Streaming Erasure Codes under Mismatched Source-Channel Frame Rates

Pratik Patil, Ahmed Badr and Ashish Khisti
School of Electrical and Computer Engineering
University of Toronto
Toronto, ON, M5S 3G4, Canada
Email: {ppatil, abadr, akhisti}@comm.utoronto.ca

***Abstract*—Streaming erasure codes sequentially encode a source stream into channel packets over a burst erasure channel and guarantee that each source frame is recovered within a fixed decoding delay. We study streaming codes when M channel packets need to be transmitted between successive source frames. This extends earlier works which exclusively focus on the case when $M = 1$. We obtain a general upper bound on the associated streaming capacity and show that it can be achieved for sufficiently large decoding delays using a *layered* code. For the minimum possible decoding delay we also establish the streaming capacity and show that it can be obtained using a repetition code.**

I. Introduction

In media streaming applications, a source stream is encoded into channel packets by a transmitter which are then are sent over a noisy channel and the receiver expects each source packet to be decoded within a fixed decoding delay. This delay constraint presents many interesting coding problems. Traditional block codes, e.g. maximum distance separable codes, are not ideal in streaming applications as an entire codeword must be received to decode the corresponding source symbols and the receiver cannot afford to wait for the end of the entire codeword. Codes such as the digital fountain codes are also not suitable as they require the entire source data before the output stream is reproduced. Also sequential reconstruction of the source stream is not guaranteed.

In [1] (Chapter 8), [2], [3] a new class of codes called streaming erasure codes (*SCo*) is developed. The transmitter observes a source stream with one source frame arriving per each channel packet sent. The source stream is encoded in a causal manner with rate R. The channel is modeled as a burst erasure channel that introduces an erasure burst of maximum B channel packets starting at an arbitrary time. The decoder is required to output each source packet within a delay of T packets. A fundamental relationship between R, B, and T is established and *SCo* codes are constructed that achieve this tradeoff. The above setup has been extended in various directions in [4], [5], [6].

In this paper we generalize the above setup when a total of M channel packets need to be transmitted between the arrival of successive source frames. One practical motivation for our model is that in many network architectures the size of a channel packet is restricted and upper layer packets are fragmented into smaller packets before the transmission. The

Fig. 1. System under consideration

M packets between two successive source frames will be considered as a macro-packet. Note that $M = 1$ corresponds to the previously studied setup. We assume a burst-erasure channel where a burst of length B can occur across any window of B consecutive channel packets, which may span multiple macro-packets and can start anywhere within the macro-packet. For such a scenario we study the associated steaming capacity. First we obtain a general upper bound on the achievable rate in terms of M, B and T. Then we construct codes with rates that achieve the upper bound for a certain range of parameter values. Our main code construction involves a layered approach by dividing each source frame into urgent and non-urgent source symbols and applying different levels of error correction to each.

II. System Model

We consider discrete time slots $i \in \mathbb{Z}_{\geq 0}$. All symbols considered in this paper are elements from a common base field $\mathbb{F}_q$. Figure 1 depicts the system under consideration.

Encoder: The encoder receives a stream of i.i.d. source packets $\mathbf{s}[i]$ at the beginning of time slot i. Each $\mathbf{s}[i]$ is causally encoded into M channel packets $\mathbf{x}[i,j] \in \mathbb{F}_q^n$, $j = [1, M]$, i.e.,

$$\mathbf{x}[i,j] = f_i^j(\mathbf{s}[0], \mathbf{s}[1], \cdots, \mathbf{s}[i]) \tag{1}$$

where $f_i^j(\cdot)$ is a deterministic encoding function. We refer to the set of these M channel packets as one macro-packet $\mathbf{X}[i,:] = \big(\mathbf{x}[i,1], \mathbf{x}[i,2], \cdots, \mathbf{x}[i,M]\big) \in \mathbb{F}_q^{n\times M}$.

Channel: The channel packets are sent through a burst erasure channel and the received packets are denoted by $\mathbf{y}[i,j]$. The actual (channel) time slot associated with the packet $\mathbf{x}[i,j]$ is $t = iM + j - 1$. The channel is such that it may introduce an erasure burst of maximum B channel packets starting at an arbitrary time slot t_s and ending at $t_f \leq t_s + B - 1$. $\mathbf{y}[i,j] = \mathbf{x}[i,j]$, when the packet is not erased and $\mathbf{y}[i,j] = \star$, when the packet is erased, i.e.,

$$\mathbf{y}[i,j] = \begin{cases} \star & \text{for } t_s \leq iM + j - 1 \leq t_f \\ \mathbf{x}[i,j] & \text{otherwise} \end{cases} \tag{2}$$

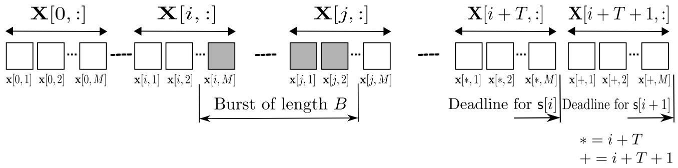

Fig. 2. General burst pattern

The erasure burst can start at any arbitrary channel packet $\mathbf{x}[i,j]$ for $j \in [1, M]$ and can span multiple macro-packets as shown in Figure 2. We denote the set of channel packets corresponding to the macro-packet $\mathbf{X}[i,:]$ by $\mathbf{Y}[i,:] = \big(\mathbf{y}[i,1], \mathbf{y}[i,2], \cdots, \mathbf{y}[i,M]\big)$.

Decoder: The decoder attempts to decode the original source stream using the received channel packets within allowed delay of T macro-packets which means that the receiver must be able to decode the source packet $\mathbf{s}[i]$ at the end of the macro-packet $i+T$. The decoder implements a decoding function $g_i(\cdot)$ such that,

$$\mathbf{s}[i] = g_i(\mathbf{Y}[i+T,:], \mathbf{Y}[i+T-1,:], \cdots, \mathbf{Y}[0,:]). \tag{3}$$

Let $\mathsf{H}(\mathbf{s}[i]) = \mathsf{H}(\mathbf{s})$. Rate of the streaming code is defined as the ratio of the entropy of the source packet to the size of the macro-packet, $R = \frac{\mathsf{H}(\mathbf{s})}{n \cdot M}$. An optimal streaming erasure code is the one that achieves the maximum rate (streaming capacity) for a given choice of (M, B, T).

III. Upper Bound

Theorem 1. *For the given streaming setup with M, T and B of the form $bM + B'$ where $b \in \mathbb{Z}_{\geq 0}$, $B' = [1, M-1]$, the streaming capacity C is upper bounded by*

$$C \leq R^{+} = 1 - \frac{B}{M(T+b+1)}. \tag{4}$$

Proof: We use the technique of periodic erasure channel ([2]). Consider periodic bursts each of length B with a guard interval of $M(b+T+1) - B$ as shown in Figure 3. We let the first burst start at $\mathbf{x}[0,1]$. By definition we require all lost packets due to this burst in the first period to be recovered by the macro-packet $T+b+1$. Once these erased packets are recovered, we can treat these erasures as never happened and simply repeat the technique for the next period and so on. Therefore we must have that the information used to recover source packets in one period must equal the information contained within the unerased channel packets in that period. Therefore for any streaming code with parameters (M, B, T), the following relationship must hold:

$$T_{period} \cdot \mathsf{H}(\mathbf{s}) \leq (MT_{period} - B) \cdot n \cdot M \tag{5}$$

From the definition of rate R, we have that for any achievable rate (and thus for streaming capacity C)

$$C = R^{+} \leq \frac{M(T+b+1) - (bM+B')}{M(T+b+1)} \tag{6}$$

which completes the proof. ∎

Remark 1. In Theorem 1, burst lengths $B = bM$ are not considered. The bound derived above also applies to such bursts, but we can easily tighten it to get

$$C \leq R^{+} = \frac{T}{T+b}. \tag{7}$$

Further we can show that this upper bound is achievable using (b, T) *SCo* codes with a simple interleaving of factor M.

Remark 2. Note that we can let the first burst start at other positions in the first period. However such burst positions result in loose upper bounds. The burst position considered in the proof results in the strictest bound.

Remark 3. Note that the minimum delay (T_{min}) for any positive rate is b.

IV. Code Constructions

We treat the cases $b = 0$ and $b \geq 1$ separately as the former special case is simpler.

A. $b = 0$

Below we construct a code with rate $\frac{M(T+1)-B}{M(T+1)}$ (obtained by substituting $b = 0$ in equation 4). As pointed in Section II, all symbols are from $\mathbb{F}_q$.

Encoding: Let $k = M(T+1) - B$ and $n = T+1$. We split each source packet $\mathbf{s}[i]$ into k symbols, $\mathbf{s}[i] = (s_1[i], s_2[i], \cdots, s_k[i])^{\dagger}$[1]. We then apply a (nM, k, T) convolutional code to generate the macro-packet $\mathbf{X}[i,:]$ as follows,

$$\mathbf{x}^{\text{vec}}[i] = (\mathbf{x}^{\dagger}[i,1], \ldots, \mathbf{x}^{\dagger}[i,M])^{\dagger} = \left(\sum_{j=0}^{T} \mathbf{s}^{\dagger}[i-j] \cdot \mathbf{G}_j\right)^{\dagger} \tag{8}$$

where $\mathbf{x}^{\text{vec}}[i]$ denotes the vectorization of $\mathbf{X}[i,:]$ (stacking the columns of $\mathbf{X}[i,:]$ on top of one another) and $\mathbf{G}_0, \cdots, \mathbf{G}_T \in \mathbb{F}_q^{k \times nM}$. Thus each macro-packet contains $nM = M(T+1)$ symbols. The rate of this code is $R = \frac{k}{nM} = \frac{M(T+1)-B}{M(T+1)}$ as required. The first $T+1$ macro-packets can be expressed as

$$\left[\mathbf{x}^{\text{vec},\dagger}[0], \cdots, \mathbf{x}^{\text{vec},\dagger}[T]\right] = \left[\mathbf{s}^{\dagger}[0], \cdots, \mathbf{s}^{\dagger}[T]\right] \cdot \mathbf{G}_T^{s} \tag{9}$$

where

$$\mathbf{G}_T^{s} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_T \\ 0 & \mathbf{G}_0 & \cdots & \mathbf{G}_{T-1} \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & \mathbf{G}_0 \end{bmatrix}. \tag{10}$$

We consider matrices $\mathbf{G}_i$ such that they form a strongly-MDS code [7]. Particularly we consider systematic codes in which the matrices $\mathbf{G}_i$ take the form

$$\mathbf{G}_0 = \left[\mathbf{I}_{k \times k} \quad \mathbf{H}_0\right], \ \mathbf{G}_i = \left[\mathbf{0}_{k \times k} \quad \mathbf{H}_i\right], \ i = [1, T], \tag{11}$$

where $\mathbf{H}_i \in \mathbb{F}_q^{k \times B}$ generate B parity check symbols, $\mathbf{q}[i] = (q_1[i], q_2[i], \cdots, q_B[i])^{\dagger} \in \mathbb{F}_q^{B}$, i.e.,

$$\mathbf{q}[i] = \left(\sum_{j=0}^{T} \mathbf{s}^{\dagger}[i-j] \cdot \mathbf{H}_j\right)^{\dagger}. \tag{12}$$

[1] $\dagger$ denotes the transpose of a vector.

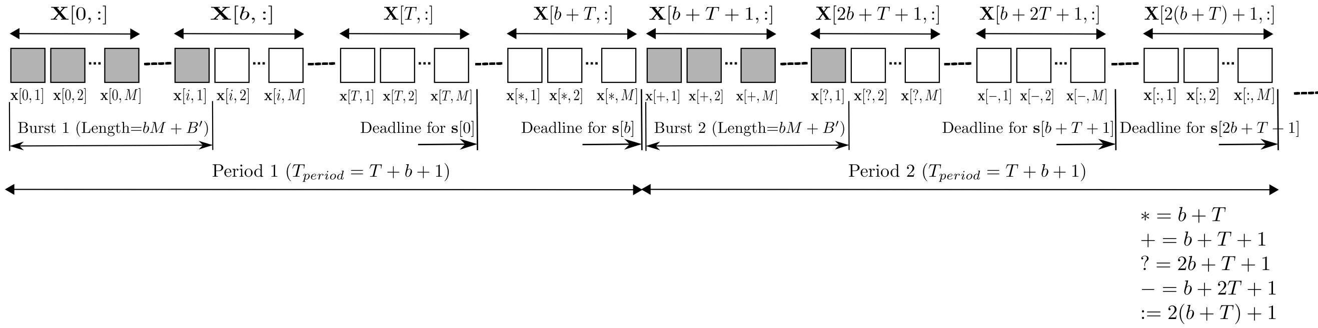

Fig. 3. Periodic erasure patterns

Decoding: We show that the above code indeed corrects any erasure burst of length B within the deadline. Since the code is time invariant, it suffices to consider only the burst patterns that start within the macro-packet $\mathbf{X}[0,:]$. There are maximum M such burst patterns. Note that since $b=0$, $1 \leq B \leq M-1$. Thus the burst can erase symbols from at most two adjacent macro-packets. The key idea in the decoding is that in all burst positions we have enough number of unerased parity symbols (equations) to recover the lost symbols (unknowns) within the decoding delay T.

Consider any burst pattern that starts within the macro-packet $\mathbf{X}[0,:]$. Using [8, Corollary 3.2], if we have enough parity symbols within the decoding window, we can completely recover from the erasure burst. We consider the decoding window from the macro-packet $i=0$ to the macro-packet $i=T$ with $nM(T+1)$ total symbols. We have the following two cases.

1) Burst is such that none of the parity symbols in the decoding window are erased. In this case, number of symbols erased is $B(T+1)$ (each channel packet contains $(T+1)$ symbols and B such channel packets are erased). Number of parity symbols within decoding window is $B(T+1)$ (each macro-packet contains B parities and none of them are erased). Hence we can completely recover within the deadline.
2) Burst is such that it erases r parities. In this case, number of symbols erased is $B(T+1)-r$. Number of unerased parities within the decoding window is $B(T+1)-r$. Thus all such bursts are recoverable as well.

Remark 4. Note that the above construction *simultaneously* recovers all the erased packets. If the burst spans the macro-packets $i=0$ and $i=1$ then both $\mathbf{s}[0]$ and $\mathbf{s}[1]$ are simultaneously recovered at the end of the macro-packet T even though the delay of $\mathbf{s}[1]$ is later. As we will show next for $b \geq 1$, *sequential* recovery in general is necessary to achieve the upper bound in equation 4.

B. $b \geq 1$

Our target rate in this case is $R=\frac{M(T+b+1)-B}{M(T+b+1)}$ (upper bound).

Encoding: Let $k=M(T+b+1)-B$ and $n=(T+b+1)$. Each source packet is split into $k=M(T+b+1)-B$ symbols. We refer to first B symbols of each source packet as urgent symbols, $\mathbf{u}[i]=(s_1[i],s_2[i],\cdots,s_B[i])^\dagger$ and rest as non-urgent symbols (represented by $\mathbf{v}^{\text{vec}}[i]$). All urgent symbols of the source packet $\mathbf{s}[i]$ are placed in the first channel packet $\mathbf{x}[i,1]$ of the macro-packet $\mathbf{X}[i,:]$[2]. Overall we construct a systematic (nM,k,T) convolutional code generating $(n-k)=B$ parity check symbols, $\mathbf{p}[i]=(p_1[i],p_2[i],\cdots,p_B[i])^\dagger$. The code construction involves two layers for the parity symbols. First layer is a repetition code of urgent symbols $\mathbf{u}[i]$ with delay T. Second layer is a $(k-B,B,T)$ strongly-MDS code applied to the non-urgent symbols $\mathbf{v}^{\text{vec}}[i]$ forming B parity symbols $\mathbf{q}[i]$ similar to the case of $b=0$ (subsection IV-A), i.e.,

$$\mathbf{q}[i]=\left(\sum_{j=0}^{T}\mathbf{v}^{\text{vec},\dagger}[i-j]\cdot\mathbf{H}_j\right)^\dagger \tag{13}$$

where $\mathbf{H}_0,\cdots,\mathbf{H}_T \in \mathbb{F}_q^{k-B\times B}$ and the corresponding generator matrices $\mathbf{G}_i$ form a strongly-MDS code. Thus the parity symbols $\mathbf{p}[i]$ for the streaming code consist of two parts, $\mathbf{p}[i]=\mathbf{q}[i]+\mathbf{u}[i-T]$ (addition is over $\mathbb{F}_q$). We put all the parity check symbols into the last channel packet $\mathbf{x}[i,M]$ of each macro-packet[3] and divide the non-urgent symbols $\mathbf{v}^{\text{vec}}[i]$ to form $\mathbf{v}[i,1]$ to $\mathbf{v}[i,M]$. Overall arrangement of symbols in the macro-packet $\mathbf{X}[i,:]$ is as shown below.

$$\mathbf{X}[i]=\begin{bmatrix}\mathbf{u}[i]\\ \mathbf{v}[i,1]\end{bmatrix}\Bigg|\;\mathbf{v}[i,2]\;\Bigg|\;\mathbf{v}[i,3]\;\Bigg|\;\cdots\;\Bigg|\begin{bmatrix}\mathbf{v}[i,M]\\ \mathbf{p}[i]\end{bmatrix}. \tag{14}$$

Decoding: Similar to the case of $b=0$, we only need to worry about M different burst patterns that start at $\mathbf{x}[0,j]$ for $j=[1,M]$. Key idea here is that we recover all the erased urgent symbols at their respective deadlines (that's why the name urgent), e.g. $\mathbf{u}[0]$ at T. All other erased non-urgent symbols are decoded before. Below we explain in detail the decoding for different burst positions starting at $\mathbf{x}[0,j]$.

[2]Here we implicitly assumed that all urgent symbols are accommodated in the first channel packet. Hence we require that $B \leq T+b+1$ i.e., $T \geq b(M-1)+B'-1$ (let's call this T').

[3]For all the parity symbols to be accommodated in the last channel packet, we require that $B \leq T+b+1$, i.e., $T \geq T'$ same as the condition in the footnote above.

1) $j = 1$:
Decoding of urgent symbols - In this burst pattern, urgent symbols $\mathbf{u}[0]$ are erased. We use $\mathbf{p}[T]$ to recover them. We can do this provided $\mathbf{q}[T]$ is known (which is a function of non-urgent symbols $\mathbf{v}^{\text{vec}}[0], \cdots, \mathbf{v}^{\text{vec}}[T]$). Parity symbols $\mathbf{p}[j]$, $j > T$ are used to decode rest of the erased urgent symbols $\mathbf{u}[j-T]$.
Decoding of non-urgent symbols - Next we show that we can recover all the erased non-urgent symbols by macro-packet $T-1$. Using [8, Corollary 3.2] similar to the case with $b = 0$, if we have enough unerased parity symbols, we can recover erased non-urgent symbols (as $\mathbf{q}$ is constructed by applying strongly-MDS code to $\mathbf{v}^{\text{vec}}$). We consider a window of length $(nM - B)T$ (from $\mathbf{x}[0,1]$ to $\mathbf{x}[T-1, M]$). In this window we can recover from an erasure burst of no longer than BT non-urgent symbols. Number of non-urgent symbol erased is equal to $B(T+b+1) - (b+1)B = BT$ which is exactly what we can recover from. Thus we can recover all the erased non-urgent symbols by macro-packet $T-1$ and then at each subsequent deadline we can recover corresponding urgent symbols.
2) $2 \le j \le M - B' + 1$:
Decoding of urgent symbols - For these burst patterns, $\mathbf{x}[0,1]$(and thus $\mathbf{u}[0]$) is available at the decoder. So we no longer need $\mathbf{p}[T]$ to recover $\mathbf{u}[0]$. Similar to the case above, we use parity symbols $\mathbf{p}[j]$, $j > T$ to decode rest of the erased urgent symbols $\mathbf{u}[j-T]$.
Decoding of non-urgent symbols - Since we have $\mathbf{u}[0]$, we can use $\mathbf{p}[T]$ to recover the erased non-urgent symbols. Thus we consider a bigger window of length $(nM-B)(T+1)$. Therefore the recoverable burst length become $B(T+1)$. Number of non-urgent symbols erased is $B(T+b+1) - bM = B(T+1)$ (since $\mathbf{u}[0]$ is no longer erased) which is exactly equal to the recoverable burst length.
3) $j > (M - B' + 1)$:
Decoding of urgent symbols - These burst patterns include $\mathbf{x}[b+1,1]$. $\mathbf{x}[0,1]$ is still unerased. So the recovery of urgent symbols is the same as the case above.
Decoding of non-urgent symbols - Similar to the case above, we can use $\mathbf{p}[T]$ for the recovery of non-urgent symbols. So while the recoverable length remains the same $(B(T+1))$, the number of non-urgent erasures is B symbols less than what we have in the case above (as $\mathbf{x}[b+1,1]$ contains B urgent symbols). Hence we easily recover all the erased non-urgent symbols as required.

Remark 5. It is easy to check that we can achieve a rate $R' = \frac{M(T+1)-B}{M(T+1)}$ using random linear codes for this case. This is strictly less than the rate achieved by the construction above. For the case with $b = 0$, both our codes and random linear codes achieve the same rate.

Remark 6. Note that although our model only considers a single erasure burst, as with *SCo*, our constructions correct multiple erasure-bursts separated with a sufficient guard interval.

So far we have constructed codes whose rates meet the upper bound except for when $b \ge 1$ and $T_{min} \le T \le T'$. In the next section, we consider a special case of the minimum decoding delay with $T = T_{min}$.

V. Case with minimum decoding delay

When $T = T_{min} = b$, we obtain a tighter upper bound than that given by equation 4. We also propose a code construction to achieve the new upper bound.

Theorem 2. For any burst $B = bM + B'$ and delay $T = b$, the streaming capacity is given by

$$R = \begin{cases} \frac{M-B'}{M} & B' \ge \frac{M}{2} \\ \frac{1}{2} & B' < \frac{M}{2} \end{cases}. \tag{15}$$

Proof: We divide the proof of Theorem 2 into two parts.

A. Converse

Consider a channel that erases first $B = bM + B'$ channel packets $\mathbf{x}[i,1], \ldots, \mathbf{x}[i+b, B']$. Since the delay constraint for $\mathbf{s}[i]$ is $i + b$, the following equation should be satisfied,

$$\mathsf{H}(\mathbf{s}[i] | \mathbf{x}[i+b, B'+1], \ldots, \mathbf{x}[i+b, M]) = 0. \tag{16}$$

Now consider a channel erasing the channel packets $\mathbf{x}[i+b, M-B'+1], \ldots, \mathbf{x}[i+2b, M]$. The delay of $\mathbf{s}[i+b]$ is $i+2b$. Thus the following equation should be satisfied,

$$\mathsf{H}(\mathbf{s}[i+b] | \mathbf{x}[i+b, 1], \ldots, \mathbf{x}[i+b, M-B']) = 0. \tag{17}$$

Combining (16) and (17) one can write,

$$\begin{aligned} \mathsf{H}(\mathbf{s}[i], \mathbf{s}[i+b] | &\mathbf{x}[i+b,1], \ldots, \mathbf{x}[i+b, M-B'] \\ &, \mathbf{x}[i+b, B'+1], \ldots, \mathbf{x}[i+b, M]) = 0. \end{aligned} \tag{18}$$

From (18), we get $\begin{cases} 2(M-B')\mathsf{H}(\mathbf{x}) \ge 2\mathsf{H}(\mathbf{s}) & B' \ge \frac{M}{2} \\ \mathsf{H}(\mathbf{x}) \ge 2\mathsf{H}(\mathbf{s}) & B' < \frac{M}{2} \end{cases}$

Therefore $R = \frac{\mathsf{H}(\mathbf{s})}{M\mathsf{H}(\mathbf{x})} \le \begin{cases} \frac{M-B'}{M} & B' \ge \frac{M}{2} \\ \frac{1}{2} & B' < \frac{M}{2} \end{cases}$ and the converse follows.

B. Achievability

A simple repetition scheme is considered.

1) $B' < \frac{M}{2}$
We split each source packet into M symbols, $\mathbf{s}[i] = (s_1[i], \ldots, s_M[i])^\dagger$ and assign the channel packets as follows,

$$\mathbf{x}[i,j] = \begin{cases} (s_{2j-1}[i], s_{2j}[i])^\dagger & j < \frac{M+1}{2} \\ (s_M[i], s_0[i-T])^\dagger & j^* = \frac{M+1}{2} \\ (s_{2j-M-1}[i-T], s_{2j-M}[i-T])^\dagger & j > \frac{M+1}{2} \end{cases} \tag{19}$$

where $j = [1, M]$; $*$ - only when M is odd.
2) $B' \ge \frac{M}{2}$
In this case, we split each source packet into $M - B'$

$\mathbf{X}[0,:]$		$\mathbf{X}[1,:]$		$\mathbf{X}[2,:]$		$\mathbf{X}[3,:]$	
$\mathbf{x}[0,1]$	$\mathbf{x}[0,2]$	$\mathbf{x}[1,1]$	$\mathbf{x}[1,2]$	$\mathbf{x}[2,1]$	$\mathbf{x}[2,2]$	$\mathbf{x}[3,1]$	$\mathbf{x}[3,2]$
$u_1[0]$	$v_1[0,2]$	$u_1[1]$	$v_1[1,2]$	$u_1[2]$	$v_1[2,2]$	$u_1[3]$	$v_1[3,2]$
$u_2[0]$	$v_2[0,2]$	$u_2[1]$	$v_2[1,2]$	$u_2[2]$	$v_2[2,2]$	$u_2[3]$	$v_2[3,2]$
$u_3[0]$	$u_1[-3]+v_1[-3,2]+v_1[-1,1]$	$u_3[1]$	$u_1[-2]+v_1[-2,2]+v_1[0,1]$	$u_3[2]$	$u_1[-1]+v_1[-1,2]+v_1[1,1]$	$u_3[3]$	$u_1[0]+v_1[0,2]+v_1[2,1]$
$v_1[0,1]$	$u_2[-3]+v_2[-3,2]+v_2[-1,1]$	$v_1[1,1]$	$u_2[-2]+v_2[-2,2]+v_2[0,1]$	$v_1[2,1]$	$u_2[-1]+v_2[-1,2]+v_2[1,1]$	$v_1[3,1]$	$u_2[0]+v_2[0,2]+v_2[2,1]$
$v_2[0,1]$	$u_3[-3]+v_2[-2,2]+v_1[-1,2]$	$v_2[1,1]$	$u_3[-2]+v_2[-1,2]+v_1[0,2]$	$v_2[2,1]$	$u_3[-1]+v_2[0,2]+v_1[1,2]$	$v_2[3,1]$	$u_3[0]+v_2[1,2]+v_1[2,2]$

TABLE I
CODE CONSTRUCTION FOR $(M{=}2, B{=}3, T{=}3)$

symbols i.e., $\mathbf{s}[i] = (s_1[i], \ldots, s_{M-B'}[i])^\dagger$ and assign the channel packets as follows,

$$\mathbf{x}[i,j] = \begin{cases} s_j[i] & j \in [1, M-B'] \\ 0 & j \in [M-B'+1, B'] \\ s_{j-B'}[i-T] & j \in [B'+1, M] \end{cases} \tag{20}$$

where $j = [1, M]$.

In each case, by inspection we can check that the code described above is decodable within the decoding delay b. ■

Remark 7. It can easily be seen that the capacity in Theorem 2 is strictly less than the upper bound in Theorem 1.

VI. EXAMPLE

In this section we show a code construction for parameters $M = 2$, $B = 3$, $T = 3$. $B = 2+1$, so we have $b = 1$ and $B' = 1$. From Theorem 1, $C \le \frac{M(T+b+1)-B}{M(T+b+1)} = \frac{7}{10}$. We construct a code with rate $\frac{7}{10}$ as follows.

Encoding:

1) Split each source packet $\mathbf{s}[i]$ into $M(T+b+1)-B = 7$ symbols $(s_1[i], \cdots, s_7[i])^\dagger$.
2) Group these into urgent and non-urgent symbols. For each source packet $\mathbf{s}[i]$, we call first $B = 3$ symbols as urgent symbols, $\mathbf{u}[i] = (u_1[i], \cdots, u_3[i])^\dagger = (s_1[i], \cdots, s_3[i])^\dagger$. Rest of the symbols are referred to as non-urgent symbols $\mathbf{v}^{\text{vec}}[i]$. We divide them to form $\mathbf{v}[i,1] = (v_1[i,1], v_2[i,1])^\dagger = (s_4[i], s_5[i])^\dagger$ and $\mathbf{v}[i,2] = (v_1[i,2], v_2[i,2])^\dagger = (s_6[i], s_7[i])^\dagger$.
3) We place B parity symbols $\mathbf{p}[i] = (p_1[i], p_2[i], p_3[i])^\dagger$ into the last channel packet of each macro-packet. We generate these parities using two components, $\mathbf{p}[i]$=$\mathbf{q}[i]$+$\mathbf{u}[i-3]$. $\mathbf{q}[i]$ is formed either by applying strongly-MDS code to ($\mathbf{v}[i,1]$,$\mathbf{v}[i,2]$) or more directly as $(q_1[i], q_2[i], q_3[i])^\dagger = (v_1[0,2] + v_1[2,1], v_2[0,2] + v_2[2,1], v_2[1,2] + v_1[2,2])^\dagger$. Component $\mathbf{u}[i-3]$ is just a repetition code.

Decoding: Since $M = 2$, there are two burst patterns that need to be checked.

1) Burst that erases the first three channel packets
   Non-urgent symbol recovery - We recover $v_1[0,1]$, $v_2[0,1]$, $v_1[0,2]$ from $p_1[1]$, $p_2[1]$ $p_3[1]$ respectively and $v_2[0,2]$, $v_1[1,1]$, $v_2[1,1]$ from $p_3[2]$, $p_1[2]$, $p_2[2]$ respectively.
   Urgent symbol recovery - Since we have recovered all the erased non-urgent symbols, we can easily recover $\mathbf{u}[0]$ from $\mathbf{p}[3]$ and $\mathbf{u}[1]$ from $\mathbf{p}[4]$ at their respective deadlines.
2) Burst that erases the next three channel packets (i.e., $\mathbf{x}[0,2], \mathbf{x}[1,1], \mathbf{x}[1,2]$)
   Non-urgent symbol recovery - Since $\mathbf{u}[0]$ is unerased, we can use $\mathbf{p}[3]$ for non-urgent symbol recovery. We recover $v_1[0,2]$, $v_2[0,2]$, $v_1[1,1]$, $v_2[1,1]$ from $p_1[3]$, $p_2[3]$, $p_1[2]$, $p_2[2]$ respectively and $v_2[1,2]$ from $p_3[3]$. Finally we use previously decoded $v_2[0,2]$ to obtain $v_1[1,2]$ from $p_3[2]$.
   Urgent symbol recovery - $\mathbf{u}[0]$ is not erased and $\mathbf{u}[1]$ is easily recovered from $\mathbf{p}[4]$ since all the erased non-urgent symbols are already decoded.

VII. CONCLUSION

This paper studies delay constrained streaming erasure codes for mismatched source-channel frame rates where M channel packets need to be transmitted between two successive source frames. Using the technique of periodic erasure channel, a general upper bound on the streaming capacity is obtained. Streaming codes are then constructed for different cases of parameter choices. Our proposed codes are capacity achieving for sufficiently large decoding delays ($T \ge T'$) and the special case of the minimum decoding delay ($T = T_{min}$). Finding the streaming capacity for the decoding delays between T_{min} and T' is left for future work.

REFERENCES

[1] E. Martinian, "Dynamic Information and Constraints in Source and Channel Coding," *Ph.D. Thesis*, MIT, September 2004
[2] Emin Martinian and Carl-Erik W. Sundberg, "Burst Erasure Correction Codes With Low Decoding Delay," *IEEE Transactions on Information Theory*, October 2004
[3] Emin Martinian and Mitchell Trott, "Delay-optimal Burst Erasure Code Construction," *International Symposium on Information Theory*, (Nice, France), July 2007
[4] A. Badr, A. Khisti and E. Martinian, "Diversity Embedded Streaming Erasure Codes (DE-SCo): Constructions and Optimality," in *IEEE Global Telecommunications Conf.*, Miami, FL, USA, 2010
[5] A. Badr, D. Lui and A. Khisti, "Multicast Streaming Codes (Mu-SCo) for Burst Erasure Channels," *Allerton Conf.*, Monticello, IL, USA, 2010
[6] A. Badr, A. Khisti, W. Tan and J. Apostoloupolos, "Streaming Codes for Channels with Burst and Isolated Erasures," *INFOCOM*, Turin, Italy, 2013
[7] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, "Strongly MDS convolutional codes," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 584598, 2006
[8] Virtudes Toms, Joachim Rosenthal, Roxana Smarandache, "Decoding of Convolutional Codes over the Erasure Channel," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 90-108, Jan. 2012