High Rate Streaming Codes Over the Three-Node Relay Network

M. Nikhil Krishnan[1], Gustavo Kasper Facenda[1], Elad Domanovitz[1] and Ashish Khisti
Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON M5S 3G4, Canada
E-mail: {nikhilkrishnan.m@gmail.com}

Wai-Tian Tan and John Apostolopoulos
Enterprise Networking Innovation Labs
Cisco Systems
San Jose, CA 95134, USA

2021 IEEE Information Theory Workshop (ITW) | 978-1-6654-0312-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITW48936.2021.9611490

***Abstract*—In this paper, we investigate streaming codes over a three-node relay network. Source node transmits a sequence of message packets to the destination via a relay. Source-to-relay and relay-to-destination links are unreliable and introduce at most N_1 and N_2 packet erasures, respectively. Destination needs to recover each message packet with a strict decoding delay constraint of T time slots. We propose streaming codes under this setting for all feasible parameters $\{N_1, N_2, T\}$. Relay naturally observes erasure patterns occurring in the source-to-relay link. In our code construction, we employ a channel-state-dependent relaying strategy, which rely on these observations. In a recent work, Fong et al. provide streaming codes featuring channel-state-independent relaying strategies, for all feasible parameters $\{N_1, N_2, T\}$. Our schemes offer a strict rate improvement over the schemes proposed by Fong et al., whenever $N_1 < N_2$.**

I. Introduction

Reliable communication with low-latency is critical in many applications, such as audio/video streaming, virtual gaming and tele-medicine. Data packets, which are generated in a sequential fashion, need to be communicated to the receivers over an unreliable packet erasure channel, with strict decoding deadlines. Due to high round-trip delays, methods involving retransmission, such as automatic repeat request (ARQ) are not suitable. For that reason, the literature has considered forward error correction (FEC) schemes a more appropriate solution. In particular, recent literature has studied streaming codes, which are packet-level FECs designed to ensure reliability against packet erasures under a tight decoding delay constraint.

The first paper to consider such codes was [1], where the authors study a point-to-point system consisting of a source and destination. In this work, the authors consider streaming codes which tolerate a packet erasure burst of length at most B and derive an upper bound on the achievable rate of streaming codes. A family of optimal codes, namely, maximally-short codes have been proposed for a wide range of parameters $\{B, T\}$. In a subsequent work [2], the authors prove tightness of the rate upper bound in [1] by providing optimal streaming codes for all $\{B, T\}$. Badr et al. considers a more general, sliding-window-based, packet erasure model where, in any sliding window of W consecutive time slots, there can be either (i) at most N erasures at arbitrary positions or else (ii) an erasure burst which affects at most B consecutive time slots. The paper provides a rate upper bound for streaming codes over the sliding window model and also proposes near-optimal streaming codes. The works [3]–[6] provide rate-optimal streaming codes under the sliding window erasure model. There are several works such as [7]–[10] which study various other models for low-delay communication systems. In contrast to the existing literature which focusses on point-to-point networks, our focus in this paper is on a topology introduced in a recent paper [11]. Fong et al. [11] propose a generalization of the point-to-point networks to a three-node architecture for streaming codes, which consists of a source, a relay and a destination. This kind of a topology is often present in content delivery networks [11], [12]. An extension of the three-node relay network to a multi-hop network may be found in [13].

In [13], the authors introduce the concept of a channel-state-dependent streaming code, in which the relays adapt to the erasure pattern observed. In this paper, we show that this idea can be used to improve the rate previously presented in [11] for the three-node network. Note that, although the general idea of state dependent codes was presented in [13], using it to actually improve the rate still requires novel ideas, which are highlighted in Section III.

II. Setting

In this section, we formally introduce the problem setting. We use the following notation throughout the paper. The set of non-negative integers is denoted by $\mathbb{Z}_+$. The finite field with q elements is denoted by $\mathbb{F}_q$. The set of l-dimensional column vectors over $\mathbb{F}_q$ is denoted by $\mathbb{F}_q^l$. For $a, b \in \mathbb{Z}_+$, we use $[a : b]$ to denote $\{i \in \mathbb{Z}_+ \mid a \leq i \leq b\}$. Naturally, we set $[a : \infty] \triangleq \{i \in \mathbb{Z}_+ \mid i \geq a\}$.

Consider a three node setup consisting of a source, relay and destination. All packet communication happening in source-to-relay and relay-to-destination links are assumed to be instantaneous, i.e., with no propagation delays. In each discrete time slot $t \in [0 : \infty]$, soure has a *message packet* $\underline{m}(t) \in \mathbb{F}_q^k$ available, which needs to be communicated to the destination via relay. Towards this, at time-t, source invokes a source-

[1] Equal contribution.

978-1-6654-0312-2/21/$31.00 ©2021 IEEE

side encoder $\mathcal{E}_{\rm S}(t) : \underbrace{\mathbb{F}_q^k \times \cdots \mathbb{F}_q^k}_{t+1 \text{ times}} \to \mathbb{F}_q^{n_1}$ to produce a *source packet* $\underline{x}(t) \in \mathbb{F}_q^{n_1}$, which is obtained as a function of message packets $\{\underline{m}(t')\}_{t' \in [0:t]}$. Source transmits $\underline{x}(t)$ to the relay over a packet erasure channel. Let $\underline{x}_{\rm R}(t)$ denote the packet received by relay. We have:

$$\underline{x}_{\rm R}(t) = \begin{cases} *, & \text{if } \underline{x}(t) \text{ is erased,} \\ \underline{x}(t), & \text{otherwise.} \end{cases}$$

In time-t, once relay receives $\underline{x}_{\rm R}(t)$, it produces a *relay packet* $\underline{y}(t) \in \mathbb{F}_q^{n_2}$ by invoking a relay-side encoder:

$$\mathcal{E}_{\rm R}(t) : \underbrace{\mathbb{F}_q^{n_1} \cup \{*\} \times \cdots \times \mathbb{F}_q^{n_1} \cup \{*\}}_{t+1 \text{ times}} \to \mathbb{F}_q^{n_2}.$$

The relay packet $\underline{y}(t)$ is a function of packets $\{\underline{x}_{\rm R}(t')\}_{t' \in [0:t]}$. Relay transmits $\underline{y}(t)$ to the destination in time-t. Owing to erasures in relay-to-destination link, the packet $\underline{y}_{\rm D}(t)$ received by destination in time-t is given by:

$$\underline{y}_{\rm D}(t) = \begin{cases} *, & \text{if } \underline{y}(t) \text{ is erased,} \\ \underline{y}(t), & \text{otherwise.} \end{cases}$$

At time-$(t+T)$, destination uses decoder:

$$\mathcal{D}(t) : \underbrace{\mathbb{F}_q^{n_2} \cup \{*\} \times \cdots \times \mathbb{F}_q^{n_2} \cup \{*\}}_{t+1+T \text{ times}} \to \mathbb{F}_q^k$$

to obtain an estimate $\hat{\underline{m}}(t) \in \mathbb{F}_q^{k \times 1}$ of $\underline{m}(t)$ as a function of received packets $\{\underline{y}_{\rm D}(t')\}_{t' \in [0:t+T]}$. The decoder is delay-constrained as $\underline{m}(t)$ has to be estimated by time-$(t+T)$. The tuple $(\{\mathcal{E}_{\rm S}(t)\}, \{\mathcal{E}_{\rm R}(t)\}, \{\mathcal{D}(t)\})$ constitutes an $(n_1, n_2, k, T)_q$ *streaming code*. Rate of an $(n_1, n_2, k, T)_q$ streaming code is naturally defined to be $\frac{k}{\max\{n_1, n_2\}}$.

Definition II.1 (Erasure Sequences)**.** A source-relay erasure sequence denoted by $e_{\rm S}^\infty \triangleq \{e_{{\rm S},t}\}_{t \in [0:\infty]}$ is a binary sequence, where $e_{{\rm S},t} = 1$ iff $x_{\rm R}(t) = *$. Similarly, a relay-destination erasure sequence $e_{\rm R}^\infty \triangleq \{e_{{\rm R},t}\}_{t \in [0:\infty]}$ will have $e_{{\rm R},t} = 1$ iff $y_{\rm D}(t) = *$

Definition II.2 (N-Erasure Sequences)**.** Let $N \in \mathbb{Z}_+$. A source-relay erasure sequence $e_{\rm S}^\infty$ is defined to be an N-erasure sequence if $\sum_{t \in [0:\infty]} e_{{\rm S},t} \le N$. Similarly, $e_{\rm R}^\infty$ is an N-erasure sequence if $\sum_{t \in [0:\infty]} e_{{\rm R},t} \le N$.

Definition II.3 ((N_1, N_2)-Achievability)**.** An $(n_1, n_2, k, T)_q$ streaming code is defined to be (N_1, N_2)-achievable if it is possible to perfectly reconstruct all message packets (i.e., $\hat{\underline{m}}(t) = \underline{m}(t)$ for all t) at the destination in presence of (i) any N_1-erasure sequence $e_{\rm S}^\infty$ and (ii) any N_2-erasure sequence $e_{\rm R}^\infty$.

It may be noted that for an (N_1, N_2)-achievable $(n_1, n_2, k, T)_q$ streaming code, we have $N_1 + N_2 \le T$. This is because, if $N_1 + N_2 > T$, in presence of erasure of erasure patterns $e_{\rm S}^\infty = \{\underbrace{0, \ldots, 0}_{i}, \underbrace{1 \ldots, 1}_{N_1}, 0, \ldots\}$ and $e_{\rm R}^\infty = \{\underbrace{0, \ldots, 0}_{i+N_1}, \underbrace{1 \ldots, 1}_{N_2}, 0, \ldots\}$, it is impossible for the destination to recover $\underline{m}(i)$ by time-$(i+T)$.

In a recent work [11], the authors provide $(n_1, n_2, k, T)_q$ streaming codes for all parameters $\{T, N_1, N_2\}$ which yield rate:

$$R_{T,N_1,N_2} \triangleq \frac{T+1-N_1-N_2}{T+1-\min\{N_1, N_2\}}. \quad (1)$$

The codes presented in [11] are *state-independent* in the sense that relay-side encoding at time-t performed by $\mathcal{E}_{\rm R}(t)$ does not depend on the erasure pattern $\{e_{{\rm S},t'}\}_{t' \in [0:t]}$ observed thus far by the relay. In contrast, in the present paper, we consider *state-dependent* streaming codes for all parameters $\{T, N_1, N_2\}$. If $N_1 \ge N_2$, rate achievable by our codes match (1). However, when $N_1 < N_2$, our codes offer a strict rate improvement over (1).

Remark II.1. Error protection provided by (N_1, N_2)-achievable (n_1, n_2, k, T) streaming codes may appear to be limiting, as they consider only N_1 erasures across all time slots $[0 : \infty]$ in source-relay link and N_2 erasures across all time slots $[0 : \infty]$ in relay-destination link. However, owing to the delay-constrained decoder, these codes can in fact recover from any e_S^∞, e_R^∞ which satisfy: $\sum_{t'=i}^{i+T} e_{S,t} \le N_1$ and $\sum_{t'=i}^{i+T} e_{R,t} \le N_2$ for all $i \in [0 : \infty]$. i.e., in any sliding window of $T{+}1$ consecutive time slots, source-relay and relay-destination links see at most N_1 and N_2 erasures, respectively.

III. Proposed Coding Scheme

In this section, we present the code construction and the results of such streaming code. For space and clarity reasons, we present an example that highlights the main concept ideas and a general sketch of the code construction algorithm. A complete description of the scheme can be found in [].

Theorem 1. *For any N_1, N_2 and T, there exists an (N_1, N_2)-achievable $(n_1, n_2, k, T)_q$ streaming code with rate $R = \min(R_1, R_2)$ where*

$$R_1 = \frac{T+1-N_1-N_2}{T+1-N_2} \quad (2)$$

$$R_2 = \frac{T+1-N_2}{T+1+\sum_{i=0}^{N_1} \frac{N_1-i}{T+1-N_2-(N_1-i)} + \delta} \quad (3)$$

and where δ is an overhead which goes to 0 as $q \to \infty$.

Remark III.1. *In Theorem 1, the δ term represents the fact that, since the relay is changing its coding strategy according to the erasure pattern that has occurred in the link from source to relay, that erasure pattern must also be informed to the destination by the relay. In order to keep it simpler, we present our examples and highlight the main ideas assuming the destination is given the erasure pattern that has occurred from source to relay. In the full proof, we show that the δ term can be easily bounded and goes to 0.*

Corollary 1. *For any T and $N_2 > N_1$, for a sufficiently large q, there exists an (N_1, N_2)-achievable channel-state-dependent $(n_1, n_2, k, T)_q$ streaming code that achieves a rate (strictly) higher than $R = \frac{T+1-N_1-N_2}{T+1-N_1}$ which is the rate achieved by channel-state-independent (N_1, N_2)-achievable streaming codes [11].*

A. Main Ideas

Our streaming code differs from the one presented in [11] mainly in its relaying function. The encoding function from source to relay follows the same core idea. However, the relaying function differs greatly by employing the knowledge of the erasure pattern that has occurred. Below, we highlight the two main concepts which allow us to exploit that knowledge.

1) Variable Rate: In our coding scheme, from relay to destination, each message packet is transmitted using a different rate, based on the erasure pattern its respective source packet has been subject to in the link from source to relay. In [11], each message is transmitted assuming it has been subject to the worst case erasure pattern, which leads to every message being transmitted with an effective delay $T' = T - N_1$ from relay to destination. However, note that, for example, if a source packet has not been erased from source to relay, the relay can transmit the message with effective delay $T' = T$, and it is guaranteed that some source packets will not be erased. Transmitting these packets with a higher rate will lead to an overall higher rate, as will be seen later. Note that it can also lead to a variable streaming code rate, however, we handle that by zero padding.

2) Noisy Relaying: In [11], the relay only transmits information about symbols it has already fully recovered. In this work, the relay transmits symbols that contain interference caused by past messages. Because the destination is guaranteed to recover packets sequentially, this interference is guaranteed to be cancelled by the deadline.

B. Example

Consider, for example, a network with $N_1 = 2$, $N_2 = 3$ and $T = 6$. Let us consider $k = 24$, that is, each message packet consists of 24 symbols. We denote by $m_i(t)$ the ith message symbol at time t. We use the notation $a : i : b = \{a, a+i, a+2i, \dots, b\}$ and $m_{a:i:b}(t) = \left[m_a(t), m_{a+i}(t), \dots, m_b(t)\right]$.

As we show next, the source is using the same coding scheme as [11] which amounts to a systematic transmission with diagonally interleaved block codes which results in transmitting at rate of $24/48 = 1/2$.

First, let us consider the scenario where the erasures in the first link occur in a burst. As can be seen in Fig. 1, the relay starts transmitting source packets that have not been erased immediately with rate $4/7$. This can be seen, for example, for $m(3)$. On the other hand, source packets that are subject to erasures are transmitted with lower rate, since they need to be transmitted "faster" (i.e., with a smaller effective delay). Thus, it can be seen that $\underline{m}(4)$ is transmitted with rate $3/6$ and $\underline{m}(5)$ is transmitted with rate $2/5$. This highlights the variable rate aspect of our coding strategy. Not only that, note that the relay is unable to recover the symbols $m_{2:2:16}(5)$ at time 6. If the relay would wait until it can be recovered without the interference from $\underline{m}(4)$, then $\underline{m}(5)$ would also need to be transmitted with rate $2/5$. In order to be able to transmit it with a better rate, we instead transmit the noisy symbols $m'_{2:2:16}(5) = m_{2:2:16}(5) + m_{1:2:15}(4)$. Then, at the destination, since it can recover $\underline{m}(4)$ entirely at time 10, it can cancel out the interference at time 11 and recover $\underline{m}(5)$. This highlights the concept of noisy relaying.

Time	3	4	5	6	7	8	9	10	11
Source	$m_{1:2:24}(3)$	$m_{1:2:24}(4)$	$m_{1:2:24}(5)$	$m_{1:2:24}(6)$	$m_{1:2:24}(7)$	$m_{1:2:24}(8)$	$m_{1:2:24}(9)$	$m_{1:2:24}(10)$	$m_{1:2:24}(11)$
	$m_{2:2:24}(3)$	$m_{2:2:24}(4)$	$m_{2:2:24}(5)$	$m_{2:2:24}(6)$	$m_{2:2:24}(7)$	$m_{2:2:24}(8)$	$m_{2:2:24}(9)$	$m_{2:2:24}(10)$	$m_{2:2:24}(11)$
	$m_{2:2:24}(1) + m_{2:2:24}(2)$	$m_{1:2:24}(2) + m_{2:2:24}(3)$	$m_{1:2:24}(3) + m_{2:2:24}(4)$	$m_{1:2:24}(4) + m_{2:2:24}(5)$	$m_{1:2:24}(5) + m_{2:2:24}(6)$	$m_{1:2:24}(6) + m_{2:2:24}(7)$	$m_{1:2:24}(7) + m_{2:2:24}(8)$	$m_{1:2:24}(8) + m_{2:2:24}(9)$	$m_{1:2:24}(9) + m_{2:2:24}(10)$
	$m_{1:2:24}(0) + 2m_{2:2:24}(1)$	$m_{1:2:24}(1) + 2m_{2:2:24}(2)$	$m_{1:2:24}(2) + 2m_{2:2:24}(3)$	$m_{1:2:24}(3) + 2m_{2:2:24}(4)$	$m_{1:2:24}(4) + 2m_{2:2:24}(5)$	$m_{1:2:24}(5) + 2m_{2:2:24}(6)$	$m_{1:2:24}(6) + 2m_{2:2:24}(7)$	$m_{1:2:24}(7) + 2m_{2:2:24}(8)$	$m_{1:2:24}(8) + 2m_{2:2:24}(9)$

Time	3	4	5	6	7	8	9	10	11
Relay	$m_{1:4:24}(3)$			$m_{1:4:24}(6)$	$m_{1:4:24}(7)$	$m_{1:4:24}(8)$	$m_{1:4:24}(9)$	$m_{1:4:24}(10)$	$m_{1:4:24}(11)$
	$m_{2:4:24}(2)$	$m_{2:4:24}(3)$			$m_{2:4:24}(6)$	$m_{2:4:24}(7)$	$m_{2:4:24}(8)$	$m_{2:4:24}(9)$	$m_{2:4:24}(10)$
	$m_{3:4:24}(1)$	$m_{3:4:24}(2)$	$m_{3:4:24}(3)$			$m_{3:4:24}(6)$	$m_{3:4:24}(7)$	$m_{3:4:24}(8)$	$m_{3:4:24}(9)$
	$m_{4:4:24}(0)$	$m_{4:4:24}(1)$	$m_{4:4:24}(2)$	$m_{4:4:24}(3)$			$m_{4:4:24}(6)$	$m_{4:4:24}(7)$	$m_{4:4:24}(8)$
		$P^{(0)}_{1:1:6}(4)$	$P^{(1)}_{1:1:6}(5)$	$P^{(2)}_{1:1:6}(6)$	$P^{(3)}_{1:1:6}(7)$			$P^{(6)}_{1:1:6}(10)$	$P^{(7)}_{1:1:6}(6)$
			$P^{(0)}_{1:1:6}(5)$	$P^{(1)}_{1:1:6}(6)$	$P_{1:1:6}{}^{(2)}(7)$	$P^{(3)}_{1:1:6}(8)$			$P^{(6)}_{1:1:6}(11)$
				$P^{(0)}_{1:1:6}(6)$	$P^{(1)}_{1:1:6}(7)$	$P^{(2)}_{1:1:6}(8)$	$P^{(3)}_{1:1:6}(9)$		
				$m_{2:2:24}(4)$	$m_{1:2:24}(4)$	$P^{(4)}_{1:1:12}(8) = m_{2:2:24}(4) + m_{1:2:24}(4)$	$P^{(4)}_{1:1:12}(9) = m_{2:2:24}(4) + 2m_{1:2:24}(4)$	$P^{(4)}_{1:1:12}(10) = m_{2:2:24}(4) + 3m_{1:2:24}(4)$	
				$m'_{2:2:16}(5) = m_{1:2:16}(4) + m_{2:2:16}(5)$	$m_{1:2:16}(5)$	$m_{17:1:24}(5)$	$P^{(5)}_{1:1:8}(9)$	$P^{(5)}_{1:1:8}(10)$	$P^{(5)}_{1:1:8}(11)$

Fig. 1: $T = 6,\ N_1 = 2\ N_2 = 3$ example of burst erasures in the link between source and relay

Let us now consider the scenario where the erasures are spaced. Since $\underline{m}(4)$ is subject to only one erasure, we start transmitting it attempting to transmit with rate $3/6$, similar to how the packet $\underline{m}(5)$ was transmitted in the previous example. However, another erasure occurs at time 6 - thus, we now don't have enough symbols to keep transmitting with such high rate, and instead we simply transmit the remaining symbols that have been previously recovered at time 5. Then, at time 7, we start transmitting $\underline{m}(6)$ with rate $3/6$, and we change the rate used for $\underline{m}(4)$ from $3/6$ to $2/5$, as it now has been subject to two erasures. This highlights the adaption our relaying scheme performs based on the observed erasure pattern.

Time	3	4	5	6	7	8	9	10	11	12
Source	$m_{1:2:24}(3)$	$m_{1:2:24}(4)$	$m_{1:2:24}(5)$	$m_{1:2:24}(6)$	$m_{1:2:24}(7)$	$m_{1:2:24}(8)$	$m_{1:2:24}(9)$	$m_{1:2:24}(10)$	$m_{1:2:24}(11)$	$m_{1:2:24}(12)$
	$m_{2:2:24}(3)$	$m_{2:2:24}(4)$	$m_{2:2:24}(5)$	$m_{2:2:24}(6)$	$m_{2:2:24}(7)$	$m_{2:2:24}(8)$	$m_{2:2:24}(9)$	$m_{2:2:24}(10)$	$m_{2:2:24}(11)$	$m_{2:2:24}(12)$
	$m_{2:2:24}(1) + m_{2:2:24}(2)$	$m_{1:2:24}(2) + m_{2:2:24}(3)$	$m_{1:2:24}(3) + m_{2:2:24}(4)$	$m_{1:2:24}(4) + m_{2:2:24}(5)$	$m_{1:2:24}(5) + m_{2:2:24}(6)$	$m_{1:2:24}(6) + m_{2:2:24}(7)$	$m_{1:2:24}(7) + m_{2:2:24}(8)$	$m_{1:2:24}(8) + m_{2:2:24}(9)$	$m_{1:2:24}(9) + m_{2:2:24}(10)$	$m_{1:2:24}(10) + m_{2:2:24}(11)$
	$m_{1:2:24}(0) + 2m_{2:2:24}(1)$	$m_{1:2:24}(1) + 2m_{2:2:24}(2)$	$m_{1:2:24}(2) + 2m_{2:2:24}(3)$	$m_{1:2:24}(3) + 2m_{2:2:24}(4)$	$m_{1:2:24}(4) + 2m_{2:2:24}(5)$	$m_{1:2:24}(5) + 2m_{2:2:24}(6)$	$m_{1:2:24}(6) + 2m_{2:2:24}(7)$	$m_{1:2:24}(7) + 2m_{2:2:24}(8)$	$m_{1:2:24}(8) + 2m_{2:2:24}(9)$	$m_{1:2:24}(9) + 2m_{2:2:24}(10)$

Time	3	4	5	6	7	8	9	10	11	12
Relay	$m_{1:4:24}(3)$		$m_{1:4:24}(5)$		$m_{1:4:24}(7)$	$m_{1:4:24}(8)$	$m_{1:4:24}(9)$	$m_{1:4:24}(10)$	$m_{1:4:24}(11)$	$m_{1:4:24}(12)$
	$m_{2:4:24}(2)$	$m_{2:4:24}(3)$		$m_{2:4:24}(5)$		$m_{2:4:24}(7)$	$m_{2:4:24}(8)$	$m_{2:4:24}(9)$	$m_{2:4:24}(10)$	$m_{2:4:24}(11)$
	$m_{3:4:24}(1)$	$m_{3:4:24}(3)$	$m_{3:4:24}(3)$		$m_{3:4:24}(5)$		$m_{3:4:24}(7)$	$m_{3:4:24}(8)$	$m_{3:4:24}(9)$	$m_{3:4:24}(10)$
	$m_{4:4:24}(0)$	$m_{4:4:24}(1)$	$m_{4:4:24}(2)$	$m_{4:4:24}(3)$		$m_{4:4:24}(5)$		$m_{4:4:24}(7)$	$m_{4:4:24}(8)$	$m_{4:4:24}(9)$
		$P^{(0)}_{1:1:6}(4)$	$P^{(1)}_{1:1:6}(5)$	$P^{(2)}_{1:1:6}(6)$	$P^{(3)}_{1:1:6}(7)$		$P^{(5)}_{1:1:6}(9)$		$P^{(7)}_{1:1:6}(11)$	$P^{(8)}_{1:1:6}(12)$
			$P^{(0)}_{1:1:6}(5)$	$P^{(1)}_{1:1:6}(6)$	$P^{(2)}_{1:1:6}(7)$	$P^{(3)}_{1:1:6}(8)$		$P^{(5)}_{1:1:6}(10)$		$P^{(7)}_{1:1:6}(12)$
				$P^{(0)}_{1:1:6}(6)$	$P^{(1)}_{1:1:6}(7)$	$P^{(2)}_{1:1:6}(8)$	$P^{(3)}_{1:1:6}(9)$		$P^{(5)}_{1:1:6}(11)$	
			$m_{2:2:16}(4)$							
				$m_{18:2:24}(4)$						
					$m_{1:2:24}(4)$	$P^{(4)}_{1:1:12}(8) = m_{2:2:24}(4) + m_{1:2:24}(4)$	$P^{(4)}_{1:1:12}(9) = m_{2:2:24}(4) + 2m_{1:2:24}(4)$	$P^{(4)}_{1:1:12}(10) = m_{2:2:24}(4) + 3m_{1:2:24}(4)$		
					$m_{2:2:16}(6)$	$m_{1:2:16}(6)$	$m_{17:1:24}(6)$	$P^{(5)}_{1:1:8}(9)$	$P^{(5)}_{1:1:8}(10)$	$P^{(5)}_{1:1:8}(11)$

Fig. 2: $T = 6,\ N_1 = 2\ N_2 = 3$ example of spaced erasures in the link between source and relay

In general, our scheme attempts to transmit each source packet with the maximal possible rate, and, as soon as it observes a new erasure, it reduces the rate of transmission of the affected source packets. Further, it also transmits noisy symbols when required, knowing that the noise can always be cancelled at the destination due to the nature of the delay constraint. In the following section we present the general code construction.

C. Code Construction

For given parameters $\{N_1, N_2, T\}$, we set message packet, source packet sizes as the following:

$$k \triangleq \prod_{i=0}^{N_1} T+1-N_2-i, \tag{4}$$

$$n_1 \triangleq (T+1-N_2)\prod_{i=0}^{N_1-1} T+1-N_2-i, \tag{5}$$

$$n_2 \triangleq (T+1-N_1)\prod_{i=1}^{N_1} T+1-N_2-i + \sum_{l=1}^{N_1}\prod_{i=0,i\neq l}^{N_1} T+1-N_2-i. \tag{6}$$

This choice is to ensure that every subcode from relay to destination, which have a rate of the form $(T+1-N_2-i)/(T+1-i)$, can be met with integer parameters.

Let message packet $\underline{m}(t)$ be represented as a column vector of the form:

$$\underline{m}(t) \triangleq \left[m_0(t)\ m_1(t)\ \cdots\ m_{k-1}(t)\right]^T.$$

For consistency in notation, we assume that $\underline{m}(t) \triangleq \underline{0}$, if $t < 0$.

D. Source-to-Relay Encoding

As mentioned previously, the source-to-relay encoding is fairly similar to the previous work on [11]. The only major difference is that we use multiple "layers" of the same code. This can be seen in the previous example, where we use 12 layers of a $2/4$ code, that is, we replicate a $2/4$ diagonally-interleaved MDS code twelve times.

In general, we use $\ell' = \prod_{i=0}^{N_1-1}(T+1-N_2-i)$ layers of diagonally-interleaved MDS codes with parameters $k' = T+1-N_1-N_2$ and $n' = T+1-N_2$. A complete description of the construction of such codes can be found in the full paper version in []. With such parameters, we have $n_1 = \ell' n'$, which equals (5).

Now, we make a major observation about such codes. Using Lemma 3 from [11], we know that, if $\underline{x}(i)$ has been erased, then ℓ' symbols of $\underline{m}(i)$ can be recovered at time $i+N_1$, another ℓ' symbols can be recovered at time $i+N_1+1$, and so on, until the entire message has been recovered. This observation is a guarantee independent of erasure pattern. However, considering the erasure pattern, we make a stronger claim about the recovery of symbols and, especially, "noisy symbols", which we now define as estimates.

Definition III.1. We say $\tilde{m}_j(i)$ is an *estimate* of a source symbol $m_j(i)$ if there exists a function $\Psi_{i,j}$ such that $\Psi_{i,j}(\tilde{m}_j(i), \{\underline{m}(t)\}_{t\in[0:i-1]}) = m_j(i)$.

Proposition III.1. *Assume the packet $\underline{x}(i)$ is erased. Then, ℓ' new estimates of symbols of $\underline{m}(i)$ can be recovered from each subsequent non-erased packet $\underline{x}(i')$, $i' > i$, until estimates of all symbols have been recovered.*

What we mean by "new" is that the estimates we recover from each non-erased packet are always estimates of symbols for which we did not have an estimate yet. To understand this definition and proposition, let us consider the examples given previously. First, consider the example in Fig. 2. In this example, it is straight forward that the relay can recover 12 symbols from $\underline{m}(4)$ at time 5 (from the first non-erased packet after time 4) and another 12 symbols at time 7 (from the second non-erased packet). However, in the example in Fig. 1, the relay only has access to the so-called *estimates* of 12 symbols of $\underline{m}(5)$ at time 6, since there is still interference from $\underline{m}(4)$. Nonetheless, as can be seen in the example and can be shown to hold in general, relaying these estimates is enough, since the destination will have access to previous messages by the deadline of recovery.

E. Relay-to-Destination Encoding

Relay employs two different encoding mechanisms depending on whether the source packet $\underline{x}(t)$ sent from source is successfully received (non-erased) or not (erased). In each time-t, relay transmits a relay packet $\underline{y}(t)$ which is a function of all non-erased source-to-relay source packets within the set $\{\underline{x}(t')\}_{t'\in[0:t]}$. For ease of exposition, we will view each $\underline{y}(t)$ as an unordered set of n_2 symbols, rather than a column vector.

1) $\underline{x}(t)$ is Non-Erased: If a source packet $\underline{x}(t)$ is successfully received by the relay, owing to the use of systematic source-to-relay encoding, the whole message packet $\underline{m}(t)$ of size k (see (4)) is known to the relay in time-t itself. Relay will partition $\underline{m}(t)$ into $\ell'' \triangleq \prod_{i=1}^{N_1}(T+1-N_2-i)$ message sub-packets $\{\underline{m}'^{(i)}(t)\}_{i\in[0:\ell'-1]}$, each of size $k'' \triangleq T+1-N_2$. Relay will then employ diagonal interleaving involving $[n'' \triangleq T+1, k'']$-systematic-MDS codes for each of $\{\underline{m}'^{(i)}(t)\}_{i\in[0:\ell'-1]}$ in the following manner. Let $G \triangleq [I_{k''}\ P]$ denote the generator matrix of the $[n'', k'']$-MDS code, $\underline{m}'^{(0)}(t) \triangleq [m_0'^{(0)}(t)\ m_1'^{(0)}(t)\ \cdots\ m_{k''-1}'^{(0)}(t)]^\top = [m_0(t)\ m_1(t)\ \cdots\ m_{k''-1}(t)]^\top$, $\underline{m}'^{(1)}(t) \triangleq [m_0'^{(1)}(t)\ m_1'^{(1)}(t)\ \cdots\ m_{k''-1}'^{(1)}(t)]^\top = [m_{k''}(t)\ m_{k''+1}(t)\ \cdots\ m_{2k''-1}(t)]^\top$ and so on. Let

$$[p^{(i)}(t+k'')\ p^{(i)}(t+k''+1)\ \cdots\ p^{(i)}(t+n''-1)] = \underline{m}'^{(i)}(t)^\top P.$$

Then, for all $i \in [0 : \ell''-1]$, the relay adds $m_1'^{(i)}(t)\ \cdots\ m_{k''-1}'^{(i)}(t), p^{(i)}(t+k'')\ p^{(i)}(t+k''+1)\ \cdots\ p^{(i)}(t+n''-1)$ to $\underline{y}(t), \underline{y}(t+1), \ldots, y(t+n'-1) \triangleq \underline{y}(t+T)\}$, respectively. Thus, each non-erased source packet $\underline{x}(t)$ contributes ℓ'' symbols to each of the relay packets $\underline{y}(t), \underline{y}(t+1), \ldots, y(t+n'-1)$. Note that there is a slight difference (apart from the difference in MDS code parameters) in the way message symbols are arranged in the diagonal interleaving techniques employed at source-side encoder and relay-side encoder. In the source from relay link, symbols of each message sub-packet $\underline{m}^{(i)}(t)$ appear vertically (within the same coded sub-packet). However, in relay-side diagonal interleaving, symbols of each message sub-packet appear diagonally, i.e., they are part of the same MDS codeword.

If $\underline{x}(t)$ is erased, relay has no information of $\underline{m}(t)$ in time-t and relay will follow a different encoding mechanism. Relay will include $\{C(t;1), C(t;2), \ldots, C(t;T)\}$ as a part of

$\underline{y}(t+1), \underline{y}(t+2), \ldots, \underline{y}(t+T)$, respectively. Here, each $C(t;j)$ is a set of code symbols (to be viewed as a column vector) computed by relay, as a function of all non-erased source packets in time slots $[0 : t+j]$. The size of each $C(t;j)$ can vary anywhere in $[0 : \ell']$. In the remainder of this section, we will discuss (i) how to determine $C(t;j)$'s, (ii) how we obtain a relay packet size which matches (6) and (iii) how recoverability of each $\underline{m}(t)$ is guaranteed at destination by time-$(t+T)$ despite the possibility of N_2 erasures in relay-to-destination link.

2) $\underline{x}(t)$ is Erased: Let $\mathcal{I}_t \triangleq \{t_1, t_2, \ldots, t_{T+1-N_2-N_1}\}$ denote the first $T+1-N_2-N_1$ time slots in $[t+1 : t+T]$ during which there are no erasures in source-to-relay link. Based on Proposition III.1, relay has access to $\ell' j$ estimates of symbols of $\underline{m}(t)$ by time-t_j, $j \in [1 : T+1-N_2-N_1]$.

Recall that $C(t;i)$, $i \in [1 : T]$ consists of a set of code symbols which are to be included a part of relay packet $\underline{y}(t+i)$. Each $C(t;i)$ has size $\alpha_{t,i} \in [0 : \ell']$ and is computed purely as a function of estimates of $\underline{m}(t)$ received in time slots $\{t_j, j \in [1 : T+1-N_2-N_1] \mid t_j \leq t+i\}$. Each $\alpha_{t,i}$ is determined on-the-fly by relay in time-$(t+i)$ based on erasure pattern in the source-relay link in time slots $[t : t+i]$. $C(t;i)$ is obtained by "slicing" a codeword of a systematic MDS code in the following manner. Consider a "long" systematic $[n_{\text{long}}, k_{\text{long}}]$-MDS code, where $n_{\text{long}} \triangleq \sum_{i \in [1:T]} \alpha_{t,i}, k_{\text{long}} \triangleq (T+1-N_2-N_1)\ell' = k$. The length-$n_{\text{long}}$ row-vector $C(t)^\top \triangleq [C(t;1)^\top \; C(t;2)^\top \; \cdots \; C(t;T)^\top]$ is then a codeword of this $[n_{\text{long}}, k_{\text{long}}]$-MDS code. Initial k code symbols of $C(t)^\top$ are k estimates of the symbols in $\underline{m}(t)$. Precisely, the first ℓ' code symbols of $C(t)^\top$ are the ℓ' estimates of $\underline{m}(t)$ determined by relay in time-t_1, the next ℓ' code symbols are the ℓ' estimates determined in time-t_2 and so on. The last $n_{\text{long}} - k$ code symbols of $C(t)^\top$ are MDS parity symbols obtained as a function of the initial k code symbols of $C(t)^\top$. In the following, we discuss how $\{\alpha_{t,i}\}$ are determined, which essentially completes the description of relay-to-destination encoder.

Consider time slots $[t : t+T]$. By assumption, $\underline{x}(t)$ is erased and there can be at most $N_1 - 1$ more erasures in time slots $[t+1 : t+T]$ (in source-to-relay link). For $j \in [1 : N_1-1]$, let $t+v_j$ denote the j-th time slot within $[t+1 : t+T]$ where there is an erasure. If there are only $l < N_1 - 1$ erasures in time slots $[t+1 : t+T]$, we set $v_{j'} \triangleq T+1$, $j' \in [l+1 : N_1-1]$. Also, let $v_0 \triangleq 1$, $v_{N_1} \triangleq T$. Let $\kappa_t(t+i)$ denote the cumulative number of estimations of message symbols in $\underline{m}(t)$ available to relay by time-$\{t+i\}$. The values of $\alpha_{t,i}$'s are obtained by relay in the following manner:

- *Step-1*: Initialize $i = 1$. Go to next step.
- *Step-2*: Let the number of erasures in time slots $[t+1 : t+i-1]$ be $j^* \in [0 : N_1-1]$. If $\underline{x}(t+i)$ is not erased or $\kappa_t(t+i) = k$, $\alpha_{t,i} = \frac{k}{T-N_2-j^*} \triangleq \ell_{j^*}$. Go to next step.
- *Step-3* If $\underline{x}(t+i)$ is erased and $\kappa_t(t+i) < k$, $\alpha_{t,i} = \min\{\ell_{j^*}, \kappa_t(t+i) - \sum_{a \in [1:i-1]} \alpha_{t,a}\}$. Go to next step.
- *Step-4* Increment i by 1. If $i \leq T$, go to Step-2.

To illustrate, let us again look at the examples. Let us analyze the code used for the transmission of packet $\underline{m}(4)$. In Fig. 1, at time 5, we have no symbols to transmit, so we have $\alpha_{4,1} = 0$. Afterwards, we transmit at constant $\alpha_{4,i} = \frac{24}{2} = 12$, where $2 = T - N_2 - 1$. On the other hand, in Fig. 2, we have $\alpha_{4,1} = 8$, because $j^* = 0$ (recall that j^* represents how many erasures happened from time $t+1$ up to $t+i-1$). Then, $\alpha_{4,2} = 4$, because $x(6)$ is erased, thus we only transmit the remaining symbols. Finally, we have $\alpha_{4,i} = 12$ for all other packets. Since in both examples we recover 12 symbols of $\underline{m}(4)$ from each non-erased packet, we always have enough symbols. This holds in general, since $\alpha_{t,i} \leq \ell'$.

This is just a greedy algorithm such that as many symbols are included in $C(t;i)$ subject to following constraints:

1) $\alpha_{t,i} \leq \ell_{j^*}$,
2) $C(t;i)$ is a function of message symbol estimates of $\underline{m}(t)$ obtained by relay in non-erased time slots among $[t+1 : t+i]$,
3) $C(t)^\top \triangleq [C(t;1)^\top \; C(t;2)^\top \; \cdots \; C(t;T)^\top]$ is a codeword of a systematic $[n_{\text{long}}, k_{\text{long}}]$-MDS code. Initial k code symbols k message symbol estimates of $\underline{m}(t)$.

3) Worst-Case Length of Relay Packets: Using this code construction, we can easily find the maximum packet length and zero-pad the packets with smaller length. The analysis is omitted due to space limitations, but it leads to (6) plus a small overhead of $\log_q \binom{T+1}{N_1}$, used to inform the destination about the erasure pattern that has occurred in the link from source to relay. By making $q \to \infty$, we have n_2 equal to (6). This shows that the rate achieved by our coding scheme is as given in Theorem 1.

4) Recoverability of $\underline{m}(t)$ at Destination by Time-$(t+T)$: Now, we should show that, for any erasure patterns in each link, the proposed scheme is (N_1, N_2)-achievable, which would then complete the theorem. We present a sketch of the analysis here, and the complete proof can be found in [].

Proposition III.2. *Using our coding scheme, if there are at most N_2 erasures from relay to destination, the destination is able to recover an estimate $\underline{\tilde{m}}(t)$ of $\underline{m}(t)$ at time $t+T$.*

The sketch is as follows: first, consider the scenario where $\underline{x}(t)$ has not been erased. Then, since the source packet $\underline{m}(t)$ is transmitted using layers of a $[T+1, T+1-N_2]$ MDS code, it can be recovered if there are at most N_2 erasures (note $n - k = N_2$). On the other hand, if $\underline{x}(t)$ has been erased, we may consider two different scenarios. In the first scenario, assume the relay always has enough symbols to transmit. Then, it can be shown that, in the long MDS code, the number of symbols in any $T - N_2$ non-erased packets is at least k, thus we can recover k estimates of symbols of $\underline{m}(t)$. On the other hand, if the relay does not have enough symbols at some point (e.g., in Fig. 2), then we can count the number of erased symbols and subtract it from the total number of symbols. Doing so will again lead to the number of non-erased symbols being at least k. Thus, in any scenario, we can always recover k estimates of $\underline{m}(t)$. Finally, by noting that $\underline{\tilde{m}}(0)$ is exactly $\underline{m}(0)$ (as all previous packets are known from assumption), and applying an induction argument, all original source packets can be recovered with delay T.

REFERENCES

[1] E. Martinian and C. W. Sundberg, "Burst erasure correction codes with low decoding delay," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2494–2502, 2004.

[2] E. Martinian and M. Trott, "Delay-Optimal Burst Erasure Code Construction," in *Proc. IEEE Int. Symp. Inf. Theory, ISIT*. IEEE, 2007, pp. 1006–1010.

[3] S. L. Fong, A. Khisti, B. Li, W. Tan, X. Zhu, and J. G. Apostolopoulos, "Optimal Streaming Codes for Channels With Burst and Arbitrary Erasures," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4274–4292, 2019.

[4] M. N. Krishnan, D. Shukla, and P. V. Kumar, "Rate-Optimal Streaming Codes for Channels With Burst and Random Erasures," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4869–4891, 2020.

[5] D. Dudzicz, S. L. Fong, and A. Khisti, "An explicit construction of optimal streaming codes for channels with burst and arbitrary erasures," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 12–25, 2020.

[6] M. N. Krishnan, V. Ramkumar, M. Vajha, and P. V. Kumar, "Simple Streaming Codes for Reliable, Low-Latency Communication," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 249–253, 2020.

[7] D. Leong and T. Ho, "Erasure coding for real-time streaming," in *Proc. IEEE Int. Symp. Inf. Theory, ISIT*. IEEE, 2012, pp. 289–293.

[8] R. Mahmood, A. Badr, and A. Khisti, "Streaming Codes for Multiplicative-Matrix Channels With Burst Rank Loss," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 5296–5311, 2018.

[9] A. Badr, D. Lui, A. Khisti, W. Tan, X. Zhu, and J. G. Apostolopoulos, "Multiplexed Coding for Multiple Streams With Different Decoding Delays," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4365–4378, 2018.

[10] N. Adler and Y. Cassuto, "Burst-Erasure Correcting Codes With Optimal Average Delay," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 2848–2865, 2017.

[11] S. L. Fong, A. Khisti, B. Li, W. Tan, X. Zhu, and J. G. Apostolopoulos, "Optimal Streaming Erasure Codes Over the Three-Node Relay Network," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 2696–2712, 2020.

[12] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 1021–1036, 2012.

[13] E. Domanovitz, A. Khisti, W. Tan, X. Zhu, and J. G. Apostolopoulos, "Streaming Erasure Codes over Multi-hop Relay Network," in *Proc. IEEE International Symposium on Information Theory, ISIT*. IEEE, 2020, pp. 497–502.