ISIT 2009, Seoul, Korea, June 28 - July 3, 2009

On multicasting with streaming burst-erasure codes

Ashish Khisti and Jatinder Pal Singh
Deutsche Telekom Inc. R&D Lab USA
Los Altos, CA, 94040
Email: {ashish.khisti, jatinder.singh}@telekom.com

***Abstract*—We study a multicast extension of streaming burst-erasure codes previously proposed for the single user setting. There are two receivers each interested in the common stream. Each receiver's channel however has a different burst parameter and likewise each receiver tolerates a different delay; both receivers are interested in a common stream. We develop two upper bounding approaches on the streaming multicast capacity. The first upper bound is developed by introducing an erasure channel that introduces periodic erasure bursts, which can corrected due to the multicast property. The second upper bound is based on information theoretic inequalities and is tight at the minimum delay point. Finally we propose a simple multicast code construction by combining the parity checks of two single-user codes.**

I. Introduction

Streaming burst-erasure codes developed in [1], [2], [3] encode live streams with causal encoding and delay constrained decoding. An incoming stream of source packets at the encoder is mapped into a stream of channel packets in a causal manner — the channel packet $\mathsf{x}[i]$ depends on source $\mathsf{s}[0], \ldots, \mathsf{s}[i]$. The channel introduces one burst of contiguous erasures. The receiver observes all packets reliably and without any delay except for a burst of B packets that get erased. The decoder outputs each source packet within a delay of T packets. In other words source packet i is produced by time $i + T$. The *streaming capacity* defined as the maximum rate that can be supported on this channel is characterized in terms of B and T. A complete characterization of the streaming capacity in this setup is provided in [2], [3].

In related work, Sahai [4] considers the impact of peak decoding-delay constraints on achievable error exponents over discrete memoryless channels. The problem of multicasting to two or more receivers over i.i.d. erasure channels with *average* decoding delay constraints has been recently studied in [5], [6].

In this paper we study the design of burst-erasure codes to simultaneously multicast a common stream to two heterogeneous receivers. Each receiver's channel is affected by an erasure-burst of different length. Likewise each receiver experiences a different delay. Both the receivers are interested in the common stream. One application for such codes is video transmission over wireless channels. The wireless channel introduces bursts of packet losses due to outage. When multicasting a common video stream to several receivers, it is of interest to develop such multicast codes. One other relevant scenario where burst erasures are common is peer-to-peer networks. Whenever any node gets disconnected the connected child nodes experience a burst of erasures until a suitable parent is selected again.

II. Problem Formulation

We first review the single user setup in [1], [2], [3].

A. Single receiver case

The encoder receives a stream of i.i.d. source packets $\{\mathsf{s}[t]\}_{t\geq 0}$, each packet is over an alphabet $\mathcal{S}$. It produces a stream of *channel* packets $\{\mathsf{x}[t]\}_{t\geq 0}$. The channel packet at time t depends on on the source packets $\mathsf{s}[0], \mathsf{s}[1], \ldots, \mathsf{s}[t]$ i.e.,

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

for some sequence of functions $f_t : \mathcal{S}^t \to \mathcal{X}$. Throughout this paper we only focus on deterministic mappings $f_t(\cdot)$. The channel introduces a single erasure burst of length B at some arbitrary time i.e., for some $j \geq 0$,

$$\mathsf{y}[t] = \begin{cases} \star, & t \in [j, j+B-1] \\ \mathsf{x}[t], & \text{otherwise.} \end{cases} \tag{2}$$

A decoder with delay $T \geq 0$ outputs the source packet $\mathsf{s}[t]$ at time $t + T$ i.e., there exists a sequence of decoding functions $g_t : \mathcal{Y}^{t+T} \to \mathcal{S}$, such that $\hat{\mathsf{s}}[t] = g_t\left(\mathsf{y}[0], \ldots, \mathsf{y}[t], \ldots, \mathsf{y}[t+T]\right)$ and

$$\Pr(\hat{\mathsf{s}}[t] \neq \mathsf{s}[t]) = 0, \quad \forall t \geq 0. \tag{3}$$

The quantity of interest is the information rate defined as

$$R = \frac{H(\mathsf{s})}{H(\mathsf{x})} \ \text{ bits/symbol.} \tag{4}$$

The maximum attainable rate such that there exists a sequence of encoding and decoding functions that satisfy (3) is the *streaming capacity*. As established in [1],

$$C = \begin{cases} \frac{T}{T+B}, & T \geq B \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Remark 1: Note that for a given source and channel rates, the proposed codes equivalently minimized the peak decoding delay at the decoder. Also even though the setup considers only a single erasure burst, it is possible to show [3] that the resulting codes can correct multiple bursts, provided they are separated by more than $T + B$ symbols. Finally note that even though the focus is on erasures the constructions naturally generalize when no more than B consecutive symbols pass through a noisy channel [7].

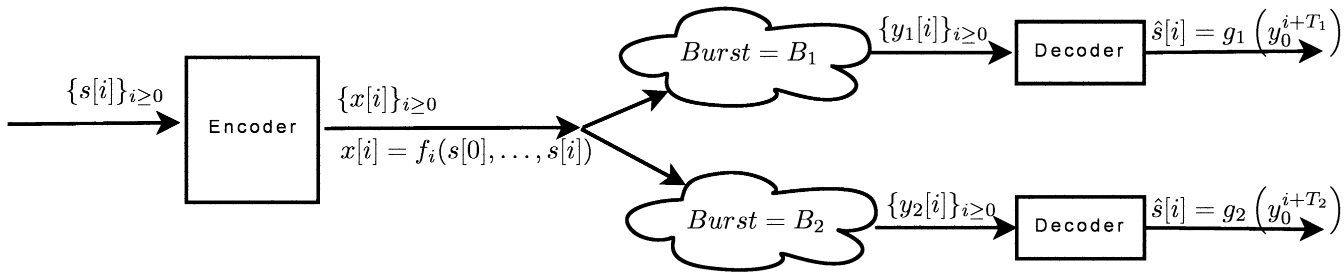

Fig. 1. The extension of streaming capacity setup to the case when there are two receivers. The source stream is mapped into a stream of channel packets by the encoder. Both the receivers observe these packets via their channels. The channel introduces an erasure-burst of length B_i, and it tolerates a delay of T_i, for $i = 1, 2$.

B. Two receiver case

Fig. 1 shows the setup of interest. The encoder receives a stream of source packets $\{\mathsf{s}[t]\}_{t\geq 0}$ that needs to be reproduced at both the receivers. The channel packets $\{\mathsf{x}[t]\}_{t\geq 0}$ are produced causally from the source stream via a function $f_t(\cdot)$ c.f. (1). Denote the channel output at receiver i at time t by $\mathsf{y}_i[t]$. The channel of receiver i introduces an erasure-burst of length B_i i.e., for some $j_i \geq 0$, we have that

$$\mathsf{y}_i[t] = \begin{cases} \star, & t \in [j_i, j_i + B_i - 1] \\ \mathsf{x}[t], & \text{otherwise,} \end{cases} \tag{6}$$

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

$$\hat{\mathsf{s}}_i[t] = \gamma_{it}\left(\mathsf{y}_1[0], \mathsf{y}_1[1], \ldots, \mathsf{y}_1[t + T_i]\right), \quad i = 1, 2, \tag{7}$$

and

$$\Pr(\mathsf{s}_i[t] \neq \hat{\mathsf{s}}_i[t]) = 0, \quad \forall t \geq 0,\ i = 1, 2. \tag{8}$$

As before, we study the streaming capacity for the two-user setup.

III. Multicast code construction

Our code construction exploits the systematic nature of the single user burst-erasure codes and concatenates the parity checks of the two users. This results in the following lower bound on the streaming capacity

Theorem 1: An achievable rate for the multicast streaming capacity using the burst-erasure codes is

$$R = \frac{1}{1 + \sum_{i=1}^{2} \frac{B_i}{T_i}} - o_S(1), \tag{9}$$

where the term $o_S(1)$ vanishes to zero as the size of the source alphabet $|\mathcal{S}|$ goes to infinity.

Before describing our construction, we summarize for the convenience of the reader, the single-user code construction in [3].

A. Single user burst-erasure codes

A single user code of rate $T/(T+B)$ is constructed by the following steps

1) Construct a systematic $(B+T, T)$ block-code that can correct a burst erasure of length B within a delay of T. An explicit construction of such codes is provided in [3], [2].
2) Split each source symbol $\mathsf{s}[i]$ into T sub-symbols and apply the block code designed above on a diagonally interleave sub symbol stream.

B. Numerical Examples for Single User and Multicast Codes

It is instructive to illustrate the multicast code construction via a numerical example. The single user code constructions for $(B, T) = (1, 2)$ and $(B, T) = (2, 4)$ as well as the resulting multicast code are illustrated in Fig. 2.

Note that for $(B, T) = (1, 2)$, a systematic block code that can correct a single erasure with a delay of $T = 2$ is $(\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_0 \oplus \mathsf{v}_1)$. By inspection it can be verified that if any one symbol is erased, the remaining symbols can be recovered within a delay of $T = 1$. We construct a streaming code from the block code as follows: split each symbol $\mathsf{s}[i]$ into two sub-symbols $(\mathsf{s}_0[i], \mathsf{s}_1[i])$. After diagonal interleaving, as shown in Fig. 2(a), the resulting coded symbol is

$$\mathsf{x}[i] = (\mathsf{s}_0[i], \mathsf{s}_1[i], \mathsf{s}_0[i-2] \oplus \mathsf{s}_1[i-1]).$$

Similarly we show the construction of a streaming code for $B = 2$ and $T = 4$ in Fig. 2(b). Here the corresponding systematic block code is $(\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \mathsf{v}_3, \mathsf{v}_0 \oplus \mathsf{v}_2, \mathsf{v}_1 \oplus \mathsf{v}_3)$. For the streaming code, each source symbol is split into four sub-symbols $(\mathsf{s}_0[i], \mathsf{s}_1[i], \mathsf{s}_2[i], \mathsf{s}_3[i])$ and diagonally interleaved to apply the block code giving $\mathsf{x}[i] = (\mathsf{s}_0[i], \mathsf{s}_1[i], \mathsf{s}_2[i], \mathsf{s}_3[i], \mathsf{s}_0[i-4] \oplus \mathsf{s}_2[i-2], \mathsf{s}_1[i-4] \oplus \mathsf{s}_3[i-2])$.

Note that both these code constructions have a systematic part and a parity check part. The construction of an encoder that simultaneously serves two users, with parameters $(B_1, T_1) = (1, 2)$ and $(B_2, T_2) = (2, 4)$, as illustrated in Fig. 2(c) exploits this systematic code-structure. Each source symbol is split into four sub-symbols $(\mathsf{s}_0[i], \mathsf{s}_1[i], \mathsf{s}_2[i], \mathsf{s}_3[i])$. The parity checks for (B_1, T_1) are generated by treating $\mathsf{t}_0[i] = (\mathsf{s}_0[i], \mathsf{s}_1[i])$ and $\mathsf{t}_1[i] = (\mathsf{s}_2[i], \mathsf{s}_3[i])$ as super-symbols. The resulting parity check symbols are,

$$\mathsf{p}_1[i] = \mathsf{t}_0[i-2] \oplus \mathsf{t}_1[i-1] = \begin{pmatrix} \mathsf{s}_0[i-2] \oplus \mathsf{s}_2[i-1] \\ \mathsf{s}_1[i-2] \oplus \mathsf{s}_3[i-1] \end{pmatrix}.$$

The parity check symbols for (B_2, T_2) using the above code

$\boxed{\mathsf{s}_0[i-2]}$	$\mathsf{s}_0[i-1]$	$\mathsf{s}_0[i]$	$\mathsf{s}_0[i+1]$	$\mathsf{s}_0[i+2]$	$\mathsf{s}_0[i+3]$
$\mathsf{s}_1[i-2]$	$\boxed{\mathsf{s}_1[i-1]}$	$\mathsf{s}_1[i]$	$\mathsf{s}_1[i+1]$	$\mathsf{s}_1[i+2]$	$\mathsf{s}_1[i+3]$
$\mathsf{s}_0[i-4]\oplus\mathsf{s}_1[i-3]$	$\mathsf{s}_0[i-3]\oplus\mathsf{s}_1[i-2]$	$\boxed{\mathsf{s}_0[i-2]\oplus\mathsf{s}_1[i-1]}$	$\mathsf{s}_0[i-1]\oplus\mathsf{s}_1[i]$	$\mathsf{s}_0[i]\oplus\mathsf{s}_1[i+1]$	$\mathsf{s}_0[i+1]\oplus\mathsf{s}_1[i+2]$

(a) Burst-erasure code for $B=1$ and $T=2$. The code is formed from the block-code by splitting each source symbols into two sub-symbols of equal length and then diagonally interleaving the symbols for the block code as illustrated above.

$\boxed{\mathsf{s}_0[i-2]}$	$\mathsf{s}_0[i-1]$	$\mathsf{s}_0[i]$	$\mathsf{s}_0[i+1]$	$\mathsf{s}_0[i+2]$	$\mathsf{s}_0[i+3]$
$\mathsf{s}_1[i-2]$	$\boxed{\mathsf{s}_1[i-1]}$	$\mathsf{s}_1[i]$	$\mathsf{s}_1[i+1]$	$\mathsf{s}_1[i+2]$	$\mathsf{s}_1[i+3]$
$\mathsf{s}_2[i-2]$	$\mathsf{s}_2[i-1]$	$\boxed{\mathsf{s}_2[i]}$	$\mathsf{s}_2[i+1]$	$\mathsf{s}_2[i+2]$	$\mathsf{s}_2[i+3]$
$\mathsf{s}_3[i-2]$	$\mathsf{s}_3[i-1]$	$\mathsf{s}_3[i]$	$\boxed{\mathsf{s}_3[i+1]}$	$\mathsf{s}_3[i+2]$	$\mathsf{s}_3[i+3]$
$\mathsf{s}_0[i-6]\oplus\mathsf{s}_2[i-4]$	$\mathsf{s}_0[i-5]\oplus\mathsf{s}_2[i-3]$	$\mathsf{s}_0[i-4]\oplus\mathsf{s}_2[i-2]$	$\mathsf{s}_0[i-3]\oplus\mathsf{s}_2[i-1]$	$\boxed{\mathsf{s}_0[i-2]\oplus\mathsf{s}_2[i]}$	$\mathsf{s}_0[i-1]\oplus\mathsf{s}_2[i+1]$
$\mathsf{s}_1[i-6]\oplus\mathsf{s}_3[i-4]$	$\mathsf{s}_1[i-5]\oplus\mathsf{s}_3[i-3]$	$\mathsf{s}_1[i-4]\oplus\mathsf{s}_3[i-2]$	$\mathsf{s}_1[i-3]\oplus\mathsf{s}_3[i-1]$	$\mathsf{s}_1[i-2]\oplus\mathsf{s}_3[i]$	$\boxed{\mathsf{s}_1[i-1]\oplus\mathsf{s}_3[i+1]}$

(b) Burst-erasure code for $B=2$ and $T=4$. The code is formed from the block-code by splitting each source symbols into four sub-symbols of equal length and then diagonally interleaving the symbols for the block code as illustrated above.

$\boxed{\boxed{\mathsf{s}_0[i-2]}}$	$\mathsf{s}_0[i-1]$	$\mathsf{s}_0[i]$	$\mathsf{s}_0[i+1]$	$\mathsf{s}_0[i+2]$	$\mathsf{s}_0[i+3]$
$\boxed{\mathsf{s}_1[i-2]}$	$\boxed{\mathsf{s}_1[i-1]}$	$\mathsf{s}_1[i]$	$\mathsf{s}_1[i+1]$	$\mathsf{s}_1[i+2]$	$\mathsf{s}_1[i+3]$
$\mathsf{s}_2[i-2]$	$\boxed{\mathsf{s}_2[i-1]}$	$\boxed{\mathsf{s}_2[i]}$	$\mathsf{s}_2[i+1]$	$\mathsf{s}_2[i+2]$	$\mathsf{s}_2[i+3]$
$\mathsf{s}_3[i-2]$	$\boxed{\mathsf{s}_3[i-1]}$	$\mathsf{s}_3[i]$	$\boxed{\mathsf{s}_3[i+1]}$	$\mathsf{s}_3[i+2]$	$\mathsf{s}_3[i+3]$
$\mathsf{s}_0[i-6]\oplus\mathsf{s}_2[i-4]$	$\mathsf{s}_0[i-5]\oplus\mathsf{s}_2[i-3]$	$\mathsf{s}_0[i-4]\oplus\mathsf{s}_2[i-2]$	$\mathsf{s}_0[i-3]\oplus\mathsf{s}_2[i-1]$	$\boxed{\mathsf{s}_0[i-2]\oplus\mathsf{s}_2[i]}$	$\mathsf{s}_0[i-1]\oplus\mathsf{s}_2[i+1]$
$\mathsf{s}_1[i-6]\oplus\mathsf{s}_3[i-4]$	$\mathsf{s}_1[i-5]\oplus\mathsf{s}_3[i-3]$	$\mathsf{s}_1[i-4]\oplus\mathsf{s}_3[i-2]$	$\mathsf{s}_1[i-3]\oplus\mathsf{s}_3[i-1]$	$\mathsf{s}_1[i-2]\oplus\mathsf{s}_3[i]$	$\boxed{\mathsf{s}_1[i-1]\oplus\mathsf{s}_3[i+1]}$
$\mathsf{s}_0[i-4]\oplus\mathsf{s}_2[i-3]$	$\mathsf{s}_0[i-3]\oplus\mathsf{s}_2[i-2]$	$\boxed{\mathsf{s}_0[i-2]\oplus\mathsf{s}_2[i-1]}$	$\mathsf{s}_0[i-1]\oplus\mathsf{s}_2[i]$	$\mathsf{s}_0[i]\oplus\mathsf{s}_2[i+1]$	$\mathsf{s}_0[i+1]\oplus\mathsf{s}_2[i+2]$
$\mathsf{s}_1[i-4]\oplus\mathsf{s}_3[i-3]$	$\mathsf{s}_1[i-3]\oplus\mathsf{s}_3[i-2]$	$\boxed{\mathsf{s}_1[i-2]\oplus\mathsf{s}_3[i-1]}$	$\mathsf{s}_1[i-1]\oplus\mathsf{s}_3[i]$	$\mathsf{s}_1[i]\oplus\mathsf{s}_3[i+1]$	$\mathsf{s}_1[i+1]\oplus\mathsf{s}_3[i+2]$

(c) A code construction to simultaneously support two users, one with $(B,T)=(2,4)$ and the other with $(B,T)=(1,2)$. It is formed by repeating the parity check symbols of the two codes for the single user channel. The first four rows, correspond to the information symbols. The next two rows are the parity check symbols for user 1, while the last two rows are parity check symbols for user 2.

Fig. 2. Streaming codes for single user and two user channels. Our lower bound is constructed by repeating the parity checks of the two individual codes as illustrated above.

construction are

$$\mathsf{p}_2[i]=\begin{pmatrix}\mathsf{s}_0[i-4]\oplus\mathsf{s}_2[i-2]\\ \mathsf{s}_1[i-4]\oplus\mathsf{s}_3[i-2]\end{pmatrix}.$$

The resulting code is then constructed by concatenating the information symbols with both the parity check symbols $\mathsf{p}_1[i]$ and $\mathsf{p}_2[i]$ as shown in Fig. 2(c). Note that a decoder on channel 1 can recover from a burst of length $B=1$ with a delay of $T=2$ by simply ignoring the parity checks $\mathsf{p}_2[i]$ and using the single user decoder on super-symbols $\mathsf{t}_j[i]$. Likewise the decoder on channel 2 can also recover all symbols from a burst of length $B=2$ with a delay of $T=4$ symbols by ignoring the parity checks $\mathsf{p}_1[i]$ and using the single user decoder on the remaining symbols.

The construction for general burst-delay parameters is a rather straightforward extension of the previous example.

C. Multicast code construction for general parameters

We first consider the code construction when $(B_1,T_1)=(B,T)$ and $(B_2,T_2)=(\alpha B,\alpha T)$ for some integer $\alpha>1$. We split the source symbol into T sub-symbols: $\mathsf{s}_0[i],\ldots,\mathsf{s}_{\alpha T-1}[i]$. If required the last super-symbol is padded with zeros so that it has the same size as the remaining symbols. We generate parity-check symbols for the first channel by defining the super-symbol

$$\mathsf{t}_j[i]=(\mathsf{s}_{\alpha j}[i],\mathsf{s}_{\alpha j+1}[i],\ldots,\mathsf{s}_{\alpha(j+1)-1)}),\quad j=0,1,\ldots,T-1$$

and generating the parity checks for the single-user (B,T) code. Since each super-symbol constitutes of α symbols and the (B,T) code produces a total of B parity-checks, the total number of parity symbols due to channel 1 is αB. For channel 2, we generate additional αB parity checks from the single-user $(\alpha B,\alpha T)$ code. The overall code is formed by concatenating all the parity check symbols. The resulting rate of this code is

$$R=\frac{\alpha T}{2\alpha B+\alpha T}=\frac{T}{2B+T}.$$

The decoder on channel 1 applies single user decoder for channel 1 on super symbols $\mathsf{t}_j[i]$ and ignore the parity checks included for channel 2. Likewise the decoder on channel 2 applies single user decoder for channel 2 and ignores the parity checks included for channel 1.

For general values of (B_1,T_1) and (B_2,T_2), we let T be the lowest common multiple of T_1 and T_2, and divide $\mathsf{s}[i]$ into

T sub-symbols of equal size. Again the last symbol is padded with zeros if required. Provided the alphabet is sufficiently large, there is negligible loss in this step. We apply the code for (B_1, T_1) on super-symbols $\mathsf{t}_{1j}[i]$ obtained by clustering T/T_1 information symbols, and apply the code for (B_2, T_2) on super-symbols $\mathsf{t}_{2j}[i]$ obtained by clustering T/T_2 information symbols. The resulting parity check symbols are concatenated as before and the decoders operate on the respective super-symbols. The rate of the resulting code is

$$R = \frac{T}{T + T\frac{B_1}{T_1} + T\frac{B_2}{T_2}} - o_S(1) = \frac{1}{1 + \sum_{i=1}^{2} \frac{B_i}{T_i}} - o_S(1), \tag{10}$$

where the term $o_S(1)$ accounts for the rate loss due to potential zero padding of the last symbol.

IV. Upper Bound via periodic erasure channel

In this section we establish an upper bound on the streaming capacity by introducing a periodic erasure channel. This approach builds upon the converse in [2] for the single user case. While our proposed approach can be extended to general burst-delay parameters, in the following, we only consider "proportional" burst-delay parameters as defined below.

Theorem 2: Suppose that the two receivers in section II-B have parameters $(B_1, T_1) = (B, T)$ and $(B_2, T_2) = (\alpha B, \alpha T)$ for some integer $\alpha \geq 2$ and $T \geq B$. Then the streaming capacity is upper bounded by $C \leq C^+$, where

$$C^+ = 1 - \frac{\alpha B}{\alpha T + (\alpha - 1)B}. \tag{11}$$

Remark 2: Note that an immediate consequence of the above upper bound (11) is that the multicast streaming capacity is in general strictly smaller than the single user streaming capacity. In particular, for each $\alpha \geq 2$, the upper bound above is strictly below the $\frac{T}{T+B}$, the single user streaming capacity.

Numerical Example: We illustrate the basic idea behind our proof via a simple example. Consider the case $(B_1, T_1) = (1, 1)$ and $(B_2, T_2) = (2, 2)$. Consider a channel that erases every two out of three symbols i.e., $y[k] = \star$, if $k \ mod \ 3 \neq 2$ and $y[k] = x[k]$ otherwise. This corresponds to $B = 1$ in Fig. 3. As we now argue, for this channel, using the two decoders and the multicast encoder we can recover all the source symbols. Clearly for any sequence of codes that recovers all erasures on this periodic erasure-channel, the rate is upper bounded by 1/3, which is consistent with Theorem 2. To show that all the symbols $\{\mathsf{x}[t]\}$ are recovered, consider the symbols at time $t = 0, 1, 2$. Among these $\mathsf{x}[0]$ and $\mathsf{x}[1]$ are erased and only $\mathsf{x}[2]$ is observed. We use the decoder for receiver 2 to recover $\mathsf{s}[0]$ (and hence $\mathsf{x}[0] = f_0(\mathsf{s}[0])$) with a delay of two symbols i.e., from $\mathsf{x}[2]$. Now it only remains to recover $\mathsf{s}[1]$. Since we have already recovered $\mathsf{x}[0]$, we can use the decoder of receiver 1, to recover $\mathsf{s}[1]$ by time 2. Thus by time 2 both the erased symbols are recovered. Since the channel is a periodic erasure channel, this argument can be continued to recover all the symbols.

General Case: The proof of the general case follows via a straightforward extension of this reasoning. We begin by showing the following Lemma:

Lemma 1: Suppose there exists a sequence of encoding functions $\{f_t(\cdot)\}$ and decoding functions $\{\gamma_{1t}(\cdot)\}$ and $\{\gamma_{2t}(\cdot)\}$ that satisfies (8). Then for the encoding function $\{f_t(\cdot)\}$ there exists a sequence of decoding functions $\gamma_t(\cdot)$ that can reproduce the source symbols $\mathsf{s}[t]$, over a channel with periodic bursts as stated below

$$\mathsf{y}[t] = \begin{cases} \star, & t \in [T_k, T_k + \alpha B - 1] \\ \mathsf{x}[t], & t \in [T_k + \alpha B, T_{k+1} - 1] \end{cases} \tag{12}$$

where $T_k = k\alpha T + k(\alpha - 1)B$, $k = 0, 1, \ldots$

An illustration of the periodic-burst channel when $\alpha = 2$ is shown in Fig. 3. Note that the result of Lemma 1 directly implies that the capacity is upper bounded by

$$C \leq 1 - \frac{\alpha B}{\alpha T + (\alpha - 1)B}.$$

To establish Lemma 1, it suffices to show that by time $T_k - 1$, the receiver is able to recover symbols $\mathsf{x}[0], \ldots, \mathsf{x}[T_k - 1]$. We first show that by time $T_1 - 1$, the receiver is able to recover symbols $\mathsf{x}[0], \ldots, \mathsf{x}[T_1 - 1]$. Since only symbols $\mathsf{x}[0], \ldots, \mathsf{x}[\alpha B - 1]$ are erased by time $T_1 - 1$ we focus on these symbols.

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

V. Capacity at minimum delay point

The minimum delay point occurs when $T_i = B_i$. In this case both the receivers require the least possible delay that can be supported by the respective channels. The following characterizes the capacity in this case.

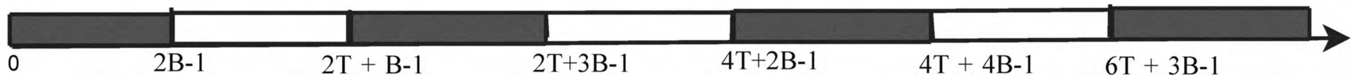

Fig. 3. A periodic burst erasure channel for $\alpha = 2$. The channel all symbols transmitted in the shaded regions. An encoder that can simultaneously satisfy two multicast users with parameters (B, T) and $(2B, 2T)$ can also be used to recover packets on the periodic burst channel.

Theorem 3: The streaming multicast capacity at the minimum delay point i.e., when $T_1 = B_1$ and $T_2 = B_2$ equals 1/3.

Note that the capacity can be achieved via simple repetition,

$$\mathsf{x}[i] = (\mathsf{s}[i], \mathsf{s}[i - T_1], \mathsf{s}[i - T_2])$$

The upper bound (11) in the previous section reduces to $C^+ = \frac{\alpha-1}{2\alpha-1}$ and is not tight for $\alpha > 2$ and hence a different approach is needed. In the remainder of the section we show the following,

Lemma 2: For any sequence of encoding functions and decoding functions that support $T_1 = B_1$ and $T_2 = B_2$, we have that $H(\mathsf{x}_k) \geq 3H(\mathsf{s})$ for each $k \geq \max(T_1, T_2)$.

Note that the upper bound of $1/3$ immediately follows from the above lemma. For convenience we let $T_1 = a$ and $T_2 = b$ and assume that $b > a$. In what follows we use the notation s_b^a to denote the subsequence $(\mathsf{s}_a, \mathsf{s}_{a+1}, \ldots, \mathsf{s}_b)$. Note that for any sequence of encoding and decoding functions we have that for all $k \geq b$ the following conditions hold:

$$H(\mathsf{s}_k | \mathsf{s}_0^{k-1}, \mathsf{x}_k) = 0 \tag{13}$$

$$H(\mathsf{s}_{k-a} | \mathsf{s}_0^{k-a-1}, \mathsf{x}_k) = 0 \tag{14}$$

$$H(\mathsf{s}_{k-b} | \mathsf{s}_0^{k-b-1}, \mathsf{x}_k) = 0 \tag{15}$$

where (13) follows since if channel 1 erases symbols x_{k+1}^{k+a} then s_k must be decoded from x_k and the previously decoded symbols, (14) from the fact that if channel 1 erases x_{k-a}^{k-1}, then s_{k-a} must be recovered using x_k and the previously decoded symbols and (15) from the fact that the symbol s_{k-b} must be recovered by time k using x_k and symbols upto time $k-b-1$ when channel 2 erases x_{k-b}^{k-1}.

Conditions (13)-(15) can be combined to establish the following, whose proof cannot be included due to space constraints.

Lemma 3: For each $k \geq b$ we have that

$$\begin{aligned} H(\mathsf{x}_k) \geq\; & H(\mathsf{x}_k, \mathsf{s}_{k-b}^k | \mathsf{s}_0^{k-b-1}) \\ & - H(\mathsf{s}_{k-b+1}^{k-a-1} | \mathsf{s}_0^{k-b}, \mathsf{x}_k) - H(\mathsf{s}_{k-a+1}^{k-1} | \mathsf{s}_0^{k-a}, \mathsf{x}_k) \end{aligned} \tag{16}$$

To complete the proof it suffices to show that the right hand side of (16) is at-least equal to $3H(\mathsf{s})$. The first term in this expression can be written as follows,

$$\begin{aligned} & H(\mathsf{x}_k, \mathsf{s}_{k-b}^k | \mathsf{s}_0^{k-b-1}) = H(\mathsf{x}_k, \mathsf{s}_{k-b}, \mathsf{s}_{k-b+1}^{k-a-1}, \mathsf{s}_{k-a}, \mathsf{s}_{k-a+1}^{k-1}, \mathsf{s}_k | \mathsf{s}_0^{k-b-1}) \\ & = H(\mathsf{x}_k, \mathsf{s}_{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k | \mathsf{s}_0^{k-b-1}) \\ & \quad + H(\mathsf{s}_{k-b+1}^{k-a-1}, \mathsf{s}_{k-a+1}^{k-1} | \mathsf{x}_k, \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) \\ & = H(\mathsf{x}_k, \mathsf{s}_{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k | \mathsf{s}_0^{k-b-1}) + H(\mathsf{s}_{k-b+1}^{k-a-1} | \mathsf{x}_k, \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) \\ & \quad + H(\mathsf{s}_{k-a+1}^{k-1} | \mathsf{x}_k, \mathsf{s}_0^{k-a}, \mathsf{s}_k) \\ & = H(\mathsf{x}_k | \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) + H(\mathsf{s}_{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k | \mathsf{s}_0^{k-b+1}) \\ & \quad + H(\mathsf{s}_{k-b+1}^{k-a-1} | \mathsf{x}_k, \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) + H(\mathsf{s}_{k-a+1}^{k-1} | \mathsf{x}_k, \mathsf{s}_0^{k-a}, \mathsf{s}_k) \\ & = H(\mathsf{x}_k | \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) + 3H(\mathsf{s}) \\ & \quad + H(\mathsf{s}_{k-b+1}^{k-a-1} | \mathsf{x}_k, \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) + H(\mathsf{s}_{k-a+1}^{k-1} | \mathsf{x}_k, \mathsf{s}_0^{k-a}, \mathsf{s}_k) \end{aligned} \tag{17}$$

where we use the fact that the symbols s_i are i.i.d. in the last step. Substituting the last expression in (16) we have that

$$\begin{aligned} & H(\mathsf{x}_k) \geq 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^{k-b}, \mathsf{s}_{k-a}, \mathsf{s}_k) \\ & \quad - I(\mathsf{s}_{k-b+1}^{k-a-1}; \mathsf{s}_k, \mathsf{s}_{k-a} | \mathsf{x}_k, \mathsf{s}_0^{k-b}) - I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{s}_k | \mathsf{x}_k, \mathsf{s}_0^{k-a}) \\ & = 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^{k-a}, \mathsf{s}_k) + I(\mathsf{x}_k; \mathsf{s}_{k-b+1}^{k-a-1} | \mathsf{s}_0^{k-b}, \mathsf{s}_k, \mathsf{s}_{k-a}) \\ & \quad - I(\mathsf{s}_{k-b+1}^{k-a-1}; \mathsf{s}_k, \mathsf{s}_{k-a} | \mathsf{x}_k, \mathsf{s}_0^{k-b}) - I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{s}_k | \mathsf{x}_k, \mathsf{s}_0^{k-a}) \\ & = 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^{k-a}, \mathsf{s}_k) - I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{s}_k | \mathsf{x}_k, \mathsf{s}_0^{k-a}) \\ & \quad + I(\mathsf{x}_k; \mathsf{s}_{k-b+1}^{k-a-1} | \mathsf{s}_0^{k-b}, \mathsf{s}_k, \mathsf{s}_{k-a}) - I(\mathsf{s}_{k-b+1}^{k-a-1}; \mathsf{s}_k, \mathsf{s}_{k-a} | \mathsf{x}_k, \mathsf{s}_0^{k-b}) \\ & = 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^{k-a}, \mathsf{s}_k) - I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{s}_k | \mathsf{x}_k, \mathsf{s}_0^{k-a}) \\ & \quad + I(\mathsf{x}_k, \mathsf{s}_0^{k-b}; \mathsf{s}_{k-b+1}^{k-a-1}) \end{aligned} \tag{18}$$

$$\begin{aligned} & \geq 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^{k-a}, \mathsf{s}_k) - I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{s}_k | \mathsf{x}_k, \mathsf{s}_0^{k-a}) \\ & = 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^k) \\ & \quad + I(\mathsf{x}_k; \mathsf{s}_{k-a+1}^{k-1} | \mathsf{s}_0^{k-a}, \mathsf{s}_k) - I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{s}_k | \mathsf{x}_k, \mathsf{s}_0^{k-a}) \\ & = 3H(\mathsf{s}) + H(\mathsf{x}_k | \mathsf{s}_0^k) + I(\mathsf{s}_{k-a+1}^{k-1}; \mathsf{x}_k, \mathsf{s}_0^{k-a}) \end{aligned} \tag{19}$$

Here (18) follows from the fact that s_{k-a+1}^{k-1} is independent of $(\mathsf{s}_0^{k-b}, \mathsf{s}_k, \mathsf{s}_{k-a})$ and similarly (19) follows from the fact that s_0^{k-a} is independent of s_k. Finally since the last two terms in (19) are non-negative the desired inequality $H(\mathsf{x}_k) \geq 3H(\mathsf{s})$ follows.

Acknowledgments

Emin Martinian provided very insightful feedback on the results in this paper.

References

[1] Emin Martinian and Carl-Erik W. Sundberg, "Burst Erasure Correction Codes With Low Decoding Delay," IEEE Transactions on Information Theory, October 2004

[2] Emin Martinian, "Dynamic Information and Constraints in Source and Channel Coding," Ph.D. Thesis, Massachusetts Inst. of Technology, September 2004

[3] Emin Martinian and Mitchell Trott, "Delay-optimal Burst Erasure Code Construction," International Symposium on Information Theory, (Nice, France) July 2007

[4] A. Sahai, "Anytime Information Theory", PhD Thesis, Massachusetts Inst. of Technology, 2001

[5] J. Sundararajan, D. Shah, M. Medard, "Feedback-based online network coding", submitted to IEEE Transactions on Information Theory, April 2009

[6] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in Proc. of NetCod, 2008

[7] E. Martinian, Private Communication, 2009