common source stream. We study the capacity as a function of these burst and delay parameters. We note that

The corresponding author is Ashish Khisti (ashish.khisti@gmail.com). This work was supported by a discovery grant from National Science and Engineering Research Council of Canada, Hewlett Packard through a HP-IRP Award and an Ontario Early Researcher Award. Preliminary results of this work were presented at the 2010 Allerton Conference on Communication, Control, and Computing, Montecillo, IL.

our model can also be viewed as a compound setup involving a single receiver with two different channels. However instead of the worst-case scenario, the delay in our setup depends on the channel realization. One application of our setup is in error concealment for adaptive media playback [6]. Such techniques adjust the play-out rate as a function of the receiver buffer size, so that a temporary increase in delay can be naturally accommodated.

The multicast setup has been introduced previously in [7], [8]. Necessary and sufficient conditions under which the multicast capacity equals the single user capacity of the stronger user i.e., user 1 were established. In particular it was shown that if the delay of user 2 satisfies $T_2 \geq \frac{B_2}{B_1}T_1 + B_1$, then the multicast capacity equals the capacity when user 1 is alone present. A new construction, Diversity Embedded Streaming Codes (DE-SCo) was introduced that achieves this bound. The DE-SCo construction exploits the relatively large value of T_2 to apply two single user MS codes to the source stream, and superimposes the resulting parity checks in a fashion that they do not interfere with one another. One of the codes is simply an optimal single user MS code for user 1, while the other code for user 2 is constructed by taking advantage of the side information available to this receiver from the first MS code.

In this paper we provide a more thorough treatment of the multicast capacity for all burst and delay parameters. In particular we classify the system into two different operating regimes. The *large delay regime* corresponds to the case when $T_1 \geq B_2$ or $T_2 \geq B_1 + B_2$ i.e., one of the delays is sufficiently large. For this case we characterize the capacity, and show that it can be obtained through certain modifications of the single user MS codes and the DE-SCo construction. Our key observation is that in this regime without loss of optimality, the delay of one of the users can be reduced up to a certain critical threshold, to which the previously proposed constructions are applicable. For the *low delay regime*, when $T_1 < B_2$ and $T_2 < B_1 + B_2$, a new code construction is proposed and shown to be optimal when $T_2 \geq T_1 + B_1$. In this regime our proposed code cannot keep the parity checks for the two users non-interfering, as was done in DE-SCo. In order to account for the additional interference generated due to overlapping parity checks, we construct and embed a third set of parity checks. The construction of these parity checks must also satisfy the causality and delay constraints, which makes the analysis particularly challenging. Furthermore we also remark that the upper bounds on the capacity in the low-delay regime are based on techniques that are significantly more difficult than [7], [8]. The converse in [7], [8], requires constructing a periodic erasure channel (PEC) and establishing that every erased packet in this channel can be recovered at the destination. Such an approach does not yield the tightest upper bound for the low-delay regime. Our converse proofs are based on an information theoretic argument and capture certain new tensions that arises due to the decoding constraints of the two receivers in the low-delay regime.

In the rest of the paper, Section II discusses related works and Section III introduces the streaming setup. We review the results on single-user MS codes as well as the diversity embedded streaming codes [7], [8] in Section IV and summarize our main results in Section V. In Section VI we provide an information theoretic converse for the single user capacity, which is useful in the subsequent proofs. In Section VII we establish the capacity in the large delay regime while the subsequent sections treat the low-delay regime. For the case when $T_2 \geq T_1 + B_1$ we establish the capacity by presenting the code construction in Section VIII and the corresponding converse in Section IX. The case when $T_2 < T_1 + B_1$ is treated in Sections X and XI. We establish upper and lower bounds on the capacity in Section X, establish the capacity in the special cases when $T_1 = B_1$ and $T_2 = B_2$ in Sections XI-A and XI-B respectively, and present a conjecture on the capacity in Section XI-C. We finally present the conclusions in Section XII.

II. RELATED WORK

We review prior works on low-delay codes for streaming for the interest of the reader, and discuss the differences with the present work. References [3]–[5] introduce Maximally Short codes (MS) codes for the burst-erasure channel that introduces an erasure burst of maximum length B , and with a decoding delay of T . These constructions involve a two step approach. In the first step a block code is constructed with certain low-delay properties, and then interleaved to construct a streaming code. We will briefly review these codes in Section IV. In [9]–[11], low-delay codes for a sliding window channel model with burst and isolated erasures are proposed. A fundamental tradeoff between the burst erasure and isolated erasure correction properties of any code is established, and a new class of codes, Maximum Distance And Span (MiDAS) codes, that achieves a near-optimal tradeoff is proposed. These codes involve a layered code design, as opposed to the block-code construction of MS Codes.

The setup in [3]–[5] considers the case when one source packet arrives in each time-slot and one channel packet must be transmitted in each slot. References [11]–[14] consider the case where the source arrival and channel transmission rates are mismatched. In particular, $M > 1$ channel packets must be transmitted by the encoder between two successive source packets. References [11], [12] consider the decoding delay in terms of the source packets and characterize the capacity for the case of burst-erasure channels. The associated code constructions are based on layering and involve Strongly-MDS codes [15] as constituent codes. References [13], [14] study a similar setup when the decoding delay is with respect to channel packets. For the burst erasure model, diagonally interleaved block codes are shown to be optimal when gaps between successive bursts are sufficiently small. For the i.i.d. erasure model a family of time-invariant intra-session codes are proposed with a performance that is close to an upper bound.

In [16], [17], Lui et al. consider a model where the transmitter and receiver are connected through multiple parallel links. Each link is assumed to be a burst erasure channel that introduces a burst of maximum length B . The capacity is characterized

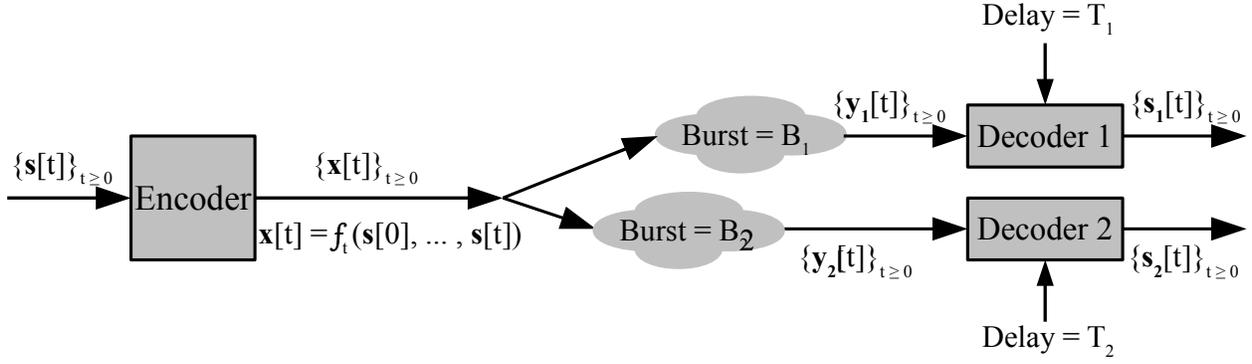


Fig. 1. The source stream $\{s[i]\}$ is causally mapped into an output stream $\{x[i]\}$. Both the receivers observe these symbols via their channels. The channel introduces an erasure-burst of length B_i , and each receiver tolerates a delay of T_i , for $i = 1, 2$.

in some special cases and joint coding across the sub-channels is required to attain the capacity. In [18], the problem of having multiple erasure bursts within each coding block is studied. It is shown that the delay to recover any individual symbol not only depends on the number of bursts within a coding block, but also on whether the source symbols are encoded causally or non-causally.

As discussed previously, the two-user multicast model was proposed in [7], [8], but only the case when the multicast capacity equals the single-user capacity of the stronger receiver was considered. In this paper we consider the more general question on how the capacity behaves as a function of the burst and delay parameters i.e., B_1 , T_1 , B_2 and T_2 . We note that some preliminary results of this paper appeared in the conference version [19].

In the broader literature, problems involving real-time coding and compression have been studied from many different perspectives. We briefly discuss some such approaches, although these are not directly relevant to the present work. Schulman [20] and Sahai [21] study coding techniques based on *tree codes* in a streaming setup with discrete memoryless channels. Sukhavasi and Hassibi [22] have proposed linear time-invariant tree codes for the class of i.i.d. erasure channels, which are attractive due to low decoding complexity. However, these works focus on i.i.d. channels and applications to control systems. Adaptations of rateless codes for streaming are also studied in the literature. In [23], [24], the use of rateless codes with overlapped sliding windows are considered for real-time requirements. In [25], a generalization of rateless codes is proposed that involves Unequal Error Protection (UEP) and provides unequal recovery times. In [26], [27], the performance of LT codes for single-server streaming to diverse users is investigated. Different users have different channel conditions as well as different decoding capabilities. Optimization of the degree distribution is proposed and solved using linear programming. For the case of feedback, Sahai [28] showed that for Discrete Memoryless Channels (DMC), feedback generally provides dramatic gains in the error exponents when fixed end-to-end delay is considered. In [29], the authors proposed a new scheme which combines the benefits of network coding and ARQ by acknowledging degrees of freedom instead of original packets. In [30]–[32], real-time streaming over blockage channels with delayed feedback is studied. A multi-burst transmission protocol is proposed which achieves a non-trivial tradeoff between the delay and throughput within this framework.

III. SYSTEM MODEL

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

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

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

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

where $i \in \{1, 2\}$, $j_i \geq 0$ and \star denotes an erasure. Furthermore, user i tolerates a delay of T_i , i.e., there should exist a sequence of decoding functions $\gamma_{1t}(\cdot)$ and $\gamma_{2t}(\cdot)$ such that

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

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 uniformly sampled over some finite alphabet $\mathcal{S} = \mathbb{F}_q^k$. The channel symbols $\mathbf{x}[t]$ belong to some alphabet $\mathcal{X} = \mathbb{F}_q^n$.

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., $R = \frac{\log |\mathcal{S}|}{\log |\mathcal{X}|} = \frac{k}{n}$, and the multicast streaming capacity is the maximum achievable rate.

Definition 1 (Multicast Streaming Capacity). *A rate R is achievable if there exists a streaming code of this rate over some field-size q such that if the channel introduces a burst of length B_i for $i \in \{1, 2\}$, every source symbol $\mathbf{s}[t]$ for $t \geq 0$ can be decoded with a delay of¹ T_i . Such code is called a $\{(B_1, T_1), (B_2, T_2)\}$ Multicast Streaming Code (Mu-SCo). The maximum of all achievable rates is the multicast streaming capacity and is denoted by $C(B_1, T_1, B_2, T_2)$.*

Remark 1. *Note that the considered model assumes a single erasure burst on each channel. However, the proposed constructions correct multiple erasure bursts provided that each erasure pattern corresponds to a burst erasure of maximum length B_1 followed by a guard interval (with no erasures) of length T_1 , or alternately corresponds to a burst of maximum length B_2 followed by a guard interval of length T_2 . Furthermore note that if the actual guard interval after the burst B_i is only of length $\tilde{T}_i < T_i$, then a $\{(B_1, \tilde{T}_1), (B_2, \tilde{T}_2)\}$ can be used over such a channel².*

Without loss of generality, we assume throughout the paper that $B_2 > B_1$. We only consider the burst-erasure channel model in this paper. More general channel models that include both burst and isolated erasures can be potentially tackled using a layered coding approach as discussed in [9]–[11]. However such extensions will not be considered in this work.

IV. BACKGROUND

To keep the paper self contained, we first briefly review the single user scenario [3]–[5]. We point the reader to these references as well as a summary in [8] for a more exhaustive treatment.

A. Single User Capacity

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

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

where T_1 and B_1 are replaced by T and B for simplicity.

The associated (B, T) Maximally-Short (MS) code construction involves the following steps.

- **Maximum Distance Separable (MDS) Code:** We start by constructing a $(T, T - B)$ systematic MDS code over a finite field \mathbb{F}_q . We note that a $(T, T - B)$ MDS code is capable of correcting B erasures in arbitrary locations (including burst erasures). The corresponding generator matrix can be expressed as,

$$\mathbf{G} = [\mathbf{I}_{T-B} \quad \mathbf{H}] \quad (5)$$

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

- **Low Delay-Burst Erasure Block Code (LD-BEBC):** We construct a systematic $(T+B, T)$ LD-BEBC from the previously constructed $(T, T - B)$ MDS Code with the generator matrix given by,

$$\mathbf{G}^* = \begin{bmatrix} \mathbf{I}_B & \mathbf{0}_{B \times (T-B)} & \mathbf{I}_B \\ \mathbf{0}_{(T-B) \times B} & \mathbf{I}_{T-B} & \mathbf{H} \end{bmatrix}, \quad (6)$$

where $\mathbf{0}_{a \times b}$ is the $a \times b$ all zeros matrix. Thus after splitting the information symbols $\mathbf{b} \in \mathbb{F}_q^T$ into two groups,

$$\mathbf{b} = (\mathbf{u}, \mathbf{n}), \quad \text{where } \mathbf{u} \in \mathbb{F}_q^B \text{ and } \mathbf{n} \in \mathbb{F}_q^{T-B}, \quad (7)$$

the resulting codeword is given by

$$\mathbf{d} = \mathbf{b} \cdot \mathbf{G}^* = (\mathbf{u}, \mathbf{n}, \mathbf{u} + \mathbf{n} \cdot \mathbf{H}) = (\mathbf{b}, \mathbf{r}), \quad (8)$$

where we have used (7) and introduced $\mathbf{r} = \mathbf{u} + \mathbf{n} \cdot \mathbf{H}$ to denote the parity check symbols in \mathbf{d} in the last step. One can show that the codeword \mathbf{d} has the property that it is capable of correcting any burst of length B with a delay of at most T symbols.

- **Diagonal Interleaving:** In this step, we convert the LD-BEBC constructed in the second step to a streaming code. We start by splitting each source symbol $\mathbf{s}[t] \in \mathbb{F}_q^T$ into T sub-symbols, i.e., $\mathbf{s}[t] = (s_0[t], \dots, s_{T-1}[t])$, where $s_j[t] \in \mathbb{F}_q$ for

¹We note that the capacity is zero whenever $T_1 < B_1$ or $T_2 < B_2$. Thus we will assume throughout that $T_i \geq B_i$ in this paper.

²A similar approach is also considered in the single user case in [11].

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

TABLE I

A (2,3) MS CODE CONSTRUCTION IS ILLUSTRATED WHERE EACH SOURCE SYMBOL $s_{\cdot}[i]$ IS DIVIDED INTO THREE SUB-SYMBOLS $s_0[\cdot]$, $s_1[\cdot]$ AND $s_2[\cdot]$ AND A (5,3) LD-BEBC CODE IS THEN APPLIED ACROSS THE DIAGONAL TO GENERATE TWO PARITY CHECK SUB-SYMBOLS GENERATING A RATE $3/5$ MS CODE. EACH COLUMN CORRESPONDS TO ONE CHANNEL SYMBOL.

$j \in \{0, \dots, T-1\}$. We then apply a $(T+B, T)$ LD-BEBC diagonally as follows. The information vector is constructed by collecting sub-symbols diagonally as follows,

$$\mathbf{b}_t = (s_0[t], s_1[t+1], \dots, s_{T-1}[t+T-1]), \quad (9)$$

and the corresponding diagonal codeword $\mathbf{d}_t = \mathbf{b}_t \mathbf{G}^* = (\mathbf{b}_t, \mathbf{r}_t)$ is then constructed according to (8). The resulting parity check sub-symbols in \mathbf{r}_t ,

$$\mathbf{r}_t = (r_0[t], \dots, r_{B-1}[t]) = (p_0[t+T], \dots, p_{B-1}[t+T+B-1]), \quad (10)$$

and

$$p_j[i] = s_j[i-T] + h_j(s_B[i-j-T+B], s_{B+1}[i-j-T+B+1], \dots, s_{T-1}[i-j-1]), \quad j = 0, \dots, B-1. \quad (11)$$

where $h_j(\mathbf{v})$ denotes the mapping produced by multiplying the vector \mathbf{v} by the j th column in \mathbf{H} in (5). The parities $p_j[i] \in \mathbb{F}_q$ are then appended diagonally to the source stream to produce the channel input stream. The channel symbol at time t is given by $\mathbf{x}[t] = (\mathbf{s}[t], \mathbf{p}[t])$, where $\mathbf{p}[t] = (p_0[t], \dots, p_{B-1}[t])$ is the parity check symbol at time t .

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

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

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

The converse is based on a periodic erasure channel (PEC) argument, similar to the upper bounding technique used in classical burst-noise channels [33, Section 6.10]. The basic idea is to amplify the effect of a single erasure burst into a periodic erasure channel and use the capacity of such a channel as an upper bound. We complement this argument with a rigorous information theoretic proof for Theorem 1 in Section VI. The information theoretic proof is more general and provides a tighter upper bound when we consider the multicast setup.

Table I illustrates a (2,3) MS code capable of correcting a burst erasure of length $B = 2$ with a delay of $T = 3$. Each source symbol is divided into $T = 3$ sub-symbols, $\mathbf{s}[i] = (s_0[i], s_1[i], s_2[i])$. A (5,3) LD-BEBC (6) of the form $(b_0, b_1, b_2, b_0+b_2, b_1+b_2)$ is then applied diagonally to the source sub-symbols, i.e., the corresponding diagonal codeword (12) is given by

$$\mathbf{d}_t = (s_0[t], s_1[t+1], s_2[t+2], s_0[t] + s_2[t+2], s_1[t+1] + s_2[t+2]) \quad (13)$$

The channel input at time t is given by

$$\mathbf{x}[t] = (s_0[t], s_1[t], s_2[t], s_0[t-3] + s_2[t-1], s_1[t-3] + s_2[t-2]). \quad (14)$$

The resulting channel input stream for $t \in [i-1, i+4]$ is illustrated in Table I. Note that the rate of this code is $\frac{T}{T+B} = \frac{3}{5}$ as it introduces two parity check sub-symbols for each three source sub-symbols.

For decoding, suppose that the channel introduces a burst erasure of length $B = 2$, i.e., $\mathbf{x}[i-1]$ and $\mathbf{x}[i]$ are erased. The decoder proceeds by recovering $s_2[i]$ and $s_2[i-1]$ at time $i+1$. At time $i+2$, $s_0[i-1]$ and $s_1[i-1]$ can be recovered both with a delay of $T = 3$. Likewise, $s_0[i]$ and $s_1[i]$ can be recovered with a delay of 3 from parity-check symbols at time $i+3$.

B. DE-SCO Construction

In earlier work [8], Badr et. al consider the proposed multicast setup when the delay of the weaker user, user 2, is sufficiently large.

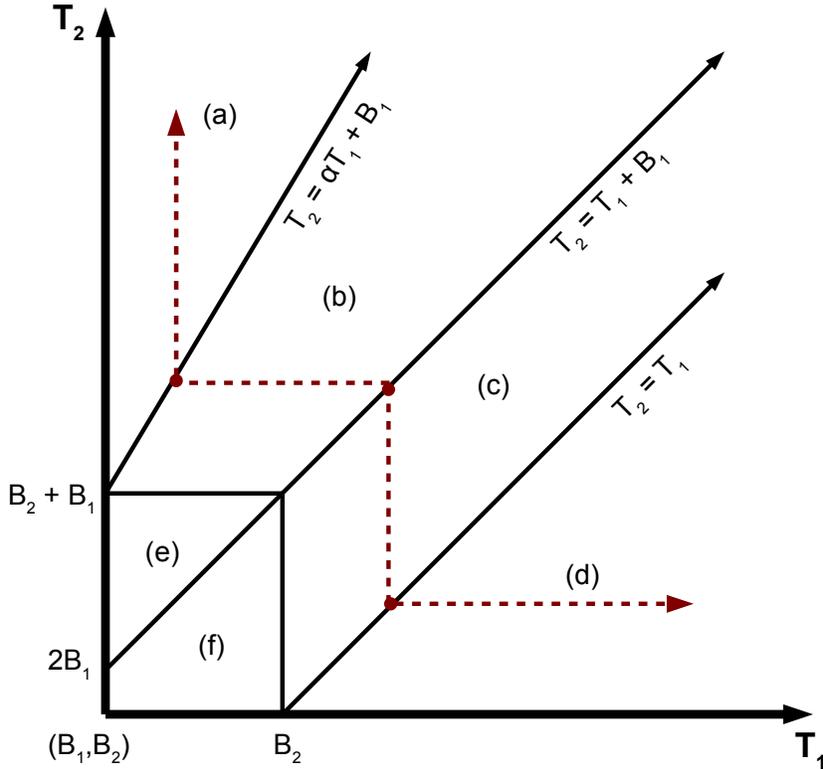


Fig. 2. Capacity behavior in the (T_1, T_2) plane. We hold B_1 and B_2 as constants with $(B_2 > B_1)$, so the regions depend on the relation between T_1 and T_2 only. The red dashed line shows the contour of constant capacity in regions (a), (b), (c) and (d).

Theorem 2 (Badr et. al [8]). *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:*

$$C_1 = \frac{T_1}{T_1 + B_1}. \quad (15)$$

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 opposite-diagonal and then combining these parity checks in a suitable manner. As mentioned previously, a key property exploited in the DE-SCo construction is that the delay T_2 is sufficiently large, so that the parity checks of the two codes when superimposed do not interfere with one another. We omit the details of the encoding and decoding steps as they are rather involved and refer the reader to [8]. A converse argument is also provided in [8] to establish that T_2 is indeed the smallest possible threshold to achieve the rate of C_1 .

V. MAIN RESULTS

We divide our results into two main regimes, the large-delay regime and the low-delay regime, which are treated separately below.

A. 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:

$$T_1 \geq B_2, \quad (\text{or}) \quad T_2 \geq B_1 + B_2. \quad (16)$$

we have been able to determine the multicast capacity as stated in Theorem 3 below. 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 (Multicast Capacity in Large-Delay Regime). *When the delays T_1 and T_2 satisfy (16) and $B_2 > B_1$ the multicast capacity is given by*

$$C = \begin{cases} C_1, & T_2 \geq \alpha T_1 + B_1, \\ \frac{T_2 - B_1}{T_2 - B_1 + B_2}, & T_1 + B_1 \leq T_2 \leq \alpha T_1 + B_1, \\ \frac{T_1}{T_1 + B_2}, & T_1 \leq T_2 \leq T_1 + B_1, \\ C_2, & T_2 \leq T_1. \end{cases} \quad (17)$$

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

The proof of Theorem 3 appears in Section VII. The achievability scheme involves a suitable application of the DE-SCo construction in Theorem 2 and single user MS codes. In particular, we exploit the following observation.

Remark 2. *In each of the four cases in (17) 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 property allows us to use code constructions at the two dominating points on each constant-capacity contour. Note that the first dominating point is on the line $T_2 = \alpha T_1 + B_1$ where the DE-SCo construction in Theorem 2 can be effectively used with certain modifications to account for possibly non-integer valued code parameters. The second dominating point is on the line $T_2 = T_1$, where a single-user MS Code in Theorem 1 can be used. We provide the details in Section VII.*

The converse is based on a periodic erasure channel (PEC) argument. By simultaneously using the decoding constraints at both the receivers we show that all erasures associated with a certain PEC can be recovered. Thus the capacity of this channel serves as an upper bound for the multicast setup. As an example, consider the point $\{(B_1, T_1), (B_2, T_2)\} = \{(2, 4), (3, 6)\}$. This point satisfies $T_2 = T_1 + B_1$ and hence its capacity, from the third case in Theorem 3, is given by $C = \frac{4}{7}$. For the converse, we consider a PEC with a period of 7, where the first three symbols are erased, and followed by four non-erased symbols. We show the first period below.

$$\cancel{x[0]} \quad \cancel{x[1]} \quad \cancel{x[2]} \quad x[3] \quad x[4] \quad x[5] \quad x[6] \quad (18)$$

We argue that using a code $C_1 = (2, 4)$ and $C_2 = (3, 6)$ we can recover all the three erasures in this period by time 6. In particular by applying C_2 , the source symbol $s[0]$, and in turn $x[0]$, is recovered by time $T = 6$. At this point the decoder is left with an erasure burst of length 2. Upon applying C_1 , the sources symbols $s[1]$ and $s[2]$ are recovered at time 5 and 6 respectively. Thus all the three erasures in the first period are recovered at time 6. By repeating this argument in each subsequent period, it follows that all the source symbols are recovered by the decoder. Thus the capacity of the above periodic erasure channel, which is $4/7$, constitutes an upper bound for the multicast setup. We note that a similar argument was also used in [8] in the proof of Theorem 2. Nevertheless we provide the details in Lemma 2 for completeness.

B. 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.,

$$B_1 \leq T_1 < B_2, \quad (\text{and}) \quad B_2 \leq T_2 < B_1 + B_2. \quad (19)$$

This regime is more challenging compared to the large-delay regime and is the main focus of the paper. We further split the low-delay regime into two regions, (e) and (f) as illustrated in Fig. 2. The capacity is characterized in region (e) as stated in Theorem 4, whereas in region (f), upper and lower bounds are provided in Theorem 5. Furthermore, the capacity in region (f) for the special cases $T_1 = B_1$ and $T_2 = B_2$ is provided in Propositions 1 and 2 respectively.

Theorem 4 (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,*

$$C_e = \frac{T_1}{2T_1 + B_1 + B_2 - T_2}. \quad (20)$$

\square

The complete proof for Theorem 4 is divided into two main parts. The achievability scheme is provided in Section VIII while the converse is given in Section IX.

The expression in (20) can be interpreted as follows. Consider the special case when $T_2 = B_2$. The rate in (20) can be attained using a simple concatenation of two single user codes, a (B_1, T_1) MS code and a (T_2, T_2) repetition code. When $T_2 \neq B_2$, the parity check streams of these codes must overlap in $T_2 - B_2$ sub-symbols in order to attain (20). Our code construction in Section VIII is based on this observation and involves embedding an additional set of parity-check symbols to clear the interference due to such overlap.

The converse in Section IX involves a new insight of revealing some of the source symbols to a virtual decoder to obtain a tighter bound than the periodic erasure channel argument used in the converse proof of Theorem 3. As an example, consider the multicast point $\{(B_1, T_1), (B_2, T_2)\} = \{(4, 5), (7, 10)\}$ that belongs to region (e). To obtain the tightest upper bound, we consider a PEC of period length 12 with the first 7 symbols in each period being being erased:

$$\cancel{x[0]} \ \cancel{x[1]} \ \cancel{x[2]} \ \cancel{x[3]} \ \cancel{x[4]} \ \cancel{x[5]} \ \cancel{x[6]} \ x[7] \ x[8] \ x[9] \ x[10] \ x[11] \quad (21)$$

The recovery at the decoder is based on the following sequence of steps using $\mathcal{C}_1 = (4, 5)$ and $\mathcal{C}_2 = (7, 10)$:

$$\begin{aligned} & s[0], s[1] \quad (\text{at time } 10, 11 \text{ using } \mathcal{C}_2) \\ & \quad s[2] \quad (\text{revealed to the decoder}) \\ & s[3], s[4], s[5], s[6] \quad (\text{at time } 8, 9, 10, 11 \text{ using } \mathcal{C}_1) \end{aligned} \quad (22)$$

Note that in the above erasure pattern, after $s[0]$ and $s[1]$ have been recovered using \mathcal{C}_2 , the decoder sees an erasure burst of length 5. To apply \mathcal{C}_2 , the decoder will need to go beyond time $t = 11$, which will result in a weaker upper bound. Similarly \mathcal{C}_1 cannot be applied as well since it can only handle an erasure burst of length 4. Our new upper bound involves *revealing* $s[2]$ to the decoder to enable it to proceed using \mathcal{C}_1 for recovering the remaining symbols. Thus for the above pattern, a total of 6 source symbols are recovered from 5 channel symbols. Thus the rate of 5/11 constitutes an upper bound.

A rigorous information theoretic proof based on the above approach is provided in Section IX.

The remainder of the low-delay regime is called region (f). For this region, we provide a general upper and lower bounds on the capacity. The capacity remains open except in the special cases of either $T_1 = B_1$ or $T_2 = B_2$.

Theorem 5 (Bounds on Capacity in Region (f)). *The multicast streaming capacity in region (f) defined by $B_2 \leq T_2 \leq T_1 + B_1$ and $B_1 \leq T_1 < B_2$ is upper and lower bounded as follows,*

$$C_f^- \leq C_f \leq C_f^+, \quad (23)$$

where the lower bound is given by,

$$C_f^- = \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \quad (24)$$

and the upper bound is given by,

$$C_f^+ = \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)}. \quad (25)$$

□

We note that the rate expression in (24) is the same as the capacity expression in region (e) in Theorem 4. The code construction is essentially the same as in Section VIII, but requires a modification in the decoder of user 1 as discussed in Section X-A. The proof of the upper bound is given in Section X-B and also involves similar arguments as that used in the converse proof of Theorem 4.

The bounds in Theorem 5 do not coincide in general. We identify special cases when each is tight as stated in the following propositions.

Proposition 1 (Capacity in Region (f) at $(T_1 = B_1)$). *The multicast streaming capacity in region (f) at the minimum delay case for user 1 ($T_1 = B_1$) is given by,*

$$C_{f(T_1=B_1)} = C_f^+. \quad (26)$$

□

To establish Proposition 1, we provide the encoding and decoding steps of the code construction achieving the rate in (26) in Section XI-A. The code is obtained by concatenating the parity-check symbols of a (T_1, T_1) repetition code and a $(B_2 - B_1, T_2 - B_1)$ MS code. The converse is already proved in Theorem 5.

Proposition 2 (Capacity in Region (f) at $(T_2 = B_2)$). *The multicast streaming capacity in region (f) at the minimum delay case for user 2 ($T_2 = B_2$) is given by,*

$$C_{f(T_2=B_2)} = C_f^- = \frac{T_1}{2T_1 + B_1}. \quad (27)$$

□

The achievability scheme is the same as that in Theorem 5 provided in Section VIII since substituting $T_2 = B_2$ in (24) gives (27). The proof of the converse part for Proposition 2 is provided in Section XI-B. The technique is significantly different than earlier converses and involves carefully double-counting the redundancy arising from the recovery of certain source symbols. To illustrate the main idea consider the point $\{(B_1, T_1), (B_2, T_2)\} = \{(2, 3), (4, 4)\}$, which belongs to region

	Region	Capacity Expression	Code Construction	Converse Proof
Large-Delay Regime $T_1 \geq B_2$, (or) $T_2 \geq B_1 + B_2$	Region (a): Theorem 3 $T_2 \geq \alpha T_1 + B_1$	$C_a = \frac{T_1}{T_1 + B_1}$	DE-SCo	PEC
	Region (b): Theorem 3 $T_1 + B_1 < T_2 < \alpha T_1 + B_1$	$C_b = \frac{T_2 - B_1}{T_2 - B_1 + B_2}$	DE-SCo + Source Expansion	PEC
	Region (c): Theorem 3 $T_1 < T_2 \leq T_1 + B_1$	$C_c = \frac{T_1}{T_1 + B_2}$	(B_2, T_1) MS Code	PEC
	Region (d): Theorem 3 $T_2 \leq T_1$	$C_d = \frac{T_2}{T_2 + B_2}$	(B_2, T_2) MS Code	PEC
Low-Delay Regime $B_1 \leq T_1 < B_2$, (and) $B_2 \leq T_2 < B_1 + B_2$	Region (e): Theorem 4 $T_2 \geq T_1 + B_1$	$C_e = \frac{T_1}{2T_1 + B_1 + B_2 - T_2}$	Partial Concatenation	Revealing
	Region (f) $T_2 < T_1 + B_1$ Theorem 5, Proposition 1, 2	$C_f^- \leq C_f \leq C_f^+$ $C_f^- = \frac{T_1}{2T_1 + B_1 + B_2 - T_2}$ $C_f^+ = \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)}$ $C_{f(T_1=B_1)} = C_f^+$ $C_{f(T_2=B_2)} = C_f^-$	Partial Concatenation - Simple Concatenation -	- Revealing - Double-Counting

TABLE II

SUMMARY OF CAPACITY EXPRESSIONS, CODE CONSTRUCTIONS AND CONVERSE PROOFS OF ALL REGIONS IN THE CONSIDERED MULTICAST MODEL WITH TWO USERS OF PARAMETERS $\{(B_1, T_1), (B_2, T_2)\}$. THE ACRONYM PEC STANDS FOR ‘‘PERIODIC ERASURE CHANNEL’’ AND $\alpha = \frac{B_2}{B_1}$.

(f). We consider a periodic erasure channel of period length $B_2 + T_1 = 7$ with the first $B_2 = 4$ symbols in each period being erased. The first period is given as follows:

$$\cancel{x[0]} \quad \cancel{x[1]} \quad \cancel{x[2]} \quad \cancel{x[3]} \quad x[4] \quad x[5] \quad x[6] \quad (28)$$

Upon using the two codes $\mathcal{C}_1 = (2, 3)$ and $\mathcal{C}_2 = (4, 4)$, the recovery sequence is as follows:

$$\begin{aligned} x[4] &\rightarrow s[0] \quad (\text{using } \mathcal{C}_2) \\ x[5] &\rightarrow s[1] \quad (\text{using } \mathcal{C}_2) \\ x[6] &\rightarrow s[2] \quad (\text{using } \mathcal{C}_2) \\ x[4], x[5] &\rightarrow s[2] \quad (\text{using } \mathcal{C}_1) \\ x[5], x[6] &\rightarrow s[3] \quad (\text{using } \mathcal{C}_1) \end{aligned} \quad (29)$$

In the above steps note that $s[2]$ has been recovered twice using two non-overlapping sets of channel symbols. In particular, \mathcal{C}_1 used $x[4]$ and $x[5]$ to recover $s[2]$ while \mathcal{C}_2 used $x[6]$ to recover $s[2]$. Hence, the 3 available channel symbols are used to recover effectively 5 source symbols (instead of 4) and the corresponding upper-bound is $\frac{3}{8}$. Recovering $s[2]$ twice using non-overlapping sets of channel symbols is what we refer to as *double-counting*. A rigorous information theoretic proof is provided to substantiate this intuition in Section XI-B.

Finally a conjecture on the capacity in region (f), which is consistent with all the special cases above, is discussed in Section XI-C.

This concludes the main results of the paper. For convenience of the reader, these are summarized in Table II.

VI. CONVERSE PROOF OF THEOREM 1

In this section we provide an information theoretic converse to Theorem 1. Our approach here will be useful in the multicast setup in subsequent sections. Furthermore we establish the upper bound in a slightly more general setup where we allow for (i) small error probability at the decoder and (ii) common randomness, independent of the source stream, at the encoder and decoder.

Let us use the following notation:

$$s \begin{bmatrix} b \\ a \end{bmatrix} = \begin{cases} s[a], s[a+1], \dots, s[b-1], s[b], & a \leq b \\ \emptyset, & \text{otherwise} \end{cases} \quad (30)$$

$$W_a^b = \begin{cases} W_a, W_{a+1}, \dots, W_{b-1}, W_b, & a \leq b \\ \emptyset, & \text{otherwise} \end{cases} \quad (31)$$

To establish the proof of Theorem 1, we consider a periodic erasure channel with a period length of $B+T$ channel symbols. In each period, the first B symbols are erased while the remaining T symbols are received at the decoder. In particular, the i -th period consists of the channel symbols $\mathbf{x}[i(T+B)], \dots, \mathbf{x}[(i+1)(T+B)-1]$ among which the first B symbols, $\mathbf{x}[i(T+B)], \dots, \mathbf{x}[i(T+B)+B-1]$ are erased whereas the T following symbols $\mathbf{x}[i(T+B)+B], \dots, \mathbf{x}[(i+1)(T+B)-1]$ are not erased.

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

$$\begin{aligned} V_i &= \mathbf{s} \begin{bmatrix} (i+1)(T+B)-1 \\ i(T+B) \end{bmatrix}, & V_{1,i} &= \mathbf{s} \begin{bmatrix} i(T+B)+B-1 \\ i(T+B) \end{bmatrix}, & V_{2,i} &= \mathbf{s} \begin{bmatrix} (i+1)(T+B)-1 \\ i(T+B)+B \end{bmatrix}, \\ W_i &= \mathbf{x} \begin{bmatrix} (i+1)(T+B)-1 \\ i(T+B) \end{bmatrix}, & W_{1,i} &= \mathbf{x} \begin{bmatrix} i(T+B)+B-1 \\ i(T+B) \end{bmatrix}, & W_{2,i} &= \mathbf{x} \begin{bmatrix} (i+1)(T+B)-1 \\ i(T+B)+B \end{bmatrix}, \end{aligned} \quad (32)$$

where $i \in \{0, 1, 2, \dots\}$. Note that $V_i = [V_{1,i}, V_{2,i}]$ refer to a group of source symbols, whereas $W_i = [W_{1,i}, W_{2,i}]$ is a group of channel symbols in the i -th period. We also note that the channel symbol at any time t is a causal function of source symbols, i.e.,

$$\mathbf{x}[t] = f_t(s[0], \dots, s[t], \mathbf{M}), \quad (33)$$

where \mathbf{M} is common randomness at encoder and decoder³. Fig. 3 shows the time slots that the symbols come from as well as the size of V_i and W_i .

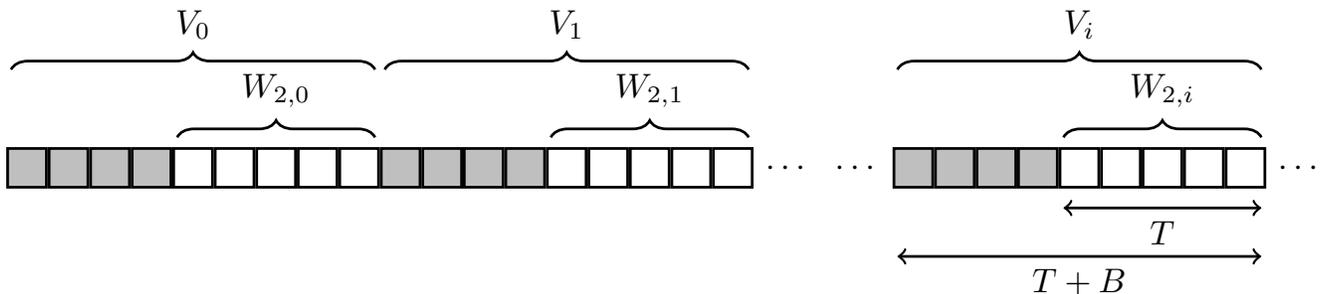


Fig. 3. The periodic erasure channel used in proving the upper bound of the single user scenario in Theorem 1, with indication of which symbols are in the groups V_i and W_i . Grey and white squares denote erased and unerased symbols respectively.

Lemma 1. Consider a (B, T) code which is capable of recovering each source symbol over a burst erasure channel, which introduces an erasure burst of maximum length B , with a delay of T symbols and a maximum error probability of ϵ , i.e., each output symbol

$$\hat{\mathbf{s}}[t] = g_t(\mathbf{y}[0], \dots, \mathbf{y}[t+T], \mathbf{M}) \quad (34)$$

satisfies

$$\Pr(\mathbf{s}[t] \neq \hat{\mathbf{s}}[t]) \leq \epsilon, \quad \forall t \geq 0. \quad (35)$$

The following conditions hold for all $n \geq i+1$,

$$H(V_{1,i} | W_0^{i-1}, W_{2,0}^n, \mathbf{M}) \leq B(H(\epsilon) + \epsilon \log |\mathcal{S}|), \quad (36)$$

$$H(V_{2,i} | W_{1,i}, W_0^{i-1}, W_{2,0}^n, \mathbf{M}) \leq T(H(\epsilon) + \epsilon \log |\mathcal{S}|), \quad (37)$$

where $V_{1,i}$, $V_{2,i}$, W_i , $W_{1,i}$ and $W_{2,i}$ is as in (32).

Proof: See Appendix A. ■

Eq. (36) considers an erasure burst that spans the interval $[i(T+B), i(T+B)+B-1]$, and uses the fact that the associated source symbols $V_{1,i}$ must be recovered when $W_{2,i}$ is received. By applying Fano's inequality to each of the B symbols in $V_{1,i}$ we obtain (36). Eq. (37) considers the recovery of the source symbols $V_{2,i}$ when the erasure burst spans $W_{1,i+1}$. Upon receiving of $W_{2,i+1}$ each of the T symbols in $V_{2,i}$ must be recovered. In general, we do not need all of channel symbols $W_{2,0}^n$ in the conditioning, but the proof is simpler if we assume that all are available.

³We note that the same upper-bound in (38) is achieved when no common randomness is assumed. Hence, we will only consider deterministic codes and \mathbf{M} will be dropped in the subsequent converse proofs.

We now consider the following chain of inequalities:

$$\begin{aligned}
n(T+B)\log|\mathcal{S}| &= H(V_0^{n-1}) \stackrel{(a)}{=} H(V_0^{n-1}|\mathbf{M}) \leq H(V_0^{n-1}, W_{2,0}^n|\mathbf{M}) \\
&= H(W_{2,0}^n|\mathbf{M}) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|V_0^{i-1}, W_{2,0}^n, \mathbf{M}) + H(V_{2,i}|V_{1,i}, V_0^{i-1}, W_{2,0}^n, \mathbf{M}) \right) \\
&\stackrel{(b)}{=} H(W_{2,0}^n|\mathbf{M}) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|W_0^{i-1}, W_{2,0}^n, \mathbf{M}) + H(V_{2,i}|W_{1,i}, W_0^{i-1}, W_{2,0}^n, \mathbf{M}) \right) \\
&\stackrel{(c)}{\leq} H(W_{2,0}^n|\mathbf{M}) + n(T+B)(H(\epsilon) + \epsilon \log|\mathcal{S}|) \\
&\stackrel{(d)}{\leq} H(W_{2,0}^n) + n(T+B)(H(\epsilon) + \epsilon \log|\mathcal{S}|) \\
&\leq (n+1)T \cdot \log|\mathcal{X}| + n(T+B)(H(\epsilon) + \epsilon \log|\mathcal{S}|), \tag{38}
\end{aligned}$$

where (a) holds due to independence of source symbols on the random variable \mathbf{M} , (b) follows by using (33), (c) follows by using Lemma 1 and (d) follows from the fact that conditioning reduces entropy.

Finally, we conclude from (38) that the rate of any (B, T) streaming erasure code must satisfy

$$R = \frac{\log|\mathcal{S}|}{\log|\mathcal{X}|} \leq \frac{n+1}{n} \cdot \frac{T}{T+B} + \frac{H(\epsilon) + \epsilon \log|\mathcal{S}|}{\log|\mathcal{X}|} \xrightarrow{n \rightarrow \infty, \epsilon \rightarrow 0} \frac{T}{T+B}, \tag{39}$$

which gives the required upper bound on the rate.

VII. MULTICAST CAPACITY IN LARGE-DELAY REGIME (THEOREM 3)

We discuss in turn the achievability and converse of Theorem 3 in this section.

A. Achievability

The case in region (a) where the capacity equals $C_1 = \frac{T_1}{T_1+B_1}$, was already addressed in Theorem 2 established in [8] using the DE-SCo construction. Region (c), sandwiched between $T_1 \leq T_2 \leq T_1 + B_1$ and $T_1 \geq B_2$ in Fig. 2 satisfies

$$C_c = \frac{T_1}{T_1 + B_2}. \tag{40}$$

In this region we can use a single user (B_2, T_1) MS code that simultaneously satisfies both the users. Clearly this code is feasible since $T_1 \geq B_2$. It satisfies user 1 since $B_2 > B_1$ and user 2 since $T_2 \geq T_1$. Similarly in region (d), defined by $T_2 \leq T_1$ and $B_2 \geq B_1$, it suffices to serve user 2 using a single user MS code of parameters (B_2, T_2) .

Thus the only remaining region of the large-delay regime in Fig. 2 is region (b). Recall that the capacity here is $C_b = \frac{T_2 - B_1}{T_2 - B_1 + B_2}$ and $T_1 + B_1 \leq T_2 \leq \alpha T_1 + B_1$ holds. Since the capacity does not depend on T_1 , we can keep reducing the value of T_1 to \tilde{T}_1 such that

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

where $\alpha = \frac{B_2}{B_1}$. This is equivalent to

$$\tilde{T}_1 = \frac{B_1}{B_2}(T_2 - B_1). \tag{41}$$

Provided that $\tilde{T}_1 \geq B_1$, and furthermore \tilde{T}_1 is an integer, we can use a $\{(B_1, \tilde{T}_1), (B_2, T_2)\}$ DE-SCo [8] 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 $T_2 \geq B_2 + B_1$ which is satisfied by every point in region (b). If \tilde{T}_1 is not an integer, a suitable *source expansion* is needed, where we split each source stream into sufficiently many, say M sub-streams such that $M\tilde{T}_1$ is an integer. We can then construct a $\{(MB_1, M\tilde{T}_1), (MB_2, MT_2)\}$ DE-SCo and show that it is feasible on the original channel. Details are relegated to Appendix B.

Example: We consider the point $\{(B_1, T_1), (B_2, T_2)\} = \{(1, 2), (2, 4)\}$, which belongs to region (b) in Theorem 3. The capacity at this point is $C = 3/5$. We note that this point is dominated by the point $\{(B_1, T'_1), (B_2, T_2)\} = \{(1, 1.5), (2, 4)\}$ which belongs to the line $T_2 = \frac{B_2}{B_1}T_1 + B_1$. However we cannot directly construct a DE-SCo for this point as T'_1 is not integer valued. Instead we construct a DE-SCo with parameters $\{(2, 3), (4, 8)\}$ with rate $3/5$ and observe using the source expansion technique in Appendix B, that the resulting code can be mapped to the original parameters.

B. Converse

For the converse we start by establishing an upper bound on the multicast streaming capacity in the large-delay regime as follows,

Lemma 2. 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

$$C^+ = \begin{cases} \frac{T_2 - B_1}{T_2 - B_1 + B_2} & T_2 > T_1 + B_1, \\ \frac{T_1}{T_1 + B_2} & T_2 \leq T_1 + B_1, \end{cases} \quad (42)$$

Proof: The proof of Lemma 2 is provided in Appendix C. It involves simultaneously using the decoding constraints of both the receivers and a periodic erasure channel (PEC) argument similar to the single-user case. ■

We further tighten the upper bound in (42) using the fact that the multicast streaming capacity cannot exceed the single user capacity on any of the two links:

$$C^U = \min \{C^+, C_1, C_2\} = \begin{cases} C_1 \triangleq C_a, & T_2 \geq \alpha T_1 + B_1 \\ \frac{T_2 - B_1}{T_2 - B_1 + B_2} \triangleq C_b, & T_1 + B_1 < T_2 < \alpha T_1 + B_1 \\ \frac{T_1}{T_1 + B_2} \triangleq C_c, & T_1 < T_2 \leq T_1 + B_1 \\ C_2 \triangleq C_d, & T_2 \leq T_1. \end{cases} \quad (43)$$

where recall that $\alpha = \frac{B_2}{B_1}$. This completes the proof of the converse.

VIII. ACHIEVABILITY SCHEME IN REGION (E) (THEOREM 4)

We show that for any point in region (e) in Fig. 2 that satisfies, $T_1 + B_1 \leq T_2 \leq B_2 + B_1$ and $B_1 \leq T_1 < B_2$, there exists a multicast code that achieves the following rate:

$$C_e = \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \quad (44)$$

Towards this end we parametrize the T_2 and B_2 as

$$T_2 = T_1 + B_1 + m, \quad B_2 = T_1 + k + m, \quad m \geq 0, k \in [0, B_1] \quad (45)$$

Substituting into (44), the capacity simplifies to

$$C_e = \frac{T_1}{2T_1 + k}. \quad (46)$$

In our construction we first consider two codes, one for each user as follows.

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

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

- Let \mathcal{C}_1 be a (B_1, T_1) MS code, as described in Section IV applied to the source symbols $\mathbf{s}[i]$ producing B_1 parity check sub-symbols

$$\mathbf{p}^1[i] = (p_0^1[i], \dots, p_{B_1-1}^1[i]) \quad (47)$$

at each time by combining the source sub-symbols diagonally:

$$p_j^1[i] = s_j[i - T_1] + h_j^1(s_{B_1}[i - j - T_1 + B_1], \dots, s_{T_1-1}[i - j - 1]), \quad j = \{0, 1, \dots, B_1 - 1\}, \quad (48)$$

where recall that $h_j^1(\cdot)$ denote a linear combination of the sub-symbols $s_{B_1}[i - j - T_1 + B_1], \dots, s_{T_1-1}[i - j - 1]$ as in (11).

- Let \mathcal{C}_2 be simple repetition code applied to the source symbols $\mathbf{s}[i]$ with a delay of T_2 , i.e.

$$\mathbf{p}^2[i] = (p_0^2[i], \dots, p_{T_1-1}^2[i]) = (s_0[i - T_2], \dots, s_{T_1-1}[i - T_2]) = \mathbf{s}[i - T_2]. \quad (49)$$

- Concatenate the two streams $\mathbf{p}^1[\cdot]$ and $\mathbf{p}^2[\cdot]$ in (47) and (49) with a *partial* overlap of

$$T_2 - B_2 = B_1 - k$$

symbols as illustrated in (50). The two streams of parity checks $\mathbf{p}^1[\cdot]$ and $\mathbf{p}^2[\cdot]$ are concatenated with the last $B_1 - k$ rows of the first added to upper most $B_1 - k$ rows of the second.

$$\tilde{\mathbf{x}}[i] = \begin{bmatrix} s_0[i] \\ \vdots \\ s_{T_1-1}[i] \\ p_0^1[i] \\ \vdots \\ p_{k-1}^1[i] + s_0[i - T_2] \\ \vdots \\ p_{B_1-1}^1[i] + s_{B_1-k-1}[i - T_2] \\ s_{B_1-k}[i - T_2] \\ \vdots \\ s_{T_1-1}[i - T_2] \end{bmatrix} \quad (50)$$

Note that $\tilde{\mathbf{x}}[i]$ consists of a total of T_1 sub-symbols of $\mathbf{s}[i]$, B_1 parity-check sub-symbols of $\mathbf{p}^1[i]$ and T_1 parity-check sub-symbols of $\mathbf{p}^2[i]$ with an overlap of $B_1 - k$ sub-symbols. Thus the rate associated with $\tilde{\mathbf{x}}[i]$ satisfies (46).

As we will see, the above construction needs to be further extended by embedding a third code onto the non-overlapping parity checks of \mathcal{C}_2 . To motivate this we begin by considering the decoding at both the receivers.

A. Decoding at Receiver 1

At receiver 1 suppose that the erasure burst spans the interval $[i - B_1, i - 1]$. We need to reconstruct each $\mathbf{s}[t]$ for $t \in [i - B_1, i - 1]$ with a delay of T_1 . The parity checks $\mathbf{p}^1[\cdot]$ that will be required at the decoder span $\mathcal{I}_1 = [i, i + T_1 - 1]$. We claim that the overlapping parity checks $\mathbf{s}_2[t]$ for $t \in \mathcal{I}_1$, as illustrated in (50) are not erased and hence can be cancelled out. In particular, consider $t = i$. The overlapping $\mathbf{s}[i - T_2]$ is clearly not erased since $T_2 \geq B_2 > B_1$, by definition. At the other extreme when $t = i + T_1 - 1$, we have:

$$i + T_1 - T_2 - 1 \leq i - B_1 - 1 \quad (51)$$

since $T_2 \geq B_1 + T_1$ in region (e). Thus $\mathbf{s}[t - T_2]$ is again associated before the start of the erasure burst and is not erased. Thus it follows that for each $t \in \mathcal{I}_1$ the overlapping $\mathbf{s}[t - T_2]$ in (50) is not erased and can be cancelled out to recover $\mathbf{p}^1[t]$ with no additional delay. At this point one can use the decoder associated with \mathcal{C}_1 , which is a (B_1, T_1) MS Code, to recover each erased source symbol with a delay of T_1 .

B. Decoding at Receiver 2

At receiver 2 we suppose that the erasure burst spans the interval $[i - B_2, i - 1]$. We are required to reconstruct each $\mathbf{s}[t]$ for $t \in [i - B_2, i - 1]$ with a delay of T_2 . Recall that the parity check $\mathbf{p}^2[t] = \mathbf{s}[t - T_2]$ is merely a repetition code with a shift of T_2 . Thus if we can cancel the overlapping $\mathbf{p}^1[t]$ for $t \in [i - B_2 + T_2, i + T_2 - 1]$, we can recover each erased source symbol with a delay of T_2 . Now note that using (45) we have that $T_2 - B_2 = B_1 - k$ and $0 \leq k \leq B_1$ holds. We can partition the interval $[i, i + T_2 - 1]$ into three sub-intervals as follows:

- $\mathcal{J}_1 = [i, i + T_2 - B_2 - 1]$. For each $t \in \mathcal{J}_1$ we have that $t - T_2 \leq i - B_2 - 1$, and thus the overlapping $\mathbf{s}[t - T_2]$ is not erased. Thus one can cancel the overlapping $\mathbf{s}[t - T_2]$ and recover $\mathbf{p}^1[t]$ in (50). Note that in this interval the portion of $\mathbf{s}[t - T_2]$ that does not overlap with $\mathbf{p}^1[\cdot]$ is not used in the recovery.
- $\mathcal{J}_2 = [i + T_2 - B_2, i + T_1 - 1]$. For each $t \in \mathcal{J}_2$ we have that both $\mathbf{s}[t - T_2]$ are erased and the overlapping $\mathbf{p}^1[t]$ involve source symbols in the interval $[t - T_1, t - 1]$, which may be erased. In this case one cannot recover sub-symbols $s_j[t - T_2]$ for $j = 0, \dots, B_1 - k - 1$. However we recover $s_j[t - T_2]$ for $j = B_1 - k, \dots, T_1 - 1$. Thus a subset of $\mathbf{s}[t - T_2]$ cannot be recovered.
- $\mathcal{J}_3 = [i + T_1, i + T_2 - 1]$. For each $t \in \mathcal{J}_3$, we have that $\mathbf{p}^1[t]$ involves source symbols in the interval $[t - T_1, t - 1]$, since the memory of the code \mathcal{C}_1 is T_1 (see Section IV). Since these source symbols appear after the erasure burst and hence are erased, one can cancel the overlapping $\mathbf{p}^1[t]$ and thus recover $\mathbf{p}^2[t] = \mathbf{s}[t - T_2]$. Thus it follows that the erased source symbol $t - T_2$ is recovered from (50).

To summarize the above steps, when the erasure burst spans the interval $[i - B_2, i - 1]$ at receiver 2, the code construction in (50) is not able to recover source sub-symbols $s_j[t - T_2]$ for $j \in \{0, \dots, B_1 - k - 1\}$ and $t \in \{i + B_1 - k, \dots, i + T_1 - 1\}$. These $(B_1 - k) \cdot (T_1 - B_1 + k)$ sub-symbols are illustrated by the shaded box in Fig. 4. Secondly, in the interval $\mathcal{J}_1 = [i, i + B_1 - k - 1]$, all the parity check symbols $\mathbf{p}^2[t]$ correspond to the source symbols that are not erased. The total number of sub-symbols that do not overlap with $\mathbf{p}^1[t]$, illustrated by the region with hatched lines in Fig. 4 also equals $(B_1 - k) \cdot (T_1 - B_1 + k)$, the same

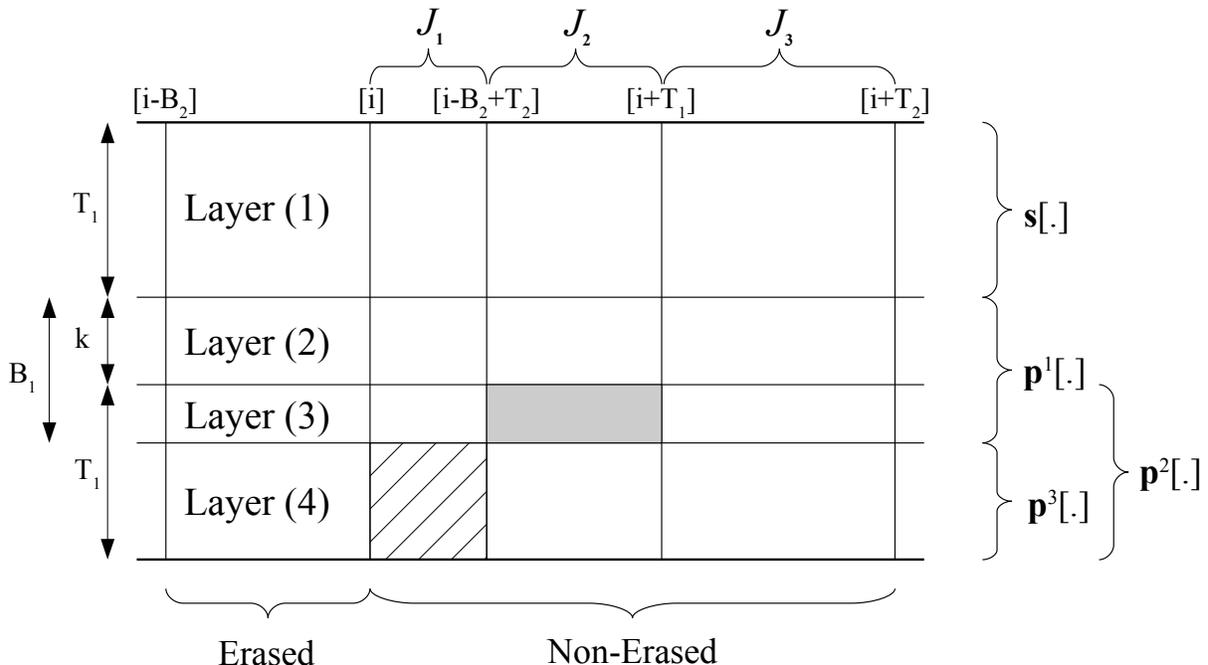


Fig. 4. A graphical illustration of the structure of the code construction. The labels on the right show the layers spanned by each set of parity check sub-symbols. The labels at the bottom show the intervals in which each set of parity check sub-symbols combine erased source sub-symbols. Note that the construction $\tilde{\mathbf{x}}[i]$ in (50) involves an overlap between $\mathbf{p}^1[\cdot]$ and $\mathbf{p}^2[\cdot]$ as shown. The shaded symbols cannot be recovered at user 2. To recover these we use a third layer of parity check symbols $\mathbf{p}^3[\cdot]$ that are embedded in the last $T_1 - (B_1 - k)$ rows as shown.

as the number of shaded sub-symbols. We next describe our code construction that uses these available positions to recover the missing source symbols.

C. Construction of \mathcal{C}_3

We extend our construction of $\tilde{\mathbf{x}}[i]$ to incorporate a third set of parity check symbols $\mathbf{q}[i] = (q_0[i], \dots, q_{T_1 - (B_1 - k) - 1}[i])$ so that the transmitted symbol $\mathbf{x}[i]$ is expressed as follows:

$$\mathbf{x}[i] = \begin{bmatrix} s[i] \\ p_0^1[i] \\ \vdots \\ p_{k-1}^1[i] + s_0[i - T_2] \\ \vdots \\ p_{B_1-1}^1[i] + s_{B_1-k-1}[i - T_2] \\ s_{B_1-k}[i - T_2] + q_0[i] \\ \vdots \\ s_{T_1-1}[i - T_2] + q_{T_1 - (B_1 - k) - 1}[i] \end{bmatrix} \quad (52)$$

We show that by judiciously selecting $\mathbf{q}[\cdot]$, these parity check symbols in \mathcal{J}_1 can be used to recover the parity symbols $\mathbf{p}^1[t]$ for $t \in \mathcal{J}_2$ and ultimately all the source symbols. As remarked before, the number of available parity sub-symbols of $\mathbf{q}[\cdot]$ in \mathcal{J}_1 , which are denoted by the region with the hatched lines is sufficient to recover the erased sub-symbols of $\mathbf{p}^1[\cdot]$ in the interval \mathcal{J}_2 , which are shaded in Fig. 4. However in the construction of $\mathbf{q}[\cdot]$ one needs to satisfy two additional properties in the streaming construction.

- *Causality*: Each $\mathbf{q}[t]$ must only depend on the source symbols up to time t .

- *Delay Constraint:* Each $\mathbf{p}^1[t]$ for $t \in \mathcal{J}_2$ must be reconstructed with zero delay, so that the underlying source symbol $\mathbf{s}[t - T_2]$ can be recovered with a delay of T_2 .

We discuss the construction of $\mathbf{q}[\cdot]$ separately in the cases when $T_1 \leq 2(B_1 - k)$, and when $T_1 > 2(B_1 - k)$ below.

1) *Case (A) $T_1 \leq 2(B_1 - k)$:* We define $T_3 = B_1 - k$ and $B_3 = T_1 - (B_1 - k)$, and consider a code \mathcal{C}_3 that is a (B_3, T_3) single-user MS code applied to the last $B_1 - k$ parity check symbols of $\mathbf{p}^1[\cdot]$ i.e., to $(p_k^1[i], \dots, p_{B_1-1}^1[i])$. Such a code is feasible since $B_3 \leq T_3$, due to $T_1 \leq 2(B_1 - k)$. Thus we let

$$\mathbf{p}^3[i] = (p_0^3[i], \dots, p_{T_1-(B_1-k)-1}^3[i]) \quad (53)$$

by combining the last $B_1 - k$ parity check sub-symbols, $(p_k^1[\cdot], \dots, p_{B_1-1}^1[\cdot])$, diagonally, i.e.,

$$p_j^3[i] = p_{k+j}^1[i - T_3] + h_j^3(p_{k+B_3}^1[i - j - T_3 + B_3], \dots, p_{k+T_3-1}^1[i - j - 1]), \quad (54)$$

for $j = \{0, 1, \dots, T_1 - (B_1 - k) - 1\}$ and $h_j^3(\cdot)$ involves the linear combination associated with the MS code, as defined in (11).

We note that the code \mathcal{C}_3 is a MS code. It can recover a burst of length B_3 , spanning the interval $\mathcal{J}_2 = [i + T_3, i + T_3 + B_3 - 1]$ with a delay of T_3 , provided the following conditions are satisfied:

- The associated parity check symbols $\mathbf{p}^3[t]$ in the interval $t \in [i + T_1, i + T_1 + T_3 - 1]$ are available.
- The symbols $\mathbf{p}^1[t]$ for $t \in \mathcal{J}_1 = [i, i + T_3 - 1]$ which is the period of length $T_3 = T_2 - B_2$ symbols preceding the burst are available to the decoder.

As noted before, the interfering symbols of $\mathbf{p}^2[t]$ for $t \in \mathcal{J}_1$ can be cancelled out and thus $\mathbf{p}^1[t]$ for $t \in \mathcal{J}_1$ are available. Furthermore, in order to retrieve the required $\mathbf{p}^3[t]$ from the hatched lines in Fig. 4, we apply a backward shift of T_1 source symbols and let

$$\mathbf{q}[t] = \mathbf{p}^3[t + T_1] \quad (55)$$

With the above mapping, each required $\mathbf{p}^3[t]$ for $t \in [i + T_1, i + T_1 + T_3 - 1]$ can be retrieved from the corresponding $\mathbf{q}[t - T_1]$, which span the interval \mathcal{J}_1 and hence are available. However the choice of $\mathbf{q}[t]$ in (55) does not satisfy the causality condition. This is because $\mathbf{p}^3[t + T_1]$ can potentially depend on source symbols after time t whereas $\mathbf{q}[t]$ must only depend on the source symbols up to time t . Therefore we need to modify (55) to only send the *causal* part of $\mathbf{p}^3[t + T_1]$ at time t as discussed next.

Definition 2 (Causal and Non-Causal Parts of a Parity-Check). *Consider a linear parity check sub-symbol $p_j[t_1]$ generated over source stream $\mathbf{s}[t]$. For any $t_2 \leq t_1$ we can express*

$$p_j[t_1] = \overleftarrow{p}_j[t_1]_{t_2} + \overrightarrow{p}_j[t_1]_{t_2} \quad (56)$$

where $\overleftarrow{p}_j[t_1]_{t_2}$ and $\overrightarrow{p}_j[t_1]_{t_2}$ denote the causal and non-causal parts of the parity check $p_j[t_1]$ with respect to t_2 , respectively.

The causal part, $\overleftarrow{p}_j[t_1]_{t_2}$, is obtained by replacing all source symbols $\mathbf{s}[t]$ from time $t > t_2$ with zeros in $p_j[t_1]$, whereas the non-causal part, $\overrightarrow{p}_j[t_1]_{t_2}$, is obtained by replacing all source symbols from time $t \leq t_2$ with zeros in $p_j[t_1]$.

Recall that in (56) we use the fact that $p_j[t_1]$ is a linear combination of source sub-symbols (see (11)).

Example 1. *Consider the parity check $p_0[5] = s_0[1] + s_1[2] + s_2[3] + s_3[4]$. The causal and non-causal parts with respect to $t_2 = 2$ are given by,*

$$\begin{aligned} \overleftarrow{p}_0[5]_2 &= s_0[1] + s_1[2] \\ \overrightarrow{p}_0[5]_2 &= s_2[3] + s_3[4], \end{aligned} \quad (57)$$

respectively. In particular, $\overleftarrow{p}_0[5]_2$ is equal to $p_0[5]$ after removing all source sub-symbols later than time $t = 2$ whereas $\overrightarrow{p}_0[5]_2$ is the same but after removing all source sub-symbols from time before and including time $t = 2$.

Hence, we replace the parity-check symbol $\mathbf{p}^3[i + T_1]$ with its causal part $\overleftarrow{\mathbf{p}}^3[i + T_1]_i$ in (55) as discussed in Definition 2, i.e.,

$$\mathbf{q}[i] = \overleftarrow{\mathbf{p}}^3[i + T_1]_i = \left(\overleftarrow{p}_0^3[i + T_1]_i, \dots, \overleftarrow{p}_{T_1-(B_1-k)-1}^3[i + T_1]_i \right). \quad (58)$$

We now revisit the decoding analysis for user 2 in Section VIII-B. Suppose that the symbols in the interval $\mathcal{I}_2 = [i - B_2, i - 1]$ are erased by the channel of user 2. The main steps at the decoder are summarized as follows. We refer to the four layers as illustrated in Fig. 4.

- **Step (1) (Recovery of $\mathbf{p}^1[\cdot]$):** The parity checks $\overleftarrow{\mathbf{p}}^3[t + T_1]_t$ in the interval $t \in \mathcal{J}_1 = [i, i + B_1 - k - 1]$ (which correspond to the hatched region in Fig. 4), are used to recover the last $B_1 - k$ sub-symbols of $\mathbf{p}^1[t]$ for $t \in \mathcal{J}_2 = \{i + B_1 - k, \dots, i + T_1 - 1\}$ by time t . These correspond to the shaded symbols in Fig 4.
- **Step (2) (Removal of $\mathbf{p}^1[\cdot]$):** Subtract $\mathbf{p}^1[t]$ in layer (3) in the interval $t \in \mathcal{J}_2 \cup \mathcal{J}_3 = [i + B_1 - k, i + T_2 - 1]$ and recover the underlying sub-symbols of $\mathbf{p}^2[t]$ that overlap with them.

- **Step (3) (Removal of $\mathbf{p}^3[\cdot]$):** Compute and subtract $\mathbf{p}^3[t]$ in layer (4) in the interval $t \in \mathcal{J}_2 \cup \mathcal{J}_3 = [i + B_1 - k, i + T_2 - 1]$ and recover the underlying sub-symbols of $\mathbf{p}^2[t]$ that overlap with them.
- **Step (4) (Recovery using $\mathbf{p}^2[\cdot]$):** Use $\mathbf{p}^2[t]$ for $t \in [i + B_1 - k, i + T_2 - 1]$ to recover the erased source symbols, $(s[i - B_2], \dots, s[i - 1])$.

Having summarized the four steps, we provide a justification for each of them below. Step (1) is the most elaborate step and is established in Appendix D. We summarize it in the following lemma.

Lemma 3. *When the erasure burst spans the interval $\mathcal{I}_2 = [i - B_2, i - 1]$, the decoder at receiver 2 can recover each of the overlapping parity sub-symbols, $p_j^1[t]$, for $t \in \mathcal{J}_2 = [i + B_1 - k, i + T_1 - 1]$ and $j \in \{k, \dots, B_1 - 1\}$ by time t , using $\overleftarrow{p}_j^3[t + T_1]|_t$ for $t \in \mathcal{J}_1 = [i, i + B_1 - k - 1]$ and the unerased source symbols starting from time i .*

Proof: See Appendix D. ■

Step (2) (Removal of $\mathbf{p}^1[\cdot]$): Once the overlapping parity check sub-symbols of \mathcal{C}_1 in the interval $\mathcal{J}_2 = [i + B_1 - k, i + T_1 - 1]$ have been recovered in Step (1), they can be cancelled to recover the parity check sub-symbols of \mathcal{C}_2 . Those symbols $\mathbf{p}^1[t]$ appearing in the interval $t \in \mathcal{J}_3 = [i + T_1, i + T_2 - 1]$ are clearly functions of unerased source symbols (since the underlying MS code has a memory of T_1); these can be computed by the decoder, and cancelled at time t to recover the parity check sub-symbols of \mathcal{C}_2 .

Step (3) (Removal of $\mathbf{p}^3[\cdot]$): Recall that \mathcal{C}_3 is a (B_3, T_3) MS code with a memory of $T_3 = B_1 - k$. We show that the interfering $\mathbf{p}^3[\cdot]$ in the interval $\mathcal{J}_2 \cup \mathcal{J}_3 = [i + T_3, i + T_2 - 1]$ can be computed by the decoder and cancelled. Consider the parity check at time $i + T_3$, $\overleftarrow{\mathbf{p}}^3[i + B_1 - k + T_1]|_{i+B_1-k}$. Following (54), it involves parity check sub-symbols $\mathbf{p}^1[t]$ of \mathcal{C}_1 of time

$$i + B_1 - k - T_3 + T_1 = i + T_1$$

and later. In computing the above we subtract T_3 from the time index, as this is the memory of the code, and add T_1 as this corresponds to the backward shift. Furthermore as argued in Step (2) above, the parity check sub-symbols of \mathcal{C}_1 at time $i + T_1$ and later only combine unerased source symbols. Thus the overlapping $\mathbf{p}^3[t]$ can be computed and cancelled out as claimed. In a similar fashion all other such symbols in the interval $[i + T_3, i + T_2 - 1]$ can be cancelled out.

Step (4) (Recovery using $\mathbf{p}^2[\cdot]$): Using the previous two steps each $p_j^2[t]$ for $j \in \{0, \dots, T_1 - 1\}$ and $t \in \{i + B_1 - k, \dots, i + T_2 - 1\}$ can be recovered by time t . Since $\mathbf{p}^2[i + B_1 - k] = \mathbf{s}[i + B_1 - k - T_2] = \mathbf{s}[i - B_2]$ it follows that each erased source symbol can be recovered with a delay of T_2 .

This completes the decoding analysis at receiver 2. Since the analysis of decoder 1 remains unchanged, the achievability proof is complete when $T_1 \leq 2(B_1 - k)$ holds.

2) *Case (B): $T_1 > 2(B_1 - k)$:* To complete the proof, it remains to consider the case $T_1 > 2(B_1 - k)$. In this case $B_1 - k < T_1 - (B_1 - k)$, therefore the MS code with $B_3 = T_1 - (B_1 - k)$, and $T_3 = B_1 - k$ constructed before is no longer feasible. We begin by expressing $T_1 - (B_1 - k)$ as follows:

$$T_1 - (B_1 - k) = r(B_1 - k) + q, \quad 0 \leq q < (B_1 - k) \text{ and } r \geq 1. \quad (59)$$

We construct a total of $r + 1$ codes $\mathcal{C}_{3,n}$ as follows. For $n = 1, \dots, r$ we let $\mathcal{C}_{3,n}$ to be a $(B_1 - k, B_1 - k)$ repetition code applied to the last $B_1 - k$ sub-symbols in $\mathbf{p}^1[i]$, shifted back by $n(B_1 - k)$ i.e., the corresponding parity check symbols are given by:

$$\mathbf{p}^{3,n}[i] = (p_0^{3,n}[i], \dots, p_{B_1-k-1}^{3,n}[i]) = (p_k^1[i + n(B_1 - k)], \dots, p_{B_1-1}^1[i + n(B_1 - k)]). \quad (60)$$

Let $\mathcal{C}_{3,r+1}$ be a $(B_{3,r+1}, T_{3,r+1}) = (q, B_1 - k)$ MS code again applied on the last $B_1 - k$ parity check sub-symbols $(p_k^1[i], \dots, p_{B_1-1}^1[i])$ 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, $(p_k^1[\cdot], \dots, p_{B_1-1}^1[\cdot])$ as in (54) after replacing B_3 and T_3 with $B_{3,r+1}$ and $T_{3,r+1}$ respectively.

Concatenate the set of streams $\mathbf{p}^{3,n}[\cdot]$ for $n = 1, \dots, r$ and $\mathbf{p}^{3,r+1}[\cdot]$ after introducing a shift of $\Delta_{3,r+1} = -T_1$ in the later and keeping its causal part. The output symbol at time i is as in (52) where,

$$\mathbf{q}[i] = (q_0[i], \dots, q_{T_1-(B_1-k)-1}[i]) = (\overleftarrow{\mathbf{p}}^{3,1}[i]|_i, \dots, \overleftarrow{\mathbf{p}}^{3,r}[i]|_i, \overleftarrow{\mathbf{p}}^{3,r+1}[i + T_1]|_i) \quad (61)$$

is the concatenation of the causal part of the $r + 1$ parity check sub-streams for the codes $\mathcal{C}_{3,n}$ for $n = 1, \dots, r + 1$, respectively. We let \mathcal{C}_3 be the result of concatenating $\{\mathcal{C}_{3,1}, \dots, \mathcal{C}_{3,r+1}\}$ and let $\mathbf{p}^3[t]$ be the result of concatenating the parity sub-symbols as in (61).

In the decoding analysis we only need to revisit steps (1) and (3) in the case $T_1 \leq 2(B_1 - k)$ above. We show that Lemma 3 continues to hold for the above construction in Appendix D, i.e., each $\mathbf{p}^1[t]$ for $t \in [i + B_1 - k, i + T_1 - 1]$ can be decoded with zero delay using the parity sub-symbols of $\mathbf{p}^3[t]$ in the interval $t \in [i, i + B_1 - k - 1]$.

In step (3) we need to show that each $\mathbf{p}^3[t]$ for $t \in [i + B_1 - k, i + T_2 - 1]$ can be cancelled to recover the associated $\mathbf{p}^2[t]$.

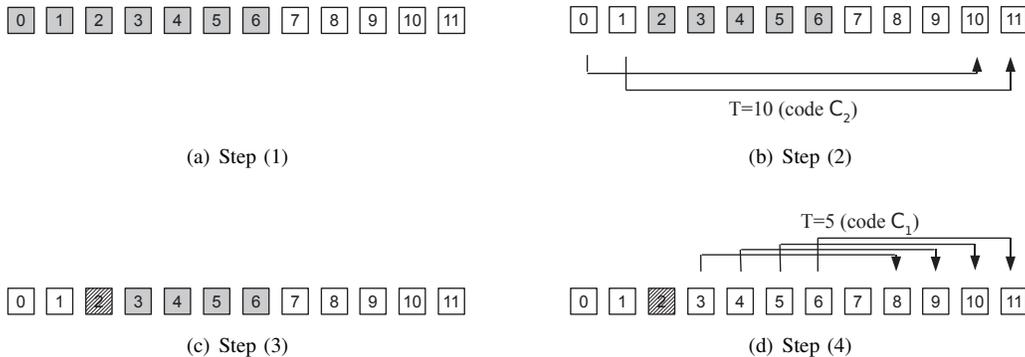


Fig. 5. Main steps of finding the upper bound for the $\{(4, 5), (7, 10)\}$ point lying in region (e) through one period illustration of the Periodic Erasure Channel. Grey and white squares denote erased and unerased symbols respectively while hatched squares denote symbols revealed to the receiver.

For each $n = 1, 2, \dots, r$ we have from (60) that

$$\overleftarrow{\mathbf{p}}^{3,n}[t]_t = \mathbf{p}^1[t + n(B_1 - k)]_t \quad (62)$$

We note that if $t + n(B_1 - k) \geq i + T_1$ then since \mathcal{C}_1 has a memory of T_1 it follows that $\mathbf{p}^1[t + n(B_1 - k)]$ only involves source symbols $\mathbf{s}[\cdot]$ after time i , which are not erased. Thus these parity sub-symbols can be computed by the decoder. If instead $t + n(B_1 - k) < i + T_1$ then each such $\overleftarrow{\mathbf{p}}^{3,n}[t + n(B_1 - k)]_t$ is recovered at time $i + B_1 - k - 1$ as it is recovered via Lemma 3, and hence is available to the decoder. Finally for $n = r + 1$ the decoder can compute and cancel $\overleftarrow{\mathbf{p}}^{3,r+1}[t + T_1]_t$ in a manner analogous to step (3) in case (A).

D. Examples

In Appendix E we provide two examples of the code constructions with parameters $\{(4, 5), (7, 10)\}$ and $\{(3, 5), (7, 9)\}$ corresponding to the two cases when $T_1 \leq 2(B_1 - k)$ and $T_1 > 2(B_1 - k)$ discussed above.

IX. CONVERSE PROOF IN REGION (E) (THEOREM 4)

We start by considering the example $\{(4, 5), (7, 10)\}$ illustrating the steps of the converse proof. Then, we will provide a rigorous converse proof for any point in region (e). We again use the periodic erasure channel technique with a period of length 12 and assume that the first 7 of these symbols are erased. With 7 erasures, code $\mathcal{C}_2 = (7, 10)$ can recover the first two symbols $\mathbf{s}[0]$ and $\mathbf{s}[1]$ by time 10 and 11, respectively (cf. Fig. 5(b)). Since code $\mathcal{C}_1 = (4, 5)$ is not capable of recovering the remaining 5 erasures, we reveal the first symbol at time $t = 2$ to the decoder. Now, \mathcal{C}_1 can recover the source symbols $\mathbf{s}[3], \dots, \mathbf{s}[6]$ by times 8 to 11, respectively (i.e., incurring a delay of 5 symbols). Finally the unerased symbols in $t \in [7, 11]$ are guaranteed to be recovered using the $(7, 10)$ code in the following period. Thus a total of 5 unerased channel symbols are sufficient to recover 6 erased source symbol. Therefore one can see that a rate of $5/11$ upper bounds the capacity of this channel.

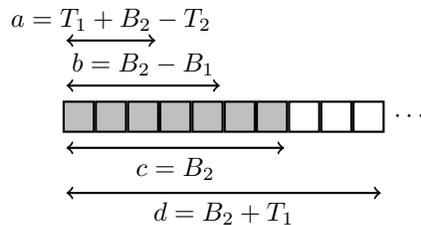


Fig. 6. One period of the periodic erasure channel used to prove an upper bound on capacity in region (e). Grey and white squares denote erased and unerased symbols respectively..

For the general case, one period of the periodic erasure channel to be used is shown in Fig. 6. Each period has B_2 erasures

followed by T_1 non-erasures. We can assign

$$\begin{aligned} a &= T_1 + B_2 - T_2, & b &= B_2 - B_1, & c &= B_2, & d &= B_2 + T_1 \quad (\text{period length}), \\ V_{1,i} &= \mathbf{s} \begin{bmatrix} id+a-1 \\ id \end{bmatrix}, & V_{2,i} &= \mathbf{s} \begin{bmatrix} id+b-1 \\ id+a \end{bmatrix}, & V_{3,i} &= \mathbf{s} \begin{bmatrix} id+c-1 \\ id+b \end{bmatrix}, & V_{4,i} &= \mathbf{s} \begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix} \\ W_{1,i} &= \mathbf{x} \begin{bmatrix} id+a-1 \\ id \end{bmatrix}, & W_{2,i} &= \mathbf{x} \begin{bmatrix} id+b-1 \\ id+a \end{bmatrix}, & W_{3,i} &= \mathbf{x} \begin{bmatrix} id+c-1 \\ id+b \end{bmatrix}, & W_{4,i} &= \mathbf{x} \begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix} \\ V_i &= (V_{1,i}, V_{2,i}, V_{3,i}, V_{4,i}), & W_i &= (W_{1,i}, W_{2,i}, W_{3,i}, W_{4,i}) \end{aligned}$$

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

$$R \leq \frac{n+1}{n} \cdot \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \xrightarrow{n \rightarrow \infty} \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \quad (63)$$

The more formal proof is as follows. We start by defining the capability of the $\mathcal{C}_1 = (B_1, T_1)$ and $\mathcal{C}_2 = (B_2, T_2)$ codes in i th period. In particular, a similar argument to that used in the proof of Lemma 1 can be used to get the following for $n \geq i+1$,

$$\begin{aligned} H(V_{1,i}|W_0^{i-1}, W_{4,0}^n) &= 0, & H(V_{3,i}|W_{1,i}, W_{2,i}, W_0^{i-1}, W_{4,0}^n) &= 0, \\ H(V_{4,i}|W_{1,i}, W_{2,i}, W_{3,i}, W_0^{i-1}, W_{4,0}^n) &= 0 \end{aligned} \quad (64)$$

Hence,

$$\begin{aligned} n(2T_1 + B_1 + B_2 - T_2) \log |\mathcal{S}| &= H(V_{1,0}^{n-1}, V_{3,0}^{n-1}, V_{4,0}^{n-1}) \stackrel{(a)}{=} H(V_0^{n-1}) - H(V_{2,0}^{n-1}) \\ &\leq H(V_0^{n-1}, W_{4,0}^n) - H(V_{2,0}^{n-1}) \\ &= H(W_{4,0}^n) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|V_0^{i-1}, W_{4,0}^n) + H(V_{2,i}|V_{1,i}, V_0^{i-1}, W_{4,0}^n) \right. \\ &\quad \left. + H(V_{3,i}|V_{1,i}, V_{2,i}, V_0^{i-1}, W_{4,0}^n) + H(V_{4,i}|V_{1,i}, V_{2,i}, V_{3,i}, V_0^{i-1}, W_{4,0}^n) - H(V_{2,i}|V_{2,0}^{i-1}) \right) \\ &\stackrel{(b)}{\leq} H(W_{4,0}^n) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|V_0^{i-1}, W_{4,0}^n) + H(V_{3,i}|V_{1,i}, V_{2,i}, V_0^{i-1}, W_{4,0}^n) \right. \\ &\quad \left. + H(V_{4,i}|V_{1,i}, V_{2,i}, V_{3,i}, V_0^{i-1}, W_{4,0}^n) \right) \\ &\stackrel{(c)}{=} H(W_{4,0}^n) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|W_0^{i-1}, W_{4,0}^n) + H(V_{3,i}|W_{1,i}, W_{2,i}, W_0^{i-1}, W_{4,0}^n) \right. \\ &\quad \left. + H(V_{4,i}|W_{1,i}, W_{2,i}, W_{3,i}, W_0^{i-1}, W_{4,0}^n) \right) \\ &\stackrel{(d)}{=} H(W_{4,0}^n) \leq (n+1)T_1 \cdot \log |\mathcal{X}|, \end{aligned} \quad (65)$$

where (a) follows from the fact that source symbols are i.i.d., (b) follows using the fact that conditioning reduces entropy, (c) holds since channel symbols are causal functions of source symbols (cf. (1)) and (d) follows by using (64).

Finally, we conclude that the rate of any $\{(B_1, T_1), (B_2, T_2)\}$ code in region (e) must satisfy,

$$R = \frac{\log |\mathcal{S}|}{\log |\mathcal{X}|} \leq \frac{n+1}{n} \cdot \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \xrightarrow{n \rightarrow \infty} \frac{T_1}{2T_1 + B_1 + B_2 - T_2} \quad (66)$$

which matches the our upper bound on the rate in (20).

X. UPPER AND LOWER BOUNDS IN REGION (F) (THEOREM 5)

A. Lower Bound

We note that the lower bound C_f^- in Theorem 5 is the same expression as the capacity C_e in Theorem 4. We argue that the code construction in region (e) can also be used in region (f). The only difference between the two regions is that the condition $T_2 \geq T_1 + B_1$ is satisfied in region (e) whereas $T_2 < T_1 + B_1$ holds in region (f). Upon examining the decoding analysis for the code in region (e), we note that the requirement $T_2 \geq T_1 + B_1$ is not used in the analysis of decoder 2. It is however used in the analysis of decoder 1 and in particular in (51). Fortunately it turns out that this condition is not necessary in the decoding analysis for user 1. As such only the condition $T_2 > T_1$ is required, which holds in region (f) since $T_1 < B_2$ holds in this region and $B_2 \leq T_2$ holds by definition. Thus we only need to explain a modified decoding procedure such that $T_2 \geq T_1 + B_1$ is not required.

We consider a channel that introduces a burst of length B_1 in the interval $[i - B_1, i - 1]$ and show that the source symbols $s[i - B_1 + j]$ for $j \in \{0, \dots, B_1 - 1\}$ are recovered at time $i - B_1 + T_1 + j$.

Since C_1 is a (B_1, T_1) MS code, it suffices to show that in the interval $t \in [i, i - B_1 + T_1 + j]$ the parity symbols $\mathbf{p}^1[t]$ are all available by time $i - B_1 + T_1 + j$ by cancelling the interfering $\mathbf{p}^2[t] = s[t - T_2]$ in this period.

We establish the above property recursively starting with $j = 0$ and considering the interval $t \in [i, i + T_1 - B_1]$. The parity-check symbols of C_2 , given by $\mathbf{p}^2[t] = s[t - T_2]$, are not erased. This follows since $t - T_2 \leq i + T_1 - B_1 - T_2 < i - B_1$ since $T_2 > T_1$. Hence the overlapping $\mathbf{p}^1[t]$ in this interval can be recovered and used to recover $s[i - B_1]$ at time $i + T_1 - B_1$.

For each $j > 0$, we assume recursively that the source symbols up to time $i - B_1 + j - 1$ are recovered and the parity checks $\mathbf{p}^1[\cdot]$ in the interval $[i, i - B_1 + T_1 + j - 1]$ are also available. We claim that at time $t = i - B_1 + j + T_1$, the source symbol $s[i - B_1 + j]$ and the parity check $\mathbf{p}^1[i - B_1 + j + T_1]$ can also be recovered. Note that the interfering parity check $\mathbf{p}^2[i - B_1 + j + T_1]$ is the source symbol $s[i - B_1 + j + T_1 - T_2]$ which has been recovered since $i - B_1 + j + T_1 - T_2 \leq i - B_1 + j - 1$ as $T_2 > T_1$. Thus this symbol can be cancelled out to compute $\mathbf{p}^1[i - B_1 + j + T_1]$ and in turn the source symbol $s[i - B_1 + j]$ can be recovered.

The claim now follows.

B. Upper Bound

The proof of the upper bound in region (f) uses a revealing argument similar to that used in the converse proof in region (e) provided in Section IX. We shall use Fig. 7 to illustrate one period of the periodic erasure channel used in this proof. One period contains B_2 erasures followed by $T_2 - B_1$ non-erasures, for a total of $B_2 + T_2 - B_1$ symbols.

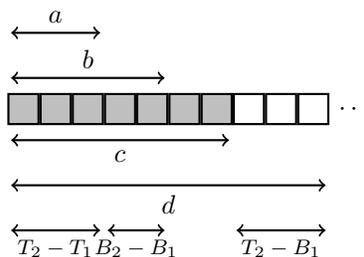


Fig. 7. One period of the periodic erasure channel used to prove the first upper bound in region (f). Grey and white squares denote erased and unerased symbols respectively.

The first $B_2 - B_1$ source symbols can be recovered with code C_2 , from $\mathbf{x} \begin{bmatrix} B_2 + T_2 - B_1 - 1 \\ B_2 \end{bmatrix}$, 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 C_1 recovers the next $T_2 - T_1$ source symbols, which is $\mathbf{s} \begin{bmatrix} B_2 - B_1 + T_2 - T_1 - 1 \\ B_2 - B_1 \end{bmatrix}$. 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 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 $\mathbf{s} \begin{bmatrix} B_2 + T_2 - B_1 - 1 \\ B_2 \end{bmatrix}$. 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 (e) proof, we can suppose that the upper bound on the capacity is:

$$R = \frac{\log |\mathcal{S}|}{\log |\mathcal{X}|} \leq \frac{n + 1}{n} \cdot \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)} \xrightarrow{n \rightarrow \infty} \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)} \quad (67)$$

For the formal proof, we assign the following:

$$a = B_2 - B_1, \quad b = B_2 - B_1 + T_2 - T_1, \quad c = B_2, \quad d = B_2 + T_2 - B_1 \quad (\text{period length}),$$

$$\begin{aligned}
V_{1,i} &= \mathbf{s} \begin{bmatrix} id+a-1 \\ id \end{bmatrix}, & V_{2,i} &= \mathbf{s} \begin{bmatrix} id+b-1 \\ id+a \end{bmatrix}, & V_{3,i} &= \mathbf{s} \begin{bmatrix} id+c-1 \\ id+b \end{bmatrix}, & V_{3,i} &= \mathbf{s} \begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix} \\
W_{1,i} &= \mathbf{x} \begin{bmatrix} id+a-1 \\ id \end{bmatrix}, & W_{2,i} &= \mathbf{x} \begin{bmatrix} id+b-1 \\ id+a \end{bmatrix}, & W_{3,i} &= \mathbf{x} \begin{bmatrix} id+c-1 \\ id+b \end{bmatrix}, & W_{4,i} &= \mathbf{x} \begin{bmatrix} (i+1)d-1 \\ id+c \end{bmatrix} \\
V_i &= (V_{1,i}, V_{2,i}, V_{3,i}, V_{4,i}), & W_i &= (W_{1,i}, W_{2,i}, W_{3,i}, W_{4,i})
\end{aligned}$$

A similar argument to that used in the proof of Lemma 1 can be applied to $\mathcal{C}_1 = (B_1, T_1)$ and $\mathcal{C}_2 = (B_2, T_2)$ codes, i.e., for $n \geq i + 1$,

$$\begin{aligned}
H(V_{1,i}|W_0^{i-1}, W_0^n) &= 0, & H(V_{2,i}|W_{1,i}, W_0^{i-1}, W_0^n) &= 0, \\
H(V_{4,i}|W_{1,i}, W_{2,i}, W_{3,i}, W_0^{i-1}, W_0^n) &= 0
\end{aligned} \tag{68}$$

Hence,

$$\begin{aligned}
n(2(T_2 - B_1) + (B_2 - T_1)) \log |\mathcal{S}| &= H(V_{1,0}^{n-1}, V_{2,0}^{n-1}, V_{4,0}^{n-1}) \stackrel{(a)}{=} H(V_0^{n-1}) - H(V_{3,0}^{n-1}) \\
&\leq H(V_0^{n-1}, W_{4,0}^n) - H(V_{3,0}^{n-1}) \\
&= H(W_{4,0}^n) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|V_0^{i-1}, W_{4,0}^n) + H(V_{2,i}|V_{1,i}, V_0^{i-1}, W_{4,0}^n) \right. \\
&\quad \left. + H(V_{3,i}|V_{1,i}, V_{2,i}, V_0^{i-1}, W_{4,0}^n) + H(V_{4,i}|V_{1,i}, V_{2,i}, V_{3,i}, V_0^{i-1}, W_{4,0}^n) - H(V_{3,i}|V_{3,0}^{i-1}) \right) \\
&\stackrel{(b)}{\leq} H(W_{4,0}^n) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|V_0^{i-1}, W_{4,0}^n) + H(V_{2,i}|V_{1,i}, V_0^{i-1}, W_{4,0}^n) \right. \\
&\quad \left. + H(V_{4,i}|V_{1,i}, V_{2,i}, V_{3,i}, V_0^{i-1}, W_{4,0}^n) \right) \\
&\stackrel{(c)}{=} H(W_{4,0}^n) + \sum_{i=0}^{n-1} \left(H(V_{1,i}|W_0^{i-1}, W_{4,0}^n) + H(V_{2,i}|W_{1,i}, W_0^{i-1}, W_{4,0}^n) \right. \\
&\quad \left. + H(V_{4,i}|W_{1,i}, W_{2,i}, W_{3,i}, W_0^{i-1}, W_{4,0}^n) \right) \\
&\stackrel{(d)}{=} H(W_{4,0}^n) \leq (n+1)(T_2 - B_1) \cdot \log |\mathcal{X}|,
\end{aligned} \tag{69}$$

where (a) follows from the fact that source symbols are i.i.d., (b) follows using the fact that conditioning reduces entropy, (c) follows by using (1) and (d) follows by using (68).

In other words,

$$R = \frac{\log |\mathcal{S}|}{\log |\mathcal{X}|} \leq \frac{n+1}{n} \cdot \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)} \xrightarrow{n \rightarrow \infty} \frac{T_2 - B_1}{2(T_2 - B_1) + (B_2 - T_1)} \tag{70}$$

Therefore, (70) governs the rate of any $\{(B_1, T_1), (B_2, T_2)\}$ code in region (f).

XI. SPECIAL CASES IN REGION (F)

We present the capacity region in region (f) for the special cases when $T_1 = B_1$ and $T_2 = B_2$ and then present a conjecture on the capacity in the general case in this section.

A. Achievability Scheme in Region (f) at $T_1 = B_1$ (Proposition 1)

For the special case when $T_1 = B_1$, we provide a code construction that attains the upper bound C_f^+ in this section. We begin with an example of $\{(4, 4), (5, 6)\}$ Mu-SCo construction of rate $2/5$, as shown in Table III. We split each source symbol into two sub-symbols as shown. A $(4, 4)$ repetition code is applied resulting in the first two rows of parity checks and then a $(B_2 - B_1, T_2 - T_1) = (1, 2)$ MS code 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, suppose 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.

1) *Code Construction:* Our proposed code construction, which achieves the minimum delay for user 1, i.e., $T_1 = B_1$ is as follows

- Split each source symbol $\mathbf{s}[i]$ into $T_2 - B_1 = T_2 - T_1$ sub-symbols

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

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

TABLE III

MU-SCO CONSTRUCTION FOR $(B_1, T_1) = (4, 4)$ AND $(B_2, T_2) = (5, 6)$. THIS POINT ACHIEVES THE UPPER BOUND C_f^+ GIVEN IN THEOREM 5 AS STATED IN PROPOSITION 1 SINCE $T_1 = B_1 = 4$.

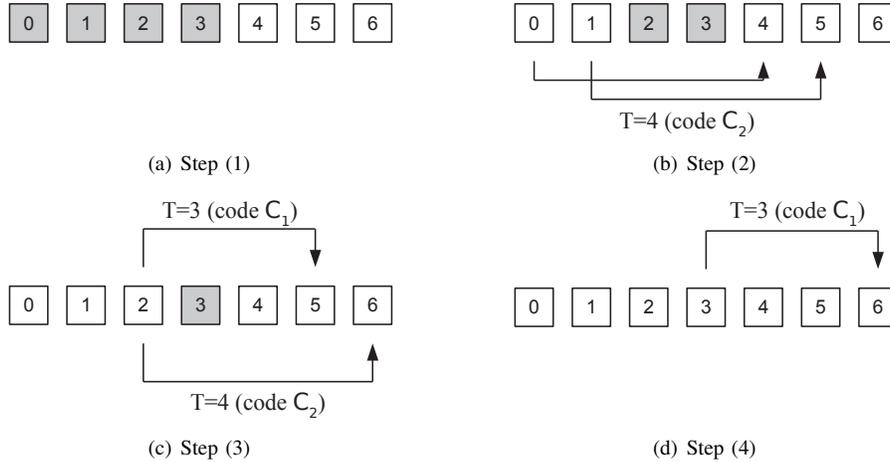


Fig. 8. Main steps of finding the upper bound for the $\{(2, 3), (4, 4)\}$ point lying in region (f) through one period illustration of the Periodic Erasure Channel. Grey and white squares denote erased and unerased symbols respectively. Note that the symbol 2 is recovered by both codes C_1 and C_2 .

- Let C_1 be the single user $(B_1, T_1) = (T_1, T_1)$ MS code obtained by repeating the source symbols them to produce $T_2 - T_1$ parity check sub-symbols, i.e.,

$$\mathbf{p}^1[i] = (p_0^1[i], \dots, p_{T_2-T_1-1}^1[i]) = (s_0[i-T_1], \dots, s_{T_2-T_1-1}[i-T_1]) = \mathbf{s}[i-T_1]. \quad (71)$$

- Let C_2 be a $(B_2 - B_1, T_2 - T_1)$ MS code applied to the source symbols $\mathbf{s}[i]$ and then constructing $(B_2 - B_1)$ parity checks $\mathbf{p}^2[i] = (p_0^2[i], \dots, p_{B_2-B_1-1}^2[i])$ at each time by combining the source sub-symbols diagonally.
- Concatenate the two streams $\mathbf{p}^1[\cdot]$ and $\mathbf{p}^2[\cdot]$ after introducing a shift of T_1 in the second stream. The output symbol at time i is $\mathbf{x}[i] = (\mathbf{s}[i], \mathbf{p}^1[i], \mathbf{p}^2[i-T_1])$.

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

2) *Decoding at User 1:* A burst erasure of length B_1 can be directly recovered using the stream of parity checks $\mathbf{p}^1[\cdot]$ produced by code 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.

3) *Decoding at User 2:* Suppose that the symbols $\mathbf{x}[i-B_2], \dots, \mathbf{x}[i-1]$ 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 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 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 C_1 repetition code. This completes the decoding analysis for user 2.

B. Converse Proof in Region (f) at $T_2 = B_2$ (Proposition 2)

In contrast to the special case when $T_1 = B_1$, where the upper bound C_f^+ in Theorem 5 is shown to be tight, we show that in the case when $T_2 = B_2$ the lower bound C_f^- is the true capacity. We do this by presenting a tighter upper bound, which depends on double recovery of some source symbols, once using code C_1 and another using C_2 . We illustrate the main idea of such converse through considering the specific point $\{(2, 3), (4, 4)\}$ shown in Fig. 8. We start by considering a periodic erasure

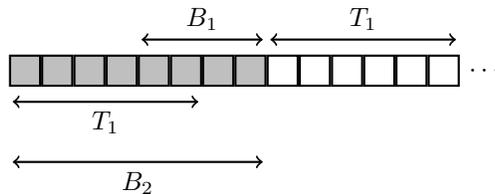


Fig. 9. One period of the periodic erasure channel used to prove an upper bound on capacity in region (f) for the special case $T_2 = B_2$. Grey and white squares denote erased and unerased symbols respectively.

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 $s[0]$ and $s[1]$ 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). Step (3) in Fig. 8 gives the main idea of this converse. Since, there are two remaining erasures, the source symbol $s[2]$ 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 $s[4]$, $s[5]$ and $s[6]$ 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, one period of the corresponding periodic erasure channel to be used for proving the upper bound is given in Fig. 9. Each period has B_2 erasures followed by T_1 non-erasures.

In Fig. 9, we have the first period of the erasure channel. The key is to show that the received channel symbols $\mathbf{x} \begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ alone can recover all of the source symbols in the period, but there is enough information in the channel symbols to recover some of the source symbols twice. The fact that we have two decoders allows some of the source symbols to be decoded by mutually exclusive groups of channel symbols, but when we put all of the channel symbols together, the redundant information in the channel symbols does affect the maximum achievable rate of the code.

The source symbols 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 9 shows, the first two groups of source symbols overlap. The overlap consists of the symbols $\mathbf{s} \begin{bmatrix} T_1-1 \\ B_2-B_1 \end{bmatrix}$. 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 symbols $\mathbf{s} \begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ using only the symbols $\mathbf{x} \begin{bmatrix} B_2+T_1-1 \\ B_2 \end{bmatrix}$ and does not require any future channel symbols.

Assuming that what we have just described is possible, then we have T_1 channel symbols recovered $2T_1 + B_1$ source symbols. We should be able to write the relation:

$$(2T_1 + B_1) \cdot \log |\mathcal{S}| \leq T_1 \cdot \log |\mathcal{X}|$$

$$R = \frac{\log |\mathcal{S}|}{\log |\mathcal{X}|} \leq \frac{T_1}{2T_1 + B_1} \quad (72)$$

The formal proof shows that this is indeed possible. We can split the proof into three major parts.

1. The source symbols $\mathbf{s} \begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix}$ and $\mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ B_2-B_1 \end{bmatrix}$ can be recovered from the channel symbols $\mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}$ using the (B_2, B_2) and (B_1, T_1) decoders respectively, i.e.,

$$H\left(\mathbf{s} \begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix} \middle| \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}\right) = 0, \quad (73)$$

$$H\left(\mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ B_2-B_1 \end{bmatrix} \middle| \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix} \mathbf{s} \begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix}\right) = 0. \quad (74)$$

Next, we can write

$$\begin{aligned} H\left(\mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}\right) &= H\left(\mathbf{s} \begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix}, \mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ B_2-B_1 \end{bmatrix}, \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}\right) - H\left(\mathbf{s} \begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix} \middle| \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}\right) \\ &\quad - H\left(\mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ B_2-B_1 \end{bmatrix} \middle| \mathbf{s} \begin{bmatrix} B_2-B_1-1 \\ 0 \end{bmatrix}, \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}\right) \\ &\stackrel{(a)}{=} H\left(\mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ 0 \end{bmatrix} \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix}\right) \\ &= H\left(\mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ 0 \end{bmatrix}\right) \\ &= H\left(\mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ 0 \end{bmatrix}\right) + H\left(\mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ B_2 \end{bmatrix} \middle| \mathbf{s} \begin{bmatrix} 2B_2-B_1-T_1-1 \\ 0 \end{bmatrix}, \mathbf{x} \begin{bmatrix} 2B_2-B_1-1 \\ 0 \end{bmatrix}\right), \end{aligned} \quad (75)$$

where we used (73) and (74) to remove the negative terms before step (a).

2. In this step, we show that two source symbols $\mathbf{s}[m - B_2]$ and $\mathbf{s}[m - T_1]$ can be recovered from each added channel packet

$\mathbf{x}[m]$ for $m \in \{2B_2 - B_1, \dots, B_2 + T_1 - 1\}$. We start by establishing the following lemma.

Lemma 4. *The following inequality is true for all $m \geq 2B_2 - B_1 - 1$:*

$$\sum_{i=B_2}^m H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\left[\begin{smallmatrix} m-B_2 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} m-T_1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{x}\left[\begin{smallmatrix} m \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} m-B_2 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} m-T_1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} m-B_2 \\ 0 \end{smallmatrix}\right]\right) \quad (76)$$

Proof: See Appendix F. ■

We then substitute $m = B_2 + T_1 - 1$ into (76)

$$\sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i]) \geq H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right). \quad (77)$$

3. We can recover $\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right]$ from $\mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right]$ given the previous channel symbols $\mathbf{x}\left[\begin{smallmatrix} B_2-1 \\ 0 \end{smallmatrix}\right]$ using decoder 2, so we can write

$$H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ 0 \end{smallmatrix}\right]\right) = 0. \quad (78)$$

Using (78), we continue with (77) to get,

$$\begin{aligned} \sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i]) &\geq H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) \\ &= H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2-1 \\ 0 \end{smallmatrix}\right]\right) \\ &\quad - H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ 0 \end{smallmatrix}\right]\right) \\ &\stackrel{(d)}{=} H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2-1 \\ 0 \end{smallmatrix}\right]\right) \\ &= H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2-1 \\ 0 \end{smallmatrix}\right]\right) \\ &\quad + H\left(\mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2-1 \\ 0 \end{smallmatrix}\right]\right) \\ &= H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) + H\left(\mathbf{x}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2 \end{smallmatrix}\right] \middle| \mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right] \mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2-B_1 \end{smallmatrix}\right] \mathbf{x}\left[\begin{smallmatrix} B_2-1 \\ 0 \end{smallmatrix}\right]\right) \\ &\geq H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) \end{aligned} \quad (79)$$

where step (d) makes use of (78).

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

$$\begin{aligned} \sum_{i=B_2}^{B_2+T_1-1} H(\mathbf{x}[i]) &\geq H\left(\mathbf{s}\left[\begin{smallmatrix} T_1-1 \\ 0 \end{smallmatrix}\right]\right) + H\left(\mathbf{s}\left[\begin{smallmatrix} B_2+T_1-1 \\ B_2-B_1 \end{smallmatrix}\right]\right) \\ T_1 \cdot \log |\mathcal{X}| &\geq (2T_1 + B_1) \cdot \log |\mathcal{S}| \\ R &= \frac{\log |\mathcal{S}|}{\log |\mathcal{X}|} \leq \frac{T_1}{2T_1 + B_1} \end{aligned} \quad (80)$$

which matches the upper bound in (27).

C. Conjectured Capacity in Region (f)

Conjecture 1. *For any given point $\{(B_1, T_1), (B_2, T_2)\}$ in region (f) defined by $B_2 \leq T_2 \leq T_1 + B_1$ and $B_1 \leq T_1 < B_2$, the capacity is given by,*

$$C_f = \frac{T_1}{2T_1 + \frac{B_2 - T_1}{T_2 - T_1} B_1}. \quad (81)$$

One can see that the capacity in (81) coincides with the capacity results in Propositions 1 and 2 by simply substituting $T_1 = B_1$ and $T_2 = B_2$ respectively in (81). It also maintains the continuity in capacity at the edges of region (f) with regions (c) and (e). In particular, the results of substituting $B_2 = T_1$ and $T_2 = T_1 + B_1$ in (81) match the second case in (17) and (20), respectively.

For a given B_1 and T_1 , the conjectured capacity is constant for all points satisfying,

$$\frac{B_2 - T_1}{T_2 - T_1} = k \Rightarrow T_2 = \frac{1}{k}(B_2 - T_1) + T_1. \quad (82)$$

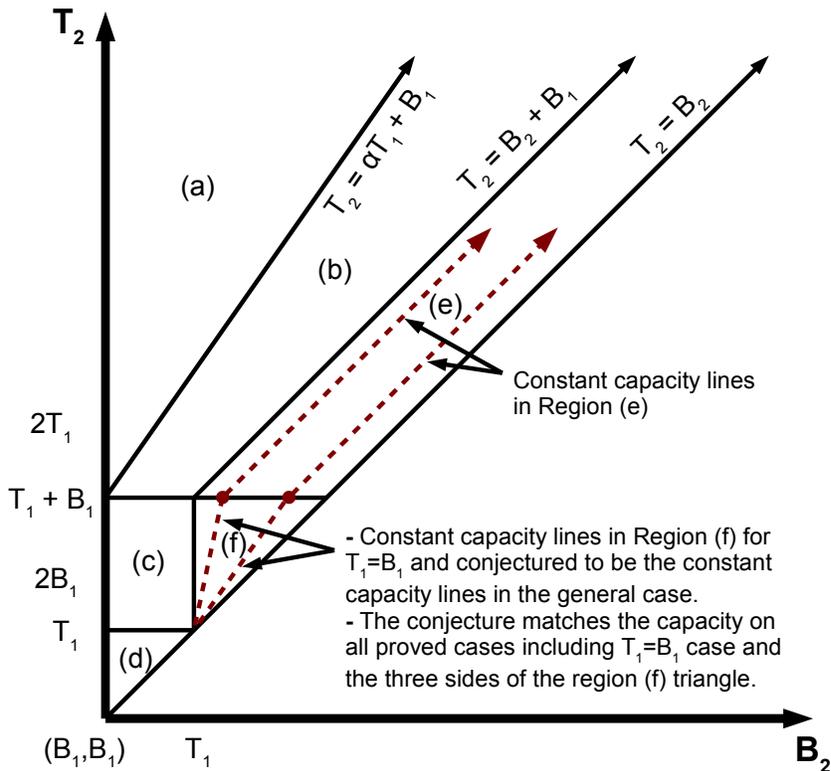


Fig. 10. Capacity behavior in the (B_2, T_2) plane. We hold B_1 and T_1 as constants, so the regions depend on the relation between T_2 and B_2 only. The dashed line gives the contour of constant capacity in region (e) as well as in the special case of $T_1 = B_1$ in region (f).

Hence, the capacity can be shown to be constant on the straight lines passing through the point $(B_2, T_2) = (T_1, T_1)$ and slope $1/k$ in Fig. 10. However, this point is not included since C_f in (81) is undetermined at $(B_2, T_2) = (T_1, T_1)$. The capacity at this point is $\frac{1}{2}$ and can be obtained from the capacity in region (c) given by (40).

Also note that the capacity expression in region (e), 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 region (e), it is natural to fix B_1 and T_1 and classify the various regions as shown in Fig. 10. Observe that the streaming capacity for any point in region (e) is constant across the 45-degree line and is equal to the multicast upper bound at the lowest point on the horizontal line, $T_2 = T_1 + B_1$, separating regions (e) and (f) in Fig. 10.

An example of region (f) with $(B_1, T_1) = (10, 16)$ is illustrated in Fig. 11. Different points denote different B_2 and T_2 values as shown on the x and y axes respectively. The fraction at each point is the conjectured capacity as per (81). One can see that the conjectured capacity is constant on the straight lines passing through the point $(B_2, T_2) = (T_1, T_1) = (16, 16)$ but not including it.

XII. CONCLUSION

We study a multicast extension of the low-delay codes for streaming over burst erasure channels. We observe an interesting interplay between the delay of the two receivers from a capacity point of view. In particular, in the large-delay regime we characterize the capacity and observe that the delay of one of the receivers can be reduced up to a certain critical value without reducing the capacity. This enables us to use simple modifications of previously proposed codes to establish the capacity. In the low-delay regime a new code construction is proposed and it is shown to be optimal in a subset of this regime. New upper bounds that are tighter than previously proposed techniques are developed and shown to be tight for a certain class of parameters. Finally in the cases where the exact capacity has not been characterized a conjecture is also provided.

While the focus of this work has been on the case of burst erasure channels only, we believe that the code constructions can be extended in a natural way to deal with both burst and isolated erasure patterns. For the single user case, it has been recently shown [11] that a low delay code for the burst erasure channel can be extended using a layered construction to correct both burst and isolated erasures and is shown to outperform baseline codes over statistical channels such as Gilbert-Elliott

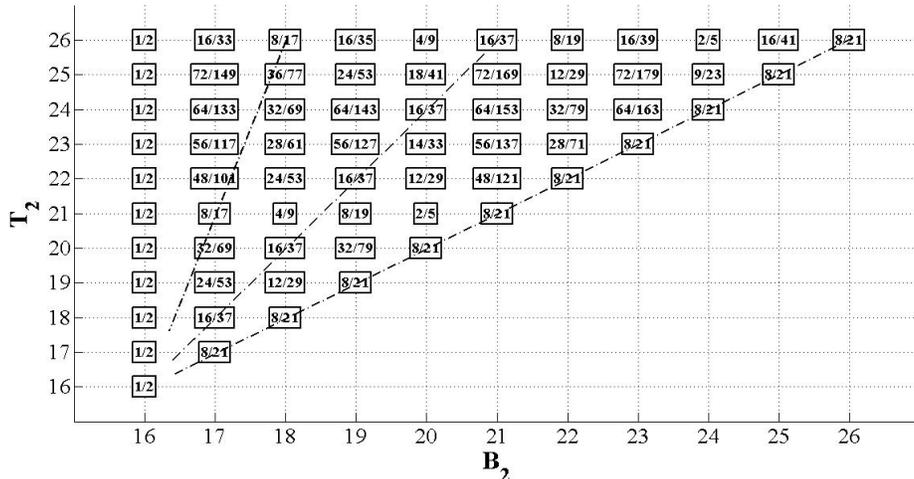


Fig. 11. An example of region (f) in the (B_2, T_2) plane for $(B_1, T_1) = (10, 16)$. The dashed lines give some examples of the contour of constant conjectured capacity in region (f). This conjecture is proved for the cases $B_2 = T_1$ which is the left vertical edge of the triangle, $T_2 = T_1 + B_1$ which is the upper horizontal edge of the triangle and $T_2 = B_2$ which is the right 45-degree edge. It is also proved for the special case of $T_1 = B_1$ which is not shown in this figure.

and Fritchman channels. It will be an interesting future work to examine whether such a layered approach can be used for multicast. More generally the constructions presented in this paper are a step towards constructing delay-universal streaming codes, where the decoding delay adapts to the channel conditions. Finally we note that the low delay codes for burst erasure channels are a class of convolutional codes with a certain column-span [11]. The multicast extension treated in this work appears to require codes with a certain column-span profile. Establishing such algebraic properties of the convolutional codes also appears to be an interesting direction for further research and might lead to a natural extension of streaming codes for multiple users.

ACKNOWLEDGEMENT

The authors thank the associate editor and the anonymous reviewers for several constructive suggestions that have improved the quality of the paper. In particular, Reviewer 1 made an elegant observation that significantly simplified the upper bound in Theorems 1, 4 and 5 of the paper. Reviewer 3 provided several deep comments on the results, including connections between regions (e) and (f), and motivated us to state the conjecture in Section XI-C of the paper.

APPENDIX A PROOF OF LEMMA 1

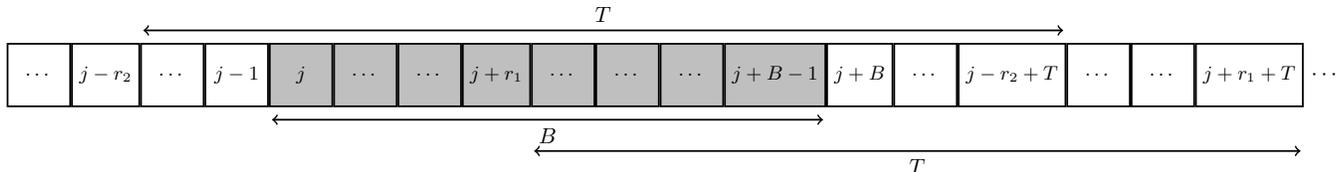


Fig. 12. A channel introducing a single burst of length B in the interval $[j, j + B - 1]$ used in proving Lemma 1. Grey and white squares denote erased and unerased symbols respectively.

We start by considering a simple channel which introduces one burst of length B in the interval $[j, j + B - 1]$ as shown in Fig. 12. A (B, T) code is capable of recovering each source symbol within a delay of T and maximum error probability ϵ as defined in (35). We are only interested in the recovery of the two sets, $\mathbf{s}[j + r_1]$ for $r_1 \in \{0, \dots, B - 1\}$ and $\mathbf{s}[j - r_2]$ for $r_2 \in \{1, \dots, T\}$. Applying Fano's inequality [34] to each of these source symbols we have that:

$$H\left(\mathbf{s}[j + r_1] | \mathbf{x} \left[\begin{smallmatrix} j-1 \\ j+B \end{smallmatrix} \right], \mathbf{M}\right) \leq H(\epsilon) + \epsilon \log |\mathcal{S}|, \quad r_1 \in [0, B - 1], \quad (83)$$

$$H\left(\mathbf{s}[j - r_2] | \mathbf{x} \left[\begin{smallmatrix} j-1 \\ j+B \end{smallmatrix} \right], \mathbf{M}\right) \leq H(\epsilon) + \epsilon \log |\mathcal{S}|, \quad r_2 \in [1, T]. \quad (84)$$

To establish Lemma 1, first recall our notation in (32),

$$\begin{aligned} V_i &= \mathbf{s} \left[\begin{smallmatrix} (i+1)(T+B)-1 \\ i(T+B) \end{smallmatrix} \right], & V_{1,i} &= \mathbf{s} \left[\begin{smallmatrix} i(T+B)+B-1 \\ i(T+B) \end{smallmatrix} \right], & V_{2,i} &= \mathbf{s} \left[\begin{smallmatrix} (i+1)(T+B)-1 \\ i(T+B)+B \end{smallmatrix} \right], \\ W_i &= \mathbf{x} \left[\begin{smallmatrix} (i+1)(T+B)-1 \\ i(T+B) \end{smallmatrix} \right], & W_{1,i} &= \mathbf{x} \left[\begin{smallmatrix} i(T+B)+B-1 \\ i(T+B) \end{smallmatrix} \right], & W_{2,i} &= \mathbf{x} \left[\begin{smallmatrix} (i+1)(T+B)-1 \\ i(T+B)+B \end{smallmatrix} \right], \end{aligned} \quad (85)$$

where $i \in \{0, 1, 2, \dots\}$. We then start from the L.H.S. of (36) and use chain rule as follows,

$$\begin{aligned} H(V_{1,i}|W_0^{i-1}, W_{2,0}^n, \mathbf{M}) &\stackrel{(a)}{\leq} H(V_{1,i}|W_0^{i-1}, W_{2,i}, \mathbf{M}) = H\left(\mathbf{s} \left[\begin{smallmatrix} i(T+B)+B-1 \\ i(T+B) \end{smallmatrix} \right] \middle| W_0^{i-1}, W_{2,i}, \mathbf{M}\right) \\ &= \sum_{r_1=0}^{B-1} H\left(\mathbf{s}[i(T+B) + r_1] \middle| \mathbf{s} \left[\begin{smallmatrix} i(T+B)+r_1-1 \\ i(T+B) \end{smallmatrix} \right], W_0^{i-1}, W_{2,i}, \mathbf{M}\right) \\ &\stackrel{(b)}{\leq} \sum_{r_1=0}^{B-1} H\left(\mathbf{s}[i(T+B) + r_1] \middle| W_0^{i-1}, W_{2,i}, \mathbf{M}\right) \\ &\stackrel{(c)}{\leq} \sum_{r_1=0}^{B-1} H\left(\mathbf{s}[i(T+B) + r_1] \middle| \mathbf{x} \left[\begin{smallmatrix} i(T+B)-1 \\ 0 \end{smallmatrix} \right], \mathbf{x} \left[\begin{smallmatrix} i(T+B)+r_1+T \\ i(T+B)+B \end{smallmatrix} \right], \mathbf{M}\right) \\ &\stackrel{(d)}{\leq} \sum_{r_1=0}^{B-1} (H(\epsilon) + \epsilon \log |\mathcal{S}|) = B(H(\epsilon) + \epsilon \log |\mathcal{S}|), \end{aligned} \quad (86)$$

where (a) and (b) uses the fact that conditioning reduces entropy and (a) uses that $n \geq i+1$, (c) follows since $\mathbf{x} \left[\begin{smallmatrix} i(T+B)+r_1+T \\ i(T+B)+B \end{smallmatrix} \right] \subseteq W_{2,i}$ for $r_1 \in [0, B-1]$, whereas (d) uses (83) at $j = i(T+B)$. This establishes (36). Using (84) a similar argument can be used to establish (37), and its proof will be omitted.

APPENDIX B SOURCE EXPANSION

We start by giving a definition of source expansion as follows.

Definition 3 (Source Expansion). A (p, r) expansion of the source stream $\mathbf{s}[\cdot]$ consists of,

- Splitting each source symbol $\mathbf{s}[i] \in \mathbb{F}_q^{rp}$ into rp sub-symbols each in \mathbb{F}_q , i.e., $\mathbf{s}[i] = (s_0[i], s_1[i], \dots, s_{rp-1}[i])$.
- Rearranging the rp sub-symbols into r groups each with p sub-symbols as follows,

$$\begin{aligned} \tilde{\mathbf{s}}[ri] &= (s_0[i], \dots, s_{p-1}[i]) \\ \tilde{\mathbf{s}}[ri+1] &= (s_p[i], \dots, s_{2p-1}[i]) \\ &\vdots \\ \tilde{\mathbf{s}}[ri+r-1] &= (s_{(r-1)p}[i], \dots, s_{rp-1}[i]), \end{aligned} \quad (87)$$

where $\tilde{\mathbf{s}}[\cdot]$ is the expanded source stream.

The relation between the decoding capability of a MS code on the original stream and that on the expanded stream is discussed in the following lemma.

Lemma 5. Consider a (p, r) expansion of the source stream $\mathbf{s}[\cdot]$ to $\tilde{\mathbf{s}}[\cdot]$. A (rB, \tilde{T}) MS code applied to $\tilde{\mathbf{s}}[\cdot]$ is capable of recovering a burst of length B symbols within a delay of $T = \lceil \frac{\tilde{T}}{r} \rceil$ on the original stream $\mathbf{s}[\cdot]$.

Proof: Suppose that a (rB, \tilde{T}) MS code be applied to $\tilde{\mathbf{s}}[\cdot]$ to generate the channel symbols $\tilde{\mathbf{x}}[\cdot]$. These symbols are multiplexed together, and the input on the channel at time i is

$$\mathbf{x}[i] = (\tilde{\mathbf{x}}[ri], \tilde{\mathbf{x}}[ri+1], \dots, \tilde{\mathbf{x}}[ri+r-1]). \quad (88)$$

We suppose the channel erases a burst of length B on the original stream in the interval $[i, i+B-1]$. By using (88) this corresponds to an erasure burst in the interval $[ri, r(i+B)-1]$ on the expanded stream of $\tilde{\mathbf{x}}[\cdot]$. This is a total of rB erasures in a burst and is thus recoverable using the (rB, \tilde{T}) MS code within a delay of \tilde{T} on the expanded stream. Each source symbol $\mathbf{s}[j]$ for $j \in \{i, i+B-1\}$ is recovered once the last corresponding expanded source symbol, $\tilde{\mathbf{s}}[rj+r-1]$ is recovered. Using the (rB, \tilde{T}) MS code, such symbol is recovered by time $rj+r-1+\tilde{T}$. This is equivalent to time $\lfloor \frac{rj+r-1+\tilde{T}}{r} \rfloor$ on the original stream (cf. (88)). Hence, $\mathbf{s}[j]$ is recovered with a delay of $\lfloor \frac{\tilde{T}+r-1}{r} \rfloor = \lceil \frac{\tilde{T}}{r} \rceil$ and the lemma follows. ■

Example 2. Table IV illustrates an example for source expansion with parameters $r = 3$, $B = 2$ and $\tilde{T} = 7$. A burst erasure of length $B = 2$ that erases $\mathbf{x}[0]$ and $\mathbf{x}[1]$ on the original stream corresponds to $rB = 6$ erased channel symbols, $\tilde{\mathbf{x}}[0], \dots, \tilde{\mathbf{x}}[5]$

Source Stream	s[0]			s[1]			s[2]			s[3]		
Expanded Source Stream	$\tilde{s}[0]$	$\tilde{s}[1]$	$\tilde{s}[2]$	$\tilde{s}[3]$	$\tilde{s}[4]$	$\tilde{s}[5]$	$\tilde{s}[6]$	$\tilde{s}[7]$	$\tilde{s}[8]$	$\tilde{s}[9]$	$\tilde{s}[10]$	$\tilde{s}[11]$
Expanded Channel Stream	$\tilde{x}[0]$	$\tilde{x}[1]$	$\tilde{x}[2]$	$\tilde{x}[3]$	$\tilde{x}[4]$	$\tilde{x}[5]$	$\tilde{x}[6]$	$\tilde{x}[7]$	$\tilde{x}[8]$	$\tilde{x}[9]$	$\tilde{x}[10]$	$\tilde{x}[11]$
Channel Stream	$\mathbf{x}[0]$			$\mathbf{x}[1]$			$\mathbf{x}[2]$			$\mathbf{x}[3]$		
Expanded Stream Recovery							\downarrow $\tilde{s}[0]$ $\tilde{s}[1]$			\downarrow $\tilde{s}[2]$		
Original Stream Recovery										\downarrow \downarrow $s[0]$		

TABLE IV

A SOURCE EXPANSION EXAMPLE WITH PARAMETERS $(p, 3)$. EACH SOURCE SYMBOL $s[i]$ IS EXPANDED INTO 3 SYMBOLS $\tilde{s}[3i]$, $\tilde{s}[3i + 1]$ AND $\tilde{s}[3i + 2]$. A $(3B, \tilde{T})$ IS THEN APPLIED TO $\tilde{s}[\cdot]$ TO GENERATE $\tilde{x}[\cdot]$. THE CHANNEL SYMBOLS IN THE ORIGINAL STREAM IS DENOTED BY $\mathbf{x}[i] = (\tilde{x}[3i], \tilde{x}[3i + 1], \tilde{x}[3i + 2])$.

on the expanded stream. Hence, the $(rB, \tilde{T}) = (6, 7)$ MS code is capable of recovering the erased source symbols within a delay of $\tilde{T} = 7$. In particular, the source symbols $\tilde{s}[0]$, $\tilde{s}[1]$ and $\tilde{s}[2]$ belonging to $s[0]$ are recovered at time 7, 8 and 9 respectively on the expanded stream. Hence, $s[0]$ is recovered at time 3 on the original stream with is equivalent to a delay of $T = \lceil \frac{\tilde{T}}{r} \rceil$.

Now we can use Definition 3 and Lemma 5 to achieve the capacity C_b in region (b) for the general case when \tilde{T}_1 in (41) is not an integer. We start by expanding the source stream $s[i]$ with parameters $(p, r) = (n\tilde{T}_1, n)$ where n is the smallest integer such that $p = n\tilde{T}_1$ is an integer. We then apply a DE-SCo with parameters $\{(nB_1, n\tilde{T}_1), (n\alpha B_1, n(\alpha\tilde{T}_1 + B_1))\}$ to the expanded source stream $\tilde{s}[\cdot]$. It can be readily verified that the proposed construction satisfies both the receivers.

APPENDIX C PROOF OF LEMMA 2

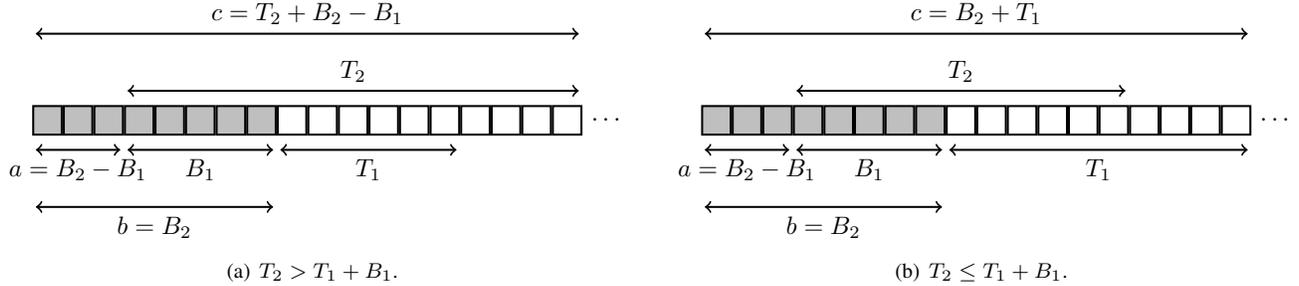


Fig. 13. One period illustration of the Periodic Erasure Channel in Fig. 3 to be used for proving the multicast upper bound provided in Lemma 2.

To establish Lemma 2, we consider the two cases $T_2 > T_1 + B_1$ and $T_2 \leq T_1 + B_1$ separately. When $T_2 > T_1 + B_1$, we consider a periodic erasure channel with period length $\mathcal{T}_p = T_2 + B_2 - B_1$. Each period has B_2 erasures followed by $T_2 - B_1$ unerased symbols as shown in Fig. 13(a). We start with the first period consisting of the channel symbols $\mathbf{x}[t]$ for $t \in [0, \mathcal{T}_p - 1]$ and the decoder proceeds as follows,

- For time $t = 0, 1, \dots, \mathcal{T}_p - 1$, the channel behaves similar to burst erasure channel with B_2 erasures. Hence, the first $B_2 - B_1$ source symbols $s[0], \dots, s[B_2 - B_1 - 1]$ can be recovered using the (B_2, T_2) code within a delay of T_2 symbols, i.e., by time $T_2, \dots, T_2 + B_2 - B_1 - 1 = \mathcal{T}_p - 1$, respectively. The corresponding channel symbols $\mathbf{x}[0], \dots, \mathbf{x}[B_2 - B_1 - 1]$ can then be computed.
- With all the previous channel symbols being recovered, the channel at time $t = B_2 - B_1, \dots, \mathcal{T}_p - 1$ behaves as a burst erasure channel with B_1 erasures. Hence, the B_1 source symbols $s[B_2 - B_1], \dots, s[B_2 - 1]$ can be recovered using the (B_1, T_1) code within a delay of T_1 symbols, i.e., by time $T_1 + B_2 - B_1, \dots, T_1 + B_2 - 1$. We note that the latest recovery time is $T_1 + B_2 - 1 < \mathcal{T}_p - 1$ since $T_2 > T_1 + B_1$.
- It remains to show that the source symbols $s[B_2], \dots, s[\mathcal{T}_p - 1]$ are also recovered. For time $t = B_2, \dots, \mathcal{T}_p + B_2 - 1$, the channel behaves as a burst erasure channel which introduces a burst of length B_2 spanning the interval $[\mathcal{T}_p, \mathcal{T}_p + B_2 - 1]$. Hence, the source symbols $s[B_2], \dots, s[\mathcal{T}_p - 1]$ can be recovered using the (B_2, T_2) code.

The above steps can be repeated across all periods. Since the length of each period is $\mathcal{T}_p = T_2 + B_2 - B_1$ and contains B_2 erasures, any $\{(B_1, T_1), (B_2, T_2)\}$ Mu-SCo with $T_2 > T_1 + B_1$ must satisfy,

$$R \leq \frac{T_2 - B_1}{T_2 - B_1 + B_2} \quad (89)$$

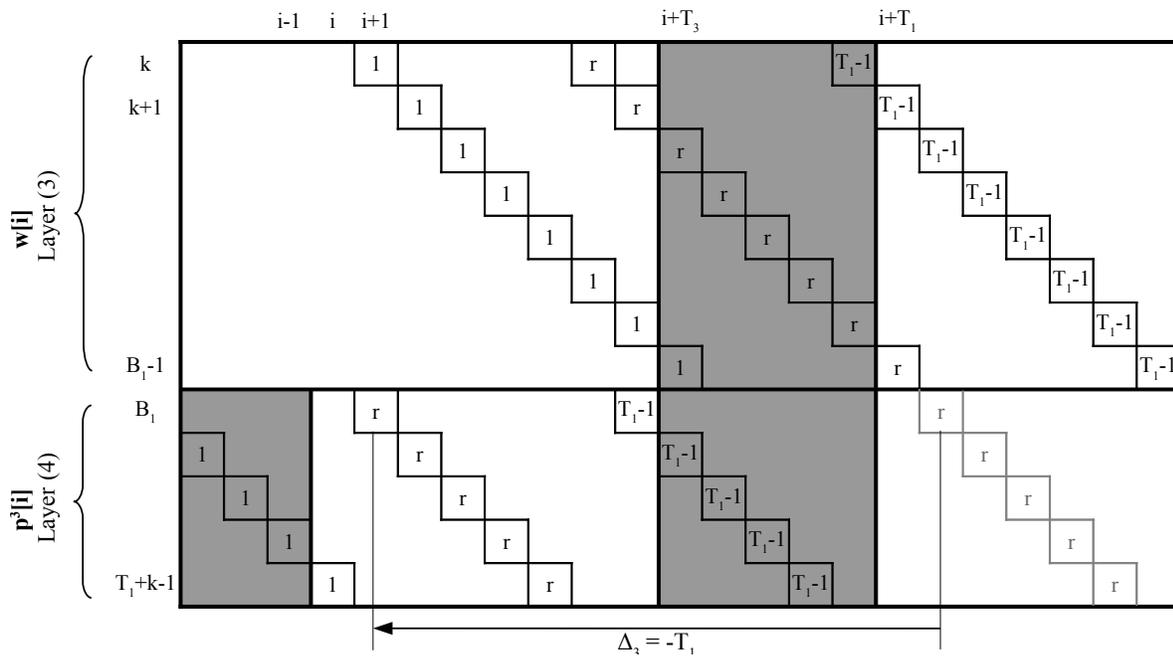


Fig. 14. Diagonal Embedding of parity checks for the construction in Section VIII. The parity checks $\mathbf{p}^3[\cdot]$ in layer (4) are generated using a (B_3, T_3) MS code to the last $B_1 - k$ parity checks of $\mathbf{p}^1[\cdot]$ in layer 3. The parity checks $\mathbf{p}^3[\cdot]$ are shifted back by T_1 units as discussed before and only the causal part of these parities are used.

which gives our upper bound on the rate.

For the case with $T_2 \leq T_1 + B_1$, a similar argument applies except that the period length is $\mathcal{T}_p = T_1 + B_2$ with B_2 erasures in each period (see Fig. 13(b)) and the corresponding upper bound is given by,

$$R \leq \frac{T_1}{T_1 + B_2}. \quad (90)$$

This completes the proof.

APPENDIX D PROOF OF LEMMA 3

We restate the lemma for convenience of the reader

Lemma. *When the erasure burst spans the interval $\mathcal{I}_2 = [i - B_2, i - 1]$, the decoder at receiver 2 can recover all the overlapping parity sub-symbols of $p_j^1[t]$ for $t \in \mathcal{J}_2 = [i + B_1 - k, i + T_1 - 1]$ and $j \in \{k, \dots, B_1 - 1\}$ by time t , using $\overleftarrow{p}_j^3[t + T_1]_t$ for $t \in \mathcal{J}_1 = [i, i + B_1 - k - 1]$ and the unerased source symbols starting from time i .*

First recall that the parity check sub-symbols of \mathcal{C}_1 that span the interval $t \in \mathcal{J}_1 = [i, i - B_2 + T_2 - 1]$ are available to the decoder as they combine $\mathbf{p}^2[t] = \mathbf{s}[t - T_2]$ that are not erased. Hence, $\overleftarrow{p}_{j_2}^3[t]$ for $j_2 \in \{0, \dots, B_3 - 1\}$ and $t \in \mathcal{J}_1$ are recovered at the decoder.

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

In our proof, it will be convenient to define the parity-check symbols of \mathcal{C}_1 that needs to be recovered as:

$$\mathbf{w}[t] = (w_0[t], \dots, w_{T_3-1}[t]) = (p_k^1[t], \dots, p_{B_1-1}^1[t]), \quad t \in \{i + T_3, \dots, i + T_1 - 1\}. \quad (91)$$

We first consider case (A), i.e., when $T_1 \leq 2(B_1 - k)$. Since \mathcal{C}_3 is a MS code 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:

$$\bar{\mathbf{d}}_r = (w_0[i + r], \dots, w_{T_3-1}[i + r + T_3 - 1], p_0^3[i + r + T_3], \dots, p_{B_3-1}^3[i + r + T_3 + B_3 - 1]), \quad r \in \{1, \dots, T_3 + B_3 - 1\}. \quad (92)$$

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

$$\mathbf{d}_r = (w_0[i + r], \dots, w_{T_3-1}[i + r + T_3 - 1], \overleftarrow{p}_0^3[i + r + T_3]_{i+r-B_3}, \dots, \overleftarrow{p}_{B_3-1}^3[i + r + T_3 + B_3 - 1]_{i+r-1}). \quad (93)$$

where recall that $\overleftarrow{p}_j[t_1]_{t_2}$ denotes the causal part of the parity check $p_j[t_1]$ w.r.t. t_2 (cf. Definition 2).

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

The following conditions are sufficient to establish Lemma 3.

- c1 The diagonals \mathbf{d}_r in (93) for $r \in \{1, \dots, T_3 + B_3 - 1\}$ span all the parity-check sub-symbols that need to be recovered, i.e., $p_j^1[\cdot]$ for $j \in \{k, \dots, B_1 - 1\}$ in the interval $\mathcal{J}_2 = [i + T_2 - B_2, i + T_1 - 1]$.
- c2 The decoder can compute the non-causal part of each parity-check $p_j^3[\cdot]$ in the interval $\mathcal{J}_2 = [i + T_2 - B_2, i + T_1 - 1]$ and reduce (93) to (92). Furthermore 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 $p_{j_1}^3[t_x]$ required for the recovery of a given parity check $w_{j_2}[t_y]$ should combine source sub-symbols $s[\cdot]$ which are both, not erased and from time earlier than t_y .
- c3 Each diagonal \mathbf{d}_r should have no more than B_3 erased sub-symbols.

For (c1), we note that the diagonal \mathbf{d}_1 covers $w_{T_3-1}[i + T_3] = p_{B_1-1}^1[i + T_3]$ 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 $w_0[i + T_3 + B_3 - 1] = p_k^1[i + T_3 + B_3 - 1]$ which is the upper right most sub-symbol that needs to be recovered. Fig. 14 easily illustrates that the diagonal \mathbf{d}_r for $r \in [1, T_3 + B_3 - 1]$ cover all of the erased sub-symbols in the interval \mathcal{J}_2 .

For (c2), consider the sub-symbols $w_0[i + r], \dots, w_{T_3-1}[i + r + T_3 - 1]$ of the diagonal \mathbf{d}_r . These involve source symbols $s[\cdot]$ from time $i + r - 1$ and earlier according to the diagonal interleaving property of \mathcal{C}_1 . Thus, one can conclude that the non-causal part of any parity-check sub-symbol $p_j^3[i + r + T_3 + j]$ with respect to $i + r - B_3 + j$ for $j \in \{0, \dots, B_3 - 1\}$ 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 (92) 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 $r \in \{1, \dots, T_1 - T_3\}$

In this range, one can see that the following symbols are available,

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

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 $r \in \{T_1 - T_3 + 1, \dots, T_3\}$

In these diagonals, the following symbols are available,

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

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 $r \in \{T_3 + 1, \dots, T_3 + B_3 - 1\}$

The available sub-symbols in these diagonals are,

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

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 (c3) follows. We note that LD-BEBC codes are capable of recovering wrap-around bursts which may start at the end of the block and wrap around to the beginning of that block.

This completes the proof when $T_1 \leq 2(B_1 - k)$.

A. $T_1 > 2(B_1 - k)$

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 (60). 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 $(p_k^1[t_1], \dots, p_{B_1-1}^1[t_1])$ for $t_1 \in \{i + (B_1 - k), \dots, i + (r + 1)(B_1 - k) - 1\} = \{i + T_2 - B_2, \dots, i + T_1 - q - 1\}$. 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 $(p_k^1[t_2], \dots, p_{B_1-1}^1[t_2])$ for $t_2 \in \{i + (r + 1)(B_1 - k), \dots, i + (r + 1)(B_1 - k) + q - 1\} = \{i + T_1 - q, \dots, i + T_1 - 1\}$ 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 $\mathcal{C}_3 = (B_3, T_3) = (T_1 - (B_1 - k), B_1 - k)$ done above, except that $B_3 = T_1 - (B_1 - k)$ is replaced by $B_{3,r+1} = q$.

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
(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)	$s_3[i-10]$	$s_3[i-9]$	$s_3[i-8]$	$s_3[i-7]$	$s_3[i-6]$	$s_3[i-5]$
	+	+	+	+	+	+
	$\overleftarrow{p}_1[i+2]+\overleftarrow{p}_3[i+4]$	$\overleftarrow{p}_1[i+3]+\overleftarrow{p}_3[i+5]$	$\overleftarrow{p}_1[i+4]+\overleftarrow{p}_3[i+6]$	$\overleftarrow{p}_1[i+5]+\overleftarrow{p}_3[i+7]$	$\overleftarrow{p}_1[i+6]+\overleftarrow{p}_3[i+8]$	$\overleftarrow{p}_1[i+7]+\overleftarrow{p}_3[i+9]$
	$s_4[i-10]$	$s_4[i-9]$	$s_4[i-8]$	$s_4[i-7]$	$s_4[i-6]$	$s_4[i-5]$
+	+	+	+	+	+	
$\overleftarrow{p}_2[i+2]+\overleftarrow{p}_3[i+3]$	$\overleftarrow{p}_2[i+3]+\overleftarrow{p}_3[i+4]$	$\overleftarrow{p}_2[i+4]+\overleftarrow{p}_3[i+5]$	$\overleftarrow{p}_2[i+5]+\overleftarrow{p}_3[i+6]$	$\overleftarrow{p}_2[i+6]+\overleftarrow{p}_3[i+7]$	$\overleftarrow{p}_2[i+7]+\overleftarrow{p}_3[i+8]$	

TABLE V

RATE 5/11 MU-SCO 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 $i - T_1$, WE USE $\overleftarrow{p}_j[i]$ INSTEAD OF $\overleftarrow{p}_j[i]|_{i-T_1}$ FOR SIMPLICITY.

APPENDIX E

EXAMPLES OF CODE CONSTRUCTION IN REGION (E)

We give the construction for two specific points in this region, Table V shows the code construction for the point $\{(4, 5), (7, 10)\}$ whereas Table VI 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): $\{(4, 5), (7, 10)\}$

Using the relations, $T_2 = T_1 + B_1 + k$ and $B_2 = T_1 + k + m$, we have that $k = m = 1$.

The code construction achieving the optimal rate of 5/11 is illustrated in Table V. In this example, we walk through the steps of both the encoder and the decoder. We note that this point denotes 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 ($s_0[\cdot], \dots, s_4[\cdot]$). A $\mathcal{C}_1 = (4, 5)$ is applied along the diagonal of such source sub-symbols producing $B_1 = 4$ parity check sub-symbols ($p_0[\cdot], \dots, p_3[\cdot]$) defined as follows,

$$\begin{aligned}
p_0[i] &= s_0[i-5] + s_4[i-1] \\
p_1[i] &= s_1[i-5] + s_4[i-2] \\
p_2[i] &= s_2[i-5] + s_4[i-3] \\
p_3[i] &= s_3[i-5] + s_4[i-4]
\end{aligned} \tag{94}$$

- The $T_1 = 5$ parity check symbols of code $\mathcal{C}_2 = (10, 10)$ which are repetitions of the source sub-symbols such that $p_j^2[i] = s_j[i-10]$ for $j \in \{0, \dots, 4\}$ are concatenated to the parity checks of \mathcal{C}_1 with partial overlap of $B_1 - k = 3$ rows as shown in Table V.
- A $\mathcal{C}_3 = (T_1 - (B_1 - k), B_1 - k) = (2, 3)$ MS code is applied to the last $B_1 - k = 3$ rows of parity check sub-symbols of \mathcal{C}_1 , ($p_1[\cdot], p_2[\cdot], p_3[\cdot]$) producing $T_1 - (B_1 - k) = 2$ parity check sub-symbols, ($p_0^3[\cdot], p_1^3[\cdot]$). 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 cannot 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 $\overleftarrow{p}_j[\cdot]|_t$) 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., $\overleftarrow{p}_0^3[i+5]|_i = \overleftarrow{p}_1[i+2]|_i + \overleftarrow{p}_3[i+4]|_i = s_1[i-3] + s_3[i-1] + s_4[i]$.

According to Fig. 4, we divide each channel symbol 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 the 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 $t \in \{i - 10, \dots, i - 6\}$. 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 \mathcal{C}_2 in the interval $[i, i + 2]$ are unerased source sub-symbols. Thus, the corresponding combined 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) MS 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 (cf. (94)).
- (c) Steps (a) and (b) show that all the parity check sub-symbols of \mathcal{C}_1 in layer (3) can be computed except for the interval $[i + 3, i + 4]$.
- (d) The parity check sub-symbols of \mathcal{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 \mathcal{C}_3 can be computed in this interval.
- (e) The parity-check sub-symbols of \mathcal{C}_3 in the interval $[i, i + 2]$,

$$\begin{pmatrix} p_0^3[i + 5] & p_0^3[i + 6] & p_0^3[i + 7] \\ p_1^3[i + 5] & p_1^3[i + 6] & p_1^3[i + 7] \end{pmatrix}, \quad (95)$$

can recover the remaining two columns of parity-check sub-symbols of \mathcal{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 \mathcal{C}_3 is a $(2, 3)$ MS code whose parity-check sub-symbols are shifted back by $T_1 = 5$.

However, only the causal part of the parity checks of \mathcal{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 MS code. Using (94), it can be seen that the recovery of the non-causal part does not require the availability of source sub-symbols after time⁴ $i + 3$. For example, the non-causal part of $p_0^3[i + 5]$ is $\vec{p}_0^3[i + 5] \Big|_i = s_4[i + 1]$ which is clearly available before time $i + 3$. Thus the non-causal portions of all the parity checks are computed and then (95) 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 $i + 4$.
- **Step (3):** Furthermore, one can see that the parity check sub-symbols of \mathcal{C}_3 interfering in the last two rows (layer (4)) starting at time $i + 3$ combine parity check sub-symbols of \mathcal{C}_1 of indices $i + 5$ and later which was shown before to combine unerased source sub-symbols (cf. (94)).

According to Step (2) and (3), the parity checks of $\mathcal{C}_2 = (10, 10)$ 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 $[i - 7, i - 1]$.

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 VI. 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 $(s_0[\cdot], \dots, s_4[\cdot])$ (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 $(p_0[\cdot], p_1[\cdot], p_2[\cdot])$ defined as

⁴A proof of this in the general case is provided in the proof of Lemma 3 in Appendix D.

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
(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)	$s_2[i-9]+\overleftarrow{p}_1[i+2]$	$s_2[i-8]+\overleftarrow{p}_1[i+3]$	$s_2[i-7]+\overleftarrow{p}_1[i+4]$	$s_2[i-6]+\overleftarrow{p}_1[i+5]$	$s_2[i-5]+\overleftarrow{p}_1[i+6]$	$s_2[i-4]+\overleftarrow{p}_1[i+7]$
	$s_3[i-9]+\overleftarrow{p}_2[i+2]$	$s_3[i-8]+\overleftarrow{p}_2[i+3]$	$s_3[i-7]+\overleftarrow{p}_2[i+4]$	$s_3[i-6]+\overleftarrow{p}_2[i+5]$	$s_3[i-5]+\overleftarrow{p}_2[i+6]$	$s_3[i-4]+\overleftarrow{p}_2[i+7]$
	$s_4[i-9]$	$s_4[i-8]$	$s_4[i-7]$	$s_4[i-6]$	$s_4[i-5]$	$s_4[i-4]$
	$\overleftarrow{p}_1[i+3]+\overleftarrow{p}_2[i+4]$	$\overleftarrow{p}_1[i+4]+\overleftarrow{p}_2[i+5]$	$\overleftarrow{p}_1[i+5]+\overleftarrow{p}_2[i+6]$	$\overleftarrow{p}_1[i+6]+\overleftarrow{p}_2[i+7]$	$\overleftarrow{p}_1[i+7]+\overleftarrow{p}_2[i+8]$	$\overleftarrow{p}_1[i+8]+\overleftarrow{p}_2[i+9]$

TABLE VI

RATE 5/11 MU-SCO 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 $i - T_1$, WE USE $\overleftarrow{p}_j[i]$ INSTEAD OF $\overleftarrow{p}_j[i]|_{i-T_1}$ FOR SIMPLICITY.

follows,

$$\begin{aligned}
p_0[i] &= s_0[i-5] + s_3[i-2] \\
p_1[i] &= s_1[i-5] + s_4[i-2] \\
p_2[i] &= s_2[i-5] + s_3[i-4] + s_4[i-3].
\end{aligned} \tag{96}$$

- 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 VI.
- 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, $r + 1 = 2$ MS 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 , $(p_1[\cdot], p_2[\cdot])$ 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 $\mathcal{C}_{3,2} = (q, B_1 - k) = (1, 2)$ MS code applied again on the last two rows of parity check sub-symbols of \mathcal{C}_1 diagonally 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 \mathcal{C}_2 (layer (4)).

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 $\overleftarrow{p}_j[\cdot]|_t$ are sent instead (cf. Table VI).

Similar to the previous example, we divide each channel symbol into four layers (cf. Fig. 4),

- 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 $t \in \{i-9, \dots, i-5\}$. 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.

- **Step (1):** 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) MS 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 (cf. (96)).

- (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 $(p_0^3[\cdot], p_1^3[\cdot])$ and a $\mathcal{C}_{3,2} = (1, 2)$ MS code producing a single parity-check sub-symbol $p_2^3[\cdot]$. At time i and $i+1$, the parity checks of $\mathcal{C}_{3,1}$,

$$\begin{pmatrix} \overleftarrow{p}_0^3[i] \\ \overleftarrow{p}_1^3[i] \end{pmatrix}_i = \begin{pmatrix} \overleftarrow{p}_1[i+2] \\ \overleftarrow{p}_2[i+2] \end{pmatrix}_i,$$

thus, $\overleftarrow{p}_1[i+2]_i$ and $\overleftarrow{p}_2[i+2]_i$ can be directly recovered, while their corresponding non-causal parts can be computed before time $i+2$. Similarly, $\overleftarrow{p}_1[i+3]_i$ and $\overleftarrow{p}_2[i+3]_i$ can be recovered at time $i+1$ and their corresponding non-causal parts can be retrieved before $i+3$. The remaining column, $(\overleftarrow{p}_1[i+4]_i, \overleftarrow{p}_2[i+4]_i)$ can be recovered using the parity checks of $\mathcal{C}_{3,2} = (1, 2)$ MS 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.

- **Step (2):** 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 $i+4$.
- **Step (3):** Remove interference in layer (4) starting at time $i+2$.

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 (cf. (96)).

According to Step (2) and (3), 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 can use the parity-checks in layer (3) and (4) to recover the erased source symbols, $\mathbf{s}[i-7], \dots, \mathbf{s}[i-1]$.

APPENDIX F PROOF OF LEMMA 4

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

$$H(\mathbf{s}[i-T_1] | \mathbf{x}_{[i-T_1+B_1]}^i, \mathbf{x}_{[0]^{i-T_1-1}}) = 0, \quad (97)$$

which follows from (83) by substituting $j = i - T_1$ and $r_1 = 0$. Also, using the (B_2, B_2) decoder, we can write,

$$H(\mathbf{s}[i-B_2] | \mathbf{x}[i], \mathbf{x}_{[0]^{i-B_2-1}}) = 0 \quad (98)$$

which again follows from (83) but with $j = i - B_2$ and $r_1 = 0$. This can be used in the following steps

$$\begin{aligned} H(\mathbf{x}[i]) &\geq H(\mathbf{x}[i] | \mathbf{x}_{[0]^{i-B_2-1}}) = H(\mathbf{s}[i-B_2], \mathbf{x}[i] | \mathbf{x}_{[0]^{i-B_2-1}}) - H(\mathbf{s}[i-B_2] | \mathbf{x}[i], \mathbf{x}_{[0]^{i-B_2-1}}) \\ &= H(\mathbf{s}[i-B_2], \mathbf{x}[i] | \mathbf{x}_{[0]^{i-B_2-1}}) \\ &= H(\mathbf{s}[i-B_2]) + H(\mathbf{x}[i] | \mathbf{s}[i-B_2], \mathbf{x}_{[0]^{i-B_2-1}}). \end{aligned} \quad (99)$$

We use mathematical induction to prove (76). For the base case, (76) at $m = 2B_2 - B_1 - 1$ is already proved by the result in (75).

For the inductive step, we assume that (76) is true for $m = j$, i.e.,

$$\sum_{i=B_2}^j H(\mathbf{x}[i]) \geq H(\mathbf{s}_{[0]^{j-B_2}}) + H(\mathbf{s}_{[B_2-B_1]^{j-T_1}}) + H(\mathbf{x}_{[B_2]}^j | \mathbf{s}_{[0]^{j-B_2}}, \mathbf{s}_{[B_2-B_1]^{j-T_1}}, \mathbf{x}_{[0]^{j-B_2}}). \quad (100)$$

We then add $H(\mathbf{x}[j+1])$ to both sides, and use (97) and (98) to recover the source symbols $\mathbf{s}[j+1-B_2]$ and $\mathbf{s}[j+1-T_1]$ respectively as follows,

$$\sum_{i=B_2}^{j+1} H(\mathbf{x}[i]) \stackrel{(a)}{\geq} H(\mathbf{s}_{[0]^{j-B_2}}) + H(\mathbf{s}_{[B_2-B_1]^{j-T_1}}) + H(\mathbf{x}_{[B_2]}^{j+1} | \mathbf{s}_{[0]^{j-B_2}}, \mathbf{s}_{[B_2-B_1]^{j-T_1}}, \mathbf{x}_{[0]^{j-B_2}})$$

$$\begin{aligned}
& + 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) \\
& - H\left(\mathbf{s}[j+1-T_1] \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 \end{bmatrix}, \mathbf{x} \begin{bmatrix} j+1-B_2 \\ 0 \end{bmatrix}\right) \\
\stackrel{(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} \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] \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{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) \\
\stackrel{(c)}{=} & 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) \tag{101}
\end{aligned}$$

Step (a) is the addition of (99) and (76), step (b) uses the fact that $j \geq 2B_2 - B_1 - 1$ and thus $B_2 - B_1 \leq j + 1 - B_2$ and thus:

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

which follows using (97), and step (c) uses the fact that the source symbols are independent of each other together with the fact that $\mathbf{s}[j+1-T_1] \notin \mathbf{s} \begin{bmatrix} j+1-B_2 \\ 0 \end{bmatrix}$ since $B_2 > T_1$ is satisfied throughout region (f). The result is in the form (76) for $m = j + 1$.

REFERENCES

- [1] T. Huang, P. Huang, K. Chen, and P. Wang. Could Skype be more satisfying? a QoE-centric study of the FEC mechanism in an internet-scale VoIP system. *IEEE Network*, 24(2):42–48, 2010.
- [2] C. Yu, Y. Xu, B. Liu, and Y. Liu. Can you SEE me now? A Measurement Study of Mobile Video Calls. In *Proceedings of IEEE Conference on Computer and Communications (INFOCOM)*, 2014.
- [3] E. Martinian. *Dynamic Information and Constraints in Source and Channel Coding*. PhD thesis, Massachusetts Institute of Technology (MIT), 2004.
- [4] E. Martinian and C. W. Sundberg. Burst erasure correction codes with low decoding delay. *IEEE Transactions on Information Theory*, 50(10):2494–2502, 2004.
- [5] E. Martinian and M. Trott. Delay-optimal burst erasure code construction. In *Proc. International Symposium on Information Theory (ISIT)*, Nice, France, July 2007.
- [6] M. Kalman, E. Steinbach, and B. Girod. Adaptive media playout for low-delay video streaming over error-prone channels. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:841 – 851, June 2004.
- [7] A. Badr, A. Khisti, and E. Martinian. Diversity embedded streaming erasure codes (de-sco): Constructions and optimality. In *Proc. Global Communications Conference (GLOBECOM)*, pages 1–5, 2010.
- [8] A. Badr, A. Khisti, and E. Martinian. Diversity embedded streaming erasure codes (DE-SCO): Constructions and optimality. *IEEE Journal on Selected Areas in Communications (JSAC)*, 29(5):1042–1054, May 2011.
- [9] A. Badr, A. Khisti, W. Tan, and J. Apostolopoulos. Streaming codes for channels with burst and isolated erasures. In *Proc. International Conference on Computer Communications (INFOCOM)*, 2013.
- [10] A. Badr, A. Khisti, W. Tan, and J. Apostolopoulos. Robust streaming erasure codes based on deterministic channel approximations. In *Proc. International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, 2013.
- [11] A. Badr, P. Patil, A. Khisti, W. Tan, and J. Apostolopoulos. Layered constructions for low-delay streaming codes. *CoRR*, abs/1308.3827, 2013.
- [12] P. Patil, A. Badr, and A. Khisti. Delay-optimal streaming codes under source-channel rate mismatch. In *Proc. Asilomar Conference on Signals, Systems & Computers*, 2013.
- [13] D. Leong and T. Ho. Erasure coding for real-time streaming. In *Proc. International Symposium on Information Theory (ISIT)*, 2012.
- [14] D. Leong, A. Qureshi, and T. Ho. On coding for real-time streaming under packet erasures. In *Proc. International Symposium on Information Theory (ISIT)*, 2013.
- [15] H. Guessing-Luerssen, J. Rosenthal, and R. Smarandache. Strongly-mds convolutional codes. *IEEE Transactions on Information Theory*, 52(2):584–598, 2006.
- [16] D. Lui, A. Badr, and A. Khisti. Streaming codes for a double-link burst erasure channel. In *Proc. Canadian Workshop on Information Theory (CWIT)*, 2011.
- [17] D. Lui. Coding theorems for delay sensitive communication over burst-erasure channels. Master’s thesis, University of Toronto, Toronto, ON, August 2011.
- [18] Z. Li, A. Khisti, and B. Girod. Correcting erasure bursts with minimum decoding delay. In *Proc. Asilomar Conference on Signals, Systems & Computers*, 2011.
- [19] A. Badr, D. Lui, and A. Khisti. Multicast streaming codes (Mu-SCO) for burst erasure channels. In *Proc. Allerton Conference on Communication, Control, and Computing*, 2010.
- [20] L. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.
- [21] A. Sahai. *Anytime Information Theory*. PhD thesis, Massachusetts Institute of Technology (MIT), 2001.
- [22] R. Sukhavasi and B. Hassibi. Linear error correcting codes with anytime reliability. In *Proc. International Symposium on Information Theory (ISIT)*, pages 1748–1752, 2011.
- [23] M. C. O. Bogino, P. Cataldi, M. Grangotto, E. Magli, and G. Olmo. Sliding-window digital fountain codes for streaming of multimedia contents. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2007.
- [24] T. Tirronen and J. Virtamo. Finding fountain codes for real-time data by fixed point method. In *Proc. International Symposium on Information Theory (ISIT)*, 2008.
- [25] N. Rahnvard, B. N. Vellambi, and F. Fekri. Rateless codes with unequal error protection property. *IEEE Transactions on Information Theory*, 53(4):1521–1532, 2007.

- [26] Y. Li and E. Soljanin. Rateless codes for single-server streaming to diverse users. In *Proc. Allerton Conference on Communication, Control, and Computing*, 2009.
- [27] Y. Li and E. Soljanin. Rateless codes for single-server streaming to diverse users. *CoRR*, abs/0912.5055, 2009.
- [28] A. Sahai. Why do block length and delay behave differently if feedback is present? *IEEE Transactions on Information Theory*, 54:1860 – 1886, May 2008.
- [29] J. K. Sundararajan and D. Shah and M. Medard. Arq for network coding. In *Proc. International Symposium on Information Theory (ISIT)*, 2008.
- [30] H. Yao, Y. Kochman, and G. W. Wornell. A multi-burst transmission strategy for streaming over blockage channels with long feedback delay. *IEEE Journal on Selected Areas in Communications (JSAC)*, 29(10):2033–2043, 2011.
- [31] Y. Kochman H. Yao and G. W. Wornell. On delay in real-time streaming communication systems. In *Proc. Allerton Conference on Communication, Control, and Computing*, 2010.
- [32] G. W. Wornell H. Yao, Y. Kochman. Delay-throughput tradeoff for streaming over blockage channels with delayed feedback. In *Proc. IEEE Military Communications Conference (MILCOM)*, 2010.
- [33] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley and Sons, 1968.
- [34] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.