

Streaming Erasure Codes over Multi-hop Relay Network

Elad Domanovitz, Ashish Khisti, Wai-Tian Tan, Xiaoqing Zhu, and John Apostolopoulos

Abstract

This paper studies low-latency streaming codes for the multi-hop network. The source is transmitting a sequence of messages (streaming messages) to a destination through a chain of relays where each hop is subject to packet erasures. Every source message has to be recovered perfectly at the destination within a delay constraint of T time slots. In any sliding window of $T + 1$ time slots, we assume no more than N_j erasures introduced by the j 'th hop channel. The capacity in case of a single relay (a three-node network) was derived by Fong [1], et al. While the converse derived for the three-node case can be extended to any number of nodes using a similar technique (analyzing the case where erasures on other links are consecutive), we demonstrate next that the achievable scheme, which suggested a clever symbol-wise decode and forward strategy, can not be straightforwardly extended without a loss in performance. The coding scheme for the three-node network, which was shown to achieve the upper bound, was “state-independent” (i.e., it does not depend on specific erasure pattern). While this is a very desirable property, in this paper, we suggest a “state-dependent” (i.e., a scheme which depends on specific erasure pattern) and show that it achieves the upper bound up to the size of an additional header. Since, as we show, the size of the header does not depend on the field size, the gap between the achievable rate and the upper bound decreases as the field size increases.

I. INTRODUCTION

Real-time interactive video streaming is an integral part of the day-to-day activity of many people in the world. Traditionally, most of the traffic on the internet is not sensitive to the typical delay induced by the network. However, as networks evolved, more and more people are using the network for real-time conversations, video conferencing, and on-line monitoring. According to [2], IP video traffic will account for 82 percent of traffic by 2022. Further, live video is projected to grow 15-fold to reach 17 percent of Internet video traffic by 2022.

All types of traffic are susceptible to errors, and therefore many applications use an error-correcting mechanism. One fundamental difference between real-time video streaming and other types of traffic is the (much more stringent) latency requirement each packet has to meet in order to provide a good user experience. A very common error-correcting mechanism is automatic repeat request (ARQ). Using ARQ means that the latency (in case of an error) is at least three times the one-way delay, which in many cases may violate the latency requirements for real-time interactive video streaming.

An alternative method for handling errors in the transmission is forward error correction (FEC). Using FEC has the potential to lower the recovery latency since it does not require communication between the receiver and transmitter. However, in many cases, when FEC is designed, the emphasis is on its error-correcting capabilities while ignoring latency constraints. Two commonly used codes are Low-density parity-check (LDPC) [3], [4] and digital fountain codes [5], [6]. The typical block length of these codes is very long (usually a few hundreds of symbols) hence precluding their use for real-time interactive applications.

Low-latency FEC are already implemented and have a noticeable impact on the quality of real-time interactive applications. For example, in [7], the FEC implemented in Skype is described, and it is argued that this mechanism is one of the main contributors to the success of this application. In another line of work, [8] derived the capacity of low-latency FEC (while denoting it as streaming codes) for a (deterministic) channel with bursts of erasures. This work was followed by a plurality of works [9]–[14] which extended the channel model to contain both bursts and arbitrary erasures while analyzing a sliding window model.

In [1], streaming codes for the three-node network were analyzed. As first step, the channel model used in this paper can be denoted as “deterministic arbitrary erasure channel”. In this channel model, the location of erasures can be arbitrary (does not necessarily occur in bursts). However, the number of erasures is (globally) upper bounded. Then, it was shown that the derived results also hold for a sliding window model in which only the number of erasures in any given window is upper bounded (i.e., the limit on the global number of erasures was removed).

When analyzing achievable schemes, in [1], a straightforward extension of point-to-point (P2P) codes to a setup with a relay was first described. In this extension, each hop uses a point-to-point code, and an optimization is carried on the allocation of the delay to be utilized by each code. Denoting this approach as “message-wise” decode and forward strategy (since each message is fully decoded at the relay prior to forwarding it), [1] showed that this scheme is inferior to a more sophisticated

E. Domanovitz and A. Khisti are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (email: elad.domanovitz@utoronto.ca, akhisti@ece.utoronto.ca)

W.-T. Tan, X. Zhu, and J. Apostolopoulos are with Cisco Systems, San Jose, CA 95134, USA.

The material in this paper was submitted (and accepted) in part to the 2020 IEEE International Symposium on Information Theory, Los Angeles, CAL.

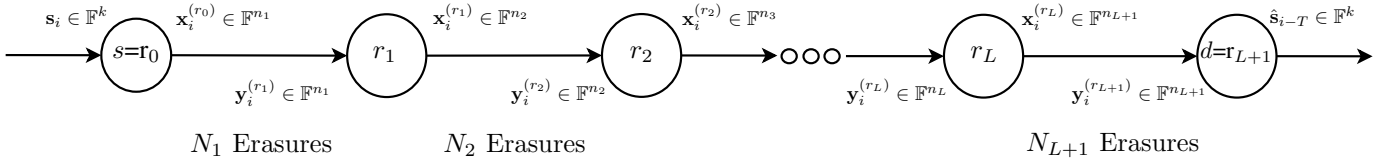


Fig. 1: Symbols generated in the L -node relay network at time i .

scheme denoted as “symbol-wise” decode and forward in which the relay forwards the recovered symbols (before the entire message can be decoded) to the destination. Further, showing that the rate achieved by “symbol-wise” decode and forward coincides with the upper bound resulted in the capacity of the three-node relay network.

Analyzing the capacity of the three-node network shows that when constraints are imposed per segment rather than globally (while meeting the same global requirements), the capacity increases. For example, as we demonstrate next, treating the network as a single-hop link with a maximum of $N = N_1 + N_2$ erasures and a total delay constraint of T symbols is worse than analyzing a three-node network where a maximum of N_1 erasures are expected in the first segment and a maximum of N_2 erasures are expected in the second segment with a total delay of T symbols. However, internet paths almost never consist of only a single relay (see, e.g., [15], [16]). Hence, designing a streaming code for a path consisting of multiple links when possible (i.e., take into account the error behavior of each link rather than aggregate across all of the links) is expected to result in improved performance guarantees.

The coding scheme described in [1], which was shown to achieve the capacity of the network, has another appealing property, which is “state-independent”, i.e., it does not depend on the specific location of erasures in the different segments. Unfortunately, as we show next, there is no straightforward extension of this scheme to a more general case (a network with more than three nodes). The scheme suggested in this paper for any number of relays is a “state-dependent” scheme, i.e., it is a scheme which reorders the symbols transmitted by each relay based on the erasure patterns that occurred in the previous links. While requiring an additional header to allow each relay to encode the received symbols transmitted to the next relay, we show that it can be easily used for any number of relays.

In this paper, we first extend the upper bound derived in [1] to the general case of a multi-hop relay network. We then describe the state-dependent scheme for the general L relay scenario and show it achieves the upper bound up to an additional overhead (a required header). We further show that the size of the header is a function of the required delay and the erasure pattern (hence it does not depend on the field size used by the code). Therefore, the gap from the upper bound decreases as the field size increases.

The rest of this paper is organized as follows. Section I-A outlines the network model of interest. Section I-B presents the formulation of streaming codes and outlines the known results for basic network models. In this Section, the problem is defined as coding over the deterministic erasure model. Section I-C presents the main results of this paper. Section I-D provides a motivating example. In this example, we show that a straightforward extension of the (state-independent) achievable scheme for a single relay results in a loss in the minimum delay that can be achieved (while maintaining the same rate) compared to the state-dependent scheme. Section II contains the proof for the upper bound on the achievable rate for a network that consists of any number of relays. Section III presents the state-dependent symbol-wise decode and forward coding scheme and contains proof on its achievable rate. Section IV provides an upper bound on the error probability when using the state-dependent symbol-wise decode and forward coding scheme when used over a channel with random (i.i.d.) erasures. Section V provides numerical results for different coding schemes used over four-node (two relay) network with random (i.i.d.) erasures. Finally, Section VI provides an extension of the presented results to the sliding window channel.

A. Network Model

A source node wants to send a sequence of messages $\{s_i\}_{i=0}^{\infty}$ to a destination node with the help of L middle nodes r_1, \dots, r_L . To ease notation we denote the source node as r_0 , and destination node as r_{L+1} . Let k be a non-negative integer, and n_1, n_2, \dots, n_{L+1} be $L+1$ natural numbers.

Each s_i is an element in \mathbb{F}^k where \mathbb{F} is some finite field. In each time slot $i \in \mathbb{Z}_+$, the source message s_i is encoded into a length- n_1 packet $\mathbf{x}_i^{(r_0)} \in \mathbb{F}^{n_1}$ to be transmitted to the first relay through the erasure channel (r_0, r_1) . The relay receives $\mathbf{y}_i^{(r_1)} \in \mathbb{F}^{n_1} \cup \{*\}$ where $\mathbf{y}_i^{(r_1)}$ equals either $\mathbf{x}_i^{(r_0)}$ or the erasure symbol “*”. In the same time slot, relay r_1 transmits $\mathbf{x}_i^{(r_1)} \in \mathbb{F}^{n_2}$ to relay r_2 through the erasure channel (r_1, r_2) . Relay r_2 receives $\mathbf{y}_i^{(r_2)} \in \mathbb{F}^{n_2} \cup \{*\}$ where $\mathbf{y}_i^{(r_2)}$ equals either $\mathbf{x}_i^{(r_1)}$ or the erasure symbol “*”. The same process continues (in the same time slot) until relay r_L transmits $\mathbf{x}_i^{(r_L)} \in \mathbb{F}^{n_{L+1}}$ to the destination r_{L+1} through the erasure channel (r_L, r_{L+1}) . To simplify the analysis we note that we assume zero propagation delay and zero processing delay for the transmission. Hence, in case no coding is applied ($n_1 = n_2 = \dots = n_{L+1} = k$) and no erasures occur, $\mathbf{y}_i^{(r_{L+1})} = s_i$. When such assumptions are relaxed, extensions to the results described in the paper can be naturally described (see, e.g. [17]).

We first assume that on the discrete timeline, each channel (r_{j-1}, r_j) introduces up to N_j arbitrary erasures respectively. The symbols generated in the L -node relay network at time i are illustrated in Figure 1.

B. Standard Definitions and Known Results

Definition 1. An $(n_1, n_2, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code consists of the following:

- 1) A sequence of source messages $\{\mathbf{s}_i\}_{i=0}^{\infty}$ where $\mathbf{s}_i \in \mathbb{F}^k$.
- 2) An encoding function $f_i^{(r_0)} : \underbrace{\mathbb{F}^k \times \dots \times \mathbb{F}^k}_{i+1 \text{ times}} \rightarrow \mathbb{F}^{n_1}$ for each $i \in \mathbb{Z}_+$, where $f_i^{(r_0)}$ is used by node r_0 at time i to encode \mathbf{s}_i according to

$$\mathbf{x}_i^{(r_0)} = f_i^{(r_0)}(\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_i).$$

- 3) A relaying function for node r_j where $j \in \{1, \dots, L\}$, $f_i^{(r_j)} : \underbrace{\mathbb{F}^{n_j} \cup \{*\} \times \dots \times \mathbb{F}^{n_j} \cup \{*\}}_{i+1 \text{ times}} \rightarrow \mathbb{F}^{n_{j+1}}$ for each $i \in \mathbb{Z}_+$, where $f_i^{(r_j)}$ is used by node r_j at time i to construct

$$\mathbf{x}_i^{(r_j)} = f_i^{(r_j)}(\mathbf{y}_0^{(r_j)}, \mathbf{y}_1^{(r_j)}, \dots, \mathbf{y}_i^{(r_j)}).$$

- 4) A decoding function $\phi_{i+T} : \underbrace{\mathbb{F}^{n_L} \cup \{*\} \times \dots \times \mathbb{F}^{n_L} \cup \{*\}}_{i+T+1 \text{ times}} \rightarrow \mathbb{F}^{n_{L+1}}$ for each $i \in \mathbb{Z}_+$ is used by node r_{L+1} at time $i+T$ to estimate \mathbf{s}_i according to

$$\hat{\mathbf{s}}_i = \phi_{i+T}(\mathbf{y}_0^{(r_{L+1})}, \mathbf{y}_1^{(r_{L+1})}, \dots, \mathbf{y}_{i+T}^{(r_{L+1})}). \quad (1)$$

Definition 2. An erasure sequence is a binary sequence denoted by $e^\infty \triangleq \{e_i\}_{i=0}^{\infty}$ where $e_i = \mathbf{1}\{\text{erasure occurs at time } i\}$.

An N -erasure sequence is an erasure sequence e^∞ that satisfies $\sum_{l=0}^{\infty} e_l = N$. Alternatively, we denote it as a N -deterministic erasure channel. The set of N -erasure sequences is denoted by Ω_N . We further denote N_1, \dots, N_{L+1} deterministic erasure network model as N_1, \dots, N_{L+1} -erasure sequences, each occur on a different channel, where for any $j \in \{1, \dots, L+1\}$, the maximal number of erasures on channel (r_{j-1}, r_j) is N_j .

Definition 3. The mapping $g_n : \mathbb{F}^n \times \{0, 1\} \rightarrow \mathbb{F}^n \cup \{*\}$ of an erasure channel is defined as

$$g_n(\mathbf{x}, e) = \begin{cases} \mathbf{x} & \text{if } e = 0, \\ * & \text{if } e = 1. \end{cases} \quad (2)$$

Denoting with $e^{j+1} \in \Omega_{N_{j+1}}$ an admissible erasure sequence in channel (r_j, r_{j+1}) , for any erasure sequence $e^{j,\infty}$ and any $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code, the following input-output relations holds for the erasure channel (r_j, r_{j+1}) for each $i \in \mathbb{Z}_+$:

$$\mathbf{y}_i^{(r_{j+1})} = g_{n_j}(\mathbf{x}_i^{(r_j)}, e_i^{j+1})$$

Definition 4. An $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code is said to be $(N_1, N_2, \dots, N_{L+1})$ -achievable if the following holds for any N_j -erasure sequence $e^{j,\infty} \in \Omega_{N_j}$ ($j \in \{1, \dots, L+1\}$), for all $i \in \mathbb{Z}_+$ and all $\mathbf{s}_i \in \mathbb{F}^k$, we have

$$\hat{\mathbf{s}}_i = \mathbf{s}_i$$

where

$$\hat{\mathbf{s}}_i = \phi_{i+T}(g_{n_{L+1}}(\mathbf{x}_0^{(r_L)}, e_0^{L+1}), \dots, g_{n_{L+1}}(\mathbf{x}_{i+T}^{(r_L)}, e_{i+T}^{L+1})) \quad (3)$$

and for previous nodes

$$\mathbf{x}_i^{(r_j)} = f_i^{(r_j)}(g_{n_j}(\mathbf{x}_0^{(r_{j-1})}, e_0^j), \dots, g_{n_j}(\mathbf{x}_i^{(r_{j-1})}, e_i^j)). \quad (4)$$

Definition 5. The rate of an $(n_1, n_2, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code is $\frac{k}{\max\{n_1, n_2, \dots, n_{L+1}\}}$.

If, for a specific j , all $N_{l \neq j} = 0$ and $N_j \neq 0$, then the L -node relay network with erasures reduces to a point-to-point packet erasure channel. It was previously shown in [9] that the maximum achievable rate of the point-to-point packet erasure channel with $N_j = N$ arbitrary erasures and delay of T symbols denoted by $C_{T,N}$ satisfies

$$C_{T,N} = \begin{cases} \frac{T-N+1}{T+1} & T \geq N \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

It was further shown that the capacity of the point-to-point channel with N arbitrary erasures and delay of T could be achieved by diagonally interleaving $(T + 1, T - N + 1)$ MDS code.

In [1], a three node relay network was analyzed, and the following theorem was shown.

Theorem 1 (Theorem 1 in [1]). *Fix any (T, N_1, N_2) . Recalling that the point-to-point capacity satisfies (5), we have*

$$\begin{aligned} C_{T, N_1, N_2} &= \min \{C_{T-N_2, N_1}, C_{T-N_1, N_2}\} \\ &= \begin{cases} \frac{T-N_1-N_2+1}{T-\min\{N_1, N_2\}+1} & T \geq N_1 + N_2 \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (6)$$

C. Main Results

In this paper we first derive an upper bound for the achievable rate in $L + 1$ -node relay network.

Theorem 2. *Assume a path with L relays. For a target overall delay of T , where the maximal number of arbitrary erasures in link (r_{j-1}, r_j) , $j \in \{1, \dots, L + 1\}$ is N_j . The achievable rate is upper bounded by*

$$R \leq \begin{cases} \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \min_j \left\{ \sum_{l=1, l \neq j}^{L+1} N_l \right\} + 1} \triangleq C_{T, N_1, \dots, N_{L+1}}^+ & T \geq \sum_{l=1}^{L+1} N_l \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

We then suggest an achievable scheme which achieves the upper bound up to a size of an overhead which is required by the scheme. Denoting

$$n_{\max} \triangleq \max_{j \in \{1, \dots, L+1\}} \left(T - \sum_{l=1, l \neq j}^{L+1} N_l + 1 \right), \quad (8)$$

we show the following Theorem.

Theorem 3. *Assume a link with L relays. For a target overall delay of T , where the maximal number of arbitrary erasures in link (r_j, r_{j+1}) , $j \in \{0, \dots, L\}$ is N_{j+1} and $T \geq \sum_{l=1}^{L+1} N_l$. When $|\mathbb{F}| \geq n_{\max}$, The following rate is achievable.*

$$R \geq \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \min_j \left\{ \sum_{l=1, l \neq j}^{L+1} N_l \right\} + 1 + \frac{n_{\max} \lceil \log(n_{\max}) \rceil}{\log(|\mathbb{F}|)}} \quad (9)$$

where n_{\max} is defined in (8).

Remark 1. *Although the deterministic erasure model is formulated in such a way that link (r_{j-1}, r_j) introduces only a finite number of erasures over the discrete timeline, the maximum coding rate remains unchanged for the following sliding window model that can introduce infinite erasures. Every message must be perfectly recovered with delay T as long as the numbers of erasures introduced by in link (r_{j-1}, r_j) in every sliding window of size $T + 1$ do not exceed N_j . This is further described in Section VI.*

We recall that for any natural numbers L and M , a systematic maximum-distance separable (MDS) $(L + M, L)$ -code is characterized by an $L \times M$ parity matrix $\mathbf{V}^{L \times M}$ where any L columns of $[\mathbf{I}_L \ \mathbf{V}^{L \times M}] \in \mathbb{F}^{L \times (L+M)}$ are independent. It is well known that a systematic MDS $(L + M, L)$ -code always exists as long as $|\mathbb{F}| \geq L + M$ [18]. To simplify notation, we sometimes denote $N_a^b = \sum_{l=a}^b N_l$. We will take all logarithms to base 2 throughout this paper. We denote the i 'th element of vector \mathbf{x} as $x[i]$, or sometime as $[x][i]$.

D. Motivating example

Consider a link with up to $N = 2$ arbitrary erasures, where the delay constraint the transmission has to meet is $T = 3$ symbols. The capacity of this link according to (5) is $C_{3,2} = 2/4$. Now, assume that in fact, this link is a three-node network ($L = 1$), where up to $N_1 = 1$ erasures occur in link (r_0, r_1) and up to $N_2 = 1$ erasures occur in link (r_1, r_2) , where transmission has to be decoded with the same overall delay of $T = 3$ symbols. This is depicted in Figure 2. The capacity of this link according to (6) is $C_{3,1,1} = 2/3$, which is better than the point-to-point link.

We start by recalling the coding scheme that was shown in [1] to achieve capacity. The example we show next is the same as provided in [1] for the scenario described above (in which the total delay $T = 3$ and the maximal number of erasures in each channel is $N_1 = N_2 = 1$).

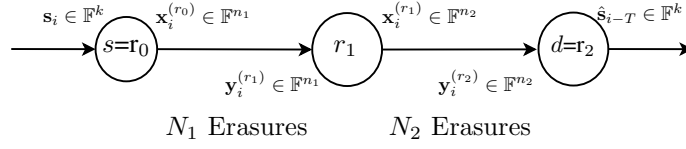


Fig. 2: A three-node relay network.

Suppose node s transmits two bits a_i and b_i at each discrete time $i \geq 0$ to node d with delay 3. For each time i , node s transmits the three-symbol packet $\mathbf{x}_i^{(r_0)} = [a_i \ b_i \ a_{i-2} + b_{i-1}]$ according to Table I where $a_j = b_j = 0$ for any $j < 0$ by convention, and the symbols highlighted in the same color are generated by the same block code.

Since channel (s, r) introduces at most $N_1 = 1$ erasure, each a_i and each b_i can be perfectly recovered by node r by time $i + 2$ and time $i + 1$ respectively. Therefore, at each time i , node r should have recovered a_{i-2} and b_{i-1} perfectly with delays 2 and 1 respectively, and it will re-encode them into another three-symbol packet $\mathbf{x}_{i+1}^{(r_1)} = [b_{i-1} \ a_{i-2} \ b_{i-3} + a_{i-3}]$. This is depicted in Table I. Since b_{i-3} , a_{i-3} and $b_{i-3} + a_{i-3}$ are transmitted by node r at time $i - 2$, $i - 1$ and i respectively, it follows from the fact $N_2 = 1$ that node d can recover a_{i-3} and b_{i-3} by time i . Consequently, this symbol-wise DF strategy achieves a rate of $2/3$.

Time	$i - 1$	i	$i + 1$	$i + 2$	$i + 3$	$i + 4$	$i + 5$
a_i	a_{i-1}	a_i	a_{i+1}	a_{i+2}	a_{i+3}	a_{i+4}	a_{i+5}
b_i	b_{i-1}	b_i	b_{i+1}	b_{i+2}	b_{i+3}	b_{i+4}	b_{i+5}
$a_{i-2} + b_{i-1}$	$a_{i-3} + b_{i-2}$	$a_{i-2} + b_{i-1}$	$a_{i-1} + b_i$	$a_i + b_{i+1}$	$a_{i+1} + b_{i+2}$	$a_{i+2} + b_{i+3}$	$a_{i+3} + b_{i+4}$

(a) Symbols transmitted by the source node s .

Time	$i - 1$	i	$i + 1$	$i + 2$	$i + 3$	$i + 4$	$i + 5$
b_{i-1}	b_{i-2}	b_{i-1}	b_i	b_{i+1}	b_{i+2}	b_{i+3}	b_{i+4}
a_{i-2}	a_{i-3}	a_{i-2}	a_{i-1}	a_i	a_{i+1}	a_{i+2}	a_{i+3}
$a_{i-3} + b_{i-3}$	$a_{i-4} + b_{i-4}$	$a_{i-3} + b_{i-3}$	$a_{i-2} + b_{i-2}$	$a_{i-1} + b_{i-1}$	$a_i + b_i$	$a_{i+1} + b_{i+1}$	$a_{i+2} + b_{i+3}$

(b) Symbols transmitted by relay node r .

Time	$i - 1$	i	$i + 1$	$i + 2$	$i + 3$	$i + 4$	$i + 5$
a_{i-3}	a_{i-4}	a_{i-3}	a_{i-2}	a_{i-1}	a_i	a_{i+1}	a_{i+2}
b_{i-3}	b_{i-4}	b_{i-3}	b_{i-2}	b_{i-1}	b_i	b_{i+1}	b_{i+2}

(c) Estimates constructed by the destination node d .TABLE I: A stateless symbol-wise DF strategy for a single relay. Symbols belong to the same code used by the source (s) and symbols belong to the same code used by the relay (r) are marked with a frame with the same style.

An important feature of this code is the fact that it is a state-less code, i.e., the structure of the code does not depend on the specific erasure pattern at any of the segments. However, if another relay is to be considered (i.e., the destination is now replaced with relay r_2), assuming up to $N_3 = 1$ erasures in link (r_2, d) , the minimal delay required to support rate $2/3$ is $T = 5$. The network of interest is depicted in Figure 3.

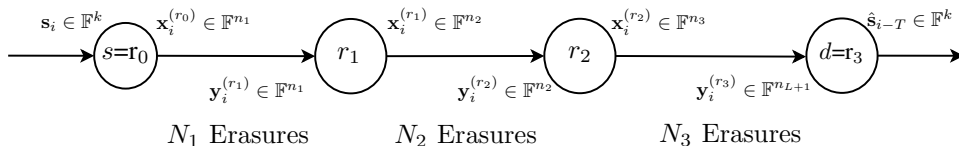


Fig. 3: A four-node relay network.

Time	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$	$i+5$
a_{i-3}	a_{i-4}	a_{i-3}	a_{i-2}	a_{i-1}	a_i	a_{i+1}	a_{i+2}
b_{i-3}	b_{i-4}	b_{i-3}	b_{i-2}	b_{i-1}	b_i	b_{i+1}	b_{i+2}
$a_{i-5} + b_{i-4}$	$a_{i-6} + b_{i-5}$	$a_{i-5} + b_{i-4}$	$a_{i-4} + b_{i-3}$	$a_{i-3} + b_{i-2}$	$a_{i-2} + b_{i-1}$	$a_{i-1} + b_i$	$a_i + b_{i+1}$

TABLE II: Symbols transmitted by node r_2 when trying to extend the stateless symbol-wise DF strategy. Symbols belong to the same code are denoted with frame with the same style.

This can be seen since basically using the coding scheme in Table I, relay r_2 can be viewed as source s with delay of $T = 3$ packets (essentially, the delay of different symbols is “reset”). Due to causality, relay r_2 can only use the coding scheme of the sender, depicted in Table II. It can be viewed that when $N_3 = 1$, symbol a_i is guaranteed to be recovered only at time $i + 5$.

In this paper, we suggest a state-dependent scheme, i.e., a scheme in which the order of symbols in each block code used (and thus the order in each diagonal) is set according to the erasure pattern in the previous link. The order of the symbols is transmitted to the receiver to allow decoding. Hence, additional overhead is required. We first show an example of the suggested scheme to the three-node network and then show how to extend it to a four-node network.

In the proposed scheme, the source r_0 uses the same code suggested in [1], i.e., a $(3, 2)$ MDS code that can recover any arbitrary single erasure in a delay of two symbols combined with diagonal interleaving (i.e., the code is applied on the diagonals). For each time i , node s transmits the three-symbol packet (symbols belong to the same block code are marked with the same color).

Time	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$
Header	123	123	123	123	123	123
a_i	a_{i-1}	a_i	a_{i+1}	a_{i+2}	a_{i+3}	a_{i+4}
b_i	b_{i-1}	b_i	b_{i+1}	b_{i+2}	b_{i+3}	b_{i+4}
$a_{i-2} + b_{i-1}$	$a_{i-3} + b_{i-2}$	$a_{i-2} + b_{i-1}$	$a_{i-1} + b_i$	$a_i + b_{i+1}$	$a_{i+1} + b_{i+2}$	$a_{i+2} + b_{i+3}$

TABLE III: Transmission of the Source in channel (r_0, r_1) . Symbols belong to the same code are marked with the same color.

When there are no erasures, relay r_1 uses the same code as the source r_0 while delaying it by one symbol, i.e. relay r_1 sends the following three-symbol packet $\mathbf{x}_{i+1}^{(r_1)} = [a_i, b_i, a_{i-2} + b_{i-1}]$.

If an erasure occurred, the relay would send any available symbols (per diagonal) in the received order until it can decode the information symbols from this block code. Then, the erased symbols will be sent. For example, assuming that a symbol at time i is erased in link (r_0, r_1) , relay r_1 will send $\mathbf{x}_{i+1}^{(r_1)} = [b_{i+1}, b_i, a_{i-2} + b_{i-1}]$ and $\mathbf{x}_{i+2}^{(r_1)} = [a_{i+1}, a_i, a_{i-1} + b_i]$ as depicted in Table IV below. Again, symbols belong to the same block code are marked with the same color. Further, headers which are different than the ones used by r_0 are marked with a frame.

Time	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$
Header	123	123	223	113	123	123
	a_{i-2}	a_{i-1}	b_{i+1}	a_{i+1}	a_{i+2}	a_{i+3}
	b_{i-2}	b_{i-1}	b_i	a_i	b_{i+2}	b_{i+3}
	$a_{i-4} + b_{i-3}$	$a_{i-3} + b_{i-2}$	$a_{i-2} + b_{i-1}$	$a_{i-1} + b_i$	$a_i + b_{i+1}$	$a_{i+1} + b_{i+2}$

TABLE IV: Transmission of relay r_1 in channel (r_1, r_2) , given that symbol i was erased when transmitted in link (r_0, r_1) . Symbols belong to the same code are marked with the same color.

We note that the erasure in time i in link (r_0, r_1) caused a change in the order of the symbols in packets $i+1$ and $i+2$. Denoting the order of symbols sent from r_0 in each code (alternatively on each diagonal) as $[1, 2, 3]$, in this example, the header of each packet is composed of the location of the symbols from each block code (the order of the symbols in each diagonal) as they would appear in the code used by r_0 .

We show next that the code used by each relay on each diagonal can be viewed as a punctured code of a code that is used by the “bottleneck” relay, i.e., the relay with the smallest rate. Hence, we note that it can be viewed as if the suggested coding scheme only changes the order of symbols per diagonal (taken from the code with the lowest rate), and it does not add or remove symbols. Therefore, using a single index per symbol suffices to allow recovery at each destination.

In our for example, as can be seen in Tables IV, at time i relay r_1 sends $\mathbf{x}_i^{(r_1)} = [a_{i-1} \ b_{i-1} \ a_{i-3} + b_{i-2}]$ with header “123” indicating that the first symbol is the first symbol (marked with underline) in the code $[a_{i-1} \ b_i \ a_{i-1} + b_i]$, the second symbol is the second symbol in the code $[a_{i-2} \ b_{i-1} \ a_{i-2} + b_{i-1}]$ and the third symbol is the code $[a_{i-3} \ b_{i-2} \ a_{i-3} + b_{i-2}]$.

However, following the erasure occurred at link (r_0, r_1) at time i , relay r_1 can not send symbol a_i at time $i + 1$ as planned (as if there was no erasure). However, it can send b_{i+1} which was not erased. With respect to the second symbol, we note that since a_{i-1} and $a_{i-1} + b_i$ were received, b_i can be recovered and used. Similarly, since a_{i-2} and b_{i-1} were received, symbol $a_{i-2} + b_{i-1}$ can be generated and used. Thus, at time $i + 1$ the relay can send $\mathbf{x}_{i+1}^{(r_1)} = [b_{i+1} \ b_i \ a_{i-2} + b_{i-1}]$. To indicate the change in order of the symbols used in the first code, the header is changed to “223” which should be indicating that the first symbol is the second symbol in the code associated with this diagonal.

At time $i + 2$, the relay can recover and send a_i . Hence, it sends $\mathbf{x}_{i+2}^{(r_1)} = [a_{i+1} \ a_i \ a_{i-1} + b_i]$ with header “113” indicating now that the second symbol is the first symbol in the code associated with this diagonal. It can be easily verified that the destination can recover the original data at a delay of $T = 3$ (assuming any single arbitrary erasure in the link between the relay and destination).

This concept can be applied to additional relays if they exist. For example consider four-node network ($L = 2$). The transmission scheme on the next relay r_2 (in this specific example), is the same as the transmission scheme of the first relay, i.e., in case there is no erasure, transmit (on each diagonal) the symbols in the same order as received, delayed by one symbol (i.e., a total delay of two symbols from the sender). If an erasure occurred before r_2 decoded the information symbols, transmit the available symbols (again, it is guaranteed that there will be enough symbols). When the information symbols can be decoded, transmit the erased symbols.

For example, in case the symbol transmitted from relay r_1 to r_2 at time $i + 2$ is erased, the suggested transmission scheme of relay r_2 is given in Table V below. Again, symbols belong to the same block code are marked with the same color. Further, headers that are different than the ones used by the relay r_1 are marked with a frame.

Time	$i - 1$	i	$i + 1$	$i + 2$	$i + 3$	$i + 4$
Header	123	123	123	223	213	113
	a_{i-3}	a_{i-2}	a_{i-1}	b_{i+1}	b_{i+2}	a_{i+2}
	b_{i-3}	b_{i-2}	b_{i-1}	b_i	a_i	a_{i+1}
	$a_{i-5} + b_{i-4}$	$a_{i-4} + b_{i-3}$	$a_{i-3} + b_{i-2}$	$a_{i-2} + b_{i-1}$	$a_{i-1} + b_i$	$a_i + b_{i+1}$

TABLE V: Transmission of relay r_2 in channel (r_2, r_3) , given that symbol $i + 2$ was erased when transmitted in link (r_1, r_2) . Symbols belong to the same code are marked with the same color.

We note that the erasure in time $i + 2$ in link (r_1, r_2) caused a change in order of the symbols in packets $i + 3$ and $i + 4$ (the order of symbols in time $i + 2$ is not the original order, yet it is the same as was transmitted from r_1 at time $i + 1$).

Since the same $(3, 2)$ code is used (with a different order of symbols which is communicated to the receiver), it can be easily verified that each packet can be recovered up to delay of $T = 4$ symbols for any arbitrary erasure in the link between the relay and the destination.

In this example, the maximal size of the header is three numbers, each taken from $\{1, 2, 3\}$. Hence, its maximal size of the header is $3 \log(3)$ bits. Since the block code used in each link transmits two bits using three bits in every channel use, we conclude that the scheme achieves a rate of

$$R = \frac{2}{3 + 3 \lceil \log(3) \rceil}. \quad (10)$$

We show next that this idea can be extended to transmitting symbols taken from any field \mathbb{F} thus in general, the achievable rate when $T = 4$ and $N_1 = N_2 = N_3 = 1$ is

$$\begin{aligned} R &= \frac{2 \cdot \log(|\mathbb{F}|)}{3 \cdot \log(|\mathbb{F}|) + 3 \lceil \log(3) \rceil} \\ &= \frac{2}{3 + \frac{3 \lceil \log(3) \rceil}{\log(|\mathbb{F}|)}}, \end{aligned} \quad (11)$$

which approaches $2/3$ as the field size increases. As we further show next, the upper bound for this scenario is indeed $2/3$.

II. PROOF OF THE UPPER BOUND

Fix any (N_1, \dots, N_{L+1}, T) . Suppose we are given an (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{j+1}, k, T)_{\mathbb{F}}$ -streaming code for some n_1, \dots, n_{j+1}, k and \mathbb{F} . Our goal is to show that

$$\frac{k}{\max\{n_1, \dots, n_{L+1}\}} \leq \min_{j \in \{1, 2, \dots, L+1\}} \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \sum_{l=1, l \neq j}^{L+1} N_l + 1}. \quad (12)$$

To this end, we let $\{\mathbf{s}_i\}_{i \in \mathbb{Z}_+}$ be i.i.d. random variables where \mathbf{s}_0 is uniform on \mathbb{F}^k . Since the $(n_1, \dots, n_{j+1}, k, T)_{\mathbb{F}}$ -streaming code is (N_1, \dots, N_{L+1}) -achievable, it follows from Definition 4 that

$$H(\mathbf{s}_i \mid \mathbf{y}_0^{(L+1)}, \mathbf{y}_1^{(L+1)}, \dots, \mathbf{y}_{i+T}^{(L+1)}) = 0 \quad (13)$$

for any $i \in \mathbb{Z}_+$ and any $e^{j, \infty} \in \Omega_{N_j}$. Consider the two cases.

Case $T < \sum_{l=1}^{L+1} N_l$:

Let $e^{j, \infty} \in \Omega_{N_j}$ be the error sequence on link (r_{j-1}, r_j) where $j \in \{1, 2, \dots, L+1\}$ such that

$$e_i^{j, \infty} = \begin{cases} 1 & \text{if } \sum_{l=1}^{j-1} N_l \leq i \leq \sum_{l=1}^j N_l - 1 \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

We note that (14) means that $\mathbf{y}_{N_1}^{(r_1)}$ is the first packet which can be used to recover \mathbf{s}_0 at r_1 . Further, $\mathbf{y}_{(N_1+N_2)}^{(r_2)}$ is the first packet which can be used to recover \mathbf{s}_0 at r_2 . Continuing the transmission across all other relays, it follows that $\mathbf{y}_{\sum_{l=1}^{L+1} N_l}^{(r_{L+1})}$ is the first packet which can be used to recover \mathbf{s}_0 . Since $T < \sum_{l=1}^{L+1} N_l$ it follows the delay constraint can not be met.

Hence, due to (14) and Definition 1, we have

$$I(\mathbf{s}_0; \mathbf{y}_0^{(L+1)}, \mathbf{y}_1^{(L+1)}, \dots, \mathbf{y}_T^{(L+1)}) = 0. \quad (15)$$

Combining (13), (15) and the assumption that $T < \sum_{l=1}^{L+1} N_l$, we obtain $H(\mathbf{s}_0) = 0$. Since \mathbf{s}_0 consists of k uniform random variables in \mathbb{F} , the only possible value of k is zero, which implies

$$\frac{k}{\max\{n_1, \dots, n_{L+1}\}} = 0. \quad (16)$$

Case $T \geq \sum_{l=1}^{L+1} N_l$:

The proof follows the footsteps of [1]. We start by generalizing the arguments given in [1] for the first and second segments to the first and last segments in our case. Then we show how similar techniques can be used to derive a constraint on the code to be used in an intermediate segment.

First Segment (link (r_0, r_1)):

First we note that for every $i \in \mathbb{Z}_+$, message \mathbf{s}_i has to be perfectly recovered by node r_1 by time $i + T - \sum_{l=2}^{L+1} N_l$ given that $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{i-1}$ have been correctly decoded by node r_1 , or otherwise a length N_2 burst erasure from time $i + T - \sum_{l=2}^{L+1} N_l + 1$ to $i + T - \sum_{l=3}^{L+1} N_l$ introduced on channel (r_1, r_2) followed by a length N_3 burst erasure from time $i + T - \sum_{l=3}^{L+1} N_l + 1$ to $i + T - \sum_{l=4}^{L+1} N_l$ introduced on channel (r_2, r_3) and so on until a length N_{L+1} burst erasure from time $i + T - N_{L+1} + 1$ to $i + T$ would result in a decoding failure for node r_1 , node r_2 and all the nodes up to the destination r_{L+1} .

Recalling that $N_2^{L+1} = \sum_{l=2}^{L+1} N_l$ it then follows that

$$H(\mathbf{s}_i \mid \{\mathbf{x}_i^{(r_0)}, \mathbf{x}_{i+1}^{(r_0)}, \dots, \mathbf{x}_{i+T-N_2^{L+1}}^{(r_0)}\} \setminus \{\mathbf{x}_{\theta_1}, \dots, \mathbf{x}_{\theta_{N_1}}\}, \mathbf{s}_0, \dots, \mathbf{s}_{i-1}) = 0 \quad (17)$$

for any $i \in \mathbb{Z}_+$ and N_1 non-negative integers denoted by $\theta_1, \dots, \theta_{N_1}$. We note that the following holds (by assuming, for example, the last N_1 symbols in every window of $T - N_2^{L+1}$ symbols starting time $i = 0$ are erased):

$$\begin{aligned} & H(\mathbf{s}_0 \mid \{\mathbf{x}_0^{(r_0)}, \mathbf{x}_1^{(r_0)}, \dots, \mathbf{x}_{T-N_2^{L+1}-N_1}^{(r_0)}\}) = 0 \\ & H(\mathbf{s}_1 \mid \{\mathbf{x}_1^{(r_0)}, \mathbf{x}_2^{(r_0)}, \dots, \mathbf{x}_{T-N_2^{L+1}-N_1}^{(r_0)}, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)}^{(r_0)}\}, \mathbf{s}_0) = 0 \\ & \quad \vdots \\ & H(\mathbf{s}_{T-N_2^{L+1}-N_1} \mid \{\mathbf{x}_{T-N_2^{L+1}-N_1}^{(r_0)}, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)+1}^{(r_0)}, \dots, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)+T-N_2^{L+1}-N_1-1}^{(r_0)}\}, \mathbf{s}_0, \dots, \mathbf{s}_{T-N_2^{L+1}-N_1-1}) \\ & = 0 \\ & H(\mathbf{s}_{T-N_2^{L+1}-N_1+1} \mid \{\mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \dots, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)+T-N_2^{L+1}-N_1}^{(r_0)}\}, \mathbf{s}_0, \dots, \mathbf{s}_{T-N_2^{L+1}-N_1}) = 0 \\ & H(\mathbf{s}_{T-N_2^{L+1}-N_1+2} \mid \{\mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \dots, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)+T-N_2^{L+1}-N_1}^{(r_0)}\}, \mathbf{s}_0, \dots, \mathbf{s}_{T-N_2^{L+1}-N_1+1}) = 0 \end{aligned}$$

$$\begin{aligned}
& \vdots \\
& H\left(\mathbf{s}_{T-N_2^{L+1}-1} \mid \left\{ \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \dots, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)+T-N_2^{L+1}-N_1}^{(r_0)} \right\}, \mathbf{s}_0, \dots, \mathbf{s}_{T-N_2^{L+1}-2}\right) = 0 \\
& H\left(\mathbf{s}_{T-N_2^{L+1}} \mid \left\{ \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \dots, \mathbf{x}_{1 \cdot (T-N_2^{L+1}+1)+T-N_2^{L+1}-N_1}^{(r_0)} \right\}, \mathbf{s}_0, \dots, \mathbf{s}_{T-N_2^{L+1}-1}\right) = 0 \\
& \vdots
\end{aligned} \tag{18}$$

Using the chain rule we have the following for each $j \in \mathbb{N}$

$$H\left(\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{T-N_2^{L+1}+(j-1)(T-N_2^{L+1}+1)} \mid \left\{ \mathbf{x}_{m \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \mathbf{x}_{1+m \cdot (T-N_2^{L+1}+1)}^{(r_0)}, \dots, \mathbf{x}_{T-N_1-N_2^{L+1}+m \cdot (T-N_2^{L+1}+1)}^{(r_0)} \right\}_{m=0}^j\right) = 0. \tag{19}$$

Alternatively we note that for all $q \in \mathbb{Z}_+$,

$$\begin{aligned}
& \left| \{q, q+1, \dots, q+T-N_2^{L+1}\} \cap \right. \\
& \left. \{m \cdot (T-N_2^{L+1}+1), 1+m \cdot (T-N_2^{L+1}+1), \dots, T-N_1-N_2^{L+1}+m \cdot (T-N_2^{L+1}+1)\}_{m=0}^j \right| \\
& = T-N_1-N_2^{L+1}+1.
\end{aligned} \tag{20}$$

Hence, (19) also follows from (17), (20) and the chain rule.

Therefore, following the arguments in [1] which we recall here and in Appendix A, we note that the (19) involves $j(T-N_2^{L+1}+1)$ source messages and $(j+1)(T-N_1^{L+1}+1)$ source packets. Therefore, the (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{L+1}k, T)_{\mathbb{F}}$ -streaming code restricted to channel (r_0, r_1) can be viewed as a point-to-point streaming code with rate k/n_1 and delay $T-N_2^{L+1}$ which can correct any N_1 erasures. In particular the point-to-point code can correct the periodic erasure sequence \tilde{e}^∞ depicted in Figure 4, which is formally defined as

$$\tilde{e}_i = \begin{cases} 0 & \text{if } 0 \leq i \bmod (T-N_2^{L+1}+1) \leq T-N_1^{L+1} \\ 1 & \text{otherwise} \end{cases} \tag{21}$$

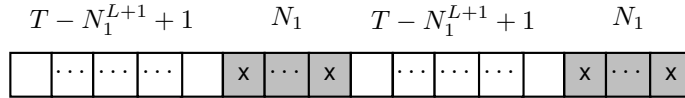


Fig. 4: A periodic erasure sequence with period $T-N_2^{L+1}+1$.

By standard arguments which are rigorously elaborated in Appendix A, we conclude that

$$\begin{aligned}
\frac{k}{n_1} & \leq \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \sum_{l=2}^{L+1} N_l + 1} \\
& = C_{T-\sum_{l=2}^{L+1} N_l, N_1}.
\end{aligned} \tag{22}$$

Last Segment (link (r_L, r_{L+1})):

In addition, for every $i \in \mathbb{Z}_+$, message \mathbf{s}_i has to be perfectly recovered from

$$\left\{ \mathbf{y}_{i+N_1^L}^{(r_{L+1})}, \mathbf{y}_{i+N_1^L+1}^{(r_{L+1})}, \dots, \mathbf{y}_{i+T}^{(r_{L+1})} \right\} \tag{23}$$

by node r_{L+1} given that $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{i-1}$ have been correctly decoded by node r_{L+1} , or otherwise a length N_1 burst erasure from time i to $i+N_1-1$ induced on channel (r_0, r_1) followed by a length N_2 burst erasure from time $i+N_1$ to $i+N_1+N_2-1$ induced on channel (r_1, r_2) and so on until a length N_L burst erasure from time $i+\sum_{l=1}^{L-1} N_l$ to $i+\sum_{l=1}^L N_l-1$ induced on channel (r_{L-1}, r_L) would result in a decoding failure for node r_{L+1} .

It follows that

$$H\left(\mathbf{s}_i \mid \left\{ \mathbf{x}_{i+N_1^L}^{(r_L)}, \mathbf{x}_{i+N_1^L+1}^{(r_L)}, \dots, \mathbf{x}_{i+T}^{(r_L)} \right\} \setminus \left\{ \mathbf{x}_{\theta_1}, \dots, \mathbf{x}_{\theta_{N_{L+1}}} \right\}, \mathbf{s}_0, \dots, \mathbf{s}_{i-1}\right) = 0 \tag{24}$$

for any $i \in \mathbb{Z}_+$ and N_{L+1} non-negative integers denoted by $\theta_1, \dots, \theta_{N_{L+1}}$. We note that for all $q \in \mathbb{Z}_+$,

$$\begin{aligned}
& \left| \{q+N_1^L, q+N_1^L+1, \dots, q+T\} \cap \right. \\
& \left. \{N_1^L+N_{L+1}+m \cdot (T-N_1^L+1), N_1^L+N_{L+1}+1+m \cdot (T-N_1^L+1), \dots, T+m \cdot (T-N_1^L+1)\}_{m=0}^j \right|
\end{aligned}$$

$$= T - N_1^L - N_{L+1} + 1 \quad (25)$$

Using (24), (25) and the chain rule, we have

$$H \left(\mathbf{s}_0, \dots, \mathbf{s}_{T-N_1^L+(j-1)(T-N_1^L+1)} \mid \left\{ \mathbf{x}_{N_1^L+N_{L+1}+m \cdot (T-N_1^L+1)}^{(r_L)}, \mathbf{x}_{N_1^L+N_{L+1}+1+m \cdot (T-N_1^L+1)}^{(r_L)}, \dots, \mathbf{x}_{T+m \cdot (T-N_1^L+1)}^{(r_L)} \right\}_{m=0}^j \right) = 0. \quad (26)$$

Therefore, following the arguments in [1] which we recall here and in Appendix A, we note that the (26) involves $j(T-N_1^L+1)$ source messages and $(j+1)(T-N_1^{L+1}+1)$ source packets. Therefore, the (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{L+1}k, T)_{\mathbb{F}}$ -streaming code restricted to channel (r_L, r_{L+1}) can be viewed as a point-to-point streaming code with rate k/n_{L+1} and delay $T - N_1^L$ which can correct any N_{L+1} erasures. In particular the point-to-point can correct the periodic erasure sequence \hat{e}^∞ depicted in Figure 5, which is formally defined as

$$\tilde{e}_i = \begin{cases} 0 & \text{if } 0 \leq i \bmod (T - N_1^L + 1) \leq T - N_1^{L+1} \\ 1 & \text{otherwise} \end{cases} \quad (27)$$

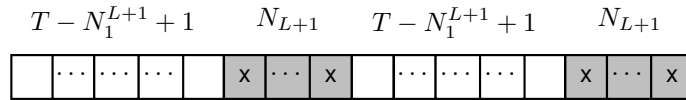


Fig. 5: A periodic erasure sequence with period $T - N_1^L + 1$.

By standard arguments which are rigorously elaborated in Appendix A, we conclude that

$$\begin{aligned} \frac{k}{n_{L+1}} &\leq \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \sum_{l=1}^L N_l + 1} \\ &= C_{T - \sum_{l=1}^L N_l, N_{L+1}}. \end{aligned} \quad (28)$$

The j 'th Segment (link (r_{j-1}, r_j)):

Now, when considering a j 'th segment (the channel (r_{j-1}, r_j)), we show again that any (N_1, \dots, N_{L+1}) achievable $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ code restricted to the channel (r_{j-1}, r_j) can be viewed as a point-to-point code which should handle any N_j erasures with a delay which we define next.

Combining the arguments given above we note that first, for every $i \in \mathbb{Z}_+$, message \mathbf{s}_i has to be perfectly recovered by node r_j by time $i + T - \sum_{l=j+1}^{L+1} N_l$ given that $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{i-1}$ have been correctly decoded by node r_j , or otherwise a length N_{j+1} burst from time $i + T - \sum_{l=j+1}^{L+1} N_l + 1$ to $i + T - \sum_{l=j+2}^{L+1} N_l$ introduced on channel (r_j, r_{j+1}) , followed by a length N_{j+2} burst from time $i + T - \sum_{l=j+2}^{L+1} N_l + 1$ to $i + T - \sum_{l=j+3}^{L+1} N_l$ introduced on channel (r_{j+1}, r_{j+2}) and so on, up to a length N_{L+1} burst from time $i - N_{L+1} + 1$ to $i + T$ introduced on channel (r_L, r_{L+1}) would result in a decoding failure for node r_j and all the nodes up to the destination r_{L+1} .

Further, in addition, for every $i \in \mathbb{Z}_+$, message \mathbf{s}_i has to be perfectly recovered from

$$\left(\mathbf{y}_{i+\sum_{l=1}^{j-1} N_l}^{(r_j)}, \mathbf{y}_{i+\sum_{l=1}^{j-1} N_l+1}^{(r_j)}, \dots, \mathbf{y}_{i+T-\sum_{l=j+1}^{L+1} N_l}^{(r_j)} \right) \quad (29)$$

by node r_j given that $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{i-1}$ have been correctly decoded by node r_j , or otherwise a length N_1 burst erasure from time i to $i + N_1 - 1$ induced on channel (r_0, r_1) followed by a length N_2 burst erasure from time $i + N_1$ to $i + N_1 + N_2 - 1$ induced on channel (r_1, r_2) and so on (up to a burst erasure N_{j-1} from time $i + \sum_{l=1}^{j-2} N_l$ to $i + \sum_{l=1}^{j-1} N_l - 1$ induced on channel (r_{j-2}, r_{j-1})) would result in a decoding failure for node r_j . These constraints are depicted in Figure 7.

Recalling that $N_1^{j-1} = \sum_{l=1}^{j-1} N_l$, and $N_{j+1}^{L+1} = \sum_{l=j+1}^{L+1} N_l$ it follows that

$$H \left(\mathbf{s}_i \mid \left\{ \mathbf{x}_{i+N_1^{j-1}}^{(r_{j-1})}, \mathbf{x}_{i+N_1^{j-1}+1}^{(r_{j-1})}, \dots, \mathbf{x}_{i+T-N_{j+1}^{L+1}}^{(r_{j-1})} \right\} \setminus \left\{ \mathbf{x}_{\theta_1}, \dots, \mathbf{x}_{\theta_{N_j}} \right\}, \mathbf{s}_0, \dots, \mathbf{s}_{i-1} \right) = 0 \quad (30)$$

for any $i \in \mathbb{Z}_+$ and N_j non-negative integers denoted by $\theta_1, \dots, \theta_{N_j}$. We note that for all $q \in \mathbb{Z}_+$,

$$\begin{aligned} &\left| \left\{ q + N_1^{j-1}, q + N_1^{j-1} + 1, \dots, q + T - N_{j+1}^{L+1} \right\} \cap \left\{ N_1^{j-1} + N_j + m \cdot (T - N_1^{j-1} - N_{j+1}^{L+1} + 1), \right. \right. \\ &\quad \left. \left. N_1^{j-1} + N_j + 1 + m \cdot (T - N_1^{j-1} - N_{j+1}^{L+1} + 1), \dots, T - N_{j+1}^{L+1} + m \cdot (T - N_1^{j-1} - N_{j+1}^{L+1} + 1) \right\}_{m=0}^j \right| \\ &= T - N_1^{j-1} - N_j - N_{j+1}^{L+1} + 1 \end{aligned} \quad (31)$$

Using (30), (31) and the chain rule, we have

$$H \left(\mathbf{s}_0, \dots, \mathbf{s}_{T-N_1^{j-1}-N_{j+1}^{L+1}+(j-1)(T-N_1^{j-1}-N_{j+1}^{L+1}+1)} \right. \\ \left. \left\{ \mathbf{x}_{N_1^{j-1}+N_j+m \cdot (T-N_1^{j-1}-N_{j+1}^{L+1}+1)}^{(r_L)}, \mathbf{x}_{N_1^{j-1}+N_j+1+m \cdot (T-N_1^{j-1}-N_{j+1}^{L+1}+1)}^{(r_L)}, \dots, \mathbf{x}_{T-N_{j+1}^{L+1}+m \cdot (T-N_1^{j-1}-N_{j+1}^{L+1}+1)}^{(r_L)} \right\}_{m=0}^j \right) \\ = 0. \quad (32)$$

Therefore, following the arguments in [1] which we recall here and in Appendix A, we note that the (32) involves $j(T - N_1^{j-1} - N_{j+1}^{L+1} + 1)$ source messages and $(j+1)(T - N_1^{L+1} + 1)$ source packets. Therefore, the (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{L+1}k, T)_{\mathbb{F}}$ -streaming code restricted to channel (r_{j-1}, r_j) can be viewed as a point-to-point streaming code with rate k/n_j and delay $T - N_1^{j-1} - N_{j+1}^{L+1}$ which can correct any N_j erasures. In particular the point-to-point can correct the periodic erasure sequence \tilde{e}^∞ depicted in Figure 6, which is formally defined as

$$\tilde{e}_i = \begin{cases} 0 & \text{if } 0 \leq i \bmod (T - N_1^{j-1} - N_{j+1}^{L+1} + 1) \leq T - N_1^{L+1} \\ 1 & \text{otherwise} \end{cases} \quad (33)$$

By standard arguments which are rigorously elaborated in Appendix A, we conclude that

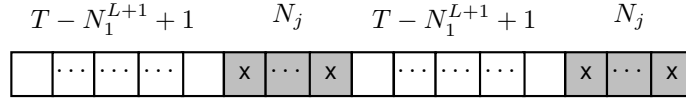


Fig. 6: A periodic erasure sequence with period $T - N_1^{j-1} - N_{j+1}^{L+1} + 1$.

$$\frac{k}{n_j} \leq \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \sum_{l=1, l \neq j}^{L+1} N_l + 1} \\ = C_{T - \sum_{l=1, l \neq j}^{L+1} N_l, N_j}. \quad (34)$$

Therefore we have

$$R \leq \frac{k}{\max_{j \in \{1, 2, \dots, L+1\}} \{n_j\}} \\ = \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \min_j \left\{ \sum_{l=1, l \neq j}^{L+1} N_l \right\} + 1} \\ = \min_j C_{T - \sum_{l=1, l \neq j}^{L+1} N_l, N_j} \\ = C_{T, N_1, \dots, N_{L+1}}^+. \quad (35)$$

III. STATE-DEPENDENT SYMBOL-WISE DF

As mentioned above, the achievable scheme we analyze is a state-dependent symbol-wise DF scheme. This scheme is composed from a block code combined with diagonal interleaving. We start with some basic definitions of point-to-point block codes which would be the basis for this scheme.

Definition 6. A point-to-point $(n, k, T)_{\mathbb{F}}$ -block code consists of the following

- 1) A sequence of k symbols $\{u[l]\}_{l=0}^{k-1}$ where $u[l] \in \mathbb{F}$.
- 2) A generator matrix $\mathbf{G} \in \mathbb{F}^{k \times n}$. The source codeword is generated according to

$$[x[0] \ x[1] \ \dots \ x[n-1]] = [u[0] \ u[1] \ \dots \ u[k-1]] \mathbf{G} \quad (36)$$

- 3) A decoding function $\varphi_{l+T} : \mathbb{F} \cup \{*\} \times \dots \times \mathbb{F} \cup \{*\} \rightarrow \mathbb{F}$ for each $l \in \{0, 1, \dots, k-1\}$, where φ_{l+T} is used by the destination at time $\min\{l+T, n-1\}$ to estimate $u[l]$ according to

$$\hat{u}[l] = \begin{cases} \varphi_{l+T}(y[0], y[1], \dots, y[l+T]) & \text{if } l+T \leq n-1 \\ \varphi_{l+T}(y[0], y[1], \dots, y[n-1]) & \text{if } l+T > n-1 \end{cases} \quad (37)$$

Definition 7. A point-to-point $(n, k, T)_{\mathbb{F}}$ -block code is said to be N -achievable if the following holds for any N -erasure sequence $e^\infty \in \Omega_N$: For the $(n, k, T)_{\mathbb{F}}$ -block code, we have

$$\hat{u}[l] = u[l] \quad (38)$$

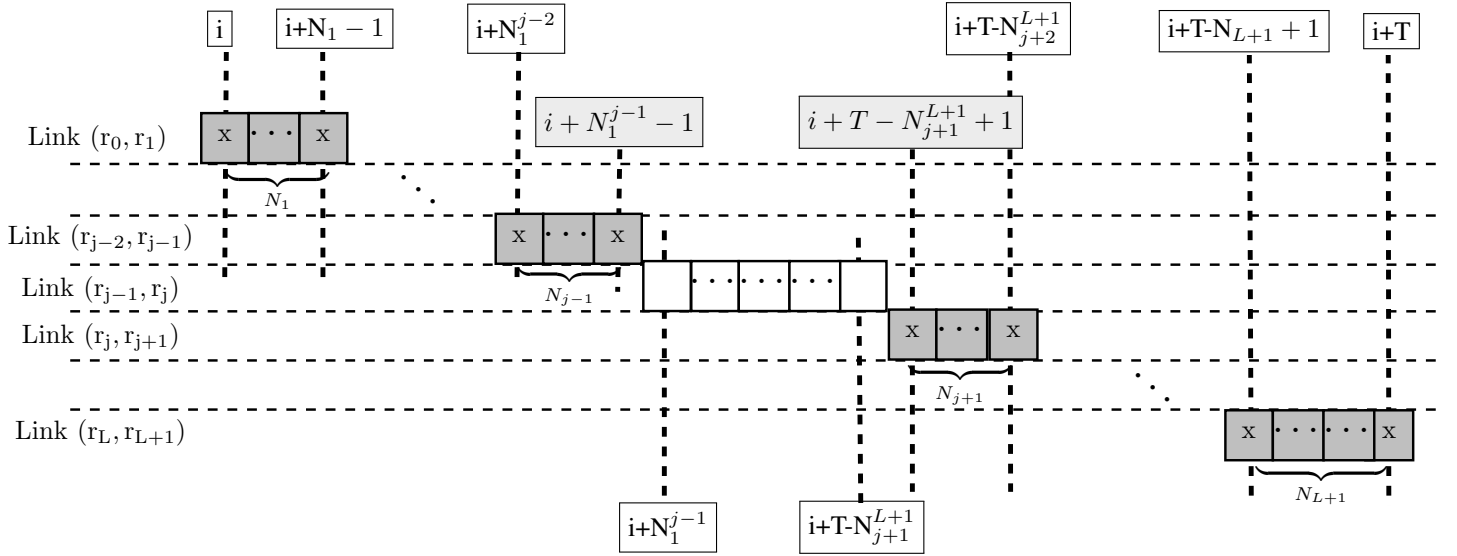


Fig. 7: Constrains imposed on transmission in link (r_{j-1}, r_j) .

for all $l \in \{0, 1, \dots, k-1\}$ and $u[l] \in \mathbb{F}$ and all $u[l] \in \mathbb{F}$, where

$$\hat{u}[l] = \begin{cases} \varphi_{l+T}(g_1(x[0], e_0), \dots, g_1(x[l+T], e_{l+T})) & \text{if } l+T \leq n-1 \\ \varphi_{l+T}(g_1(x[0], e_0), \dots, g_1(x[n-1], e_{n-1})) & \text{if } l+T > n-1 \end{cases} \quad (39)$$

with g_1 being the symbol-wise erasure function defined in (3).

Recalling from Section II that the rate in channel (r_j, r_{j+1}) is upper bounded by $R_j \leq \frac{k}{n_j}$ where

$$\begin{aligned} k &= T - \sum_{l=1}^{L+1} N_l + 1 \\ n_{j+1} &= T - \sum_{l=1, l \neq j+1}^{L+1} N_l + 1. \end{aligned} \quad (40)$$

Denoting

$$T_{j+1} = T - \sum_{l=1, l \neq j+1} N_l, \quad (41)$$

The suggested coding scheme is composed of $(n_{j+1}, k, N_1^j + T_j)_{\mathbb{F}}$ -block codes combined with diagonal interleaving. Each relay r_j is using n_{j+1} codes, each one is an $(n_{j+1}, k, N_1^j + T_j)_{\mathbb{F}}$ -block code that depends on the erasure pattern in the previous relay, i.e., each code can be different.

More formally, let $s_i[l]$ be the l 'th symbol of the source message \mathbf{s}_i and let $x_i^{(r_j)}[l]$ be the l 'th symbol of the output of encoding function $f_i^{(r_j)}$. We may say for each $i \in \mathbb{Z}_+$, a single transmission function of relay r_j constructs

$$\begin{bmatrix} x_{i+N_1^j}^{(r_j)}[0] & x_{i+N_1^j+1}^{(r_j)}[1] & \dots & x_{i+N_1^j+n_{j+1}-1}^{(r_j)}[n_{j+1}-1] \end{bmatrix} \triangleq [s_i[0] \ s_{i+1}[1] \ \dots \ s_{i+k-1}[k-1]] \times \mathbf{G}_i^{(r_j)}, \quad (42)$$

where we show next that $\mathbf{G}_i^{(r_j)}$ is a $k \times n_{j+1}$ generator matrix of a (n_{j+1}, k) MDS code. We assume that for any $i < 0$, $\mathbf{s}_i = \mathbf{0}$.

Denoting with

$$\tilde{\mathbf{s}}_i = [s_i[0] \ s_{i+1}[1] \ \dots \ s_{i+k-1}[k-1]], \quad (43)$$

an example of the diagonal interleaving of a single code is given in Table VI below.

Therefore, each transmitted packet at relay r_j is composed of n_{j+1} symbols, each of which is taken from a different code. Recalling that $[\tilde{\mathbf{s}}_i \times \mathbf{G}_i^{(r_j)}][j]$ means that we take the j 'th element from the vector resulting from multiplying $\tilde{\mathbf{s}}_i$ with the generator matrix $\mathbf{G}_i^{(r_j)}$, an example of a packet sent by relay r_j is given in (44) below.

Time	$i + N_1^j$	$i + N_1^j + 1$	\dots	$i + N_1^j + n_{j+1} - 1$
	$[\tilde{\mathbf{s}}_i \times \mathbf{G}_i^{(r_j)}][0]$			
		$[\tilde{\mathbf{s}}_i \times \mathbf{G}_i^{(r_j)}][1]$		
			\ddots	
				$[\tilde{\mathbf{s}}_i \times \mathbf{G}_i^{(r_{j+1})}][n_{j+1} - 1]$

TABLE VI: An example of a single code transmitted in r_j

$$\mathbf{x}_{i+N_1^j}^{(r_j)} = \begin{bmatrix} x_{i+N_1^j}^{(r_j)}[0] \\ x_{i+N_1^j}^{(r_j)}[1] \\ \vdots \\ x_{i+N_1^j}^{(r_j)}[n_{j+1} - 1] \end{bmatrix} = \begin{bmatrix} [\tilde{\mathbf{s}}_i \times \mathbf{G}_i^{(r_j)}][0] \\ [\tilde{\mathbf{s}}_{i-1} \times \mathbf{G}_{i-1}^{(r_j)}][1] \\ \vdots \\ [\tilde{\mathbf{s}}_{i-n_{j+1}+1} \times \mathbf{G}_{i-n_{j+1}+1}^{(r_j)}][n_{j+1} - 1] \end{bmatrix} \quad (44)$$

We note again that the specific structure of each $\mathbf{G}_i^{(r_j)}$ is defined according to the erasure pattern of the previous links. We describe next the process of generating $\mathbf{G}_i^{(r_j)}$ (required to transmit $\tilde{\mathbf{s}}_i$ in each one of the relays).

- At the sender (r_0), use an (n_1, k) MDS code. Hence, $\mathbf{G}_i^{(r_0)}$ is the generator matrix of an (n_1, k) MDS code.
- Each encoder at relay r_j ($j \in \{1, \dots, L\}$) performs the following

- 1) Store any non-erased symbols from the first N_j received symbols from link (r_{j-1}, r_j) , i.e., all non-erased symbols from

$$\left\{ x_{i+N_1^{j-1}}^{(r_{j-1})}[0], \dots, x_{i+N_1^{j-1}+N_j-1}^{(r_{j-1})}[N_j - 1] \right\}. \quad (45)$$

- 2) Start transmitting at time $i + N_1^j$ (while continuing to store the received symbols from link (r_{j-1}, r_j)). Until time $i + N_1^j + k - 2$, forward the $k - 1$ symbols received from link (r_{j-1}, r_j) by the order they were received, i.e., forward the $k - 1$ non-erased symbols from

$$\left\{ x_{i+N_1^{j-1}}^{(r_{j-1})}[0], \dots, x_{i+N_1^{j-1}+N_j+k-2}^{(r_{j-1})}[N_j + k - 2] \right\}. \quad (46)$$

Noting the $N_j + k - 2 = n_j - 1$ means that relay forwards the $k - 1$ symbols received by the order they were received, i.e., all non-erased symbols from

$$\left\{ x_{i+N_1^{j-1}}^{(r_{j-1})}[0], \dots, x_{i+N_1^{j-1}+N_j-1}^{(r_{j-1})}[n_j - 1] \right\}. \quad (47)$$

- 3) At time $N_1^j + k - 1$, recover $\tilde{\mathbf{s}}_i$. In Lemma 1 below we prove that it is feasible for any N_1, \dots, N_{L+1} -erasure sequence.
- 4) Transmit until time $N_1^j + n_{j+1} - 1$ encoded symbols. The encoded symbols should be non-erased symbols from (n_{\max}, k) MDS code to be defined below.
- 5) For each transmitted symbol, attach a header which will be defined next.

Recalling the definition of n_{\max} (8), the following Proposition sheds light on the suggested method of encoding $\tilde{\mathbf{s}}_i$ once it is recovered at r_j (step (4)) in order to guarantee that each $\mathbf{G}_i^{(r_j)}$ is a generator matrix of (n_{j+1}, k) MDS code and further simplify the header required to allow decoding.

Proposition 1. All block codes used by nodes r_j where $j \in \{0, \dots, L\}$ to transmit $\tilde{\mathbf{s}}_i$ can be generated by puncturing and applying a permutation to the (n_{\max}, k) MDS code which is associated with rate $C_{T, N_1, \dots, N_{L+1}}^+$.

This proposition holds since k (the number of information symbols) is the same for all codes. Denoting $N_{\max} = \max_j N_j$, we recall that the MDS code (n_{\max}, k) which is the MDS code associated with rate $C_{T, N_1, \dots, N_{L+1}}^+$ can correct any N_{\max} erasures. Puncturing any $N_{\max} - N_j$ columns from the generator matrix of this code results with a code that can correct any N_j erasures.

In fact, denoting with \mathbf{G}_{\max} the generator matrix of (n_{\max}, k) MDS code, $\mathbf{G}_i^{(r_j)}$ can be viewed as taking n_{j+1} columns from \mathbf{G}_{\max} and apply permutation on the order of the columns. The specific columns taken from \mathbf{G}_{\max} and their order is defined by the specific erasure pattern which occurs.

Further, following Proposition 1 we define the header as a number which indicates the location of the column from \mathbf{G}_{\max} that was used to generate this symbol. Thus, the header attached to each symbol transmitted from each relay is a number in the range $[1, \dots, n_{\max}]$.

Thus we have the following Lemma.

Lemma 1. For any N_1, \dots, N_{L+1} -erasure sequence, any $j \in \{0, \dots, L\}$ and any $i \in \mathbb{Z}_+$, $\mathbf{G}_i^{(r_j)}$ is a generator matrix of (n_{j+1}, k) MDS code.

Proof. From the construction, $\mathbf{G}_i^{(r_0)}$ is an (n_1, k) MDS code. Hence we assume by induction that $\mathbf{G}_i^{(r_{j-1})}$ is an (n_j, k) MDS code and show that $\mathbf{G}_i^{(r_j)}$ is an (n_{j+1}, k) MDS code.

We note that the only non-trivial steps in generating $\mathbf{G}_i^{(r_j)}$ are steps (2) and (3). Assuming N_1, \dots, N_{L+1} -erasure sequence means that for any $j \in \{0, \dots, L\}$

$$e^{j,\infty} \in \Omega_{N_j} \quad (48)$$

i.e., that the maximal number of erasures in line (r_{j-1}, r_j) is N_j . Since we assumed $\mathbf{G}_i^{(r_{j-1})}$ is an (n_j, k) MDS code it follows that it is guaranteed that k symbols out of the n_j transmitted symbols will not be erased. Further, since relay r_{j-1} starts sending the coded symbols at time $i + N_1^{j-1}$ and relay r_j starts forwarding the non-erased symbols received from r_{j-1} at time $i + N_1^{j-1} + N_j$ (after buffering any non erased symbols from the first N_j coded symbols) it is guaranteed that relay r_j could forward the $k - 1$ non-erased coded symbols sent from r_{j-1} .

In step (3), relay r_j needs to recover all k information symbols at time $i + N_1^j + k - 1$. We note that this step is feasible since, assuming $\mathbf{G}_i^{(r_{j-1})}$ is the generator matrix of an (n_j, k) MDS code, any of its k information symbols can be recovered from any k non-erased symbols. We note that relay r_{j-1} transmits its code in time indices

$$[i + N_1^{j-1}, \dots, i + N_1^{j-1} + n_j - 1] \quad (49)$$

therefore, the last symbol of this code is received at relay r_j at index

$$\begin{aligned} i + N_1^{j-1} + n_j &= i + N_1^{j-1} + T - \sum_{l=1, l \neq j}^{L+1} N_l + 1 \\ &= i + N_1^{j-1} + N_j + (T - \sum_{l=1}^{L+1} N_l + 1) \\ &= i + N_1^j + k. \end{aligned} \quad (50)$$

These regions are depicted in Figure 8 below.

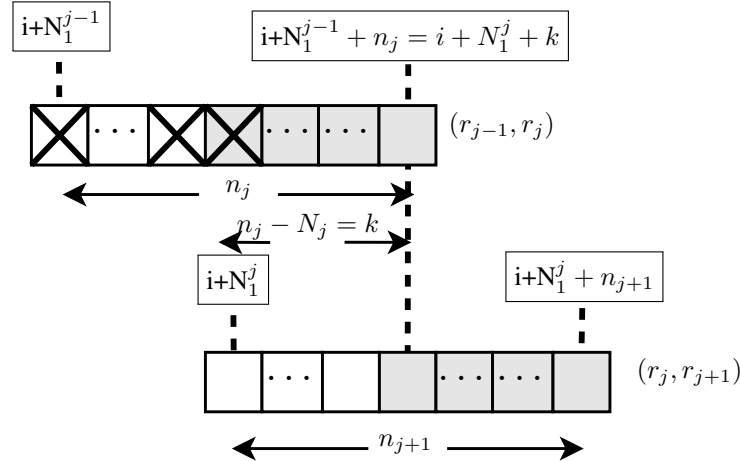


Fig. 8: Two regions of transmission in link (r_j, r_{j+1}) . The symbols with white background are symbols forwarded from link (r_{j-1}, r_j) . The shaded symbols are transmitted after the k information symbols are decoded hence they are either symbols erased in link (r_{j-1}, r_j) or additional (independent) linear combinations of the information symbols.

Since we assumed $\mathbf{G}_i^{(r_{j-1})}$ is an (n_j, k) MDS code, it follows its k information symbols can be recovered from any k symbols received. However, in order to recover the information symbols, relay r_j needs to know the structure of $\mathbf{G}^{(r_{j-1})}$. Using the header attached to each received symbol relay r_j it is guaranteed that relay r_j could recover (all k) information symbols at time $i + N_1^j + k - 1$.

Further, since after the recovery of the k symbols, r_j adds $n_{j+1} - k$ unique columns from \mathbf{G}_{\max} the process described above results with $\mathbf{G}^{(r_j)}$ which is the generator matrix of (n_{j+1}, k) MDS code which can recover any N_{j+1} erasures. \square

For specific examples on the modification of the block codes as a function of rate change between links, see Appendix B. Next we show the following lemma.

Lemma 2 (Based on Lemma 3 in [1]). *Suppose $T \geq N$, and let $k \triangleq T - N + 1$ and $n \triangleq k + N$. For any \mathbb{F} such that $|\mathbb{F}| \geq n$, there exists an N -achievable point-to-point $(n, k, T)_{\mathbb{F}}$ -block code.*

Proof. The proof follows directly from the definitions of MDS code. Any (n, k) MDS code is $(n, k, n - 1)$ block code. Thus, any (n, k) MDS code is $(n - k)$ -achievable where all symbols can be decoded by the end of the code block. \square

Recalling that when transmitting \tilde{s}_i , relay r_j starts its transmission at time $i + N_1^j$ we have the following corollary.

Corollary 1. *Recalling that relay r_j (for any $j \in \{0, \dots, L\}$) starts transmitting the coded symbols of \tilde{s}_i at time $i + N_1^j$, it follows that for any N_1, \dots, N_{L+1} -erasure sequence the code used in each relay r_j to transmit \tilde{s}_i is N_{j+1} -achievable $(n_{j+1}, k, N_1^j + T_{j+1})_{\mathbb{F}}$ point-to-point block code, i.e., all the symbols of \tilde{s}_i can be decoded at relay r_{j+1} by delay of*

$$\begin{aligned} i + N_1^j + T_{j+1} &= i + N_1^j + T - \sum_{l=1, l \neq j+1}^{L+1} N_l \\ &= i + T - \sum_{l=j+2}^{L+1} N_l. \end{aligned} \quad (51)$$

We take a closer look at the channel between the last relay and the destination (r_L, r_{L+1}) . Following Corollary 1 we have

Corollary 2. *For any $i \in \mathbb{Z}_+$, and for any N_1, \dots, N_{L+1} -erasure sequence, \tilde{s}_i can be decoded at the destination at time*

$$i + T - \sum_{l=L+2}^{L+1} N_l = i + T, \quad (52)$$

at the destination, i.e., using the construction suggested above, \tilde{s}_i can be decoded at the destination with delay of T for any $e^{L+1, \infty} \in \Omega_{N_{L+1}}$.

Next, we show that this Corollary means that the construction suggested above generates a $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code which is also N_1, \dots, N_{L+1} -achievable.

Lemma 3. *The streaming code resulting from using $\mathbf{G}_i^{(r_j)}$ defined above in each relay $j \in \{0, \dots, L\}$ for every $i \in \mathbb{Z}_+$ is a $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code which is also N_1, \dots, N_{L+1} -achievable.*

Proof. Our goal is to show that the destination (r_{L+1}) can recover $\mathbf{s}_i = [s_i[0], s_i[1], \dots, s_i[k]]$ based on

$$\left[\mathbf{y}_0^{(r_{L+1})}, \mathbf{y}_1^{(r_{L+1})}, \dots, \mathbf{y}_{i+T}^{(r_{L+1})} \right] = \left[g_{n_{L+1}} \left(\mathbf{x}_0^{(r_L)}, e_0^{L+1} \right), \dots, g_{n_{L+1}} \left(\mathbf{x}_{i+T}^{(r_L)}, e_{i+T}^{L+1} \right) \right] \quad (53)$$

Following Lemma 1 it follows each $\mathbf{G}_i^{(r_j)}$ is n_{j+1}, k MDS code. From Corollary 2 it follows that for any $i \in \mathbb{Z}_+$, $s_i[0]$ (which is the first element in \tilde{s}_i) can be recovered with a delay of T for any N_1, \dots, N_{L+1} -erasure sequence. Similarly we note that $s_i[1]$ can be recovered with a delay of $T - 1$ and $s_i[k - 1]$ can be recovered with delay of $T - k$. Thus, we conclude that the $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code resulting from the construction described above is a $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code which is also N_1, \dots, N_{L+1} -achievable. \square

Thus to prove Theorem 3, we need to show that when $|\mathbb{F}| \geq n_{\max}$, $\mathbf{G}_i^{(r_j)}$ can be generated at each relay r_j for $j \in \{0, \dots, L\}$ and analyze the overall rate (by bounding the rate of the additional header).

Proof of Theorem 3. We first note that, as mentioned in Section I-A, an (n_{\max}, k) MDS code exists as long as $|\mathbb{F}| \geq n_{\max}$. Therefore, following Lemma 1, when $|\mathbb{F}| \geq n_{\max}$ it follows that for any N_1, \dots, N_{L+1} -erasure sequence, any $j \in \{0, \dots, L\}$ and any $i \in \mathbb{Z}_+$, there exists $\mathbf{G}_i^{(r_j)}$ which is a generator matrix of (n_{j+1}, k) MDS code.

Following Lemma 3 it follows that streaming code resulting from using $\mathbf{G}_i^{(r_j)}$ defined above in each relay $j \in \{0, \dots, L\}$ for every $i \in \mathbb{Z}_+$ is a $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code which is also N_1, \dots, N_{L+1} -achievable. The rate in relay r_j , without taking the size of the header into account, is $\frac{k}{n_{j+1}}$. Thus, from Definition 5, the overall rate of transmission is upper bounded by

$$\begin{aligned} R &\leq \min_{j \in \{0, 1, \dots, L\}} \frac{k}{n_{j+1}} \\ &= C_{T, N_1, \dots, N_L}^+. \end{aligned} \quad (54)$$

The header attached to each packet sent from relay r_j is composed from stacking the n_{j+1} headers used by each symbol generated from a $(n_{j+1}, k, N_1^j + T_j)_{\mathbb{F}}$ block code which is part of each transmission packet. As we defined above, this header is a number from $[1, \dots, n_{\max}]$. Hence the size of the header is $n_{j+1} \log(n_{\max})$ bits. We further note that the size of the header is upper bounded by $n_{\max} \log(n_{\max})$.

To conclude, each node transmits n_{j+1} coded symbols (each taken from field \mathbb{F}) along with $n_{j+1} \log(n_{j+1})$ bits of header to transfer k information symbols (each taken from field \mathbb{F}). The overall rate is

$$\begin{aligned} R &\geq \min_j \frac{k \cdot \log(|\mathbb{F}|)}{n_{j+1} \cdot \log(|\mathbb{F}|) + n_{\max} \lceil \log(n_{\max}) \rceil} \\ &= \frac{T - \sum_{l=1}^{L+1} N_l + 1}{\max_j \left\{ T - \sum_{l=1, l \neq j}^{L+1} N_l + 1 \right\} + \frac{n_{\max} \lceil \log(n_{\max}) \rceil}{\log(|\mathbb{F}|)}} \\ &= \frac{T - \sum_{l=1}^{L+1} N_l + 1}{T - \min_j \left\{ \sum_{l=1, l \neq j}^{L+1} N_l \right\} + 1 + \frac{n_{\max} \lceil \log(n_{\max}) \rceil}{\log(|\mathbb{F}|)}} \end{aligned} \quad (55)$$

where n_{\max} is defined in (8). \square

IV. AN UPPER BOUND ON LOSS PROBABILITY ATTAINED BY STATE-DEPENDENT SYMBOL-WISE DF FOR RANDOM ERASURE

In Section III, the state-dependent symbol-wise DF scheme was described, and a lower bound on its achievable rate was derived while assuming a deterministic erasure model. In this section, we develop an upper bound on the average loss probability when this scheme is applied over channels with random (i.i.d.) erasures.

Let $s_i[0], s_i[1], \dots, s_i[k-1]$ be the k source symbols transmitted by node r_0 at each discrete time i . We note that for the $n_1, (n_1, k)$ MDS codes used by the sender, the following property holds:

- For every $s_i[v]$ located at the $(v+1)$ th position of the length- k packet transmitted at time i by the $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code over (r_0, r_1) , $\hat{s}_i^{(r_1)}[v]$ is generated by the relay, at the latest, at time $i - v + n_1 - 1$ (i.e., after transmission of n_1 symbols from r_0). If there are at most N_1 erasures inside the window $\{i - v, i - v + 1, \dots, i - v + n_1 - 1\}$, then $\hat{s}_i^{(r_1)}[v] = s_i[v]$.

Hence, s_i can be fully recovered at relay r_1 if for all $v \in \{0, 1, \dots, k-1\}$, in any window $\{i - v, i - v + 1, \dots, i - v + n_1 - 1\}$, there are at most N_1 erasures. We bound the loss probability by analyzing the probability in which in the window $\{i - k + 1, i - k + 2, \dots, i + n_1 - 1\}$ there are at most N_1 erasures.

Since the state-dependent symbol-wise DF encode the same information symbols (per diagonal) in each relay, we note that in the general case, when transmitting the $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ streaming code over (r_{j-1}, r_j) :

- $\hat{s}_i^{(r_j)}[v]$ is generated by the relay, at the latest, at time $i + N_1^{j-1} - v + n_j - 1$ (i.e., after transmission of n_j symbols from relay r_{j-1}). If there are at most N_j erasures inside the window $\{i + N_1^{j-1} - v, i - v + 1, \dots, i + N_1^{j-1} - v + n_j - 1\}$, $\hat{s}_i^{(r_j)}[v] = s_i[v]$.

Hence, s_i can be fully recovered at relay r_j if for all $v \in \{0, 1, \dots, k-1\}$, in any window $\{i + N_1^{j-1} - v, i + N_1^{j-1} - v + 1, \dots, i + N_1^{j-1} - v + n_j - 1\}$ there are at most N_j erasures. Similar to [1], we bound the loss probability by analyzing the probability in which in the window $\{i + N_1^{j-1} - k + 1, i + N_1^{j-1} - k + 1, \dots, i + N_1^{j-1} + n_j\}$ there are at most N_j erasures.

Denoting the average Loss probability as

$$P_{T, N_1, N_2, \dots, N_{L+1}} \triangleq \lim_{M \rightarrow \infty} \frac{1}{M} \Pr\{\hat{s}_i \neq s_i\} \quad (56)$$

achieved by the above state-dependent symbol-wise DF strategy under the random erasure model. Define $\alpha_j = \Pr(e_0^j = 1)$ to be the erasure probability in link (r_{j-1}, r_j) . According to the achievability conditions we have

$$\Pr\left(\hat{s}_i \neq s_i \mid \sum_{u=i-k+1}^{i+n_1+1} e_u^1 \leq N_1, \sum_{u=i-k+1}^{i+n_2+1} e_u^2 \leq N_2, \dots, \sum_{u=i-k+1}^{i+n_{L+1}+1} e_u^{L+1} \leq N_{L+1}\right) = 0 \quad (57)$$

for every $i \geq T - N_1^{L+1}$. Since

$$\begin{aligned} &\Pr\left(\left\{\sum_{u=i-k+1}^{i+n_1+1} e_u^1 > N_1\right\} \cup \left\{\sum_{u=i-k+1}^{i+n_2+1} e_u^2 > N_2\right\} \cup \dots \cup \left\{\sum_{u=i-k+1}^{i+n_{L+1}+1} e_u^{L+1} > N_{L+1}\right\}\right) \\ &\leq \sum_{u=N_1+1}^{2k+2N_1+1} \binom{2k+2N_1+1}{u} (\alpha_1)^u (1 - \alpha_1)^{2k+2N_1+1-u} \end{aligned}$$

$$\begin{aligned}
& + \sum_{u=N_2+1}^{2k+2N_2+1} \binom{2k+2N_2+1}{u} (\alpha_2)^u (1-\alpha_2)^{2k+2N_2+1-u} + \dots \\
& + \sum_{u=N_{L+1}+1}^{2k+N_{L+1}+1} \binom{2k+N_{L+1}+1}{u} (\alpha_{L+1})^u (1-\alpha_{L+1})^{2k+N_{L+1}+1-u}
\end{aligned} \tag{58}$$

It follows that

$$\begin{aligned}
P_{T,N_1,N_2,\dots,N_{L+1}} & \leq \kappa_1(T, N_1, \dots, N_{L+1}) \cdot (\alpha_1)^{N_1+1} + \kappa_2(T, N_1, \dots, N_{L+1}) \cdot (\alpha_2)^{N_2+1} + \dots \\
& + \kappa_{L+1}(T, N_1, \dots, N_{L+1}) \cdot (\alpha_{L+1})^{N_{L+1}+1}
\end{aligned} \tag{59}$$

where $\kappa_j(T, N_1, \dots, N_{L+1})$ does not depend on α_j (or on any other α_k for any $k \neq j$). Hence, $P_{T,N_1,N_2,\dots,N_{L+1}}$ decays exponentially fast in $\min\{N_1 + 1, N_2 + 1, \dots, N_{L+1} + 1\}$.

V. NUMERICAL RESULTS

In this section, we show the performance of the state-dependent symbol-wise DF scheme on a random model. We consider a statistical $L + 1$ -node relay network where i.i.d. erasures are independently introduced to all channels. We denote with α_j the probability of experiencing an erasure in each time slot for channel (r_{j-1}, r_j) .

Similar to [1], we will compare state-dependent symbol-wise DF with message-wise DF and instantaneous forwarding, which we briefly recall. In message-wise DF, all the symbols in the same source message are decoded by relay r_j subject to the delay constraint T_j such that $\sum_j T_j \leq T$. The overall rate of message-wise DF is

$$R_{T,N_1,N_2,\dots,N_{L+1}}^{\text{Message}} = \max_{(T_1,\dots,T_{L+1}): \sum_j T_j \leq T} \min\{C_{T_1,N_1}, C_{T_2,N_2}, \dots, C_{T_{L+1},N_{L+1}}\}. \tag{60}$$

More precisely, we consider message-wise DF scheme constructed by concatenating $L + 1$ streaming codes where the j 'th code is a $(n_j^{\text{Message}}, k, T_j)_{\mathbb{F}}$ -streaming code.

We also consider an instantaneous forwarding (IF) strategy, which uses a point-to-point streaming code over the $L + 1$ -node relay network as if the network is a point-to-point channel. More specifically, under the IF strategy, the source transmits symbols generated by the streaming code and relay r_j forwards every symbol received from relay r_{j-1} in each time slot. The overall point-to-point channel induced by the IF strategy experiences an erasure if either one of the channels experiences an erasure. This results with rate

$$R^{\text{IF}} = C_{T, \sum_l N_l}. \tag{61}$$

We first study the error-correcting capabilities of all schemes in case of having two relays with a rate equal to $2/3$ and maximal total delay of $T = 9$. We further simulate a symmetric topology, i.e., we assume the same error probability for all segments. Since we assume symmetric topology, we focus on schemes that have the same error-correcting capabilities for all segments.

- State-dependent symbol-wise DF can support $(N_1, N_2, N_3) = 2$ erasures in each segment which is constructed using concatenating three $(6, 4, 5)_{\mathbb{F}}$ streaming code. While the rate of the code is strictly lower than $2/3$ due to the overhead it uses, as we noted above, it approaches $2/3$ as $|\mathbb{F}|$ increases.
- Message-wise DF can support $(N_1, N_2, N_3) = 1$ erasures in each segment which is constructed using concatenating three $(3, 2, 2)_{\mathbb{F}}$ streaming code. As mentioned in [11], higher rate codes (such as $(4, 3, 3)_{\mathbb{F}}$) are excluded since $(3, 2, 2)_{\mathbb{F}}$ can correct more erasure patterns.
- For IF, we note that since $C_{9, \sum_j N_j} = \frac{10 - \sum_j N_j}{10}$ and we require $C_{9, \sum_j N_j} \geq 2/3$ we get $\sum_j N_j \leq 3$. Hence IF uses a $(10, 7, 9)_{\mathbb{F}}$ streaming code.

In Figure 9 we plot the frame loss ratio for state-dependent symbol-wise DF, Message-wise DF and for IF. We further plot the upper bound for State-dependent symbol-wise DF derived in Section IV with $R = 2/3$, $T \leq 9$ and $N_1 = N_2 = N_3 = 2$ while assuming $\forall j : \alpha_j = \alpha$.

Since enforcing rate $2/3$ resulted with low overall delay ($T = 6$ for example in case of message-wise DF), we further plot the performance of all schemes where we force $T = 9$ and allow the rate to be greater than or equal to $2/3$ (while trying to find the smaller rate possible). Figure 10 depicts all schemes when the rates are as close as possible to $2/3$ (from above) with $T = 9$. We note that, again, state-dependent symbol-wise DF outperforms all other methods.

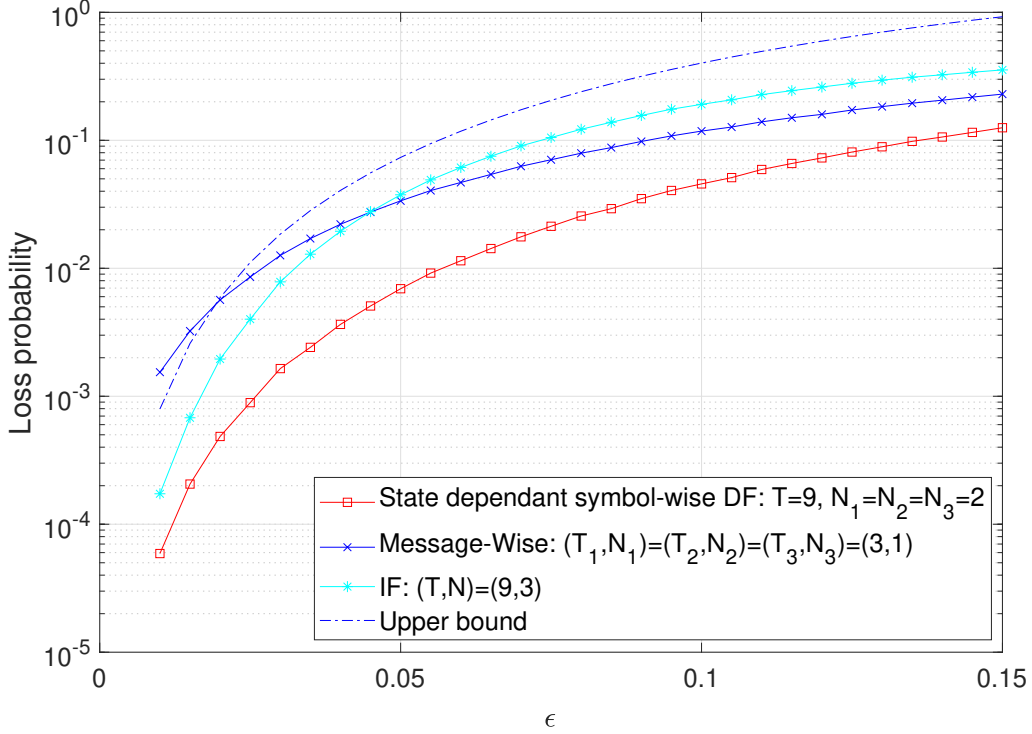


Fig. 9: Four-node (two relays) network loss probability for state-dependent symbol-wise DF, message-wise DF and IF with $T \leq 9$, rate $2/3$ and largest $N_1 + N_2 + N_3$ where α denotes the symmetric erasure probability

VI. EXTENSION TO SLIDING WINDOW MODEL

Consider the following sliding window model. For each $j \in \{1, \dots, L+1\}$, channel (r_{j-1}, r_j) introduces at most N_j erasures in any period of $T+1$ consecutive time slots (sliding window of size $T+1$).

Under the sliding window described above, denote with $C_{T,N_1,\dots,N_{L+1}}^{+,sw}$ as the upper bound on the achievable rate for $(N_1, N_2, \dots, N_{L+1})$ channel. We further denote R^{sw} as the achievable rate under the sliding window model.

Our goal is to show that

$$C_{T,N_1,\dots,N_{L+1}}^{+,sw} \leq C_{T,N_1,\dots,N_{L+1}}^+ \quad (62)$$

and

$$R^{sw} = R. \quad (63)$$

With respect to the upper bound, since for any $j \in \{1, \dots, L+1\}$, N_j -erasure sequence can be introduced by channel (r_{j-1}, r_j) in the sliding window model (62) holds.

Next, we show that the state-dependent scheme can achieve the same rate under the sliding window model. As was shown in Section III, for any $j \in \{1, \dots, L+1\}$, each symbol can be recovered as long as channel (r_{j-1}, r_j) introduces at most N_j erasures in a window of size n_j . From (34) we have $n_j < T+1$, it follows that all L conditions hold under the sliding window model thus the state-dependent code can recover all symbols and hence (63) holds.

VII. CONCLUDING REMARKS

Streaming codes became an integral part of real-time interactive video streaming applications. The ability to improve the quality of user-experience while meeting stringent latency constraints helped to transform these applications from a niche to one of the fastest-growing segments of IP traffic.

The problem of quantifying the capacity of streaming codes was first addressed for point-to-point links. Later it was extended for three-node networks and in this work, we extended the upper bound of streaming codes to multi-hop relay networks (i.e., for networks with any number of nodes) and further, we suggested an achievable scheme which achieves this upper bound with a gap that vanishes as the field size used gets larger. While for three-node networks, capacity can be achieved with “state-independent” symbol-wise DF coding scheme, we showed that this scheme could not be easily extended to more than three-nodes and suggested a symbol-wise DF “state-dependent” coding scheme.

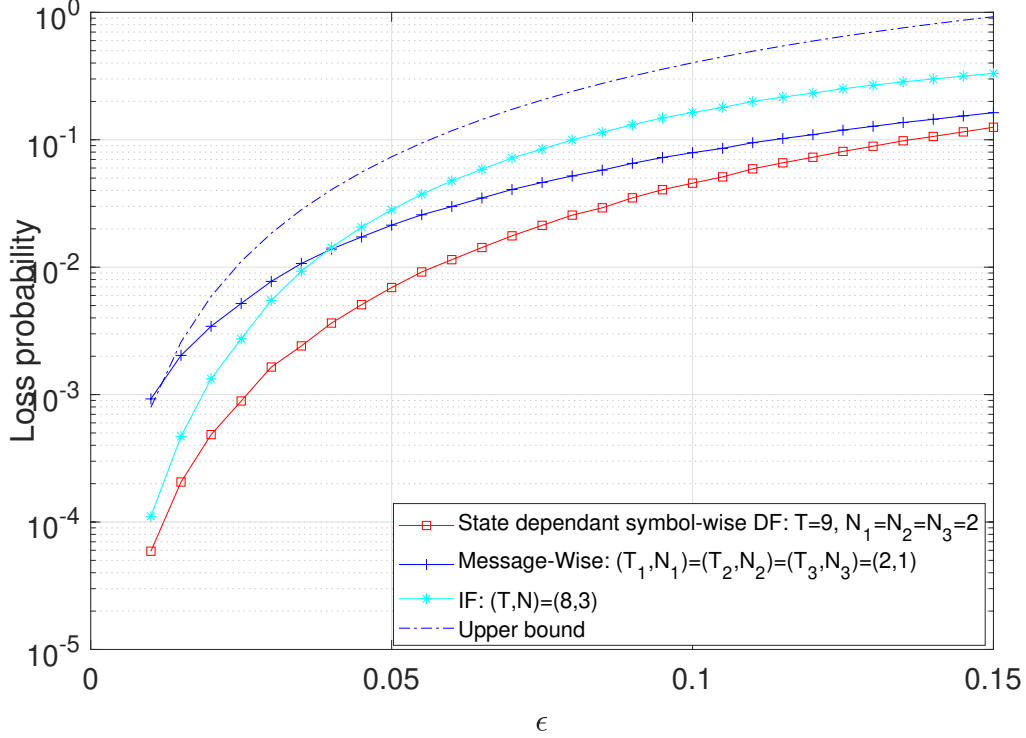


Fig. 10: Four-node (two relays) network loss probability for state-dependent symbol-wise DF, message-wise DF and IF with delay $T = 9$, $R \geq 2/3$ and largest $N_1 + N_2 + N_3$ where α denotes the symmetric erasure probability

While starting by describing the upper bound and proving the achievable scheme over “deterministic arbitrary erasure channel”, Section VI outlined that these results also hold for a sliding window model. While this model is a simplified model for a random erasure model, Section IV showed that the error probability of the achievable scheme could be upper bounded for random channel and Section V provided numerical results showing that the suggested coding scheme outperforms other simple options such as message-wise DF and instantaneous forwarding.

Packet erasures can occur either in sparse patterns or in a bursty manner. The first works on streaming codes analyzed the case of burst erasure channel. Later, works studied channels which could have either a burst or sparse erasures in a given window. Future work may explore streaming codes in a setup with relays in this case. Another avenue is to study streaming codes for more complicated networks.

APPENDIX A DERIVATIONS IN THE CONVERSE PROOF OF THEOREM 2

Deviation of (22): Since the (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code restricted to channel (r_0, r_1) can be viewed as a point-to-point streaming code with rate k/n_1 and delay $T - N_2^{L+1}$ which can correct the periodic erasure sequence \tilde{e}^∞ illustrated in Figure 4, it follows from the arguments in [10] Section IV-A that (22) holds. For the sake of completeness, we present a rigorous proof below.

Using (19), we have

$$|\mathbb{F}|^{k \times j(T - N_2^{L+1} + 1)} \leq |\mathbb{F}|^{n \times (j+1)(T - N_1^{L+1} + 1)} \quad (64)$$

because $j(T - N_2^{L+1} + 1)$ source messages can take $|\mathbb{F}|^{k \times j(T - N_2^{L+1} + 1)}$ values and $(j+1)(T - N_1^{L+1} + 1)$ source packets can take at most $|\mathbb{F}|^{n \times (j+1)(T - N_1^{L+1} + 1)}$ values for each j . Taking logarithm on both sides of (64) followed by dividing both sides by j , we have

$$k(T - N_2^{L+1} + 1) \leq n(1 + 1/j)(T - N_1^{L+1} + 1) \quad (65)$$

Since (65) holds for all $j \in \mathbb{N}$, it follows that (22) hold.

Deviation of (28): Since the (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code restricted to channel (r_L, r_{L+1}) can be viewed as a point-to-point streaming code with rate k/n_{L+1} and delay $T - N_1^L$ which can correct the periodic erasure

sequence \hat{e}^∞ illustrated in Figure 5, it follows from the arguments in [10] Section IV-A that (28) holds. For the sake of completeness, we present a rigorous proof below.

Using (26), we have

$$|\mathbb{F}|^{k \times j(T - N_1^L + 1)} \leq |\mathbb{F}|^{n \times (j+1)(T - N_1^{L+1} + 1)} \quad (66)$$

because $j(T - N_1^L + 1)$ source messages can take $|\mathbb{F}|^{k \times j(T - N_1^L + 1)}$ values and $(j+1)(T - N_1^{L+1} + 1)$ source packets can take at most $|\mathbb{F}|^{n \times (j+1)(T - N_1^{L+1} + 1)}$ values for each j . Taking logarithm on both sides of (66) followed by dividing both sides by j , we have

$$k(T - N_1^L + 1) \leq n(1 + 1/j)(T - N_1^{L+1} + 1) \quad (67)$$

Since (67) holds for all $j \in \mathbb{N}$, it follows that (28) hold.

Deviation of (32): Since the (N_1, \dots, N_{L+1}) -achievable $(n_1, \dots, n_{L+1}, k, T)_{\mathbb{F}}$ -streaming code restricted to channel (r_{j-1}, r_j) can be viewed as a point-to-point streaming code with rate k/n_j and delay $T - N_1^{j-1} - N_{j+1}^{L+1}$ which can correct the periodic erasure sequence \check{e}^∞ illustrated in Figure 6, it follows from the arguments in [10] Section IV-A that (34) holds. For the sake of completeness, we present a rigorous proof below.

Using (32), we have

$$|\mathbb{F}|^{k \times j(T - N_1^{j-1} - N_{j+1}^{L+1} + 1)} \leq |\mathbb{F}|^{n \times (j+1)(T - N_1^{L+1} + 1)} \quad (68)$$

because $j(T - N_1^L + 1)$ source messages can take $|\mathbb{F}|^{k \times j(T - N_1^{j-1} - N_{j+1}^{L+1} + 1)}$ values and $(j+1)(T - N_1^{j-1} - N_{j+1}^{L+1} + 1)$ source packets can take at most $|\mathbb{F}|^{n \times (j+1)(T - N_1^{L+1} + 1)}$ values for each j . Taking logarithm on both sides of (66) followed by dividing both sides by j , we have

$$k(T - N_1^{j-1} - N_{j+1}^{L+1} + 1) \leq n(1 + 1/j)(T - N_1^{L+1} + 1) \quad (69)$$

Since (69) holds for all $j \in \mathbb{N}$, it follows that (28) hold.

APPENDIX B

EXAMPLES FOR RATE CHANGE IN RELAY

As mentioned above, relay r_j may need to increase or decrease the rate of the code used by relay r_{j-1} . Below, we show examples for the following two cases:

- $\frac{k}{n_{j+2}} > \frac{k}{n_{j+1}}$. This means that $n_{j+2} < n_{j+1}$, i.e., that the block size of the MDS code used by relay r_{j+1} is smaller than the block size used by relay r_j . At time $i + T - \sum_{l=1}^{L+1} N_l + 1$, node r_{j+1} can recover the original data and send any of the erased symbols of the code used by r_j .

An example is given in Table VII for $N_{j+1} = 2$, $N_{j+2} = 1$, $T' = 4$ (where $T' = T - \sum_{l=1, l \neq j+1, j+2} N_l$). We note that in this example $k = T - \sum N_l + 1 = T' - N_{j+1} - N_{j+2} = 2$.

Relay r_{j+1} forwards the first $k - 1 = 1$ symbols it receives. At $i + N_1^j + 3$ the relay can recover the original data, hence from this point it sends (for example) the erased symbols.

$i + N_1^j$	$i + N_1^j + 1$	$i + N_1^j + 2$	$i + N_1^j + 3$	$i + N_1^j + 4$
Link (r_j, r_{j+1})				
a_i				
	b_{i+1}			
		$f^1(a_i, b_{i+1})$		
			$f^2(a_i, b_{i+1})$	
Link (r_{j+1}, r_{j+2})				
		b_{i+1}		
			a_i	
				$f^1(a_i, b_{i+1})$

TABLE VII: Example of increasing the rate between links. In this example, $N_{j+1} = 2$, $N_{j+2} = 1$, $T' = 4$, hence $\frac{k}{n_{j+2}} = 2/4 < 2/3 = \frac{k}{n_{j+1}}$. Assuming symbol $i + N_1^j$ and $i + N_1^j + 2$ were erased when transmitted in link (r_j, r_{j+1}) . Parity symbol are shaded.

- $\frac{k}{n_{j+2}} < \frac{k}{n_{j+1}}$. This means that $n_{j+2} > n_{j+1}$, i.e., the block size of the code used by relay r_{j+1} is larger than the block size used by relay r_j . At time $i + T - \sum_{l=1}^{L+1} N_l + 1$, relay r_{j+1} can again recover the original data and hence transmit additional $n_{j+2} - k$ symbols needed to allow handling any N_{j+2} erasures in the link (r_{j+1}, r_{j+2}) .

An example is given in Table VIII for $N_{j+1} = 1$, $N_{j+2} = 2$, $T' = 4$ (where, again, $T' = T - \sum_{l=1, l \neq j+1, j+2}$). Relay r_{j+1} forwards the first $k - 1 = 1$ symbols it receives. At $i + N_1^j + 2$, the relay can recover the original data, hence from this point it sends (for example) the erased symbols while adding parity symbols to reach the required rate.

$i + N_1^j$	$i + N_1^j + 1$	$i + N_1^j + 2$	$i + N_1^j + 3$	$i + N_1^j + 4$
Link (r_j, r_{j+1})				
a_i				
	b_{i+1}			
		$f^1(a_i, b_{i+1})$		
Link (r_{j+1}, r_{j+2})				
	b_{i+1}			
		a_i		
			$f^1(a_i, b_{i+1})$	
				$f^2(a_i, b_{i+1})$

TABLE VIII: Example of reducing rate between nodes. In this example, $N_{j+1} = 1$, $N_{j+2} = 2$, $T' = 4$, hence $\frac{k}{n_{j+1}} = 2/3 > 2/4 = \frac{k}{n_{j+2}}$. Assuming symbol $i + N_1^j$ was erased when transmitted in link (r_j, r_{j+1}) . Parity symbols are shaded.

REFERENCES

- [1] S. L. Fong, A. Khisti, B. Li, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Optimal streaming erasure codes over the three-node relay network," *IEEE Transactions on Information Theory*, 2019.
- [2] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, 2018.
- [3] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [4] D. J. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics letters*, vol. 32, no. 18, pp. 1645–1646, 1996.
- [5] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE, 2002, pp. 271–280.
- [6] A. Shokrollahi, "Raptor codes," *IEEE transactions on information theory*, vol. 52, no. 6, pp. 2551–2567, 2006.
- [7] T.-Y. Huang, P. Huang, K.-T. Chen, and P.-J. Wang, "Could skype be more satisfying? a QoE-centric study of the FEC mechanism in an internet-scale VoIP system," *IEEE Network*, vol. 24, no. 2, pp. 42–48, 2010.
- [8] 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.
- [9] 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.
- [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] S. L. Fong, A. Khisti, B. Li, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Optimal streaming codes for channels with burst and arbitrary erasures," *IEEE Transactions on Information Theory*, 2019.
- [12] 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.
- [13] 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.
- [14] D. Leong and T. Ho, "Erasure coding for real-time streaming," in *2012 IEEE International Symposium on Information Theory Proceedings*, 2012, pp. 289–293.
- [15] F. Begtasovic and P. Van Mieghem, "Measurements of the hopcount in internet," in *PAM2001, A workshop on Passive and Active Measurements, Amsterdam, the Netherlands, April 23-24, 2001.*
- [16] A. Mukaddam and I. H. Elhajj, "Hop count variability," in *2011 International Conference for Internet Technology and Secured Transactions*, 2011, pp. 240–244.
- [17] A. Badr, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Perfecting protection for interactive multimedia: A survey of forward error correction for low-delay interactive applications," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 95–113, 2017.
- [18] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes.* Elsevier, 1977, vol. 16.