Adaptive relaying for streaming erasure codes in a three node relay network

Gustavo Kasper Facenda , M. Nikhil Krishnan, Elad Domanovitz, Silas L. Fong , Ashish Khisti, Wai-Tian Tan and John Apostolopoulos

***Abstract*—This paper investigates adaptive streaming codes over a three-node relayed network. In this setting, a source node transmits a sequence of message packets to a destination with help of a relay. The source-to-relay and relay-to-destination links are unreliable and introduce at most N_1 and N_2 packet erasures, respectively. The destination node must recover each message packet within a strict delay constraint T. The paper presents a new construction of streaming codes for all feasible parameters $\{N_1, N_2, T\}$. Our work improves upon the construction in Fong et al. by adapting the relaying strategy based on the erasure patterns from source to relay. Specifically, the code employs the notion of symbol estimates, which allows the relay to forward information about symbols before it can decode that symbol, and variable-rate encoding, which decreases the rate used to encode a packet as more erasures affect that packet. The codes proposed in this paper achieve rates higher than the ones proposed by Fong et al. whenever $N_2 > N_1$, and achieve the same rate when $N_2 \leq N_1$, in which case the rate is optimal. The paper also presents an upper bound on the achievable rate that takes into account erasures in both links in order to bound the rate in the second link. The upper bound is shown to be tighter than a trivial bound that considers only the erasures in the second link.**

***Index Terms*—Cloud Computing, Streaming, Low-Latency, Symbol-Wise Decode-and-Forward, Adaptive Relay, Forward Error Correction, Packet Erasure Channel, Relayed Network**

I. INTRODUCTION

A number of emerging applications including online real-time gaming, real-time video streaming (video conference with multiple users), healthcare (under the name tactile internet), and general augmented reality require efficient low-latency communication. In these applications, data packets are generated at the source in a sequential fashion and must be transmitted to the destination under strict latency constraints. When packets are lost over the network, significant amount of error propagation can occur and suitable methods for error correction are necessary.

There are two main approaches for error correction due to packet losses in communication networks: Automatic repeat request (ARQ) and Forward error correction (FEC). ARQ is not suitable when considering low latency constraints over long distances, as the 3-way delay may be larger than the required delay constraint. For that reason, FEC schemes are considered more appropriate candidates. The literature has studied codes with strict decoding-delay constraints—called streaming codes—in order to establish fundamental limits of reliable low-latency communication under a variety of packet-loss models. Previous works have studied particular, useful cases. In [2], the authors studied a point-to-point (i.e., two nodes—source and destination) network under a maximal burst erasure pattern. In [3], the authors have studied, separately, burst erasures and arbitrary erasures. In [4], the authors have extended the erasure pattern, allowing for both burst erasures and arbitrary erasures. In particular, it was shown that random linear codes [5] are optimal if we are concerned only with correcting arbitrary erasures. Other works that have further studied various aspects of low-latency streaming codes include [6]–[18].

While most of the prior work on streaming codes has focused on a point-to-point communication link, a network topology that is of practical interest involves a relay node between source and destination, that is, a three-node network. This topology is motivated by numerous applications in which a gateway server, able to decode and encode data, connects two end nodes. Motivated by such considerations, streaming codes for such a setting were first introduced in [19], which derived the time-invariant capacity for the three-node setting, and further extended to a multi-hop network in [20].

However, the work in [19] is constrained to time-invariant codes, in particular, the relay does not exploit the knowledge about the erasure pattern it has observed in order to improve its coding scheme. On the other hand, the work in [20] allows for channel adaptation, however, applying the scheme presented in that work in the reduced three-node relayed network does not improve the rate of the streaming code above [19], and gains are only observed in the multi-hop setting.

In the present paper, we study a three-node communication network involving a source, a destination and a relay node, where the encoding function at the relay node is allowed to adapt to the erasure patterns on the source-relay link. Unlike the approach in [19], our proposed method enables the relay node to transmit information about packets as soon as it is available, which in turn can lead to more efficient error correction on the relay-destination link. The main challenge in the adaptation policy is that it should handle every feasible erasure pattern on the source-relay link and yet be tractable. Furthermore, the upper bound in [19] for time-invariant re-

Gustavo Kasper Facenda, Elad Domanovitz and Ashish Khisti are with the University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: gustavo.k.facenda@gmail.com)

M. Nikhil Krishnan is with the Department of Electrical Engineering, Indian Institute of Technology Palakkad, Palakkad, Kerala 678623, India (email: nikhilkrishnan.m@gmail.com).

Silas L. Fong is with Qualcomm Technologies Inc., Bridgewater, NJ 08807, USA.

Wai-Tian Tan and John Apostolopoulos were with Cisco Systems, San Jose, CA 95134 USA. They are now with Google.

Preliminary results for this paper have been published in [1].

laying schemes does not hold [21], [22] when adaptation is allowed on the relay-destination link. We develop a new non-trivial optimization-based upper bound that aims to find the "worst case" erasure pattern among all erasure patterns that must be corrected. However, this optimization problem is intractable, and instead we propose a heuristic to find a solution to the problem. This heuristic provides a tighter upper bound than a trivial bound that only considers the erasures from relay to destination.

A. Related Works and Applications

Our work follows the same adversarial packet erasure channel model used in previous works such as [2]–[4], [15], [16]. In these works, there is a limit on the number of erasures that may occur, and the goal is to achieve error-free communication within the strict delay constraint. Adaptation of the encoding strategies has been studied in [23], [24] in order to adapt to changing channel statistics. In these works, such adaptation is performed using random linear codes. Relay adaptation has been studied in [20] in order to transmit over a multi-hop setting. Our work also relates to streaming codes with variable-size messages [12], [18], where the source needs to adapt to the different message sizes that it observes.

The setting we study, with an intermediate relay between a source and destination, can be used to model communication between a user and a server. In such scenario, it is common that the user communicates with a nearby node that is connected to the same network as the server, and this node then communicates with the server through an internal network. In this case, the link from source to relay models the path from the user to this intermediate node, and the link from relay to destination models the path from it to the server, or vice-versa. In many applications where such a network setting is common, low latency is desirable—frequently, reducing latency is the reason the internal network is built, so the routing can be optimized to reduce delay (e.g. Riot Games' network [25] or WTFast network [26]), rather than the number of hops, which is usually desired by regular Internet Service Providers.

Furthermore, the impact of latency on the user experience in applications such as cloud gaming, where a cloud server performs the computationally intensive tasks such as video rendering, and then transmits only the video output to the player, has been widely studied [27]–[33]. This latency has different sources, such as propagation delay, hardware delay, server-side processing delay and communications delay, and reducing many of them has been studied, such as server-side delay [34], video-encoding optimization [35] and, as mentioned previously, reducing propagation delay by building internal networks optimized to reduce latency. However, reducing the communication delay seems to be understudied, in particular, the delay caused by packet losses. Considering that the round-trip-delay often represents more than 20% of the delay budget in these applications [36], re-transmissions represent a significant cost in the delay budget. Using streaming codes could make these re-transmissions unnecessary, freeing up a significant fraction of the delay budget.

Similar scenarios appear naturally in other settings, such as virtual and augmented reality, where latency has been linked to motion sickness [37], and, again, e.g. in VR cloud computing, the user communicates with an intermediate node which then communicates with the server.

II. System Model

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 $\{i \in \mathbb{Z}_+ \mid a \le i \le b\}$. Naturally, we set $[a:\infty] \triangleq \{i \in \mathbb{Z}_+ \mid i \ge a\}$.

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 $t \in [0:\infty]$, the source has a *message packet* $\underline{m}(t) \in \mathbb{F}_q^k$ available, which needs to be communicated to the destination via relay. We assume the packet consists of k independent *symbols* drawn uniformly from $\mathbb{F}_q$, and that each packet is independent from each other. For simplicity, we assume that $\underline{m}(t) \triangleq \underline{0}$, if $t < 0$. Towards this, at time-t, source invokes a source-side encoder $\mathcal{E}_{\mathrm{S}}(t) : \underbrace{\mathbb{F}_q^k \times \cdots \mathbb{F}_q^k}_{t+1 \text{ times}} \to \mathbb{F}_q^{n_1}$ to produce a *source encoded 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 $\underline{x}_{\mathrm{R}}(t)$ denote the packet received by relay. We have:

$$\underline{x}_{\mathrm{R}}(t) = \begin{cases} *, & \text{if } \underline{x}(t) \text{ is erased,} \\ \underline{x}(t), & \text{otherwise.} \end{cases} \tag{1}$$

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

$$\mathcal{E}_{\mathrm{R}}(t) : \underbrace{\mathbb{F}_q^{n_1} \cup \{*\} \times \cdots \times \mathbb{F}_q^{n_1} \cup \{*\}}_{t+1 \text{ times}} \to \mathbb{F}_q^{n_2}. \tag{2}$$

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

$$\underline{y}_{\mathrm{D}}(t) = \begin{cases} *, & \text{if } \underline{y}(t) \text{ is erased,} \\ \underline{y}(t), & \text{otherwise.} \end{cases} \tag{3}$$

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

$$\mathcal{D}(t) : \underbrace{\mathbb{F}_q^{n_2} \cup \{*\} \times \cdots \times \mathbb{F}_q^{n_2} \cup \{*\}}_{t+1+T \text{ times}} \to \mathbb{F}_q^k \tag{4}$$

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

Definition 1 (Rate of an $(n_1, n_2, k, T)_q$ streaming code)**.** *The rate of an $(n_1, n_2, k, T)_q$ streaming code is defined to be* $\frac{k}{\max\{n_1, n_2\}}$.

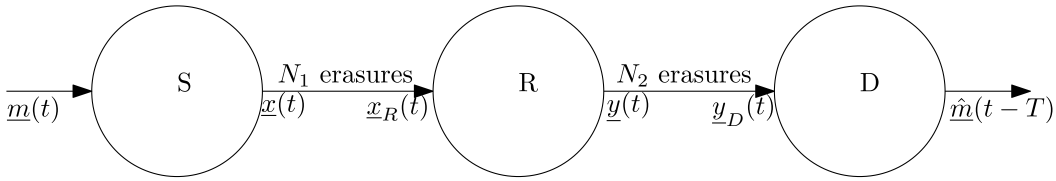

Fig. 1: Three node setting

Definition 2 (Erasure Sequences)**.** A source-relay erasure sequence denoted by $e_{\mathrm{S}}^{\infty} \triangleq \{e_{\mathrm{S},t}\}_{t\in[0:\infty]}$ is a binary sequence, where $e_{\mathrm{S},t} = 1$ iff $x_{\mathrm{R}}(t) = *$. Similarly, a relay-destination erasure sequence $e_{\mathrm{R}}^{\infty} \triangleq \{e_{\mathrm{R},t}\}_{t\in[0:\infty]}$ will have $e_{\mathrm{R},t} = 1$ iff $y_{\mathrm{D}}(t) = *$

Definition 3 (N-Erasure Sequences)**.** Let $N \in \mathbb{Z}_+$. A source-relay erasure sequence e_{S}^{∞} is defined to be an N-erasure sequence if $\sum_{t\in[0:\infty]} e_{\mathrm{S},t} \le N$. Similarly, e_{R}^{∞} is an N-erasure sequence if $\sum_{t\in[0:\infty]} e_{\mathrm{R},t} \le N$.

Definition 4 ((N_1, N_2, T)-Achievability)**.** An $(n_1, n_2, k, T)_q$ streaming code is defined to be (N_1, N_2, T)-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_{S}^{∞} and (ii) any N_2-erasure sequence e_{R}^{∞}. Similarly, a rate R is said to be (N_1, N_2, T)-achievable if there exists an $(n_1, n_2, k, T)_q$ code such that: the code is (N_1, N_2, T)-achievable and $R = \frac{k}{\max(n_1,n_2)}$.

Definition 5 ((N_1, N_2, T)-Capacity)**.** *The (N_1, N_2, T)-capacity, denoted as $C_{N_1,N_2,T}$, is the maximum of all rates that are (N_1, N_2, T)-achievable, as defined in Definition 4.*

It may be noted that, if $N_1 + N_2 > T$, the (N_1, N_2, T)-capacity is 0.

Remark 1. *Error protection provided by (N_1, N_2, T)-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.*

Remark 2. *This channel model is an approximation of a model that introduces independent and identically distributed (i.i.d.) erasures. It allows for a tractable model for which we can design codes, and codes designed for this model perform well under the statistical model it approximates. A detailed motivation is given in [19].*

III. Proposed Scheme

We start by presenting our main result on the achievable rate in the three-node relay setting. Our coding scheme has two components. In the source-to-relay link, we employ a coding scheme that is similar to [19], with a small additional "layering" step. However, in order to produce the relay-to-destination packets, [19] only makes use of message symbols that are fully decodable by the relay. In contrast, our relaying scheme allows the use of "estimates", which are message symbols that have not yet been decoded by the relay, but still provide valuable information, while still allowing the destination to be able to timely recover all packets. We present this idea, as well as other key ideas required to perform adaptation, in Section III-B. Before presenting the general construction, we introduce a toy example that demonstrates each of the main ideas, and is referred throughout the text. Finally, we present the general code construction for the relay-destination link, and analyze its worst case packet size (i.e., n_2), as well as show that it always allows the packets to be recovered within the deadline as long as the channel model holds.

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

$$R_1 = \frac{T + 1 - N_1 - N_2}{T + 1 - N_2} \tag{5}$$

$$R_2 = \frac{T + 1 - N_2}{T + 1 + \sum_{i=0}^{N_1} \frac{N_1 - i}{T+1-N_2-(N_1-i)} + \delta} \tag{6}$$

and where δ is an overhead bounded by $\frac{1}{c}\lceil (T + 1)\log_q 2\rceil$, where c is an arbitrary integer constant that controls the message packet size k. Note that the overhead goes to 0 as c increases.

Remark 3. *In Theorem 1, the δ term represents a header used to inform the destination about the erasure pattern observed, since the relay adapts its strategy according to that pattern. This header is independent of packet sizes k, n_1 and n_2, thus, if k is large, the overhead is negligible. In fact, in the worst case, the header consists of $T + 1$ bits, which is negligible in most applications.*

In order to keep it simple, in the examples and main ideas we assume that the destination has access to the erasure pattern that occurs from source to relay. In the general construction, we present a naive way to provide this information at the cost of the extra header with size δ.

Corollary 1. *For any T and $N_2 > N_1$, for a sufficiently large q, there exists an (N_1, N_2, T)-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, T)-achievable streaming codes [19].*

Throughout the section, We represent the message packet

$\underline{m}(t)$ as a column vector of the form:

$$\underline{m}(t) \triangleq \left[m_0(t)\ m_1(t)\ \cdots\ m_{k-1}(t)\right]^\top. \tag{7}$$

A. Source-to-relay encoding

In our scheme, the source node employs a construction similar to the one in [19]. In the previous work, a diagonally-interleaved maximum distance separable (MDS) code with parameters $k' = (T+1-N_1-N_2)$ and $n' = (T+1-N_2)$ was used. In our scheme, in order to match the different rates possible from relay to destination, we use multiple "layers" of this same code. Precisely, we use $\ell' \triangleq \prod_{i=0}^{N_1-1}(T+1-N_2-i)$ layers of diagonally-interleaved $[n',k']$-MDS codes with parameters k' and n'. The detailed construction of the diagonally-interleaved MDS code can be found in, e.g., [19], but in short it consists of an $[n',k']$-MDS code being diagonally interleaved such that each "diagonal" containing message symbols $m_0(t), m_1(t+1), \ldots, m_{k'-1}(t+k'-1)$ and parity symbols $p_0(k'), p_1(k'+1), \ldots, p_{r'-1}(n'-1)$, where $r' = n'-k'$, belong to the same underlying MDS block code, as illustrated in Fig. 2. That is, in the figure, the parities $p_0(k'), p_1(k'+1), \ldots, p_{r'-1}(n'-1)$ are generated according to a systematic MDS block code for which the information symbols are $m_0(0), \ldots, m_{k'-1}(k'-1)$. Then, we simply multiplex the codewords from each layer together, and obtain $k = k'\ell'$ and $n_1 = \ell' n'$. More precisely, the codewords from a layer i form "sub-packets" $\{\underline{x}^{(i)}(t)\}_{t=0}^{\infty}$. Recall that there are ℓ' layers. At time t, the ℓ' sub-packets $\{\underline{x}^{(i)}(t)\}_{i\in[0:\ell'-1]}$ are vertically stacked together (i.e., multiplexed) forming the coded packet $\underline{x}(t)$. This construction results in the following code parameters

$$k \triangleq \prod_{i=0}^{N_1} T+1-N_2-i, \tag{8}$$

$$n_1 \triangleq (T+1-N_2)\prod_{i=0}^{N_1-1} T+1-N_2-i. \tag{9}$$

Now, we make a major observation about such codes. Using Lemma 3 from [19], 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 at time $i+T-N_2$. This observation is guaranteed independent of erasure pattern, as long as at most N_1 erasures occur. However, considering the erasure pattern, we make a stronger claim about the recovery of symbols and, especially, "symbols with interference", which we now define simply as *estimates*.

Definition 6. *We say $\tilde{m}_j(i) \in \mathbb{F}_q$ is an estimate of a source symbol $m_j(i)$ if there exists a function $\Psi_{i,j}$ such that $\Psi_{i,j}(\tilde{m}_j(i), \{\underline{m}(t)\}_{t\in[0:i-1]}) = m_j(i)$. Equivalently, we have*

$$H(m_j(i)|\tilde{m}_j(i), \{\underline{m}(t)\}_{t=0}^{i-1}) = 0.$$

That is, $\tilde{m}_j(i)$ is an estimate of $m_j(i)$ if, given past ($i' < i$) message packets, we are able to recover $m_j(i)$ from $\tilde{m}_j$.

Proposition 1. *Assume that packet $\underline{x}(i)$ is erased. Then, denote by $\mathcal{I} = \{i_1, \ldots, i_{T+1-N_1-N_2}\}$ the (ordered) time indices of the first $T+1-N_1-N_2$ non-erased source encoded packets after time i, and denote by i_ν the νth element of the set. Then, at time instant i_ν, the relay has access to a set of estimates $\tilde{\mathcal{M}}^\nu$ for which the following properties hold:*

1) $|\tilde{\mathcal{M}}^\nu| = \ell'\nu$
2) $H(\underline{m}(i)|\tilde{\mathcal{M}}^\nu, \{\underline{m}(i')\}_{i'=0}^{i-1}) \leq k - \ell'\nu$

where the entropy is measured in terms of symbols in $\mathbb{F}_q$ (i.e., defined with $\log_q$) and $\ell' = \prod_{i=0}^{N_1-1}(T+1-N_2-i)$ is the number of layers of diagonally-interleaved codes, as defined previously.

Less formally, Proposition 1 is stating that ℓ' "new" estimates of symbols of $\underline{m}(i)$ can be recovered from each subsequent non-erased packet $\underline{x}(i')$, $i' > i$. Note that when $\nu = (T+1-N_1-N_2)$, we have $\ell'\nu = k$, thus, estimates of all symbols can be recovered. By new estimates, we mean that each non-erased packet provides estimates for ℓ' symbols of $\underline{m}(i)$ for which no estimates were previously available. This definition and proposition should be clearer in the example given in Section III-C. The proof is provided in the appendix.

B. Relay-to-destination: Main Ideas

We now present the three main ideas employed in our relaying scheme that allows the relay to transmit using a higher rate than in the non-adaptive schemes by employing the knowledge of the erasure pattern that has occurred. These ideas stem from Proposition 1, that is, from the fact that each non-erased packet provides some information about previous erased packets, and that the relay can decode some *estimates* before the "worst-case" erasure pattern (this will be clearer shortly), and before it can decode the "clean symbols". At this point, we wish to present the high level ideas, and then go through them in the example in the next section, before finally presenting the full construction.

1) Packet-wise variable rate: Let us denote by $N'(t) \triangleq \sum_{i=t}^{t+T-N_2} \mathbf{1}[\underline{x}_R(i) = *]$ the number of erasures in the source-to-relay link in the window $[t : t+T-N_2]$. In order to fully employ the knowledge of the erasure patterns, we must note that $N'(t)$ may be different from $N'(t')$, $t' \neq t$. In particular, assuming both $\underline{x}(t)$ and $\underline{x}(t')$ have been erased, intuitively we can see that if $N'(t) > N'(t')$, the symbols present in $\underline{x}(t)$ are subject to a "harsher" erasure pattern than the symbols in $\underline{x}(t')$. Furthermore, we must note that $N'(t) = N_1$ can not occur simultaneously for all t, that is, there are windows of length $T+1-N_2$ that contain necessarily less than N_1 erasures. With this in mind, the first adaptation we propose is the following: symbols from a packet $\underline{x}(t)$ that are subject to a harsher erasure pattern should be transmitted from relay to destination using a lower rate than symbols from a packet $\underline{x}(t')$ that is subject to a "lighter" erasure pattern. In particular, if $\underline{x}(t)$ has been erased, then the (estimates of information) symbols that belong to that packet are subject to an effective delay $T - N'(t)$ in the relay-to-destination link. As quick examples, let us consider the two extreme cases: if a packet is not erased, then those symbols may be relayed as if it was transmitted through a point-to-point network with N_2 erasures and delay constraint T. On the other hand, if a packet is erased and is subject

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2023.3254464

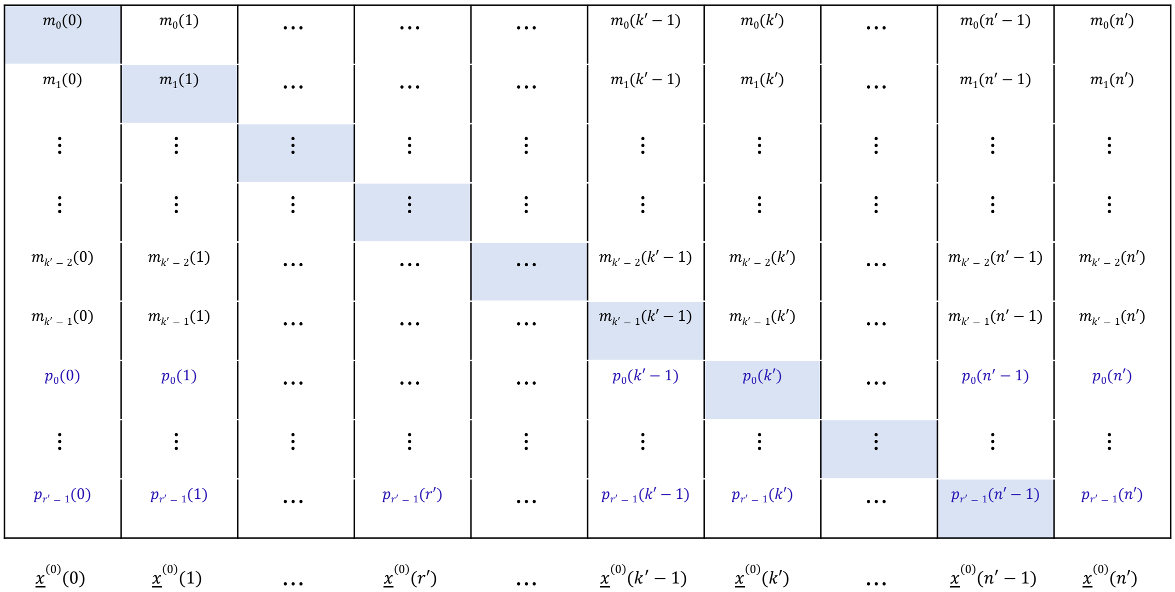

Fig. 2: An illustration of diagonal interleaving technique applied by source-side encoder to produce source encoded sub-packets $\{\underline{x}^{(i)}(t)\}_{t\in[0:\infty]}$. We illustrate here the case $i = 0$ and the same procedure will be applied for all $i \in [0 : \ell' - 1]$. Let $k' \triangleq T + 1 - N_2 - N_1, n' \triangleq T + 1 - N_2, r' \triangleq n' - k' = N_1$ and $\underline{m}^{(0)}(t) \triangleq [m_0(t)\ m_1(t)\ \cdots\ m_{k'-1}(t)]^\top$. Each diagonal is a codeword of a systematic $[n', k']$-MDS code, whose initial k' symbols are message symbols. Source encoded packet $\underline{x}(t)$ is obtained by vertically stacking ℓ' source encoded sub-packets $\{\underline{x}^{(i)}(t)\}_{i\in[0:\ell'-1]}$.

to a burst of $N'(t) = N_1$ erasures, then that packet must be relayed with effective delay $T - N_1$, and the code must still recover from any N_2 erasures, clearly reducing the rate. Then, packets may also be subject to any number of erasures between 0 and N_1. Furthermore, as we have mentioned previously, we should emphasize that not all packets can be subjected to N_1 erasures simultaneously, and therefore we can certainly gain from this adaptation. These different rates also inherently lead to variable code lengths, i.e., variable n_2. However, we handle that by zero padding for consistency with the problem definition. The relaying scheme in [19] essentially assumes $N'(t) = N_1$ for all t, which is the reason it achieves a lower rate than ours.

2) Within-message variable rate: As we saw above, because each packet $\underline{x}(t)$ may be subject to a different number of erasures $N'(t)$, each packet is transmitted with its own rate. However, we should also note that the relay only observes the erasures causally. This means that, when it first decodes a few estimates of symbols of a message packet $\underline{m}(t)$, it may not yet know $N'(t)$, because it does not know the future erasures. However, the relay should start forwarding the information about that packet as soon as it has access to estimates, otherwise there will be a decrease in the rate. Because of that, the rate used to encode each massage packet may also vary as the relay observes new erasures. In order to consistently encode using these variable rates, we use long MDS codes, which encode all k message symbols and allow for variable-rate transmission, rather than diagonally-interleaved "short" MDS codes, which were used in prior works.

3) Relaying with Interference: Finally, another main idea relies on the notion of estimates previously presented. In order to fully appreciate this idea, let us present some context about the prior work: a key insight in [19] was the introduction of symbol-wise decode-and-forward, as opposed to message-wise decode-and-forward, that is, the insight that the relay should not wait until an entire packet is recovered before it relays information about that packet, and instead it can relay information as soon as **symbols** are decoded. In our work, we go further, and we claim that the relay does not need to wait until symbols are recovered, and instead it can relay information as soon as **estimates** are decoded, which is a weaker requirement. We interpret these estimates as symbols with interference, since the past messages are undesired, and could hinder recovery of the desired symbol. On the other hand, due to the delay deadline, this interference can be canceled at the destination before the deadline of the "relevant" symbol.

C. Example

Before we proceed to the general construction, let us go through an example. Consider 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 $a : i : b$ to denote the set $\{a, a + i, a + 2i, \ldots, b\}$ and $m_{a:i:b}(t) = \big[m_a(t), m_{a+i}(t), \ldots, m_b(t)\big]$. If i does not divide $b - a$, the last index instead is the highest value smaller than b that can be written as $a + Ki$, $K \in \mathbb{Z}^+$. Thus, for example, $1 : 2 : 4$ represents $\{1, 3\}$.

The underlying code used by the source is a systematic diagonally-interleaved MDS code with $k' = (T + 1 - N_1 - N_2) = 2$ and $n' = (T + 1 - N_2) = 4$. This code is repeated $\ell' = 12$ times in order to match the possible codes used from relay to destination. Thus, we have $k = 24$ and $n_1 = 48$.

This code can be seen in Table I and Table II, on the upper side of the tables. Note that, no matter what erasure pattern occurs in the source-to-relay link, the symbols $m_{1:2:23}(t)$ can be recovered (by the relay) at time $t+3$, while the symbols $m_{2:2:24}(t)$ can be recovered at time $t+2$, which is the original result from [19]. However, as we shall see next, depending on the erasure pattern, some symbols can be recovered earlier than that.

First, let us consider the scenario where the erasures in the first link occur in a burst. Assume that source encoded packets $\underline{x}(4)$ and $\underline{x}(5)$ are erased, as illustrated in Table I. As the code is systematic, message packets which are part of non-erased source encoded packets, for example, message packets $\underline{m}(3)$ and $\underline{m}(6)$, are immediately available to the relay. On the other hand, packets that are erased, such as packets 4 and 5, are not. The message packets that are present in non-erased source encoded packets are transmitted as soon as possible, with the highest rate possible, which in this case is given by $R = 4/7$. This rate comes from the point-to-point capacity for a link with N_2 erasures and delay constraint T, i.e., $R = \frac{T+1-N_2}{T+1} = \frac{4}{7}$. In order to generate the encoded symbols, we again use an MDS code, and for analysis purposes we simply assume that each parity provides a linearly independent equation. In the bottom part of Table I, symbols belonging to packet $\underline{m}(3)$ are highlighted in blue as an example of such transmission. On the other hand, message packets for which their respective source encoded packets (i.e., where the systematic information is contained) are erased cannot be transmitted with that rate, as there is no information available to the relay yet. Intuitively, because these packets will now need to be transmitted with a lower delay, the rate must also be lower, or, from a different perspective, because the information must be more concentrated, erasures will be more costly, and therefore more redundancy is also needed. Looking at the table again, note that symbols belonging to $\underline{m}(4)$ are subject to two erasures, i.e., at times 4 and 5. Intuitively, this reduces the available delay by two, and the rate we can achieve is now $R = \frac{T-2+1-N_2}{T-2+1} = \frac{2}{5}$. Interestingly, we can think of the symbols belonging to packet $\underline{m}(5)$ as subject to only one erasure, and we can transmit with rate 3/6, following the same intuition above. These examples highlight the packet-wise variable rate aspect of our coding strategy, as different packets are transmitted with different rates, based on how many erasures they were subject to from source to relay. Finally, note that our relaying code works: any three erasures that occur from relay to destination will not stop the decoder from recovering $\underline{m}(3)$ at time 9, $\underline{m}(4)$ at time 10, or $\underline{m}(5)$ at time 11, where the decoding can be performed by simple Gaussian elimination (recall that the parities are generated according to an MDS code). For example, if relay packets 6, 7 and 9 are erased, then relay packet 8 provides 12 symbols worth of information about $\underline{m}(4)$, and 8 symbols worth of information about $\underline{m}(5)$. Later, relay packet 10 provides another 12 symbols worth of information about $\underline{m}(4)$, completing the 24 symbols (recall that $k = 24$), and another 8 symbols worth of information about $\underline{m}(5)$. Finally, relay packet 11 provides the last 8 symbols worth of information about $\underline{m}(5)$.

In the example, we should also note that, at time 6, we transmit what we call estimates, rather than "clean" symbols. Here, we can see why we referred to estimates as "symbols with interference": the message symbols from packet $\underline{m}(4)$ present in $m'_{2:2:16}(5)$ are undesired, and if packet 4 is not recovered, then this would make packet 5 also unrecoverable. However, because packet 4 is recovered at time 10, the destination can cancel this "interference", and packet 6 also effectively provides 8 symbols worth of information about $\underline{m}(5)$ **by the deadline** at time 11. Also, note that n_2 is variable and depends on the erasure pattern on the first link. For this specific pattern, we observe that packets 7, 8 and 9 all contain $n_2 = 50$ symbols (six symbols from each non-erased packet, which are five packets, then 8 symbols from packet 5 and 12 symbols from packet 4).

Now, we would like to highlight the final key idea, which is the within-message variable rate. This idea is important when the relay starts to forward a packet at some high rate, and then new erasures occur and hinder the transmission. This occurs specifically when the erasure pattern is not a (single) burst, such as illustrated in Table II. In this example, the erasures occur at times 4 and 6, but packet 5 is successfully received. Following our previous idea, at time 5, the relay transmits some (specifically, 8) symbols of $\underline{m}(4)$: in this case, it transmits $m_{2:2:16}(4)$, which are already available. This choice is because the relay has observed only one erasure so far, thus it is attempting to transmit with rate 3/6, similar to how it transmitted packet $\underline{m}(5)$ in the previous example. However, a new erasure occurs at time $t = 6$, and the relay does not have another 8 symbols worth of information about $\underline{m}(4)$, instead, it only has access to the remaining 4 symbols that were already recovered at time 5. Since these symbols are all that is available to the relay, it forwards them at time 6. Then, at time 7, it recovers the remaining 12 symbols of $\underline{m}(4)$, and transmits all of them, significantly increasing how many symbols are being transmitted (from 8 to 12), or, equivalently, reducing the rate. In the future transmissions (at times 8, 9 and 10), the relay again transmits parities containing 12 symbols worth of information about $\underline{m}(4)$. This is what we mean by "within-message variable rate": at time 5, the relay attempted to transmit $\underline{m}(4)$ with rate 3/6, with 8 symbols per packet. However, as soon as a new erasure was observed, the rate was adapted and reduced to $2/5$, that is, 12 symbols per packet. Similar to the previous example, we have $n_2 = 50$. Although this needs to hold for all possible erasure patterns, we show later that, indeed, $n_2 = 50$ is the worst case and therefore the rate of our code is $R = 24/50$.

In general, our scheme attempts to transmit each source encoded packet with the maximal possible rate, i.e., $R = \frac{T+1-N'_1-N_2}{T+1-N'_1}$, where N'_1 is the number of erasures observed so far that affect packet $\underline{m}(t)$. As soon as it observes a new erasure, (by updating this N'_1) it reduces the rate of transmission of the affected message packets. Further, it also transmits symbols with interference when required, knowing that the interference can always be cancelled at the destination due to the sequential nature of delay-constrained streaming communications. In the following section we present the

TABLE I: Example of the proposed encoding scheme in case of burst erasures in the link between source and relay for $T=6$, $N_1=2$, $N_2=3$. In this case, the two erasures in the source-to-relay link occur in a burst at times $t=4$ and $t=5$.

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

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

TABLE II: Example of the proposed encoding scheme in case of spaced erasures in the link between source and relay for $T=6$, $N_1=2$, $N_2=3$. In this case, the erasures in the source-to-relay link occur at times $t=4$ and $t=6$, with $t=5$ being successfully received.

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

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

general code construction.

D. Relay-to-Destination Encoding

For the relay-to-destination encoding, let us first state the packet size n_2, and then present the code construction, and finally show that, indeed, the code construction presented uses at most n_2 symbols per packet. Evidently, the number of message symbols is the same as in the link from source to relay, defined in (8), however, as a reminder, we include it here again.

$$k \triangleq \prod_{i=0}^{N_1} T+1-N_2-i, \tag{10}$$

$$n_2 \triangleq (T+1-N_1)\prod_{i=1}^{N_1} T+1-N_2-i + \sum_{l=1}^{N_1}\prod_{i=0,i\neq l}^{N_1} T+1-N_2-i. \tag{11}$$

Remark 4. *The choice of code parameters (in particular, k) is to ensure that every subcode from relay to destination (which have a rate of the form $(T+1-N_2-i)/(T+1-i)$, as mentioned in Section III-B) can be met with integer parameters.*

The relay employs two different encoding mechanisms depending on whether the source encoded packet $\underline{x}(t)$ sent from source is successfully received (non-erased) or not (erased). However, we note that, in case $\underline{x}(t)$ has been erased, the rate adaptation depends on how many erasures occurred and, more generally, on the erasure pattern itself, as mentioned in Section III-B. In each time-t, relay transmits a relay packet $\underline{y}(t)$ which is a function of all non-erased source-to-relay source encoded 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: In this case, since we use systematic encoding from source to relay, the entire message packet $\underline{m}(t)$ is immediately available to the relay at time

t. The relay will then partition the message packet into $\ell'' \triangleq \prod_{i=1}^{N_1}(T+1-N_2-i)$ message sub-packets, denoted as $\{\underline{m}'^{(i)}(t)\}_{i\in[0:\ell''-1]}$. That is, $\underline{m}'^{(i)}(t)$ is the ith sub-packet of the message packet $\underline{m}(t)$. From (8) and ℓ'', it follows that each sub-packet is of size $k'' \triangleq T+1-N_2$. The relay will then employ diagonal interleaving using $[n'' \triangleq T+1, k'']$-MDS codes for each sub-packet, as follows.

Let $G \triangleq [I_{k''}\ P]$ denote the generator matrix of the $[n'', k'']$-MDS code and let each message sub-packet to be represented as a column vector as follows

$$\begin{aligned}\underline{m}'^{(i)}(t) &\triangleq [m_0'^{(i)}(t)\ m_1'^{(i)}(t)\ \cdots\ m_{k''-1}'^{(i)}(t)]^\top \\ &= [m_{i\cdot k''}(t)\ m_{i\cdot k''+1}(t)\ \cdots\ m_{(i+1)k''-1}(t)]^\top\end{aligned}$$

where $i \in [0:\ell''-1]$. Similarly, let

$$\Big[p^{(i)}(t+k'')p^{(i)}(t+k''+1) \cdots\ p^{(i)}(t+n''-1)\Big] = \underline{m}'^{(i)}(t)^\top P, i \in [0:\ell''-1].$$

Then, for all $i \in [0 : \ell''-1]$, the relay appends $m_1'^{(i)}(t)\ \cdots\ m_{k''-1}'^{(i)}(t), p^{(i)}(t+k'')\ p^{(i)}(t+k''+1)\ \cdots\ p^{(i)}(t+n''-1)$ to $\underline{y}(t), \underline{y}(t+1), \ldots, y(t+n'-1) \triangleq \underline{y}(t+T)$, respectively.

That is, each sub-packet is encoded into a diagonal that goes from t up to $t+T$, where the packets at times $t' \in [t : t+k''-1]$ contain systematic symbols, and the remaining packets at times $t' \in [t+k'' : t+T]$ contain parity symbols. Note that there are exactly N_2 parity symbols for each sub-packet.

Thus, each non-erased source encoded packet $\underline{x}(t)$ contributes ℓ'' symbols to each of the relay packets $\underline{y}(t), \underline{y}(t+1), \ldots, y(t+T)$. Note that this is not a systematic streaming code[1], as the message packet $\underline{m}(t)$ is not fully included in the packet $\underline{y}(t)$. That is, in the relay-to-destination code, all symbols from each sub-packet belong to the same underlying MDS code, unlike in the source-to-relay code.

It should be easy to see that we are able to recover $\underline{m}(t)$ from any N_2 erasures using this coding scheme. The rate used is also intuitive: since no erasures occurred from source to relay, we transmit with the same rate as a point-to-point streaming code with delay constraint T and N_2 arbitrary erasures.

2) $\underline{x}(t)$ is Erased: On the other hand, if $\underline{x}(t)$ is erased, then the relay has no information of $\underline{m}(t)$ at time t, and the relay will follow a different encoding strategy. Let $C(t;j)$ be a set of code symbols (to be viewed as a column vector) computed by the relay as a function of all non-erased source encoded packets in time slots $[0 : t+j]$. The size of each $C(t;j)$ can vary from 0 up to ℓ'. Our coding strategy consists of including $C(t;i)$ as a part of $\underline{y}(t+i)$, $i \in [1:T]$. In the following, we discuss (i) how to determine $C(t;j)$, (ii) how we obtain a relay packet size which matches (11) and (iii) how we can guarantee that $\underline{m}(t)$ is recoverable at the destination at time $t+T$ under any N_2 erasures in the relay-to-destination link.

Let $\mathcal{I}_t \triangleq \{t_1, t_2, \ldots, t_{T+1-N_2-N_1}\}$ denote the set containing the first $T+1-N_2-N_1$ time slots in $[t+1 : t+T]$ during which there are no erasures in the source-to-relay link. From Proposition 1, we know that, at time t_j, the relay has access to $\ell' \cdot j$ estimates of symbols of $\underline{m}(t)$, for all $j \in [1 : T+1-N_2-N_1]$.

We start by describing an overview of our coding scheme, without specifying the sizes of each $C(t;j)$. Consider a "long" systematic $[n_{\text{long}}, k_{\text{long}}]$-MDS code, $k_{\text{long}} \triangleq k$, that is, an MDS code which encodes all k symbols from the message packet $\underline{m}(t)$ into n_{long} coded symbols. The parameter n_{long}, as we will explain now, depends on the erasure pattern in the source-relay link. Let us start by constructing a length-n_{long} row-vector $C(t)^\top$ which is a codeword of this long MDS code.

In order to construct $C(t)^\top$, the initial k code symbols of $C(t)^\top$ are k estimates of the symbols of $\underline{m}(t)$. More specifically, the first ℓ' code symbols of $C(t)^\top$ are the ℓ' estimates of $\underline{m}(t)$ determined by relay at time t_1 (i.e., from the first non-erased source encoded packet), the next ℓ' code symbols are the ℓ' estimates determined at time t_2 (second non-erased source encoded packet) and so on, up to the first k symbols of $C(t)^\top$. On the other hand, the final $n_{\text{long}} - k$ code symbols of $C(t)^\top$ are MDS parity symbols obtained as a function of the initial k code symbols of $C(t)^\top$.

Now, let us discuss how to obtain $C(t;j)$ from $C(t)^\top$. Let us define $\alpha_t(t+i)$ as the size of $C(t;i)$, and let our codeword be written as $C(t)^\top \triangleq [C(t;1)^\top\ C(t;2)^\top\ \cdots\ C(t;T)^\top]$. Then, by definition, we should have $n_{\text{long}} = \sum_{i\in[1:T]} \alpha_t(t+i)$. We now discuss how to determine each $\alpha_t(t+i)$.

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 the source-to-relay link). Let us denote by $\kappa_t(t+i)$ the cumulative number of estimations of message symbols of $\underline{m}(t)$ available to the relay at time $t+i$. Then, $\alpha_t(t+i)$ is obtained as described in Algorithm 1. In the algorithm, at time instant $t+i$, $\gamma(i)$ keeps the number of erasures that have occurred from time $t+1$ up to $t+i-1$, that is, the erasures the relay has observed in the past.

Algorithm 1 Computation of $\alpha_t(t+i)$ for $i \in [1:T]$

```
i ← 1
while i ≤ T do
    γ(i) ← number of erasures in time slots [t+1 : t+i−1].
    ℓ_γ(i) ← k / (T − N_2 − γ(i))
    if x(t+i) is not erased or κ_t(t+i) = k then
        α_t(t+i) ← ℓ_γ(i)
    else
        available ← κ_t(t+i) − Σ_{a∈[1:i−1]} α_t(t+a)
        α_t(t+i) ← min {ℓ_γ(i), available}
    end if
    i ← i+1
end while
```

To illustrate, let us analyze the code used for the transmission of packet $\underline{m}(4)$ in the example. In Table I, we start with $i=1$, that is, we shall analyze how many symbols should be transmitted at time 5, i.e., $\alpha_{4,1}$. Following our algorithm, $\underline{x}(4+1)$ is erased. So far, we have not recovered any symbols from $\underline{m}(4)$, therefore, $\kappa_4(5) = 0$, and we have $\alpha_{4,1} = 0$. This can be seen from the fact that we do not transmit any symbols

[1]While this may seem systematic in the sense that message symbols appear uncoded in later packets, in a proper systematic code, $\underline{m}(t)$ would appear entirely uncoded in $\underline{y}(t)$.

from $\underline{m}(4)$ at time 5 in the example. Afterwards, for $i = 2$, we are able to recover $\ell' = 12$ symbols, so we have $\kappa_4(6) = 12$. Following the algorithm, since $\underline{x}(6)$ is not erased, we have $\alpha_{4,2} = \frac{24}{6-3-1} = 12$. The same goes for the remaining of the transmission.

On the other hand, in Table II, we have $\alpha_{4,1} = \frac{24}{6-3-0} = 8$, and we have $\kappa_4(5) = 12$, since we recovered 12 symbols. Then, at time 6, $\underline{x}(6)$ is erased, thus, we transmit the minimum between $\ell_{j(2)} = 8$ and $\kappa_4(6) - \alpha_{4,1} = 4$, i.e., 4 symbols. Finally, we have $\alpha_{4,i} = 12$ for all other packets, since now $\ell_{\gamma(i)} = 12$, as before.

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

1) $\alpha_t(t+i) \leq \ell_{\gamma(i)} \leq \ell'$,
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. That is, the initial k code symbols are k message symbol estimates of $\underline{m}(t)$.

Note that, because of item 1, we are guaranteed to always have enough symbols to transmit because, if $\underline{x}(t+i)$ is not erased, then we recover at least as many symbols as we transmit, and if $\underline{x}(t+i)$ is erased, then we transmit the minimum between how many symbols we have available and $\ell_{\gamma(i)}$, which is by definition at most the number of symbols we have available.

Remark 5. *At first, it may seem that our relaying scheme has only two states, that is, either $\underline{x}(t)$ has been erased or not. However, note that if $\underline{x}(t)$ is erased, our transmission method is then determined by the variables $\alpha_t(t+i)$ and $\kappa_t(t+i)$. Therefore, the relaying scheme has many different states, and the state is updated depending on whether packets $\underline{x}(t+i)$, $i \in \{1, \dots, T\}$, are erased or not. This can be seen in the example in Table I, where the relaying scheme for $\underline{m}(4)$ is different from (i.e., uses a lower rate than) the one for $\underline{m}(5)$.*

3) Worst-Case Length of Relay Packets: We now wish to show that, using our code construction, the maximum packet length is at most the one described in (11). For consistency, all packets which would have a smaller packet length than the maximum are zero-padded in order to keep a constant packet length as defined in the problem statement[2].

First, let us note that, if $\underline{x}(t')$ is not erased, it appends exactly ℓ'' symbols to each relay packet in time slots from t' up to $t'+T$, where recall that $\ell'' = \frac{k}{T+1-N_2}$. Thus, at some arbitrary time t, each non-erased source encoded packet $\underline{x}(t')$, $t' \in [t-T : t]$ contributes ℓ'' symbols to $\underline{y}(t)$.

Now, let us assume that there are $i \leq N_1$ erasures at time slots $\{\tau_1, \tau_2, \dots, \tau_i\} \subseteq [t-T : t]$, where $\tau_1 < \tau_2 < \cdots < \tau_i$. Now, note that from time $\tau_{i'} + 1$ up to time t, $i - i'$ erasures have occurred, by definition. Recall also that $\alpha_t(t+i) \leq \ell_{\gamma(i)}$ in Algorithm 1. Finally, note that $\ell_0 < \ell_1 < \cdots < \ell_{N_1-1}$. From these properties, we have that

$$\alpha_{\tau_{i'}, t-\tau_{i'}} \leq \ell_{i-i'}, i' \in [1 : i]$$

where $\alpha_{\tau_{i'}, t-\tau_{i'}}$ is the number of symbols appended to $\underline{y}(t)$ due to the erasure in the link from source to relay at time $\tau_{i'}$.

Therefore, the packet length of $\underline{y}(t)$ is at most

$$\tilde{n} \leq \underbrace{(T+1-i)\ell''}_{\text{Contribution of non-erased packets}} + \underbrace{\sum_{i' \in [0:i-1]} \ell_{i'}}_{\text{Contribution of erased packets}} \tag{12}$$

$$= (T+1-i)\ell'' + \sum_{i' \in [0:i-1]} \frac{k}{T - N_2 - i'}. \tag{13}$$

Now, note that $\ell'' < \ell_0$, thus, $i = N_1$ maximizes this packet length and is therefore the worst case. We then have that $\tilde{n} \leq (T+1-N_1)\frac{k}{T+1-N_2} + \sum_{i'=0}^{N_1-1} \frac{k}{T-N_2-i'}$, which matches (11).

Finally, we should note that, since the relay changes its coding strategy according to the erasure pattern observed in the first link, this erasure pattern must also be relayed to the destination, so it knows how to decode. A naive solution is to, at time t, transmit the erasure pattern observed from time $t-T$ up to t, which is a binary sequence of length $T+1$, and does not depend on the packet size. Thus, by making the packet size go to infinity, the rate approaches

$$R_2 \triangleq \frac{k}{n_2} = \frac{T+1-N_2}{T+1+\sum_{i=0}^{N_1-1} \frac{N_1-i}{T+1-N_2-(N_1-i)}}.$$

4) Recoverability of $\underline{m}(t)$ at Destination by Time-$(t+T)$: We have shown that our code construction achieves the desired rate, however, it remains to show that our code is (N_1, N_2, T)-achievable, and therefore so is the proposed rate.

Note that Proposition 1 shows that the relay is able to recover the estimates of $\underline{m}(t)$. Now, it suffices to show that, with the proposed code construction, the destination has access to enough estimates to recover $\underline{m}(t)$ entirely.

Proposition 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 $t+T$. Furthermore, the destination is able to recover $\underline{m}(t)$ at time $t+T$.*

Proof: Note that, from relay to destination, all message packets are transmitted within the same MDS code. We split the analysis in two cases, depending on whether $\underline{x}(t)$ has been erased or not. If $\underline{x}(t)$ has not been erased, then each message sub-packet is encoded using a $[T+1, T+1-N_2]$-MDS code, and at most N_2 erasures may occur, therefore, all $(T+1-N_2)\ell'' = k$ symbols can be recovered, since each layer recovers its own $(T+1-N_2)$ symbols and there are ℓ'' layers.

On the other hand, if $\underline{x}(t)$ has been erased, we again split the analysis in two cases. In the first case, let us assume $\alpha_t(t+i) = \ell_{\gamma(i)}$ for every i, that is, we always have enough symbols to transmit the maximum we wish to. In this case, $\alpha_t(t+i) \geq \frac{k}{T-N_2}$ for all $i \in [1 : T]$. Since there are at most N_2 erasures, there are at least $T - N_2$ available packets, thus we can recover all k message symbols.

[2]This is done strictly to be consistent with the problem statement. In practice, there would be no zero-padding, and instead some packets would simply have smaller sizes, consuming less bandwidth.

Finally, the last case occurs when $\underline{x}(t)$ has been erased, but $\alpha_t(t+i) < \ell_{\gamma(i)}$ for some i. In this case, let us denote by i^* the largest i such that this condition holds. Let us state some easily verifiable facts about i^*:

1) $\underline{x}(t+i^*)$ has been erased, as otherwise we would have $\alpha_t(t+i^*) = \ell_{\gamma(i^*)}$.
2) There are $\gamma^* \triangleq \gamma(i^*)+1$ erasures in the source-to-relay link in time slots $[t+1 : t+i^*]$.
3) $\kappa_t(t+i^*) < k$, as otherwise we would have $\alpha_t(t+i^*) = \ell_{\gamma(i^*)}$.
4) $\sum_{i=1}^{i^*} \alpha_t(t+i) = \kappa_t(t+i^*)$.

Then, for $i \in [i^*+1 : T]$, we have

$$\alpha_t(t+i) \geq \frac{k}{T-N_2-\gamma^*}. \tag{14}$$

This follows from the fact that, for $i > i^*$, we have $\alpha_t(t+i) = \ell_{\gamma(i)}$ from the definition of i^* and $\gamma(i) \geq \gamma^*$ by definition of $\gamma(i)$ and γ^*.

Also, for $i \in [1 : i^*]$, we have

$$\alpha_t(t+i) \leq \frac{k}{T-N_2-(\gamma^*-1)}. \tag{15}$$

This again follows from the fact that, for $i \leq i^*$, $\gamma(i) \leq \gamma^*-1$ (recall that $\gamma(i)$ only counts erasures up to time $t+i-1$).

Also, note the following: $i^*-\gamma^*$ is exactly the number of non-erased source encoded packets from time $t+1$ up to $t+i^*$, since i^* represents the number of transmitted packets and γ^* the number of erased packets. Furthermore, we have

$$\ell'(T+1-N_1-N_2) \stackrel{(a)}{=} k \stackrel{(b)}{>} \kappa_t(t+i^*) \stackrel{(c)}{=} (i^*-\gamma^*)\ell'$$

where (a) follows by definition (8); (b) follows from item 3 above; and (c) follows from the fact that there are $i^*-\gamma^*$ non-erased packets and the relay recovers ℓ' estimates from each non-erased packet as seen in Proposition 1. This implies

$$T-N_2-i^* > N_1-1-\gamma^* \geq 0 \tag{16}$$

where the first inequality is simply from rewriting the previous inequality and the second comes from the fact that $\gamma^* \leq N_1-1$, since at most N_1-1 erasures may occur from time $t+1$ up to $t+T$ (recall that $\underline{x}(t)$ has been erased).

For simplicity, let us define as $\mathcal{I}$ the indices i such that $t+i$ has not been erased in the link from relay to destination. Then, it follows that

$$\sum_{i\in\mathcal{I}} \alpha_t(t+i) \stackrel{(a)}{\geq} \sum_{i=1}^{i^*} \alpha_t(t+i) + (T-i^*-N_2)\frac{k}{T-N_2-\gamma^*} \tag{17}$$

$$\stackrel{(b)}{=} (i^*-\gamma^*)\frac{k}{T+1-N_1-N_2} + (T-i^*-N_2)\frac{k}{T-N_2-\gamma^*} \tag{18}$$

$$\stackrel{(c)}{\geq} (i^*-\gamma^*)\frac{k}{T-N_2-\gamma^*} + (T-i^*-N_2)\frac{k}{T-N_2-\gamma^*} \tag{19}$$

$$= k \tag{20}$$

where (a) follows from the fact that, in the worst case, from conditions (14) and (15), all $i \in [1 : i^*]$ are part of $\mathcal{I}$ (otherwise we get more symbols, that is, if $i' > i^*$ and $i'' \leq i^*$, then $\alpha_t(t+i') > \alpha_t(t+i'')$), and that the remaining can all be bounded by (14); (b) follows from the fact that $\sum_{i=1}^{i^*} \alpha_t(t+i) = \kappa_t(t+i^*) = (i^*-\gamma^*)\ell' = (i^*-\gamma^*)\frac{k}{T+1-N_1-N_2}$, from item 4 above and the fact that, for each non-erased source encoded packet, the relay recovers ℓ' estimates; and (c) comes from the fact that, again, $\gamma^* \leq N_1-1$. The last step results from trivial arithmetic.

Then, owing to the use of the long $[n_{\text{long}}, k_{\text{long}} = k]$-MDS code, since at least k code symbols are available, then all k estimates of message symbols of $\underline{m}(t)$ can be recovered by the delay constraint at time $(t+T)$.

Finally, note that messages $\underline{m}(t')$, $t' < 0$ are available by definition to the destination. Therefore, at time T, since the destination is able to recover an estimate of $\underline{m}(0)$, it is also able to recover the message packet itself. Then, at time $T+1$, since it already has access to $\underline{m}(0)$, the estimate of $\underline{m}(1)$ is enough to recover the message packet as well, and so on. Thus, from this induction argument, it is easy to see that the estimates are enough for the destination to recover the message packets by the deadline. ■

Finally, we propose a naive way to inform the destination about the erasure pattern that has been observed: at every time instant t, the relay forwards the observed erasure pattern from time $t-T$ up to time t. This is a binary sequence of length $T+1$, thus it can be represented by $\lceil (T+1)\log_q 2 \rceil$ symbols. Further, it does not depend on the packet size, thus, we can make this overhead negligible by increasing the packet sizes. This can be easily achieved by multiplexing together c copies of our code[3].

Finally, since we have presented an (N_1, N_2, T)-achievable code with rate as described in Theorem 1, the proof of the theorem is complete.

IV. Upper Bound

We start the upper bound presentation by showing a toy-case example for which we show that an $(N_1 = 1, N_2 = 2, T = 4)$-achievable code must be able to recover from more than N_2 erasures in the second link under some particular conditions. Let us consider the erasure pattern in Table III. Note that, from time t to $t+T$, there are $N_2+1 = 3$ erasures. However, consider the following argument: at time $t+T$, all source encoded packets from time 0 up to $t-1$ must have been recovered. Furthermore, since $\underline{x}(t)$ has been erased, and due to causality, packet $\underline{y}(t)$ must be a function only of packets from time 0 up to $t-1$, as the relay has not yet received any information about packet $\underline{m}(t)$. Therefore, at time $t+T$, that is, at the deadline for recovery of packet $\underline{m}(t)$, the destination must be able to generate $\underline{y}(t)$, thus its erasure should not affect the recovery of $\underline{m}(t)$. This example is formalized in the following entropy equations

$$H(\underline{m}(t)|\{\underline{y}(t')\}_{t'=0}^{t-1}, \underline{y}(t+1), \underline{y}(t+3))$$

[3]For example, for $T = 5$ and $q = 2^8$, this overhead is a one-byte header, which is negligible in a 256 bytes packet.

TABLE III: A valid erasure pattern for $N_1 = 1$, $N_2 = 2$ and $T = 4$. In this example, packets t and $t+7$ are erased from source to relay, while packets t, $t+2$, $t+4$ and $t+7$ are erased from relay to destination. The key insight is that, although $N_2 = 2$, there are three erasures from time t to $t+4$, and yet the erasure pattern is valid.

Time	t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$
Source-Relay	erased							erased
Relay-Destination	erased		erased		erased			erased

$$\overset{(a)}{=} H(\underline{m}(t)|\{\underline{y}(t')\}_{t'=0}^{t-1}, \underline{y}(t+1), \underline{y}(t+3), \{\underline{m}(t')\}_{t'=0}^{t-1})$$
$$\overset{(b)}{=} H(\underline{m}(t)|\{\underline{y}(t')\}_{t'=0}^{t}, \underline{y}(t+1), \underline{y}(t+3), \{\underline{m}(t')\}_{t'=0}^{t-1})$$
$$\overset{(c)}{=} 0$$

where (a) follows from the fact that, at time $t+3$, all packets from time 0 up to $t-1$ must be recoverable, since, up to this point, at most N_2 erasures have occurred in the second link, and at most N_1 in the first link; (b) follows from the fact that, since $\underline{x}(t)$ has been erased, $\underline{y}(t)$ must be a function only of message packets up to time $t-1$, due to causality; and (c) follows from the fact that only $\underline{y}(t+2)$ and $\underline{y}(t+4)$ are missing, thus it is as if only $N_2 = 2$ erasures have occurred, and therefore $\underline{m}(t)$ must be recoverable. Then, by repeating this erasure pattern with a period of 7, and simply computing the ratio of non-erased packets to total packets in the second link, we get that $R_2 \leq \frac{4}{7}$. On the other hand, the trivial bound in this scenario would be $R_2 \leq \frac{T+1-N_2}{T+1} = \frac{3}{5}$. If we consider only time-invariant codes, then the upper bound is given by $\frac{2}{4}$, which is clearly lower than ours. However, it applies only to time-invariant codes, and in fact we can achieve higher than that using our adaptive coding scheme presented in the previous section, which, in this example, achieves $\frac{3}{5.5}$, ignoring the overhead.

The following lemma generalizes this key concept and allow us to make an induction argument in the sequence.

Lemma 1. *Assume that, at time $t+T$, the destination has access to message packets $\{\underline{m}(t')\}_{t'=0}^{t-1}$. Then, if the following conditions hold, an (N_1, N_2, T)-achievable code must be able to recover $\underline{m}(t)$.*

$$\sum_{t'=t}^{t+T} e_{S,t'} \leq N_1 \tag{21}$$
$$\sum_{t'=t}^{t+T} e_{R,t'} \leq N_2, \text{ if } e_{S,t} = 0 \tag{22}$$
$$\sum_{t'=t+1}^{t+T} e_{R,t'} \leq N_2, \text{ if } e_{S,t} = 1. \tag{23}$$

Proof: Let us split the proof in two cases. First, if $e_{S,t} = 0$, the proof follows immediately

$$H(\underline{m}(t)|\{\underline{m}(t')\}_{t'=0}^{t-1}, \{\underline{y}_D(t')\}_{t'=t}^{t+T})$$
$$=H(\underline{m}(t)|\{\underline{m}(t')\}_{t'=0}^{t-1}, \{\underline{y}_D(t')\}_{t'=t}^{t+T}, \{\underline{y}(t'), \underline{x}(t')\}_{t'=0}^{t-1}) \tag{24}$$
$$=0 \tag{25}$$

since, by assumption, the code is (N_1, N_2, T)-achievable, therefore, it must be able to recover from any N_1 erasures in the first link and N_2 erasures in the second link. In particular, since the destination has access to all previous message packets, it is able to generate all channel packets from time 0 up to $t-1$, independent of the erasures that have occurred in the past. Therefore, it is "as if" only the erasures from time t up to $t+T$ have occurred.

Now, let us consider the more interesting case, which is when $e_{S,t} = 1$. In this case, we have

$$H(\underline{m}(t)|\{\underline{m}(t')\}_{t'=0}^{t-1}, \{\underline{y}_D(t')\}_{t'=t+1}^{t+T})$$
$$=H(\underline{m}(t)|\{\underline{m}(t')\}_{t'=0}^{t-1}, \{\underline{y}_D(t')\}_{t'=t+1}^{t+T}, \{\underline{y}(t')\}_{t'=0}^{t}, \underline{x}(t')\}_{t'=0}^{t-1}) \tag{26}$$
$$=0 \tag{27}$$

where, again, since the destination has access to previous message packets, it is able to generate all the channel packets from source to relay $\{\underline{x}(t')\}_{t'=0}^{t-1}$. However, since $e_{S,t} = 1$, we have that $\underline{y}(t)$ must also be a function of $\{\underline{x}(t')\}_{t'=0}^{t-1}$, thus the destination can also generate the channel packet $\underline{y}(t)$. Then, again, it is "as if" only the erasures from time $t+1$ up to $t+T$ have occurred from relay to destination, and, since there are at most N_2 erasures in this window, packet $\underline{m}(t)$ must be recoverable. ■

We now apply this Lemma in an induction argument in order to state that not only we must be able to recover from a sliding window, as shown in e.g. [2] and recalled in Remark 1, we must be able to recover from a sliding window that, sometimes, allows for more than N_2 erasures in the second link.

Lemma 2. *If a code is (N_1, N_2, T)-achievable, then it must be able to correct any erasure patterns for which the following holds:*

$$\sum_{t'=t}^{t+T} e_{S,t'} \leq N_1, \forall t \in \mathbb{Z}_+ \tag{28}$$
$$\sum_{t'=t}^{t+T} e_{R,t'} \leq N_2, \forall t \in \{t \in \mathbb{Z}_+ : e_{S,t} = 0\} \tag{29}$$
$$\sum_{t'=t+1}^{t+T} e_{R,t'} \leq N_2, \forall t \in \{t \in \mathbb{Z}_+ : e_{S,t} = 1\} \tag{30}$$

Remark 6. *These conditions lead to a tighter upper bound than the trivial bound that only considers the second link, which is represented by the condition $\sum_{t'=t}^{t+T} e_{R,t'} \leq N_2, \forall t \in \mathbb{Z}_+$. In particular,* (30) *allows for more erasures in a window than the trivial bound allows, thus, there are less available non-erased packets and the rate must be lower. In other words, the erasure sequence used in the trivial bound meets the conditions of Lemma 2, and allowing for more possible erasure sequences can only improve the upper bound.*

Proof: This follows directly from an induction argument using Lemma 1. Note that, by definition, at time T, the destination has access to message packets $\underline{m}(t')$, $t' < 0$. Therefore, if conditions (21)-(23) hold for $t = 0$, packet $\underline{m}(0)$ is recoverable. Then, at time $T+1$, the destination has access to $\underline{m}(t)$, and if the conditions hold for $t = 1$, it can also recover $\underline{m}(1)$, and so on.

Formally, assume that, at time $\tau + T$, the destination has access to all message packets $\underline{m}(t')$, $t' < \tau$. Then, under the assumptions of this Lemma and applying Lemma 1, at time $\tau + T + 1$, the destination will have access to all message packets $\underline{m}(t')$, $t' \leq \tau$. That is, if, at any time instant, all previous message packets are known to the destination, then, under our assumptions and using Lemma 1, all following message packets will also be recoverable by the destination. Finally, remember that, by assumption from the model, $\underline{m}(t')$, $t' < 0$ is known to the destination. Therefore, under the assumptions of this Lemma, all message packets are recoverable by the destination. ∎

This Lemma provides us a framework to try and find an optimal erasure pattern pair which fulfills the desired constraints. Let us define the following optimization problem and denote its solution R_2^* as follows

$$
\begin{aligned}
R_2^* \triangleq \lim_{\tau\to\infty} \quad & \min_{e_S^\infty, e_R^\infty} \frac{1}{\tau}\sum_{t'=0}^{\tau+T}(1-e_{R,t'}) \\
\text{s.t.} \quad & \sum_{t'=t}^{t+T} e_{S,t'} \leq N_1, \forall t \in \mathbb{Z}_+ \\
& \sum_{t'=t}^{t+T} e_{R,t'} \leq N_2, \forall t \in \{t \in \mathbb{Z}_+ : e_{S,t} = 0\} \\
& \sum_{t'=t+1}^{t+T} e_{R,t'} \leq N_2, \forall t \in \{t \in \mathbb{Z}_+ : e_{S,t} = 1\}
\end{aligned} \tag{31}
$$

where the optimization is over all the valid erasure sequences that satisfy Lemma 2. We then can show the following Lemma

Lemma 3. *For any (N_1, N_2, T)-achievable streaming code with parameters k, n_1 and n_2, we have*

$$\frac{k}{n_2} \leq R_2^*. \tag{32}$$

Proof: We have shown, in Lemma 2, that if a code is (N_1, N_2, T)-achievable, it must be able to recover from an erasure sequence that meets the conditions in Lemma 2. Given a valid erasure sequence, it is easy to see that the rate must be below the ratio of non-erased packets (using the given erasure sequence) to total packets. This holds for any window of length τ, and it gets tighter as $\tau \to \infty$, as the difference between τ (total source encoded packets) and $\tau + T$ (total encoded packets) becomes negligible.

That is, the optimization is simply finding the valid erasure sequence that meets the conditions given by Lemma 2 and that minimizes the rate in the second link. ∎

In our example, an erasure sequence that satisfies Lemma 2 can be found by repeating the pattern in Table III with period 7. This can be seen by the fact that, when an erasure occurs in the first link (e.g. at time t), the constraint is that the number of erasures from time $t+1$ up to $t+4$ is less than or equal to N_2, which holds. In other windows, where there are no erasures in the first link, the constraint is that the number of erasures is at most N_2, which again holds. Then, an upper bound is simply given by the ratio between non-erased packets (4) to total packets (7) for this specific erasure pattern. However, we do not claim that the erasure pattern presented in Table III is optimal, instead, it is one valid erasure sequence which results into one valid bound.

However, solving this optimization problem is not trivial. Instead, we propose a heuristic algorithm for the adversary, which attempts to maximize the number of erasures it introduces in the relay-destination channel. The algorithm is described in Algorithm 2.

Algorithm 2 Adversary Heuristic

```
for t = 0, t++, t ≤ τ do
    num_erasures_R(t) ← Σ_{t'=t}^{t+T} e_{R,t'}
    if e_{R,t} == 1 then
        num_erasuresS(t) ← Σ_{t'=t−T}^{t} e_{S,t'}
        if num_erasures_R(t) == N_2 and num_erasures_S(t) < N_1 then
            e_{S,t} ← 1
        end if
    end if
    if (e_{S,t} == 1 and num_erasures_R(t) ≤ N_2) or (num_erasures_R(t) < N_2) then
        e_{R,t+T} ← 1
    end if
end for
```

The idea for the algorithm is simple: we introduce erasures in the second link in a greedily manner, always in the end of the window (i.e., at time $t+T$). This ensures that introducing this erasure will not make any previous window invalid, that is, it will not increase the number of erasures in any previous window to more than it is allowed. Furthermore, under our constrained framework, introducing erasures in the first link only improves the upper bound by allowing one extra erasure in the second link for the same time instant. For that reason, we only introduce erasures in the first link at times when there is already an erasure in the second link **and** that being able to introduce one extra erasure helps, that is, when the number of erasures in the window is already N_2.

Further, we note that the source is not able to adapt to any erasure pattern, thus the cut-set-like bound from [19] for the first link holds even in the adaptive case. That is, as in [19], we can upper bound the capacity by analyzing the first link and noting that the relay must recover every message packet with delay $T - N_2$, otherwise a burst of N_2 erasures in the second link makes that information impossible to recover. Thus, as in the prior works, we can bound $R_1 \leq \frac{T+1-N_1-N_2}{T+1-N_2}$. Finally, this allows us to present the following upper bound.

Theorem 2. *The (N_1, N_2, T)-capacity C_{N_1,N_2} is upper bounded by*

$$C_{N_1,N_2} \leq \min(R_1^*, R_2^*)$$

where $R_1^* \leq \frac{T+1-N_1-N_2}{T+1-N_2}$ and R_2^* is as in (31).

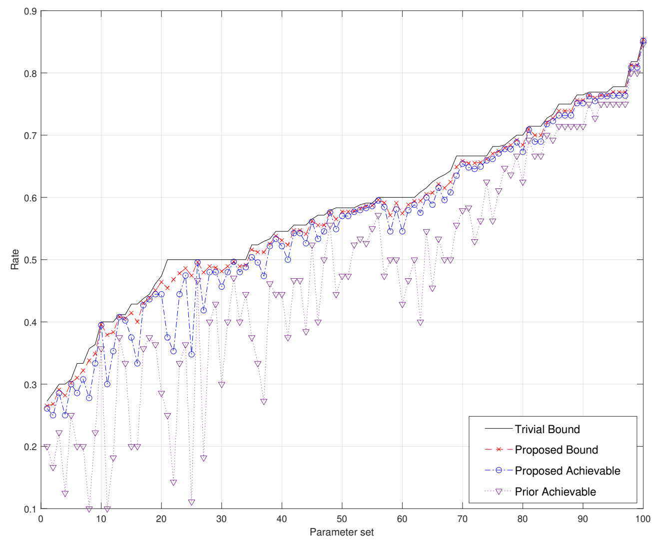

Fig. 3: Comparison between prior work and our work in terms of lower and upper bounds on the capacity of streaming codes in the three-node relayed network.

Proof: This follows immediately from the cut-set-like bound in [19] and Lemma 2. ∎

V. Results

To validate our construction, we consider multiple settings with the parameters (N_1, N_2, T) drawn randomly with the following conditions:

- $10 \geq N_2 > N_1 \geq 1$. The condition $N_2 > N_1$ is so that the second link is the bottleneck. Otherwise, both our coding scheme and the non-adaptive coding scheme from [19] are optimal.
- $N_1 + N_2 + 10 \geq T \geq N_1 + N_2$. The second condition is so that the capacity is not trivial (i.e. not zero).

With the range of parameters, we hope to observe most meaningful scenarios: N_2 and N_1 can be close or fairly separated and the delay constraint T can be tight or loose with respect to the number of erasures.

Then, for each randomly chosen set of parameters, we compute a trivial upper bound, which is obtained by reducing the relayed-setting to a point-to-point setting from relay to destination, the achievable rate from [19], our upper bound and our achievable rate. Then, for presentation, we sort the results according to the trivial bound, in increasing order, and present it in Fig. 3.

It can be seen that our achievable rate is strictly better than the non-adaptive achievable rate, with gains above 100% for some sets of parameters. We also observed that our upper bound is tighter than the trivial bound in all scenarios, although the difference might be negligible for some setting parameters.

We also present the CDF of the ratio between our achievable rate and our upper bound. As can be seen in Fig. 4, our coding scheme achieves higher than 95% of the upper bound in more than 80% of the tested scenarios, and it achieves less than 80% of the upper bound in less than 3% of the scenarios.

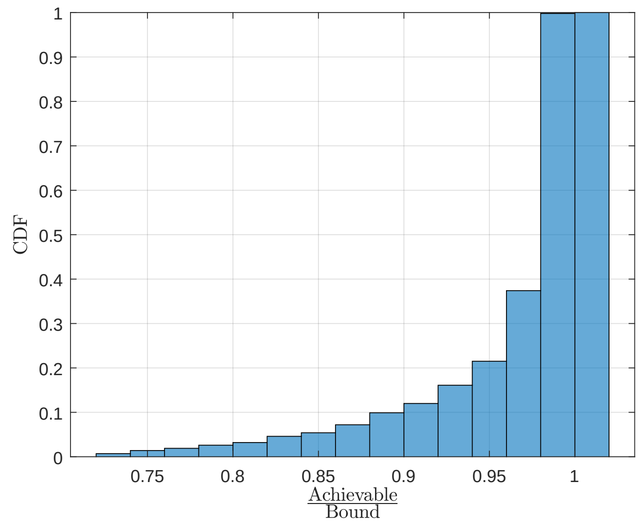

Fig. 4: CDF of proposed achievable rate divided by proposed upper bound

VI. Limitations and Future Work

In this section, we discuss some limitations of our model and our scheme, and point towards interesting future research that may improve upon these.

First, from a model perspective, ideally we would like to be able to design codes for statistical models, rather than the adversarial model. In that case, since some information packets are impossible to recover within the delay constraint, due to the possibility of all packets being erased, we must also include a packet loss rate constraint, and then design codes that meet this loss constraint within the delay constraint. However, such a setting is challenging. In fact, even analytically obtaining the loss rate of a given code construction, such as the one presented in this paper, is not trivial, and is usually done through bounds and simulations. Thus, constructing codes with these constraints is obviously of interest, but impractical with our current understanding.

Another extension that may be of interest in statistical models are codes that are able to recover from (N_1, N_2) erasures with delay T, but also from larger (N_1', N_2') erasures with extended delay T'. Currently, streaming codes in general fail to recover from more erasures than they are designed for, no matter how long the delay, however, random linear codes (for example) are able to decode from any number of erasures, given enough time. Thus, designing streaming codes with such property is also of interest.

Furthermore, our definition considers fixed-size encoded packets, even though our relaying scheme is inherently variable-size. This definition is helpful in that it is more tractable to analyze, allowing us to find a closed form expression to the achievable rate, and allowing us to prove the converse by bounding the amount of information each non-erased packet can provide. This definition is also justified in practice because it represents a limit on the maximum bandwidth allowed for the application, which is also a realistic and important limitation. Future research may be interested in performing similar analysis on different metrics, such as the average bandwidth used by the application. In particular, our scheme also presents an achievable proposal for such metrics, however, it possibly could be improved by focusing on them. Furthermore, converse techniques that focus on these different metrics are also of interest.

Many applications, such as videoconferencing, also deal with variable-size messages. However, streaming with variable-size messages has not yet been solved even in the point-to-point case, although some progress has been made in [12], [18]. Extending the relayed setting to variable-size messages is an interesting open problem, however, as the fixed-size fundamental limits are not yet completely understood, and are also of interest, we focus on these in this work.

On the actual proposed construction, one limitation is that both the packet sizes and the field sizes become significantly large for many sets of parameters. In this work, we are focused on the fundamental limits, and therefore the practicality of the construction is not prioritized. However, in order to be more practical, schemes with shorter packet sizes must be developed. At the moment, it is unclear whether this can be done without significant rate decrease. In our paper, the reason for such large packet sizes is that, for each possible number of erasures i between 0 and N_1, the relay employs a different code with rate $\frac{T+1-N_2-i}{T+1-i}$, thus k must divide $T+1-N_2-i$, for all $i \in \{0,\ldots,N_1\}$. A naive way to obtain a trade-off between packet size and rate is to, instead of adapting to all possible number of erasures, adapt only to a subset of them. More precisely, let $\mathcal{N} \subseteq \{0,\ldots,N_1\}$, we can construct a code with $k = \prod_{i\in\mathcal{N}}(T+1-N_2-i)$, possibly reducing the packet size compared to (8). Then, if the relay observes $i' \notin \mathcal{N}$ erasures, the relay adapts to the rate of the smallest number of erasures $i \in \mathcal{N}$ such that $i > i'$ (note that $N_1 \in \mathcal{N}$ is therefore required). This scheme is not the focus of our paper, thus we do not provide a general rate expression for it, however, it is easy to see that the rate will be worse if we adapt to a smaller subset of the possible number of erasures. As an example, considering the running example from Section III-C, if we only adapt to $\mathcal{N} = \{0,2\}$, not adapting to 1 erasure, we would get $k = (T+1-N_2)(T+1-N_2-N_1) = 8$, significantly smaller than the 24 symbols used in the paper, however, n_2 would (relatively) increase to 18, that is, we would get $R = 8/18 = 0.444$, smaller than the proposed $R = 24/50 = 0.48$, although still higher than $R = 0.4$ achieved in [19]. We also note that selecting $\mathcal{N} = \{N_1\}$, i.e., not adapting to any number of erasures, and simply treating all packets as if N_1 erasures have occurred, will degrade exactly to the scheme in [19]. We do not claim that this trade-off between packet size and rate is optimal, or even that the trade-off necessarily exists, and an interesting future research is to further investigate it, possibly proposing coding schemes that can achieve the same rate as ours with smaller packet sizes.

We also note that there is still a gap between our achievable result and our upper bound. One possible reason is that the upper bound inherently assumes that each non-erased packet contain n_2 symbols worth of information, however, in our construction, some packets contain less than that due to the variable-rate. Another reason is that, in general, there are windows of length $T+1$ with fewer than N_1 erasures in the first link in the erasure patterns found by our optimization. This indicates that there should be an even tighter bound that further exploits the erasures in the first link, which would enforce even lower rates in the second link.

VII. Conclusion

In our paper, we presented a novel adaptive scheme for the delay-constrained relay setting, which exploits the fact that the relay can act based on the observed erasure pattern. We also present a novel upper bound technique, which bounds the rate in the second link according to the erasure pattern observed in the first link. We compare both our achievable and upper bound to the prior work on them, and show that both are at least as good as the previous results and significantly better in some scenarios. In particular, we show that our achievable rate can be higher than twice the one from prior work, and that it is close to the upper bound most of the time.

Follow-up research includes closing the gap between achievable and upper bound, and further adapting our coding scheme to time-varying channels according to a small-frequency feedback provided from destination to the relay and from the relay to source, or directly from destination to source.

Appendix

Proof of Proposition 1: Let us focus on one layer of the source-to-relay code. Consider the sequence of source encoded sub-packets $\{\underline{x}^{(0)}(t)\}_{t\in[0:\infty]}$ produced by interleaving systematic $[n',k']$-MDS codes as shown in Fig. 2. Note that the set of code symbols $\{m_0(t-k'+1), m_1(t-k'+2),\cdots,m_{k'-1}(t),p_0(t+1),\cdots,p_{r'-1}(t+r')\}$ (for example, the diagonal highlighted in the figure) forms a codeword of the underlying $[n',k']$-MDS code.

Now, at time i_ν, consider the diagonal that contains $m_{k'-\nu}(i)$. Note that this diagonal contains $\nu-1$ symbols from *future* packets. However, by definition, i_ν is the νth non-erased packet, therefore, there are exactly ν non-erased packets between time i and time i_ν (included). Therefore, all *future* packets can be removed[4] (e.g., through Gaussian elimination). Therefore, at time instant i_ν, we may always recover the estimate $\tilde{m}_{k'-\nu}(i) = m_{k'-\nu}(i) + \sum_{t=0}^{i-1}\sum_{j=0}^{k-1}\alpha_{t,j}(i_\nu)m_j^{(0)}(t)$, where $\alpha_{t,j}(i_\nu)$ are some linear coefficients resulting from the Gaussian elimination process. Now, note that there are ℓ' layers of such codes, and all are completely independent by construction, thus at time instant i_ν, we have access to $\tilde{\mathcal{M}}^\nu = \tilde{\mathcal{M}}^{\nu-1} \cup \{\{\tilde{m}_{k'-\nu+\ell k'}\}_{\ell=0}^{\ell-1}\}$, where $\tilde{\mathcal{M}}^0 = \{\}$. This completes the proof of the first claim.

Now, consider the set $\mathcal{M}^\nu$ that contains the symbols for which the elements of $\tilde{\mathcal{M}}^\nu$ are estimates. That is, let $\mathcal{M}^\nu \triangleq \mathcal{M}^{\nu-1} \cup \{\{m_{k'-\nu+\ell k'}\}_{\ell=0}^{\ell-1}\}$, again with $\mathcal{M}^0 = \{\}$. Then, the following holds:

$$H(\underline{m}(i)|\tilde{\mathcal{M}}^\nu, \{\underline{m}(i')\}_{i'=0}^{i-1}) \overset{(a)}{=} H(\underline{m}(i)|\tilde{\mathcal{M}}^\nu, \{\underline{m}(i')\}_{i'=0}^{i-1}, \mathcal{M}^\nu) \tag{33}$$

$$\overset{(b)}{\leq} H(\underline{m}(i)|\mathcal{M}^\nu) \tag{34}$$

[4]Recall that the code is a systematic MDS code, thus its generator matrix can be written as $\mathbf{G} = [\mathbf{I}\ \mathbf{P}]$, and all square submatrices of the parity matrix $\mathbf{P}$ are invertible.

$$\stackrel{(c)}{=} H(\underline{m}(i)) - H(\mathcal{M}^{\nu}) \tag{35}$$

$$\stackrel{(d)}{=} k - \ell'\nu \tag{36}$$

where (a) follows from the fact that $\mathcal{M}^{\nu}$ can be recovered from $\tilde{\mathcal{M}}^{\nu}$ and the past messages; (b) follows from removing conditioning; (c) follows from $H(\underline{m}(i), \mathcal{M}^{\nu}) = H(\underline{m}(i)) + H(\mathcal{M}^{\nu}|\underline{m}(i)) = H(\mathcal{M}^{\nu}) + H(\underline{m}(i)|\mathcal{M}^{\nu})$ and the fact that $H(\mathcal{M}^{\nu}|\underline{m}(i)) = 0$; (d) follows from the fact that the symbols are independent and uniformly distributed in $\mathbb{F}_q$. ∎

REFERENCES

[1] M. N. Krishnan, G. K. Facenda, E. Domanovitz, A. Khisti, W. Tan, and J. Apostolopoulos, "High rate streaming codes over the three-node relay network," in *2021 IEEE Information Theory Workshop (ITW2021)*, October 2021.

[2] E. Martinian and C.-E. Sundberg, "Burst erasure correction codes with low decoding delay," *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2494–2502, 2004.

[3] D. Leong and T. Ho, "Erasure coding for real-time streaming," in *2012 IEEE International Symposium on Information Theory Proceedings*, 2012, pp. 289–293.

[4] A. Badr, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Streaming codes for channels with burst and isolated erasures," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2850–2858.

[5] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 41, no. 1. Citeseer, 2003, pp. 11–20.

[6] E. Martinian and M. Trott, "Delay-optimal burst erasure code construction," in *2007 IEEE International Symposium on Information Theory*, 2007, pp. 1006–1010.

[7] G. Joshi, Y. Kochman, and G. W. Wornell, "On playback delay in streaming communication," in *2012 IEEE International Symposium on Information Theory Proceedings*, 2012, pp. 2856–2860.

[8] N. Adler and Y. Cassuto, "Burst-erasure correcting codes with optimal average delay," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2848–2865, 2017.

[9] M. Karzand, D. J. Leith, J. Cloud, and M. Médard, "Design of FEC for low delay in 5G," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 8, pp. 1783–1793, 2017.

[10] A. Badr, P. Patil, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Layered constructions for low-delay streaming codes," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 111–141, 2017.

[11] A. Badr, A. Khisti, W.-t. Tan, X. Zhu, and J. Apostolopoulos, "FEC for VoIP using dual-delay streaming codes," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[12] M. Rudow and K. V. Rashmi, "Streaming codes for variable-size arrivals," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018, pp. 733–740.

[13] M. N. Krishnan and P. V. Kumar, "Rate-optimal streaming codes for channels with burst and isolated erasures," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 1809–1813.

[14] S. L. Fong, A. Khisti, B. Li, W. Tan, X. Zhu, and J. Apostolopoulos, "Optimal streaming codes for channels with burst and arbitrary erasures," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4274–4292, 2019.

[15] E. Domanovitz, S. L. Fong, and A. Khisti, "An explicit rate-optimal streaming code for channels with burst and arbitrary erasures," *arXiv preprint arXiv:1904.06212*, 2019.

[16] M. N. Krishnan, D. Shukla, and P. V. Kumar, "Low field-size, rate-optimal streaming codes for channels with burst and random erasures," *IEEE Transactions on Information Theory*, pp. 1–1, 2020.

[17] M. Haghifam, M. N. Krishnan, A. Khisti, X. Zhu, W.-T. Tan, and J. Apostolopoulos, "On streaming codes with unequal error protection," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 4, pp. 1165–1179, 2021.

[18] M. Rudow and K. V. Rashmi, "Streaming codes for variable-size messages," *IEEE Transactions on Information Theory*, vol. 68, no. 9, pp. 5823–5849, 2022.

[19] S. L. Fong, A. Khisti, B. Li, W. Tan, X. Zhu, and J. Apostolopoulos, "Optimal streaming erasure codes over the three-node relay network," in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 3077–3081.

[20] E. Domanovitz, A. Khisti, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Streaming erasure codes over multi-hop relay network," *CoRR*, vol. 2006.05951, 2020. [Online]. Available: https://arxiv.org/abs/2006.05951

[21] S. L. Fong, A. Khisti, B. Li, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Corrections to "optimal streaming erasure codes over the three-node relay network"," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6527–6527, 2022.

[22] G. K. Facenda, E. Domanovitz, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Streaming erasure codes over multi-access relay networks," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021.

[23] A. Cohen, G. Thiran, V. B. Bracha, and M. Médard, "Adaptive causal network coding with feedback for multipath multi-hop communications," *CoRR*, vol. abs/1910.13290, 2019. [Online]. Available: http://arxiv.org/abs/1910.13290

[24] A. Cohen, G. Thiran, V. B. Bracha, and M. Médard, "Adaptive causal network coding with feedback for multipath multi-hop communications," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 766–785, 2021.

[25] P. Maynard-Koran, "Fixing the internet for real time applications," Feb 2016. [Online]. Available: https://technology.riotgames.com/news/fixing-internet-real-time-applications-part-ii

[26] G. Hains, C. Mazur, J. Ayers, J. Humphrey, Y. Khmelevsky, and T. Sutherland, "The WTFast's gamers private network (GPN ®) performance evaluation results," in *2020 IEEE International Systems Conference (SysCon)*, 2020, pp. 1–6.

[27] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2883–2894, 2013, information System Security and Performance Modeling and Simulation for Future Mobile Networks. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717711007771

[28] P. Quax, A. Beznosyk, W. Vanmontfort, R. Marx, and W. Lamotte, "An evaluation of the impact of game genre on user experience in cloud gaming," in *2013 IEEE International Games Innovation Conference (IGIC)*, 2013, pp. 216–221.

[29] V. Clincy and B. Wilgor, "Subjective evaluation of latency and packet loss in a cloud-based game," in *2013 10th International Conference on Information Technology: New Generations*, 2013, pp. 473–476.

[30] M. Claypool and D. Finkel, "The effects of latency on player performance in cloud-based games," in *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014, pp. 1–6.

[31] I. Slivar, M. Suznjevic, L. Skorin-Kapov, and M. Matijasevic, "Empirical QoE study of in-home streaming of online games," in *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014, pp. 1–6.

[32] Z.-Y. Wen and H.-F. Hsiao, "QoE-driven performance analysis of cloud gaming services," in *2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP)*, 2014, pp. 1–6.

[33] S. Schmidt, S. Zadtootaghaj, and S. Möller, "Towards the delay sensitivity of games: There is more than genres," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, 2017, pp. 1–6.

[34] K. Lee, D. Chu, E. Cuervo, J. Kopf, S. Grizan, A. Wolman, and J. Flinn, "Outatime: Using speculation to enable low-latency continuous interaction for cloud gaming," Microsoft Research, Tech. Rep. MSR-TR-2014-115, August 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/outatime-using-speculation-to-enable-low-latency-continuous-interaction-for-cloud-gaming/

[35] I. Slivar, M. Suznjevic, and L. Skorin-Kapov, "The impact of video encoding parameters and game type on QoE for cloud gaming: A case study using the Steam platform," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, 2015, pp. 1–6.

[36] M. Carrascosa and B. Bellalta, "Cloud-gaming: Analysis of Google Stadia traffic," *CoRR*, vol. abs/2009.09786, 2020. [Online]. Available: https://arxiv.org/abs/2009.09786

[37] J.-P. Stauffert, F. Niebling, and M. E. Latoschik, "Latency and cybersickness: Impact, causes, and measures. a review," *Frontiers in Virtual Reality*, vol. 1, p. 31, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/frvir.2020.582204

Gustavo Kasper Facenda (GS'20) received the B.Sc. and the M.Sc. degrees in electrical engineering from the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, in 2017 and 2019, respectively. He is currently a PhD student at University of Toronto, Toronto, Canada. His research interests include information theory and its applications, such as multiple access channels, relay networks and streaming codes, and applications of machine learning in coding construction, encoding and decoding.

M. Nikhil Krishnan received his B.Tech. in electronics and communication engineering from the Amrita School of Engineering, Amritapuri Campus, in 2011, and his M.E. in telecommunications and the Ph.D. in electrical communication engineering from the Indian Institute of Science Bangalore, in 2013 and 2019, respectively. He was a Post-Doctoral Fellow at the Department of Electrical and Computer Engineering, University of Toronto, from 2019 to 2021. From 2021 to 2022, he worked as an Assistant Professor at the International Institute of Information Technology Bangalore. He is currently a DST-INSPIRE Faculty Fellow at the Department of Electrical Engineering, Indian Institute of Technology Palakkad. His research interests include coding for low-latency communication, coding for distributed storage, and coded computation.

Elad Domanovitz (IEEE member) received the B.Sc. degree (cum laude), M.Sc. and Ph.D. degrees in 2005, 2011, and 2020 respectively, in Electrical Engineering from Tel Aviv University, Israel. This work was done as part of a Post-Doctoral Fellowship with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. His research interests include information theory, communication theory and statistical signal processing.

Silas L. Fong (M'15) received the B.Eng., M.Phil. and Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2005, 2007 and 2011 respectively. He is currently a Staff Engineer at Qualcomm Technologies Inc., NJ, USA. From 2011 to 2013, Dr. Fong was a Postdoctoral Fellow with the Department of Electronic Engineering at City University of Hong Kong, Hong Kong. From 2013 to 2014, he was a Postdoctoral Associate with the Department of Electrical and Computer Engineering at Cornell University, Ithaca, NY. From 2014 to 2017, he was a Research Fellow with the Department of Electrical and Computer Engineering at National University of Singapore (NUS), Singapore. From 2017 to 2019, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering at University of Toronto, Toronto, ON, Canada. His research interests include information theory and its applications to communication systems such as relay networks, wireless networks, and energy-harvesting channels.

Ashish Khisti (IEEE member) received the B.A.Sc. degree from the Engineering Science Program, University of Toronto, in 2002, and the master's and Ph.D. degrees from the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2004 and 2008, respectively. Since 2009, he has been on the Faculty in the Electrical and Computer Engineering (ECE) Department, University of Toronto, where he was an Assistant Professor from 2009 to 2015, an Associate Professor from 2015 to 2019, and is currently a Full Professor. He also holds the Canada Research Chair of information theory with the ECE Department. His current research interests include theory and applications of machine learning and communication networks. He is also interested in interdisciplinary research involving engineering and healthcare.

Wai-Tian Tan received his BS, MS, and PhD degrees in electrical engineering from Brown, Stanford, and UC Berkeley, respectively. He works in the general area of multimedia networking, systems for video communications and wireless networking. He was a principal engineer in the Enterprise Networking Business in Cisco, and worked in Hewlett Packard Laboratories, Palo Alto, before joining Cisco.

John Apostolopoulos was VP/CTO of Cisco's Intent Based Networking Group (Cisco's largest business), and also founded Cisco's Innovation Labs. Coverage included wireless (Wi-Fi 6, 5G, OpenRoaming), Internet of Things, multimedia networking, software-defined WAN, and ML/AI applied to the above. Previously, John was Distinguished Technologist, and then Lab Director for the Mobile & Immersive Experience (MIX) Lab at HP Labs. The MIX Lab conducted research on novel mobile devices and sensing, mobile client/cloud multimedia computing, immersive environments, video & audio signal processing, computer vision & graphics, multimedia networking, glasses-free 3D, wireless, and user experience design. John is an IEEE Fellow, IEEE SPS Distinguished Lecturer, named "one of the world's top 100 young innovators" by MIT Technology Review, contributed to the US Digital TV Standard (Engineering Emmy Award), and his work on media transcoding in the middle of a network while preserving end-to-end security (secure transcoding) was adopted in the JPSEC standard. He published over 100 papers, receiving 5 best paper awards, and over 100 granted US patents. John was a Consulting Associate Professor of EE at Stanford. He received his B.S., M.S., and Ph.D. from MIT.