

Streaming Erasure Codes Over Multi-Access Relayed Networks

Gustavo Kasper Facenda¹, *Graduate Student Member, IEEE*, Elad Domanovitz², *Member, IEEE*,
Ashish Khisti¹, *Member, IEEE*, Wai-Tian Tan, and John Apostolopoulos, *Fellow, IEEE*

Abstract—Many emerging multimedia streaming applications involve multiple users communicating under strict latency constraints. In this paper we study streaming codes for a network involving two source nodes, one relay node and a destination node. In this paper’s setting, each source node transmits a stream of messages, through the relay, to a destination, who is required to decode the messages under a strict delay constraint. For the case of a single source node, a class of streaming codes has been proposed by Fong et al., using the concept of *delay-spectrum*. The current paper presents a novel framework, which constructs streaming codes for a relayed multi-user setting by sequentially constructing the codes for each link. This requires a characterization of the set of all achievable delay spectra for a given rate, blocklength and number of erasures, beyond the specific choice considered by Fong et al. This characterization is presented in the paper for systematic codes. Using this novel framework, the first proposed scheme involves greedily selecting the rate on the link from relay to destination and using properties of the delay-spectrum to find feasible streaming codes that satisfy the required delay constraints. A closed form expression for the achievable rate region is provided, and conditions for when the proposed scheme is optimal are established by a natural outer bound. The second proposed scheme builds upon this approach, but uses a numerical optimization-based approach to improve the achievable rate region over the first scheme. Experimental results show that the proposed schemes achieve significant improvements over baseline schemes based on single-user codes.

Index Terms—Cloud computing, streaming, low-latency, symbol-wise decode-and-forward, multi-access relay network, forward error correction, packet erasure channel, rate region.

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

Manuscript received 3 September 2021; revised 7 June 2022; accepted 26 September 2022. Date of publication 12 October 2022; date of current version 20 January 2023. An earlier version of this paper was presented at the 2021 IEEE International Symposium on Information Theory [DOI: 10.1109/ISIT45174.2021.9518192]. (*Corresponding author: Gustavo Kasper Facenda.*)

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

Wai-Tian Tan and John Apostolopoulos were with Enterprise Networking Innovation Labs, Cisco Systems, San Jose, CA 95134 USA, and also with Google, Mountain View, CA 94043 USA.

Communicated by R. D. Wesel, Associate Editor for Coding and Decoding. Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIT.2022.3214165>.

Digital Object Identifier 10.1109/TIT.2022.3214165

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 round-trip time 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 [3], the authors studied a point-to-point (i.e., two nodes—source and destination) network under a maximal burst erasure pattern. In [4], the authors have studied, separately, burst erasures and arbitrary erasures. In [5], the authors have extended the erasure pattern, allowing for both burst erasures and arbitrary erasures. In particular, it was shown that random linear codes [6] 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 [7], [8], [9], [10], [11], [12], [13], [14], [15].

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 Fong et al. [2], which derived the time-invariant capacity for the three-node setting, and further extended to a multi-hop network in [16]. A time-variant adaptive code construction has been studied in [17] and shown to improve upon the rate of [2].

The relayed setting is challenging because, under strict delay constraints, such as in the streaming applications mentioned, it is not trivial to design the operations that the relay should perform. Naive operations, such as direct-forwarding or packet-level decode-and-forward (where the relay decodes entire source packets and then re-encodes them) are shown to be sub-optimal in [2]. As an improvement to these naive strategies, the authors propose a transmission scheme denoted

as symbol-wise decode-and-forward, in which the relay decodes *symbols*, rather than *packets*, and then re-encodes them in a judicious manner to satisfy the overall delay constraint. From this idea, the authors naturally present the concept of delay-spectrum, which, at high level, represents the delay that *each symbol* is subject to, rather than the delay *the packet* is subject to, when the channel introduces erasures. In the paper, the authors propose one specific family of codes, and present the delay spectrum for such codes. These codes have a uniform delay spectrum, that is, an equal number of symbols is recovered at each feasible delay. The authors show that this family of codes is sufficient for the single-user relayed setting in order to achieve non-adaptive (or time-invariant) optimality.

In our paper, we wish to extend the relayed topology of [2] to a multiple access relay channel. This is motivated by the fact that a significant part of the mentioned applications, such as real-time gaming and video conferences, involve communications between multiple users and a common server. However, this addition of more users in the network brings new challenges. First, again, the correct operations that the relay should perform are not clear. Naive extensions, such as employing single-user codes for each user, and then multiplexing them at the relay, treating them as independent streams, or decoding the source packets from both users and then re-encoding them jointly, are sub-optimal. Therefore, a way of judiciously jointly encoding the packets from each user at the relay, while satisfying the overall delay constraint of both users, is required. Furthermore, the simple family of codes used in the previous work is no longer sufficient, because codes with non-uniform delay spectra are now required. Intuitively, this is because the channels from each user to the relay may be different, which is modeled by a different number of erasures in each link, which will cause different numbers of symbols arriving with each delay at the relay. Because of these reasons, a more detailed analysis of the delay spectrum must be performed, and we must develop a deeper understanding of how the code employed in one link is constrained by the codes employed in other links of the network. We first answer these questions, and then we show how this knowledge can be used to build an elegant framework for sequentially constructing codes in multi-user networks through a delay-spectrum-constrained code construction.

Similar to the work in [2], we focus in time-invariant streaming codes, i.e., the relay node does not change the FEC code as a function of erasures on the source-relay link. This is in contrast to the approach in [16] and [17]. This choice is motivated by a variety of reasons, including: no optimal adaptive codes are known for the single-user setting; the known achievable adaptive codes [17] for the single-user setting require overheads in order to inform the destination about the erasure pattern from source to relay; in an adaptive code, the delay pattern from source to relay depends on the observed erasure pattern, thus, when generalizing to multiple users, one must take into account numerous possible erasure patterns happening in each link from source to relay, making a general framework hard to design, and analysis extremely complex; for the previous reason, the computational complex-

ity at the relay may also be impractical for such schemes. Although we focus on the setting with two source nodes in this paper, it should be noted that the proposed framework and the converse can be directly extended to multiple source nodes, although the expressions become notationally cumbersome.

A. Related Works and Applications

From a problem formulation perspective, our work is strongly connected to previous works on streaming codes over adversarial channels such as [3], [4], [5], [14], [15]. These works use a problem formulation similar to ours, in which an error-free transmission is desired with a strict delay constraint. Works such as [18] study the adaptation of random linear codes in order to provide better performances in dynamic channels, that is, in scenarios where the channel conditions change over time. However, none of these works consider a multi-access setting.

From a setting perspective, many applications benefit from a smaller latency. For example, cloud (or stream) gaming is an application in which the heavy processing of gaming, such as physics computation and rendering, are performed in a cloud server, which then streams the video output to the player, who in turn performs actions which are streamed back to the server. The impact of latency on the user experience and performance on cloud gaming has been widely studied [19], [20], [21], [22], [23], [24], [25]. It should be noted that the overall latency is composed by many different types of delay, including local processing, propagation, error correction, server processing and display.

Companies such as Google have put considerable effort in building their own networks in order to minimize propagation delay. Riot Games has done similarly, focused on multi-player online gaming instead of cloud gaming [26]. Networks such as WTFast are also used by players in order to reduce their propagation latency and have been evaluated in papers such as [27]. Here, the delay reduction is achieved by optimized routing, rather than the longer (cheaper) routes that regular Internet Service Providers may allocate [26] (often selected to minimize number of hops, as opposed to minimize latency).

Reducing the server-side processing delay has also been studied. For example, Google and Microsoft use machine learning to predict the future inputs from the user [28], allowing the server processing to happen before it receives the input, achieving what Google calls “negative latency”. Optimization of the video encoding for cloud gaming has also been considered in the literature [29]. Other aspects of latency such as the local processing and display times are mostly influenced by the user’s hardware.

On the other hand, the error correcting delay—that is, the amount of time required in order to recover from packet losses—seems to be understudied. As shown in [30], the round-trip-delay for Stadia’s network is usually between 10 and 15ms. Under a delay constraint of 50ms, re-transmissions taking 10ms represent a significant portion of the delay budget. However, by using FEC such as the ones described in this paper, with packet delays that can be as small as $T = 5$ for considerably lossy channels, lost packets can be

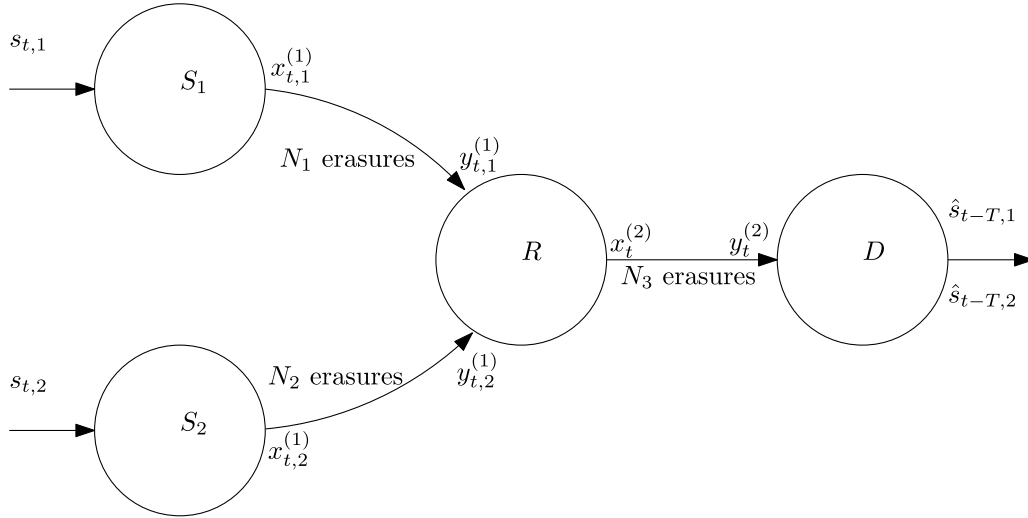


Fig. 1. Multiple access relay channel.

recovered within a few milliseconds, depending on the packet rate. This is even more significant when considering networks that are not as optimized as Google's, where round-trip delays may be considerably higher.

It should also be mentioned that a setting similar to ours has been studied in [31], where the authors consider a multi-access setting in Cloud Gaming, and show that Quality-of-Experience (QoE) can be improved by jointly optimizing different streams from different nodes. However, only resource allocation is optimized in the paper, basing the decision on the different types of games analyzed. In our work, we instead focus on optimizing the coding scheme used by the nodes and relay.

II. SYSTEM MODEL AND MAIN RESULTS

In this paper, we consider a network with two sources, one relay and one destination. Each source i wishes to transmit a sequence of messages $\{s_{t,i}\}_{t=0}^{\infty}$ to the destination through a common relay. We assume there is no direct link between sources and destination. We assume that the link between the first source and the relay introduces at most N_1 erasures, the link between the second source and the relay introduces at most N_2 erasures, and the link between relay and destination introduces at most N_3 erasures. The destination wishes to decode both source packets with a common delay T . This setting is illustrated in Fig. 1. This setting captures many practical applications, such as multi-party video conferencing, mentioned previously. Without loss of generality, we assume $N_1 \geq N_2$.

In the following, we present the formal definitions for the problem. For simplicity, we define $\mathbb{F}_e^n = \mathbb{F}^n \cup \{*\}$. The following definitions are standard and a straight-forward generalization of [2].

Definition 1: An $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code consists of the following:

- Two sequences of source messages $\{s_{t,1}\}_{t=0}^{\infty}$ and $\{s_{t,2}\}_{t=0}^{\infty}$, where $s_{t,i} \in \mathbb{F}^{k_i}$.
- Two encoding functions

$$f_{t,i} : \underbrace{\mathbb{F}^{k_i} \times \dots \times \mathbb{F}^{k_i}}_{t+1 \text{ times}} \rightarrow \mathbb{F}^{n_i}, \quad i \in \{1, 2\}$$

each used by its respective source i at time t to generate $x_{t,i}^{(1)} = f_{t,i}(s_{0,i}, s_{1,i}, \dots, s_{t,i})$.

- A relaying function

$$g_t : \underbrace{\mathbb{F}_e^{n_1} \times \dots \times \mathbb{F}_e^{n_1}}_{t+1 \text{ times}} \times \underbrace{\mathbb{F}_e^{n_2} \times \dots \times \mathbb{F}_e^{n_2}}_{t+1 \text{ times}} \rightarrow \mathbb{F}^{n_3}$$

used by the relay at time t to generate

$$x_t^{(2)} = g_t(\{y_{j,1}^{(1)}\}_{j=0}^t, \{y_{j,2}^{(1)}\}_{j=0}^t).$$

- Two decoding functions

$$\varphi_{t,1} = \underbrace{\mathbb{F}_e^{n_3} \times \dots \times \mathbb{F}_e^{n_3}}_{t+T+1 \text{ times}} \rightarrow \mathbb{F}^{k_1}$$

$$\varphi_{t,2} = \underbrace{\mathbb{F}_e^{n_3} \times \dots \times \mathbb{F}_e^{n_3}}_{t+T+1 \text{ times}} \rightarrow \mathbb{F}^{k_2}$$

used by the destination at time $t+T$ to generate

$$\hat{s}_{t,1} = \varphi_{t,1}(y_0^{(2)}, y_1^{(2)}, \dots, y_{t+T}^{(2)})$$

$$\hat{s}_{t,2} = \varphi_{t,2}(y_0^{(2)}, y_1^{(2)}, \dots, y_{t+T}^{(2)})$$

Definition 2: An erasure sequence is a binary sequence denoted by $e_i^{(1)} \triangleq \{e_{t,i}^{(1)}\}_{t=0}^{\infty}$, where

$$e_{t,i}^{(1)} = \mathbf{1}\{\text{an erasure occurs at time } t \text{ in the link from source } i \text{ to relay}\}. \quad (1)$$

Similarly, $e_t^{(2)} \triangleq \{e_t^{(2)}\}_{t=0}^{\infty}$ where

$$e_t^{(2)} = \mathbf{1}\{\text{an erasure occurs at time } t \text{ in the link from relay to destination}\} \quad (2)$$

An N -erasure sequence is an erasure sequence e that satisfies $\sum_{t=0}^{\infty} e_t = N$. In other words, an N -erasure sequence specifies N arbitrary erasures on the discrete timeline. The set of N -erasure sequences is denoted by Ω_N .

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

$$h_n(x, e) = \begin{cases} x, & \text{if } e = 0 \\ *, & \text{if } e = 1 \end{cases} \quad (3)$$

For any erasure sequences $e_i^{(1)}$ and any $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code, the following input-output relation holds for each $t \in \mathbb{Z}_+$:

$$y_{t,1}^{(1)} = h_{n_1}(x_{t,1}^{(1)}, e_{t,1}^{(1)}) \quad (4)$$

$$y_{t,2}^{(1)} = h_{n_2}(x_{t,2}^{(1)}, e_{t,2}^{(1)}) \quad (5)$$

where $e_i^{(1)} \in \Omega_{N_i}$, $i \in \{1, 2\}$. Similarly, the following input-output relation holds for each $t \in \mathbb{Z}_+$:

$$y_t^{(2)} = h_{n_3}(x_t^{(2)}, e_t^{(2)}) \quad (6)$$

where $e_t^{(2)} \in \Omega_{N_3}$.

In this paper, as stated and motivated in Section I, we focus on streaming codes for which the relaying function g_t does not adapt (change) the FEC code as function of the erasure pattern that has been observed in the link from sources to relay. We formally define time-invariant, or channel state independent codes below.

Definition 4: An $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code is channel state independent (or time-invariant) if and only if, given two fixed source sequences $\{s_{t,1}\}_{t=0}^{\infty}$ and $\{s_{t,2}\}_{t=0}^{\infty}$, the outputs of the encoding and relaying functions of this streaming code is also fixed, that is, $x_{t,1}^{(1)}$, $x_{t,2}^{(1)}$, and $x_t^{(2)}$ are independent of the erasure patterns $e_1^{(1)}$ and $e_2^{(1)}$.

Definition 5: An $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code is said to be (N_1, N_2, N_3) -achievable if, for any $e_{t,i}^{(1)}$, $i \in \{1, 2\}$ and $e_t^{(2)}$, for all $t \in \mathbb{Z}_+$ and all $s_{t,i} \in \mathbb{F}^{k_i}$, $i \in \{1, 2\}$, we have $\hat{s}_{t,i} = s_{t,i}$, $i \in \{1, 2\}$.

Definition 6: The pair of rates of an $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code is

$$\begin{aligned} R_1 &= \frac{k_1}{n} \\ R_2 &= \frac{k_2}{n} \\ n &= \max(n_1, n_2, n_3) \end{aligned}$$

Definition 7: The capacity (rate) region of an (N_1, N_2, N_3) multi-access relay network under delay constraint T is defined as the set of all rate pairs (R_1, R_2) such that there exists an (N_1, N_2, N_3) -achievable $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code, where (R_1, R_2) are defined as above.

For the remaining of the paper, when discussing the capacity region, we are referring to the capacity region of channel-state independent (time invariant) codes.

In this paper, for analysis purposes, we define four different regimes of operation based on the parameters N_1, N_2, N_3 and T . Let $C(T, N) = \frac{T+1-N}{T+1}$ denote the point-to-point capacity of streaming codes. As will be seen in Section III, the rate of the first user is bounded by $R_1 \leq C(T - N_3, N_1)$, the rate of the second user is bounded by $R_2 \leq C(T - N_3, N_2)$ and the sumrate is bounded by $R_1 + R_2 \leq C(T - N_2, N_3)$, assuming $N_1 \geq N_2$ without loss of generality. Based on these three bounds, we define our four regimes of operation.

First, we separate the regimes based on the sumrate bottleneck. If $C(T - N_2, N_3) > C(T - N_3, N_1) + C(T - N_3, N_2)$, then the sumrate constraint is active for at least one rate

pair, and we denote the regime as ‘‘source-relay bottleneck’’. Otherwise, we denote it as ‘‘relay-destination bottleneck’’.

Further, we classify each regime based on how ‘‘strong’’ the bottleneck is. For the source-relay bottleneck regimes, we denote it as ‘‘strong’’ bottleneck if we also have $C(T - N_1, N_3) \geq C(T - N_3, N_1) + C(T - N_3, N_2)$. This regime represents a scenario where the channel condition in the link from relay to destination is significantly better than both links from sources to relay. If this condition does not hold, then we denote it as a ‘‘weak’’ bottleneck.

For the relay-destination regimes, the link from relay to destination acts as a bottleneck, and the users can not simultaneously transmit at their single-user capacity. In this case, if $N_2 \geq N_3$, we denote it as a ‘‘weak’’ bottleneck, and if $N_3 > N_2$, we denote it as a ‘‘strong’’ bottleneck.

Below, we summarize each regime based on the conditions stated above.

- Strong source-relay bottleneck: $C(T - N_1, N_3) \geq C(T - N_3, N_1) + C(T - N_3, N_2)$.
- Weak source-relay bottleneck: $C(T - N_2, N_3) \geq C(T - N_3, N_1) + C(T - N_3, N_2) > C(T - N_1, N_3)$.
- Weak relay-destination bottleneck: $C(T - N_3, N_1) + C(T - N_3, N_2) \geq C(T - N_2, N_3) > C(T - N_3, N_2)$.
- Strong relay-destination bottleneck: $C(T - N_3, N_1) \leq C(T - N_2, N_3) \leq C(T - N_3, N_2)$.

For most of the paper, we focus on the Weak Relay-Destination Bottleneck regime. In Section VII we analyze the remaining regimes.

A. Main Contributions

In this paper, we derive upper and lower bounds to the capacity region of streaming codes in the described setting. Strict delay constraints are relevant in low latency applications, while the adversarial model represents worst-case scenarios, which are commonly of interest in such applications. For presentation, we split this section in three subsections, each one focused on a main contribution of the paper.

1) *Upper Bound:* The first result we present in the paper is a general upper bound for the capacity region

Theorem 1: The capacity region of time-invariant streaming codes for the four-node multi-access relayed network is upper bounded by the following three conditions

$$R_1 \leq C(T - N_3, N_1) \quad (7)$$

$$R_2 \leq C(T - N_3, N_2) \quad (8)$$

$$R_1 + R_2 \leq C(T - N_2, N_3) \quad (9)$$

where $C(T, N) = \frac{T+1-N}{T+1}$ is the capacity of a point-to-point channel subject to N erasures and strict delay T .

In order to do so, we first properly define ‘‘time invariant’’, or ‘‘channel state independent’’ streaming codes, and then we prove, from first principles, that the upper bound holds for said codes. While the arguments presented are based on the ones previously presented in [2], the prior work failed to consider channel state dependent codes, which can achieve rates higher than this bound, as shown in [17]. In our proof, we highlight where the channel state independent assumption is required, which was missed in the previous paper. This is an important

contribution, as it shows that the bound is still valid for a limited family of codes, which are of interest for reasons discussed in Section I.

2) *Delay Spectrum Analysis*: As another main contribution, in Section IV, we present an in-depth analysis of the concept of delay-spectrum. At high level, the delay spectrum maps each transmitted symbol to a delay. Consider a scenario where we wish to encode k source symbols into n channel symbols and transmit them through a channel with N erasures, and all k symbols must be recovered with delay at most T . However, we also wish to recover some of these k symbols earlier, i.e., with delay smaller than T . Evidently, no symbols can be recovered with delay smaller than N , due to a possible burst of N erasures. But, for example, could we recover $k - 1$ symbols with delay N , and only the remaining 1 symbol with delay T ? Or, more generally, how early can we recover each one of the k symbols? Alternatively, if there is an upper limit on how many symbols can be recovered with a certain delay (for example, if only a small fraction should be recovered with delay T , and most other symbols should be recovered earlier), how does that impact the total number of symbols k ? These questions are answered by our analysis in Section IV. Further, we show that, among systematic codes, the results presented in Section IV are optimal, under a strict optimality definition.

These results have direct applications in single-user multiplexed coding [32], in which a user wishes to transmit different streams with different delays to a destination. In [32], only two possible delays (for two distinct streams) are considered under a channel that allows for both isolated or burst erasures, while our results are for a general number of streams and delays, but consider only isolated erasures. Furthermore, the results in Section IV provide the tools necessary to sequentially construct codes in the multi-user setting studied in this paper.

3) *Rate Region Achievability Analysis*: As our final main contribution, we present a detailed analysis of the achievable rate region in our setting. The following theorem provides a lower bound on the capacity region in any regime of operation

Theorem 2: For any $R_1 \in [0, C(T - N_3, N_1)]$, we have

$$R_2 \geq \min(C(T - N_3, N_2), C(T - N_2, N_3) - R_1, R_2') \quad (10)$$

where

$$R_2' = \frac{N_2}{N_1} \left[\frac{N_1}{N_2} - 1 - R_1 + \frac{T + 1 - N_1 - N_3}{T + 1 - N_2} \right]. \quad (11)$$

However, we note that there is a gap between our lower and upper bounds for certain choices of N_1 , N_2 , N_3 and T . In the weak relay-destination bottleneck regime, we study the conditions for which the upper bound can be achieved and present a sufficient condition for a rate pair to be part of the capacity region. Further, aiming to reduce the gap when this condition is not met, we propose an optimization algorithm for which we present numerical results. This optimization leads to an improved lower bound, albeit still with some gap to the upper bound. We note that the results from Theorem 2, as well as the output of the optimization, hold for all regimes of operation. However, the underlying code construction used in Theorem 2 is designed with the relay-destination bottleneck

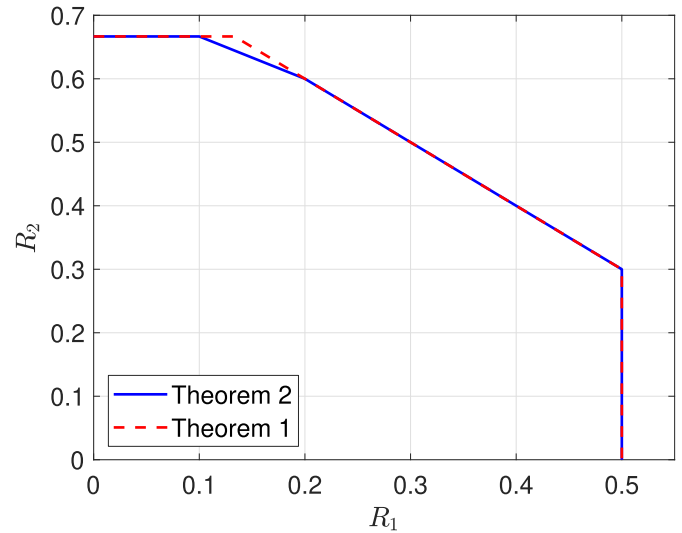


Fig. 2. Rate region for $N_1 = 3, N_2 = 2, N_3 = 1, T = 6$, where R_1 is the rate of the first user and R_2 is the rate of the second user. Theorem 1 represents the upper bound, while Theorem 2 represents a lower bound.

regime in mind, and therefore it may be suboptimal in the source-relay bottleneck regimes.

To illustrate, we compare Theorem 2 and Theorem 1 in Fig. 2. As can be seen, our lower bound is able to achieve a large portion of the upper bound. In Section V, we present two baseline schemes, and then, in Section VI, we revisit Fig. 2, comparing our schemes against these baseline schemes in detail. One particularly interesting result is that neither baseline scheme is able to achieve the sumrate upper bound in any non-trivial setting, while our schemes are able to.

In Section VII, we extend the capacity analysis for the remaining regimes of operation (other than the weak relay-destination bottleneck), also considering the baseline schemes that work well in these regimes. In particular, in both the “strong” bottleneck regimes, we derive the capacity region, that is, there is no gap between the upper and lower bounds, or, in other words, Theorem 1 is tight. In the weak source-relay bottleneck, we similarly derive a (sufficient) condition on the parameters N_1 , N_2 , N_3 and T such that the entire capacity region can be achieved.

Remark 1: The framework used in this paper in order to derive the lower bound presented in Theorem 2 and the optimization algorithm can be easily extended to scenarios with multiple users, or to a setting in which each user is subject to different delay constraints, as the tools developed in Section IV can be used to construct codes sequentially in each link, while the constraints imposed by the already-constructed codes will ensure that the overall delay constraints are always satisfied. Thus, we believe the contributions made in this paper are significantly broader than the particular setup we study.

III. UPPER BOUND

In this section, we present an upper bound on the achievable rate region. We note that, as mentioned in Section II-A, the results do not hold in general, but they do under the channel state independent assumption. This was a mistake in

the previous paper [2], which we address now by providing a formal proof from first principles in the appendix, and highlighting where the proof requires the channel state independent assumption. The high level sketch is presented in this section.

We denote by $C(T, N) = \frac{T+1-N}{T+1}$ the capacity of a single-link point-to-point channel [5]. Since each user is transmitting its own message without cooperation, we have that the following two bounds are direct extensions from [2]

$$\begin{aligned} R_1 &\leq C(T - N_3, N_1) \\ R_2 &\leq C(T - N_3, N_2). \end{aligned}$$

This is because a burst of N_3 erasures may occur from time $T - N_3 + 1$ up to T , therefore, the relay must have access to all the information about the source packets $s_{t,i}$ at time $T - N_3$, otherwise, it will not be able to relay the remaining information. Thus, we can bound each link from source i to relay as a point-to-point link with effective delay $T - N_3$ and N_i erasures, obtaining the bound above.

Furthermore, we can optimistically consider N_2 erasures in both links in the first hop, and obtain the following upper bound on the sumrate

$$R_1 + R_2 \leq C(T - N_2, N_3).$$

This follows from a similar argument. Because the relay is required to be time-invariant¹, the FEC code employed by the relay may assume that it will only recover any information about the source packets, at best, at time $t + N_2$, since a burst of N_2 erasures may happen in both links from sources to relay. Then, the link from relay to destination can be bounded by a point-to-point link with effective delay $T - N_2$ that must handle N_3 erasures.

IV. SYMBOL-WISE DECODE AND FORWARD AND DELAY SPECTRUM

In order to present our coding scheme, first let us define the notion of delay spectrum for a point-to-point, single-link code. This notion exists and is mentioned in [2], however, we make an in-depth analysis of this concept, which has not been made before. Similar analysis has been done in works such as [32], where a source wishes to transmit two streams with different delays to a destination, however, we generalize it for any number of different streams and delays, focusing on the arbitrary erasure channel instead of the burst channel.

At a high level, the delay spectrum of a (point-to-point) code represents the (worst-case) delay associated with each source symbol, rather than the delay associated with the source packet. For example, a source packet may be recoverable only with delay $T = 3$, but some of its symbols may be recovered earlier, with delay 1 or 2. A packet is only fully recovered when all of its symbols are recovered, thus the delay of a packet is equal to the maximum delay in its delay spectrum. This idea has been used in [2] and in the current paper to justify that the relay should not wait until the entire packet

¹If the relaying function is not required to be time-invariant, higher rates can be achieved, at the cost of overhead and an adaptive relaying scheme that depends on the erasure pattern observed in each link. This was missed in [2]

is recovered before transmitting information about that packet to the destination. However, in [2], only a specific family of codes was considered, which achieves a particular ‘‘uniform’’ delay spectrum, recovering one symbol at every time instant $\{N, N + 1, \dots, T\}$. Unfortunately, this is not sufficient in our setting, and we must consider a broader set of achievable delay spectra. In this section, we study a set of ‘‘optimal’’ delay spectra, which are defined formally later, but, before doing so, let us formally define the delay spectrum and relate its use to symbol-wise decode-and-forward for a relayed setting.

Definition 8: An $(n, k, \mathbf{T}, M)_{\mathbb{F}}$ point-to-point code, where $\mathbf{T} = [T[1], \dots, T[k]]$ is the delay spectrum of the code, consists of the following:

- 1) A sequence of source messages $\{s_t\}_{t=0}^{\infty}$, where $s_t \in \mathbb{F}^k$.
- 2) An encoding function

$$f : \underbrace{\mathbb{F}^k \times \dots \times \mathbb{F}^k}_{M+1 \text{ times}} \rightarrow \mathbb{F}^n$$

used by the transmitter at time t to generate

$$x_t = f(s_{t-M}, s_{t-M+1}, \dots, s_t).$$

- 3) A list of k decoding functions

$$\varphi_{t+T[j]} = \underbrace{\mathbb{F}_e^n \times \dots \times \mathbb{F}_e^n}_{t+T[j]+1 \text{ times}} \rightarrow \mathbb{F}^k$$

where $j \in \{1, 2, \dots, k\}$ is the index of the j th symbol (element of s_t), used by the receiver at time $t + T[j]$ to generate $\hat{s}_t[j]$, that is, an estimation of the j th element of s_t .

Similar to before, we assume $M \rightarrow \infty$ and refer only to $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point streaming codes.

Definition 9: An $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point code is said to achieve delay spectrum \mathbf{T} under N erasures if, for any $e'_t \in \Omega_N$,

$$\varphi_{t+T[j]}(h_n(x_0, e'_0), \dots, h_n(x_{t+T[j]}, e'_{t+T[j]})) = s_t[j]$$

For the relaying strategy, let us now introduce the concept of symbol-wise decode-and-forward. In this strategy, the relaying function employed by the code first decodes the source packets transmitted by the source, and then encodes them again. This is an extension of the symbol-wise decode-and-forward defined in [2] for the three-node network. However, the addition of a second source node adds some nuances to the strategy, as the messages relayed by the relay now must be multiplexed in some way. Below, we formally define this strategy.

For the remaining of the paper, we use $\mathbf{T}_i^{(1)}$ to denote the delay spectrum of the code used by the source node i and $\mathbf{T}^{(2)}$ to denote the delay spectrum of the code employed by the relay.

Definition 10: Assume the source nodes transmit their source messages $\{s_{t,i}\}_{t=0}^{\infty}$ to the relay using an $(n_i, k_i, \mathbf{T}_i^{(1)})_{\mathbb{F}}$ point-to-point, single-link code with decoding functions $\varphi_{t+T_i^{(1)}[j],i}^{(1)}$. Then, a relay is said to employ a symbol-wise decode-and-forward if the following holds:

- The relay employs the decoding functions $\varphi_{t+T_i^{(1)}[j],i}^{(1)}$ at time $t + T_i^{(1)}[j]$ to estimate the j -th source symbol of each source packet $\hat{s}_{t,i}[j]$.

TABLE I

EXAMPLE OF SYMBOL-WISE DECODE-AND-FORWARD FOR $N_1 = 1, N_2 = 1, N_3 = 1, T = 2$. HIGHLIGHTED IN BLUE IS PACKET $s_{0,1}$, WHICH IS RECOVERED AT TIME $t = 2$ BY THE DESTINATION

(a) Code used from source i to relay

0	1	2	3	4	5	6	7	8
$s_{0,i}[1]$	$s_{1,i}[1]$	$s_{2,i}[1]$	$s_{3,i}[1]$	$s_{4,i}[1]$	$s_{5,i}[1]$	$s_{6,i}[1]$	$s_{7,i}[1]$	$s_{8,i}[1]$
	$s_{0,i}[1]$	$s_{1,i}[1]$	$s_{2,i}[1]$	$s_{3,i}[1]$	$s_{4,i}[1]$	$s_{5,i}[1]$	$s_{6,i}[1]$	$s_{7,i}[1]$

(b) Code used from relay to destination

0	1	2	3	4	5	6	7
	$s'_1[1] = s_{0,1}[1]$	$s'_2[1] = s_{1,1}[1]$	$s'_3[1] = s_{2,1}[1]$	$s'_4[1] = s_{3,1}[1]$	$s'_5[1] = s_{4,1}[1]$	$s'_6[1] = s_{5,1}[1]$	$s'_7[1] = s_{6,1}[1]$
		$s'_1[1]$	$s'_2[1]$	$s'_3[1]$	$s'_4[1]$	$s'_5[1]$	$s'_6[1]$
	$s'_1[2] = s_{0,2}[1]$	$s'_2[2] = s_{1,2}[1]$	$s'_3[2] = s_{2,2}[1]$	$s'_4[2] = s_{3,2}[1]$	$s'_5[2] = s_{4,2}[1]$	$s'_6[2] = s_{5,2}[1]$	$s'_7[2] = s_{6,2}[1]$
		$s'_1[2]$	$s'_2[2]$	$s'_3[2]$	$s'_4[2]$	$s'_5[2]$	$s'_6[2]$

- The relay employs an $(n_3, k_1 + k_2, \mathbf{T}^{(2)})_{\mathbb{F}}$ point-to-point, single-link code to transmit a sequence of relay messages $\{s'_t\}_{t=0}^{\infty}$ to the destination.
- The relay messages are given by

$$s'_t[j'] = \hat{s}_{t-(T^{(1)}[j]),1}[j_1] \quad (12)$$

where $j_1 \in \{1, 2, \dots, k_1\}$, or

$$s'_t[j'] = \hat{s}_{t-(T^{(1)}[j]),2}[j_2] \quad (13)$$

where $j_2 \in \{1, 2, \dots, k_2\}$.

This definition simply implies that each source employs a point-to-point code, the relay attempts to decode the symbols from each source, then re-encodes them together using another point-to-point code. Before we continue, let us give a brief and simple example of a symbol-wise decode-and-forward code designed for channel parameters $N_1 = 1, N_2 = 1, N_3 = 1$ and $T = 2$. Both source nodes employ the point-to-point link presented in Table Ia. Then, the relay decodes every symbol with a delay of 1, that is, each symbol $s_{t,i}[1]$ is recovered at time $t + 1$. The relay then employs the following rule: $s'_t[1] = s_{t-1,1}[1]$ and $s'_t[2] = s_{t-1,2}[1]$. That is, the relay will now encode using a point-to-point code with $k = 2$. The first symbol of this code will be a delayed version of the source symbol coming from the first user, while the second symbol of this code will be a delayed version of the source symbol coming from the second user. This code is shown in Table I. In this example, we have $k_1 = 1, k_2 = 1, n_1 = n_2 = 2$ and $n_3 = 4$. Thus, we have $R_1 = R_2 = 1/4$. This matches the upper bound $R_1 + R_2 \leq 1/2$. From the table, it can be seen that, even with any one erasure from sources to relay, the relay is still able to recover the desired symbols with a delay of 1 time instant. Similarly, the destination is able to recover any symbol with delay at most $T = 2$ with any one erasure from relay to destination.

Furthermore, we define a concatenation of point-to-point codes.

Definition 11: A concatenation of an $(n', k', \mathbf{T}')_{\mathbb{F}}$ point-to-point code with an $(n'', k'', \mathbf{T}'')_{\mathbb{F}}$ point-to-point code is an $(n' + n'', k' + k'', [\mathbf{T}', \mathbf{T}''])_{\mathbb{F}}$ point-to-point code with the following properties

- Let $\{f'_t\}$ be the encoding function of the first code and $\{f''_t\}$ be the encoding function of the second code. The encoding function of the concatenated code is given by

$\{[f'_t, f''_t]\}$, where $[x, y]$ denotes the concatenation of a vector x and a vector y .

- Let $\{\varphi'_{t+T'[j]}\}_{j=1}^{k'}$ be the list of decoding functions of the first code and $\{\varphi''_{t+T''[j]}\}_{j=1}^{k''}$ be the list of decoding functions of the second code. The list of decoding functions of the concatenated code is given by $\{\varphi'_{t+T'[j]}\}_{j=1}^{k'} \cup \{\varphi''_{t+T''[j]}\}_{j=1}^{k''}$, where $x \cup y$ denotes the concatenation of two lists x and y .

In the previous example, in Table I, one can see such concatenation. Note that the code used from relay to destination is a concatenation of two diagonally-interleaved maximum distance separable (MDS) codes [5] with $n' = 2$ and $k' = 1$. This results in a code with $n = 4$ and $k = 2$.

Lemma 1: If there exists an $(n', k', \mathbf{T}')_{\mathbb{F}}$ point-to-point code that achieves delay spectrum \mathbf{T}' under N erasures, and an $(n'', k'', \mathbf{T}'')_{\mathbb{F}}$ point-to-point code that achieves delay spectrum \mathbf{T}'' under N erasures, then there exists an $(n' + n'', k' + k'', [\mathbf{T}', \mathbf{T}''])_{\mathbb{F}}$ point-to-point code that achieves delay spectrum $[\mathbf{T}', \mathbf{T}'']$ under N erasures.

For the remaining of this paper, all proofs that do not appear immediately after the statement of the proposition, lemma or theorem can be found in the appendix, in order of the statement appearance in the paper.

Another useful operation that can be made is simply permuting the source symbols.

Lemma 2: Assume a delay spectrum \mathbf{T} is achievable under N erasures by some code. Then, any permutation $\pi\mathbf{T}$, where π is a permutation matrix, of this delay spectrum is also achievable under N erasures.

Since any permutation of an achievable delay spectrum is also achievable, we may instead describe the delay spectrum of a code by stating how many symbols are transmitted with some delay.

Definition 12: Consider a delay spectrum $\mathbf{T} = [T[1], T[2], \dots, T[k]]$. An equally-delayed-symbols grouping description of such delay spectrum is given by a list of tuples

$$\mathbf{G} = [(T^{(g)}[1], k^{(g)}[1]), \dots, (T^{(g)}[\ell^{(g)}], k^{(g)}[\ell^{(g)}])]$$

where $\ell^{(g)}$ is the length of the list. For simplicity, we assume $T^{(g)}[1] \geq T^{(g)}[2] \geq \dots \geq T^{(g)}[\ell^{(g)}]$, therefore, $T^{(g)}[1] = \max(\mathbf{T})$ and $T^{(g)}[\ell^{(g)}] = \min(\mathbf{T})$. Furthermore, we define

- $\mathbf{T}^{(g)} = [T^{(g)}[1], \dots, T^{(g)}[\ell^{(g)}]]$ as the ordered list of possible delays.
- $\mathbf{k}^{(g)} = [k^{(g)}[1], \dots, k^{(g)}[\ell^{(g)}]]$, where $\sum_{i=1}^{\ell^{(g)}} k^{(g)}[i] = k$, as the ordered list of number of symbols associated with each delay.

Again, referring to the previous example, we could describe the code used in Table I with $\mathbf{G} = [(1, 2)]$, that is, two symbols are recovered with delay 1.

A. Single-Link Point-to-Point Results for Delay Spectrum

In this Section, we present an achievability result for single-link point-to-point codes in terms of delay spectrum.

Lemma 3 (Achievability): Let

$$\mathbf{T}^{(g)} = [T^{(g)}[1], T^{(g)}[2], \dots, T^{(g)}[\ell^{(g)}]] = [T^{(g)}[1], T^{(g)}[1] - 1, \dots, N + 1, N]$$

and $\frac{n-k}{N}$ be an integer. Then, there exists a systematic $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point single-link code that can transmit $k^{(g)}[1] = n - \frac{T^{(g)}[1]}{N}(n-k)$ symbols with delay $T^{(g)}[1]$ and $k^{(g)}[j] = \frac{n-k}{N}$ for all other delays.

The above Lemma can be used to derive the following condition on achievability:

Corollary 1: Assume there is a maximum number of symbols we are allowed to transmit at each delay $T^{(g)}[j]$, that is, $\mathbf{k}^{(g)} \leq \mathbf{k}^{\text{con}}$ is a constraint. Then, if

$$k \leq n - n \cdot N \cdot \frac{\left(1 - \sum_{\ell=1}^{j-1} \frac{k^{\text{con}}[\ell]}{n}\right)}{T^{(g)}[j] + 1} \quad \forall j \in \{1, 2, \dots, \ell^{(g)}\} \quad (14)$$

$$k \leq \sum_{\ell=1}^{\ell^{(g)}} k^{\text{con}}[\ell] \quad (15)$$

there exists an $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point single-link code that achieves the desired delay spectrum $\mathbf{T}^{(g)}$ under N erasures.

The usefulness of this corollary should be clearer when we present our Fixed-Bottleneck Symbol-wise Decode-and-forward scheme.

Furthermore, we show a lower bound on the delay spectrum.

Lemma 4 (Lower Bound): For a systematic point-to-point $(n, k)_{\mathbb{F}}$ single-link code subject to N erasures, the delay of the j -th equally-delayed group is lower bounded by

$$\mathbf{T}^{(g)}[j] \geq \frac{Nn}{n-k} \left(1 - \sum_{\ell=1}^{j-1} \frac{k^{(g)}[\ell]}{n}\right) - 1. \quad (16)$$

Corollary 2: Assume $\mathbf{T}^{(g)} = [T^{(g)}[1], T^{(g)}[1] - 1, \dots, N]$, i.e., the possible delays are separated by one, and the minimum delay is N . Then, the following holds for any $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point single-link code under N erasures

$$k^{(g)}[1] \geq n - \frac{T^{(g)}[1]}{N}(n-k), \quad \text{and} \quad k^{(g)}[\ell^{(g)}] \leq \frac{n-k}{N}$$

The implication of Corollary 2 is that there is a minimum number of symbols that need to be transmitted at the worst possible delay (i.e., $T^{(g)}[1]$), and a maximum number of symbols that can be transmitted at the smallest possible delay.

This already suggests that there is some limit to the achievable delay spectrum, and that we can not have almost all symbols transmitted with a very small delay, and very few symbols with a large delay. We now wish to understand what is the best delay spectrum we can achieve, but first, let us define the notion of “best”, or optimal, delay spectrum.

Definition 13: In this definition, for consistency, we assume all delay spectra are ordered in increasing order. Assume there exists an $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point streaming \mathcal{C} that achieves delay spectrum \mathbf{T} under N erasures. We say that the delay spectrum \mathbf{T} is optimal if and only if, for any \mathbf{T}' such that there exists an $(n, k, \mathbf{T}')_{\mathbb{F}}$ point-to-point streaming code that achieves delay spectrum \mathbf{T}' under N erasures, the following holds:

$$\mathbf{T} \leq \mathbf{T}' \quad (17)$$

where inequality is taken element-wise. In other words, every symbol is recovered by \mathcal{C} with delay less than or equal to the delay achieved by any other streaming code with the same rate, blocklength and number of erasures.

Proposition 1: Let \mathcal{C} be an $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point streaming code that achieves delay \mathbf{T} under N erasures, and let \mathcal{C}' be an $(n, k, \mathbf{T}')_{\mathbb{F}}$ point-to-point streaming code that achieves delay \mathbf{T}' under N erasures. Let the equally-delayed-symbols grouping description of \mathbf{T} be $(\mathbf{T}^{(g)}, \mathbf{k}^{(g)})$, while for \mathbf{T}' it is $(\mathbf{T}'^{(g)}, \mathbf{k}'^{(g)})$. In particular, note that $\mathbf{T}^{(g)}$ is the same for both. This is without loss of generality.²

Then

$$\sum_{\ell=j}^{\ell^{(g)}} k^{(g)}[\ell] \geq \sum_{\ell=j}^{\ell^{(g)}} k'^{(g)}[\ell] \quad \forall j \quad (18)$$

implies $\mathbf{T} \leq \mathbf{T}'$. In other words, the “better” delay spectrum transmits more symbols with lower delay.

Proof: To see it, recall that we assumed that \mathbf{T} and \mathbf{T}' are ordered. Let us prove it by showing that, if $T'[\ell'] < T[\ell]$ for some ℓ , then (18) does not hold for some j . Let ℓ' be the first symbol for which $T'[\ell'] < T[\ell]$. Let j be such that $T^{(g)}[j] = T'[\ell']$. We now count the number of symbols that are transmitted with delay up to $T'[\ell']$ in each code, from which it follows that

$$\sum_{\ell=j}^{\ell^{(g)}} k'^{(g)}[\ell] \stackrel{(a)}{\geq} \ell' \quad (19)$$

$$> \ell' - 1 \quad (20)$$

$$\stackrel{(b)}{\geq} \sum_{\ell=j}^{\ell^{(g)}} k^{(g)}[\ell] \quad (21)$$

where (a) follows from the fact that, certainly, all ℓ' symbols from 1 to ℓ' in code \mathcal{C}' are transmitted with delay at most $T'[\ell']$, because the symbols are ordered; and (b) follows from the fact that certainly the ℓ' th symbol from \mathcal{C} has a delay larger than $T'[\ell']$, therefore, at most the previous $\ell' - 1$ symbols have delay equal to or lower than $T'[\ell']$. This completes the proof. ■

²This is because we can always arbitrarily add an element $T^{(g)}[j]$ to $\mathbf{T}^{(g)}$ and have the corresponding $k^{(g)}[j]$ to be equal 0.

Example 1: Consider a code with $k = 6$, $n = 10$, $N = 1$, and $T^{(g)}[1] = 2$. Then, consider a code constructed as in Lemma 3. This code transmits $k^{(g)}[2] = 4$ symbols with delay 1, and $k^{(g)}[1] = 2$ symbols with delay 2. Let us compare this with a code with uniform delay spectrum³ that transmits $k'^{(g)}[2] = 3$ symbols with delay 1, and $k'^{(g)}[1] = 3$ symbols with delay 2. Intuitively, the first code is better. But let us arrange their delay spectra in increasing order. We have $\mathbf{T} = [1, 1, 1, 1, 2, 2]$, and $\mathbf{T}' = [1, 1, 1, 2, 2, 2]$. We then note that $T[4] < T'[4]$, which is our definition of a “better” delay spectrum. In fact, as we will see below, the first code is optimal in terms of delay spectrum, i.e., there is no other code that can transmit **any** of the k symbols with a smaller delay without decreasing the rate.

Remark 2: A natural question to ask is: does the optimal delay spectrum (always) exist? That is, is there a delay spectrum that is better than every other, for every single symbol? At first, this notion of optimality might seem too strict, however, Theorem 3 below answers this question in the affirmative for systematic codes, and in fact shows that such optimal delay spectrum matches Lemma 3.

Remark 3: The assumption that the delay spectra of the codes are ordered is justified by Lemma 2, which states that permutations of an achievable delay spectrum are also achievable. Furthermore, due to this fact, this assumption is also required, as otherwise optimality would be ill-defined for any code with more than one possible delay. This is because we would always be able to find a permutation of the code that dominates the original code in at least one position, but is dominated in another. For example, $\mathbf{T} = [1, 2]$ and $\mathbf{T}' = [2, 1]$.

Theorem 3 (Optimal Delay Spectrum): Assume $\mathbf{T}^{(g)} = [T^{(g)}[1], T^{(g)}[1] - 1, \dots, N]$, i.e., the possible delays are separated by one⁴, and the minimum delay⁵ is N . Further, assume⁶ $n - k$ is a multiple of N . Then, the optimal number of symbols transmitted with each delay under N erasures, for any $(n, k, \mathbf{T})_{\mathbb{F}}$ point-to-point systematic single-link code is given by

$$\begin{aligned} k^{(g)}[1] &= n - \frac{T^{(g)}[1]}{N}(n - k) \\ k^{(g)}[j] &= \frac{n - k}{N}, \forall j \in \{2, 3, \dots, \ell^{(g)}\}. \end{aligned} \quad (22)$$

In other words, the delay spectrum described by the above equally-delayed-symbols grouping description is optimal.

One can see that the result from Theorem 3 matches the one from Lemma 3, that is, for a systematic point-to-point code, the delay spectrum achieved by Lemma 3 is optimal.

Remark 4: The codes presented in [2] have the property $n - k = N$, and achieve a delay spectrum of the form $k^{(g)}[j] = \frac{n - k}{N} = 1$ for all j , including $j = 1$. Further, they have the

³This code could be constructed with only $n = 9$, but for the example, we assume a suboptimal code that uses $n = 10$.

⁴This assumption is without loss of generality, as $k^{(g)}[i] = 0$ can be set for some delay $T^{(g)}[i]$, and this would be equivalent to having a larger gap between the adjacent delays.

⁵This assumption is trivial, as no symbol can be recovered certainly with delay less than N , as N erasures in a burst may occur.

⁶Again, this is, in theory, without loss of generality, as n and k are design parameters and can be set to Nk' and Nn' , maintaining the rate of the code.

property that $T^{(g)}[1] = n - 1$. These properties, together, make them optimal. However, these are limiting properties, and the results presented in this section are more general. Furthermore, this is the first time optimality of such codes is shown.

V. ACHIEVABLE RATE REGION

In this section, we present lower bounds on the capacity region. For now, let us make the following assumptions on the parameters:

- $N_1 \geq N_2 \geq N_3$. Recall that we had already assumed $N_1 \geq N_2$ without loss of generality.
- $T \geq \frac{1}{2} \left(\sqrt{N_1^2 - 4N_3(N_2 - N_3)} + N_1 + 2N_2 - 2 \right)$.

These conditions ensure that we are operating in the weak relay-destination bottleneck regime⁷, as defined in Section II. Later, in Section VII, we study all the regimes of operation.

We start the section by introducing a novel scheme, in which the relay-destination link attempts to transmit at its maximal rate, denoted as Fixed-Bottleneck Symbol-Wise Decode-and-Forward (FB-SWDF). This scheme provides us with the lower bound in Theorem 2. Then, we formally present tools that allow us to naively extend the known schemes for the three-node network to a multi-access network. In particular, one of these schemes—denoted Concatenated Symbol-Wise Decode-and-Forward (CSWDF)—is able to outperform our FB-SWDF scheme for some rate pairs. This is due to the attempt of transmitting at maximal rate. Finally, we present an optimization approach, which allows us to achieve the highest known rates.

Examples of the rates achieved by each scheme and the code construction of our FB-SWDF are presented in Section V-D.

A. Fixed-Bottleneck Symbol-Wise Decode-and-Forward

To give a general idea of the scheme, let us first attempt to fix the code employed by the relay. Let us use a code with rate $R_{bn} = C(T - N_2, N_3)$, where R_{bn} is the rate of the single-link code employed by the relay. For some choice of n_3 , we then can find the delay spectrum of this code by employing Lemma 3. Note that the delay spectrum of the code employed by the relay imposes a constraint on the delay spectrum of the codes employed by the source nodes.

Further, let us also fix some code for the first user. In particular, let us attempt to transmit with rate $R_1 = C(T - N_3, N_1)$, i.e., its single-user capacity. Then again, for some choice of n_1 , using Lemma 3, we are able to find the delay spectrum of this code. This will, in turn, update the imposed constraints on the delay spectrum of the second user.

Finally, with the updated delay constraints for the second user, we can apply Corollary 1 and find an achievable rate for R_2 given the delay constraints imposed by the codes used in the relay and from the first source node to relay.

With the correct choice of n_1 , n_2 and n_3 , this coding scheme allows us to derive the following Lemma.

Lemma 5: For $R_1 = C(T - N_3, N_1)$ the following rate is achievable in the second link

$$R_2 = \min(R_2', C(T - N_2, N_3) - C(T - N_3, N_1)) \quad (23)$$

⁷The assumptions are just solving for the condition $C(T - N_3, N_1) + C(T - N_3, N_2) \geq C(T - N_2, N_3) \geq C(T - N_3, N_2)$.

where

$$R'_2 = 1 - \frac{N_2(1 - (T + 1 - N_3 - N_1)\frac{(N_2 - N_3)}{(T+1-N_2)(T+1-N_3)})}{N_1} \quad (24)$$

One can notice that this Lemma is a particular case of Theorem 2. However, we believe that introducing this coding scheme first, and then generalizing it, allows a clearer understanding of the steps required in order to achieve the rates presented. Furthermore, this Lemma is useful because it allows us to derive the following corollary, which gives a sufficient condition for the sumrate to be achieved for at least one rate pair.

Corollary 3: If $(T + 1 - N_3)(T + 1 - N_2 - N_1) \geq (T + 1 - N_3 - N_1)(N_2 - N_3)$, then the sumrate capacity $R_1 + R_2 = C(T - N_2, N_3)$ is achieved at least one point of the rate region.

Now, note that we can follow the steps mentioned in this section for any R_{bn} and R_1 . In fact, Theorem 2 is obtained by simply considering a general R_1 , instead of $R_1 = C(T - N_3, N_1)$. In the derivation of Theorem 2, we actually consider a code that might seem suboptimal for the first source node—we employ a slightly deteriorated delay spectrum, in order to obtain a uniform delay spectrum. For example, instead of transmitting 2 symbols with delay 1 and zero symbols with delay 2, we may opt to transmit 1 symbol with each delay (i.e., one symbol with delay 1, and another symbol with delay 2). Numerical results show that this choice of code does not, in fact, decrease the rate achieved by the second user. That is, the achievable rate region obtained by employing such codes in the link from first user to relay or by employing an optimal-delay-spectrum code is the same. Below, we restate Theorem 2.

Theorem: For any R_1 , the following rate is achievable in the second link in the first hop

$$R_2 = \min(C(T - N_3, N_2), C(T - N_2, N_3) - R_1, R'_2) \quad (25)$$

where

$$R'_2 = \frac{N_2}{N_1} \left[\frac{N_1}{N_2} - 1 - R_1 + \frac{T + 1 - N_1 - N_3}{T + 1 - N_2} \right] \quad (26)$$

Remark 5: We note that the rate achieved by the second user in Theorem 2 is the minimum between three rates. The first rate, $C(T - N_3, N_2)$, is the single-user bound of the second user. This term was not present in Lemma 5 because, from the assumption that we are in the relay-destination bottleneck regime, thus for $R_1 = C(T - N_3, N_1)$, the sum-rate bound is always active, instead of the single-user bound. The second term is just the sum-rate bound, as we have $R_1 + R_2 \leq C(T - N_2, N_3)$. This is the same as the first term in Lemma 5, with $R_1 = C(T - N_3, N_1)$. The third term, R'_2 , is less straight-forward, and comes from the combination of the delay spectrum of the two codes. One can verify, however, that substituting $R_1 = C(T - N_3, N_1)$ will lead to Lemma 5.

Furthermore, we are also interested in knowing which part of the capacity region (in particular, in the sumrate region) is achievable, when some of it is. This is provided by the

following Lemma, which provides another achievable rate pair. This rate pair is simply the intersection between the line defined by R'_2 in Theorem 2 and the sumrate bound.

Lemma 6: If the sumrate capacity is achieved at at least one point of the rate region, then the following point is also achievable

$$R_1 = \frac{N_2 - N_3}{T + 1 - N_2} \quad (27)$$

$$R_2 = \frac{T + 1 - 2N_2}{T + 1 - N_2} \quad (28)$$

The complete proofs of all the statements in this section are presented in the Appendix, with explicit description of the code parameters used.

B. Extensions of Known Schemes

In this section, we present a time-sharing-like tool that allows us to generalize the results in [2], and can also be used to define achievable regions based on achievable points.

Using the following Lemma, we can extend the schemes presented in [2] and, generally, show that if any two points are achievable, (practically) any linear combination between such two points is also achievable.

Lemma 7: The following lemma holds for symbol-wise decode-and-forward: assume there exists an (N_1, N_2, N_3) -achievable $(n_1, n_2, n_3, k_1, k_2, T)_{\mathbb{F}}$ -streaming code, and another (N_1, N_2, N_3) -achievable $(n'_1, n'_2, n'_3, k'_1, k'_2, T)_{\mathbb{F}}$ -streaming code. Then, for any $A, B \in \mathbb{Z}_+$, there exists an (N_1, N_2, N_3) -achievable $(An_1 + Bn'_1, An_2 + Bn'_2, An_3 + Bn'_3, Ak_1 + Bk'_1, Ak_2 + Bk'_2, T)_{\mathbb{F}}$ -streaming code.

Remark 6: While Lemma 7 allows us to achieve any convex combination of two pairs of rates (R_1, R_2) and (R'_1, R'_2) , it should be noted that, because n is defined as the maximum among n_1, n_2 and n_3 , it is possible to achieve rates higher than the simple convex combination when the largest n is different in each point of operation (i.e., each rate pair). This can be seen in Section VI for both the CSWDF and Concatenated Message-wise Decode-and-forward (CMWDF) schemes, where the achievable rate region of such schemes is not a straight line connecting the two single-user rates.

1) *Concatenated Symbol-Wise Decode-and-Forward:* The first scheme we consider is an extension of the single-user capacity-achieving codes presented in [2], which employ symbol-wise decode-and-forward in the three-node network. It is known from [2] that there exists an (N_1, N_2, N_3) -achievable code with the following parameters:

$$\begin{aligned} n_1 &= T + 1 - N_3, & n_2 &= 0, & n_3 &= T + 1 - N_1 \\ k_1 &= T + 1 - N_1 - N_3, & k_2 &= 0. \end{aligned} \quad (29)$$

Similarly, there exists an (N_1, N_2, N_3) achievable code with the following parameters:

$$\begin{aligned} n'_1 &= 0, & n'_2 &= T + 1 - N_3, & n'_3 &= T + 1 - N_2 \\ k'_1 &= 0, & k'_2 &= T + 1 - N_2 - N_3. \end{aligned} \quad (30)$$

Then, it follows from Lemma 7, that we can concatenate both single-user codes and obtain another (N_1, N_2, N_3) -achievable

streaming code. This is the construction we denote as Concatenated Symbol-wise Decode-and-forward.

Lemma 8: For any $A, B \in \mathbb{Z}_+$ and $T \geq N_1 + N_3$, there exists an (N_1, N_2, N_3) -achievable code with rate pair

$$n = \max(An_1 + Bn'_1, A n_2 + Bn'_2, A n_3 + Bn'_3) \quad (31)$$

$$R_1 = \frac{Ak_1 + Bk'_1}{n} \quad (32)$$

$$R_2 = \frac{Ak_2 + Bk'_2}{n} \quad (33)$$

where $k_1, k'_1, k_2, k'_2, n_1, n'_1, n_2, n'_2, n_3$ and n'_3 are as in (29) and (30).

Proof: The result follows directly from applying Lemma 7 to the capacity-achieving single-user codes known to be achievable from [2], and computing the resulting rate pair of the concatenated code. ■

The following Lemma presents an achievable point which, to the best of our knowledge, is the best achievable rate for R_1 such that $R_2 = C(T - N_3, N_2)$, i.e., the corner point for maximum R_2 . This rate can also be obtained using the Concatenated Symbol-wise Decode-and-forward scheme.

Lemma 9: For $R_2 = C(T - N_3, N_2)$, the following R_1 is achievable

$$R_1 = \frac{(T + 1 - N_3 - N_1)(N_2 - N_3)}{(T + 1 - N_1)(T + 1 - N_3)} \quad (34)$$

Remark 7: Note that

$$\begin{aligned} R_1 &= \frac{(T + 1 - N_3 - N_1)(N_2 - N_3)}{(T + 1 - N_1)(T + 1 - N_3)} \\ &< \frac{(T + 1 - N_3 - N_2)(N_2 - N_3)}{(T + 1 - N_2)(T + 1 - N_3)} \\ &= C(T - N_2, N_3) - C(T - N_3, N_2). \end{aligned}$$

That is, the sum rate at the point from Lemma 9 is strictly lower than the sumrate capacity.

2) *Concatenated Message-Wise Decode-and-Forward:*

Another scheme that is worth analyzing is the Concatenated Message-wise Decode-and-forward (CMWDF). Similar to the CSWDF scheme, we start by finding a single-user achievable streaming code, and then we apply Lemma 7. For this construction, however, we require the delay of all symbols to be equal in each link, that is, the relay must decode the entire source message before it can re-encode it. This effectively separates the communication in two: one with delay \tilde{T} and another with delay $T - \tilde{T}$, where \tilde{T} is a design parameter.

It then follows that there exists an (N_1, N_2, N_3) -achievable code with the following parameters:

$$\begin{aligned} n_1 &= (\tilde{T}_1 + 1)(T - \tilde{T}_1 + 1 - N_3) \\ n_2 &= 0 \\ n_3 &= (T - \tilde{T}_1 + 1)(\tilde{T}_1 + 1 - N_1) \\ k_1 &= (T - \tilde{T}_1 + 1 - N_3)(\tilde{T}_1 + 1 - N_1) \\ k_2 &= 0 \end{aligned} \quad (35)$$

and similarly, an (N_1, N_2, N_3) -achievable code with the parameters

$$\begin{aligned} n'_1 &= 0 \\ n'_2 &= (\tilde{T}_2 + 1)(T - \tilde{T}_2 + 1 - N_3) \\ n'_3 &= (T - \tilde{T}_2 + 1)(\tilde{T}_2 + 1 - N_2) \\ k'_1 &= 0 \\ k'_2 &= (T - \tilde{T}_2 + 1 - N_3)(\tilde{T}_2 + 1 - N_2) \end{aligned} \quad (36)$$

where $\tilde{T}_1 \in \{N_1, \dots, T - N_3\}$ and $\tilde{T}_2 \in \{N_2, \dots, T - N_3\}$ are design parameters. In particular, one can optimize these parameters in order to find the maximum single-user rate that can be achieved using message-wise decode-and-forward.

Lemma 10: For any $A, B \in \mathbb{Z}_+$ and $T \geq N_1 + N_3$, there exists an (N_1, N_2, N_3) -achievable code with rate pair

$$n = \max(An_1 + Bn'_1, A n_2 + Bn'_2, A n_3 + Bn'_3) \quad (37)$$

$$R_1 = \frac{Ak_1 + Bk'_1}{n} \quad (38)$$

$$R_2 = \frac{Ak_2 + Bk'_2}{n} \quad (39)$$

where $k_1, k'_1, k_2, k'_2, n_1, n'_1, n_2, n'_2, n_3$ and n'_3 are as in (35) and (36), substituting \tilde{T}_i with

$$\tilde{T}_i = \arg \max_{T'} \min(\tilde{R}_i, \tilde{R}_{bn}) \quad (40)$$

$$\tilde{R}_i = \frac{T' + 1 - N_i}{T' + 1} \quad (41)$$

$$\tilde{R}_{bn} = \frac{T - T' + 1 - N_3}{T - T' + 1}. \quad (42)$$

Proof: As before, this follows directly from applying Lemma 7 and computing the rate pair. The message-wise decode-and-forward scheme and its achievability are presented in more detail in [2]. ■

C. Optimized-Bottleneck Decode-and-Forward (OB-SWDF)

While Theorem 2 is able to achieve the sumrate capacity, and, in that case, the FB-SWDF scheme presented is optimal, there is a significant loss in rate when we can not achieve the upper bound. This is due to our greedy approach in which the rate of the code used from relay to destination is set to the optimistic $R_{bn} = C(T - N_2, N_3)$. Because we are using a high rate code in this link, it translated into a delay spectrum with more symbol with higher delays, which imposes lower rates in the source-to-relay links.

Unfortunately, analytical expressions of the rates as a function of R_{bn} , similar to Theorem 2, are hard to derive, due to a non-uniformity of the delay spectrum. For that reason, we instead tackle this problem numerically by developing an optimization framework.

Our approach is the following: we will attempt to achieve the highest possible R_2 for some given R_1 . Thus, we start by fixing some $R_1 \in [0, C(T - N_3, N_1)]$. Also, because we define the rate with $n = \max(n_1, n_2, n_3)$, we also simplify the problem, enforcing $n = n_1 = n_2 = n_3$. This is without loss of generality as we can always zero-pad symbols of the

higher-rate codes and achieve the same overall rate. This gives us the following optimization problem

$$R_2^* = \max_{R_2, R_{bn}, n} R_2 \quad (43)$$

$$\text{s.t. } \mathbf{k}_2^{(g)} \leq \mathbf{k}_{bn}^{(g)} - \mathbf{k}_1^{(g)} \quad (44)$$

where $\mathbf{k}_1^{(g)}$, $\mathbf{k}_2^{(g)}$ and $\mathbf{k}_{bn}^{(g)}$ are computed as functions of R_1 (which is fixed), R_2 , R_{bn} and n as in Lemma 3. However, the ordering in $\mathbf{k}_{bn}^{(g)}$ is slightly different from how we defined the equally-delayed-symbols grouping description previously. Now, we consider the following ordering:

$$\mathbf{T}_1^{(g)} = \mathbf{T}_2^{(g)} = [T - N_3, T - N_3 - 1, \dots, N_2 + 1, N_2] \quad (45)$$

$$\mathbf{T}_{bn}^{(g)} = [N_3, N_3 + 1, \dots, T - N_2 - 1, T - N_2]. \quad (46)$$

That is, while the delays for the symbols in the source-to-relay links are ordered in decreasing order, the delays for the relay-to-destination symbols are ordered in increasing order. Note that, by summing the delays, we get an overall delay of T . The constraint in our optimization is simply saying that, if the relay-to-destination can transmit κ symbols with delay τ , then these same κ symbols must be transmitted with delay $T - \tau$ from sources to relay⁸. Further, the constraint states that, if user 1 is already transmitting κ_1 symbols with delay $T - \tau$, then only $\kappa - \kappa_1$ remaining symbols can be transmitted by user 2 with delay $T - \tau$, in order to match the total κ symbols that the relay transmits with delay τ . The choices of \mathbf{T} in equations (45) and (46) are in order to match N_3 with $T - N_3$, $N_3 + 1$ with $T - N_3 - 1$, and so on, such that all symbols are subject to an overall delay of T (by summing the delay in the first part with the one in the second part). The ranges come from the fact that the lowest delay that a symbol can be transmitted in either link from source to relay is N_2 , while the lowest delay that a symbol can be transmitted in the link from relay to destination is N_3 , due to the possibility of a burst of erasures.

Our heuristic approach to solve this optimization problem is described below. First, we choose some large enough n such that for any choice of R_1 , R_2 and R_{bn} we meet the requirement conditions described in Lemma 3. Once R_1 and n are fixed, the delay spectrum of the code in the link from first source to relay is also fixed.

Then, we initialize the optimization algorithm with $R_{bn} = C(T - N_2, N_3)$, which again (together with n) provides us a delay spectrum, and, with the delay spectrum of the first user and the relay codes, we are able to find an achievable R_2 through Corollary 1. Note that the outcome of this initial point is exactly Theorem 2, since we have chosen to initialize with the same R_{bn} . If $R_1 + R_2 = R_{bn}$, that is, if we achieve the sumrate capacity, we stop the optimization algorithm, since nothing else needs to be done.

However, if $R_1 + R_2 < R_{bn}$, then we need to update the code used from relay to destination. We update the rate R_{bn}

⁸The sources could, in principle, transmit with a smaller delay, however, there is no necessity of doing so, thus we keep the formulation simpler by assuming the source will buffer the symbols in order to achieve exactly an overall delay of T . In practice, there is no reason to not transmit with the smallest possible delay.

using a simple bisection-like algorithm. We start by defining a lower bound $R_\ell^1 = R_1 + R_2 < R_{bn}$ and an upper bound $R_u^1 = R_{bn}$. Then, at the i th iteration, we take the following steps

- 1) Update $R_{bn}^i = \frac{R_\ell^{i-1} + R_u^{i-1}}{2}$.
- 2) Compute the delay spectrum of the relay-to-destination code with rate R_{bn}^i using Lemma 3.
- 3) Compute the delay constraint imposed on the code from the second source to the relay given the delay spectra of the codes from first source to relay and from relay to destination.
- 4) Compute R_2^i using Corollary 1.
- 5) If $R_1 + R_2^i = R_{bn}^i$, we update the lower bound $R_\ell^{i+1} = R_{bn}^i$ and keep the upper bound $R_u^{i+1} = R_u^i$. Otherwise, we update the upper bound $R_u^{i+1} = R_{bn}^i$ and keep the lower bound $R_\ell^{i+1} = R_\ell^i$.
- 6) If $R_u^{i+1} - R_\ell^{i+1} > \epsilon$, proceed to the $i + 1$ th iteration. Otherwise, end the algorithm.

Remark 8: From construction, every step will update either the lower bound (increasing it) or the upper bound (decreasing it). Each bound indeed represents a bound on our algorithm: the lower bound represents an achievable rate (i.e., indeed a lower bound), while the upper bound represents an upper bound on the achievable rate using our framework. Due to the former, the algorithm can only improve upon the FB-SWDF (which is the starting lower bound). Furthermore, the algorithm is guaranteed to eventually converge, as the gap reduces by a factor of $1/2$ every step.

Our experiments have shown that this algorithm converges fairly quickly to a high precision degree (e.g. $\epsilon = 10^{-5}$). Further, this algorithm provides the best achievable rates known to the authors. The rate outputs of the algorithm can be seen in Section VI.

D. Examples

In this example, we use the channel parameters $N_1 = 3$, $N_2 = 2$, $N_3 = 1$, $T = 6$. In order to present numerical values and a code construction, we are going to focus the example in the scenario where the first user wishes to transmit at its single-user capacity, that is, $R_1 = C(T - N_3, N_1) = 0.5$. Then, we wish to construct the codes for each user and the relay in order to maximize the rate achieved by the second user.

1) *Upper Bound:* By applying Theorem 1, we have that $R_2 \leq 0.3$. This not only gives us a benchmark, but is also useful in the code construction of our FB-SWDF, as we will greedily attempt to achieve that rate.

2) *FB-SWDF:* As explained previously, we start by fixing the rate of the code from relay to destination at $R_{bn} = C(T - N_2, N_3) = 0.8$. We also choose some value of n , in this case, $n = 30$. This choice is explained in detail in the proof of Lemma 5. With the desired rates R_1 , R_{bn} and the blocklength n , we are able to construct codes, as detailed in the proof of Lemma 3, that achieve a certain delay spectrum. In this example, the code from relay to destination is able to transmit six symbols with each delay, from 1 to 4, and the code employed by the first user transmits five symbols with

TABLE II
CAPACITY-ACHIEVING FB-SWDF CODES FOR $N_1 = 3$, $N_2 = 2$, $N_3 = 1$ AND $T = 6$

(a) Source to Relay codes. Highlighted in blue is an example of a diagonal. Note that the super-symbols $a_t[1:5]$, $b_{t+1}[1:5]$ and $c_{t+2}[1:5]$ are recovered certainly at time $t+5$, since at most $N_1 = 3$ erasures may occur. Thus a_t is subject to a delay of 5, b_{t+1} to a delay of 4 and c_{t+2} to a delay of 3 time instants

User 1: Source to relay.						
Symbol \ Time	t	t+1	t+2	t+3	t+4	t+5
$x_{t,1}^{(1)}[1] = s_{t,1}[1]$	$a_t[1:5]$	$a_{t+1}[1:5]$	$a_{t+2}[1:5]$	$a_{t+3}[1:5]$	$a_{t+4}[1:5]$	$a_{t+5}[1:5]$
$x_{t,1}^{(1)}[2] = s_{t,1}[2]$	$b_t[1:5]$	$b_{t+1}[1:5]$	$b_{t+2}[1:5]$	$b_{t+3}[1:5]$	$b_{t+4}[1:5]$	$b_{t+5}[1:5]$
$x_{t,1}^{(1)}[3] = s_{t,1}[3]$	$c_t[1:5]$	$c_{t+1}[1:5]$	$c_{t+2}[1:5]$	$c_{t+3}[1:5]$	$c_{t+4}[1:5]$	$c_{t+5}[1:5]$
$x_{t,1}^{(1)}[4]$		$c_t[1:5]$	$b_t[1:5]$ $+c_{t+1}[1:5]$	$a_t[1:5]$ $+b_{t+1}[1:5]$ $+c_{t+2}[1:5]$	$a_{t+1}[1:5]$ $+b_{t+2}[1:5]$ $+c_{t+3}[1:5]$	$a_{t+2}[1:5]$ $+b_{t+3}[1:5]$ $+c_{t+4}[1:5]$
$x_{t,1}^{(1)}[5]$			$c_t[1:5]$	$b_t[1:5]$ $+2c_{t+1}[1:5]$	$a_t[1:5]$ $+2b_{t+1}[1:5]$ $+3c_{t+2}[1:5]$	$a_{t+1}[1:5]$ $+2b_{t+2}[1:5]$ $+3c_{t+3}[1:5]$
$x_{t,1}^{(1)}[6]$				$c_t[1:5]$	$b_t[1:5]$ $+3c_{t+1}[1:5]$	$a_t[1:5]$ $+3b_{t+1}[1:5]$ $+4c_{t+2}[1:5]$
User 2: Source to relay						
$x_{t,2}^{(1)}[1] = s_{t,2}[1]$	$d_t[1:9]$	$d_{t+1}[1:9]$	$d_{t+2}[1:9]$	$d_{t+3}[1:9]$	$d_{t+4}[1:9]$	$d_{t+5}[1:9]$
$x_{t,2}^{(1)}[2]$		$d_t[1:9]$	$d_{t+1}[1:9]$	$d_{t+2}[1:9]$	$d_{t+3}[1:9]$	$d_{t+4}[1:9]$
$x_{t,2}^{(1)}[3]$			$d_t[1:9]$	$d_{t+1}[1:9]$	$d_{t+2}[1:9]$	$d_{t+3}[1:9]$

(b) Relay to Destination codes. Highlighted in blue are the packets where a_t appears. Note that it starts being transmitted at time $t+5$, and is certainly recovered by time $t+6$, thus being subject to a delay of 1 in the relay-destination link.

Relay to Destination					
Symbol \ Time	t+2	t+3	t+4	t+5	t+6
$x_t^{(2)}[1]$	$d_t[1:6]$	$d_{t+1}[1:6]$	$d_{t+2}[1:6]$	$d_{t+3}[1:6]$	$d_{t+4}[1:6]$
$x_t^{(2)}[2]$		$c_t[1:5] \frown d_t[7]$	$c_{t+1}[1:5] \frown d_t[7]$	$c_{t+2}[1:5] \frown d_t[7]$	$c_{t+3}[1:5] \frown d_t[7]$
$x_t^{(2)}[3]$			$b_t[1:5] \frown d_t[8]$	$b_{t+1}[1:5] \frown d_{t+1}[8]$	$b_{t+2}[1:5] \frown d_{t+2}[8]$
$x_t^{(2)}[4]$				$a_t[1:5] \frown d_t[9]$	$a_{t+1}[1:5] \frown d_{t+1}[9]$
$x_t^{(2)}[5]$					$d_t[1:6]$ $+c_t[1:5] \frown d_t[7]$ $+b_t[1:5] \frown d_t[8]$ $+a_t[1:5] \frown d_t[9]$

each delay from 3 to 5. These codes are presented in Table IIa. Recall that each source packet $s_{t,1}$ contains 15 symbols, and each source packet $s_{t,2}$ contains 9 symbols. For presentation, the 15 symbols from $s_{t,1}$ are split in three ‘‘super-symbols’’ a_t , b_t and c_t , each containing five symbols. Similarly, d_t denotes $s_{t,2}$. Further, for presentation, symbols from times $t' < t$ are not presented in the table. As can be seen in the table, symbols belonging to c_t can be recovered by the relay at time $t+3$, symbols belonging to b_t can be recovered at time $t+4$ and symbols belonging to a_t can be recovered at time $t+5$. Thus, 5 symbols are recovered with each delay $\{3, 4, 5\}$. The 9 symbols from d_t can be recovered at time $t+2$, thus 9 symbols are recovered with delay $\{2\}$.

Now, one can see that the symbols transmitted with delay 5 (from the first user to the relay), i.e., symbols belonging to a_t , must be matched with the symbols transmitted with delay 1 (from relay to destination) in order to achieve an overall delay of $T = 6$. Similarly, symbols transmitted with delay 4 will be matched with symbols transmitted with delay 2, and so on. Therefore, the second user has the following ‘‘budget’’: it can transmit six symbols with delay 2 (to be matched with the remaining six symbols with delay 4), one symbol with delay 3

(to be matched with the remaining one symbol with delay 3), one symbol with delay 4, and one symbol with delay 5. This gives us the following tuple

$$\mathbf{T}_2^{(g,1)} = [5, 4, 3, 2] \quad (47)$$

$$\mathbf{k}^{(con)} = [1, 1, 1, 6]. \quad (48)$$

By applying Corollary 1, we find out that we can, in fact, obtain a code with $k_2^{(1)} = 9$. In fact, it turns out that we can transmit all 9 symbols with delay 2 by simply repeating a diagonally-interleaved MDS code with rate $2/3$. However, since our goal is to recover with delay $T = 6$, we can choose to wait (i.e., buffer) some symbols, so all symbols are recovered by the destination at the same time instant. This code is presented in Table IIb. The matching of delay spectra can be seen in Table III. Recall that $R_2 = 9/30 = 0.3$ is exactly the upper bound.

3) CSWDF: For CSWDF, one way to solve this problem is by optimizing A and B such that it results in $R_1 = 0.5$ and maximizes R_2 . That is, under CSWDF, we wish to solve the following optimization problem

$$R_2^* = \max_{A,B} R_2 \quad (49)$$

TABLE III

NUMBER OF SYMBOLS TRANSMITTED WITH EACH DELAY IN EACH HOP. BLUE SYMBOLS ARE TRANSMITTED BY THE FIRST USER, WHILE GREEN SYMBOLS ARE TRANSMITTED BY THE SECOND USER

T in the first hop	T in the second hop			
	1	2	3	4
2				6
3			(5, 1)	
4		(5, 1)		
5	(5, 1)			

$$\text{s.t. } R_1 = 0.5 \tag{50}$$

$$n = \max(6A, 6B, 4A + 5B) \tag{51}$$

$$R_1 = \frac{3A}{n} \tag{52}$$

$$R_2 = \frac{4B}{n} \tag{53}$$

where the constraints come from Lemma 8. Recall that the coefficients come from (29) and (30). For example, $k_1 = T + 1 - N_1 - N_3 = 3$ is the coefficient used for R_1 . Solving this optimization leads to $A = 5$ and $B = 2$, which results in $n = 30$, $k_1 = 15$ and $k_2 = 8$, i.e., $R_1 = 0.5$ as desired, and $R_2 = 0.2667$, slightly lower than the upper bound and the rate achieved by our FB-SWDF scheme.

4) *MSWDF*: Similar to the CSWDF, this requires solving an optimization problem, however, for MSWDF we also need to optimize the parameters $\tilde{T}_i, i \in \{1, 2\}$. Doing so results in $B = 0$ and any A , that is, in order to achieve $R_1 = 0.5$, we need $R_2 = 0$ for MSWDF.

5) *Comparison*: We hope that this example demonstrates the considerably suboptimality of MSWDF and CSWDF. The major difference between our scheme and CSWDF, fundamentally, is that our scheme allows for “merging” of the streams arriving at the relay from different sources, encoding all streams within the same code. One main contribution of this paper is to present a systematic way of designing such “merging” in a delay-spectrum efficient manner, through the framework provided by Lemma 3 and Corollary 1.

VI. RESULTS

In this section, we present the rate region for three different scenarios. In all cases, the curves are labeled according to the scheme presented in each respective section previously. The curve labeled as “Time Sharing” in Fig 3 represents the best achievable region for which we have closed form expressions, which can be achieved by concatenating CSWDF and FB-SWDF.

In the first scenario, we have a small T , such that the condition in Corollary 3 is not met. This is presented in Fig. 3. Note that, in this case, no scheme is able to achieve the sumrate. In the second case, we have $T = 5$, and, by slightly increasing T this way, our FB-SWDF and OB-SWDF schemes are able to achieve the sumrate capacity in a noticeable part of the capacity region, which is shown in Fig. 4. Further increasing T increases the fraction of the sumrate capacity that we are able to achieve, as shown in Fig. 5. It should also be noted that, as T increases, the relative performance of

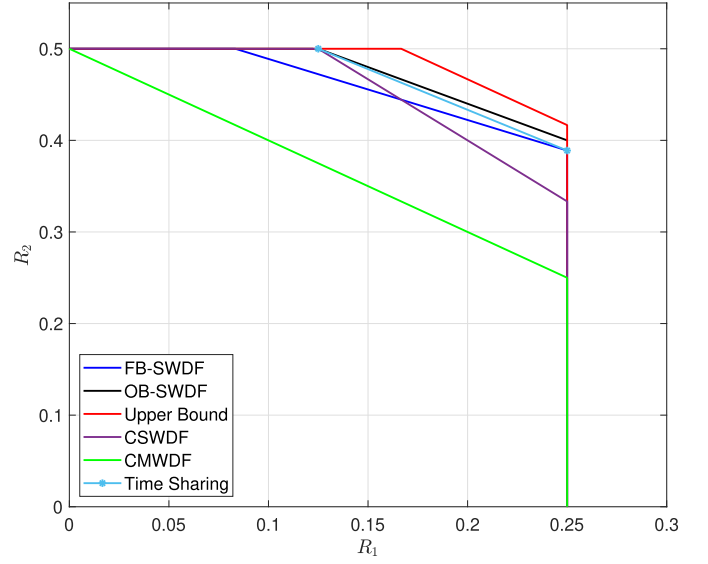


Fig. 3. Rate region for the weak relay-destination bottleneck regime with parameters $N_1 = 3, N_2 = 2, N_3 = 1, T = 4$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward.

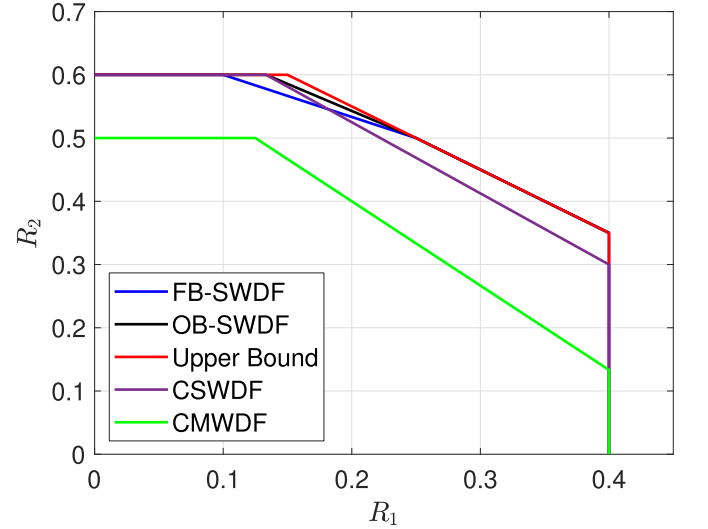


Fig. 4. Rate region for the weak relay-destination bottleneck regime with parameters $N_1 = 3, N_2 = 2, N_3 = 1, T = 5$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward.

CMWDF worsens, and for $T = 6$ it is unable to achieve even the single-user upper bound, which is also shown in [2].

In all cases, it can be seen that CSWDF is able to achieve the best result for maximal R_2 among the known achievable schemes. This has been observed in all settings we have experimented, although it remains to be proven. On the other hand, for maximal R_1 , CSWDF is noticeably inferior, and OB-SWDF can achieve a rate increase in the order of

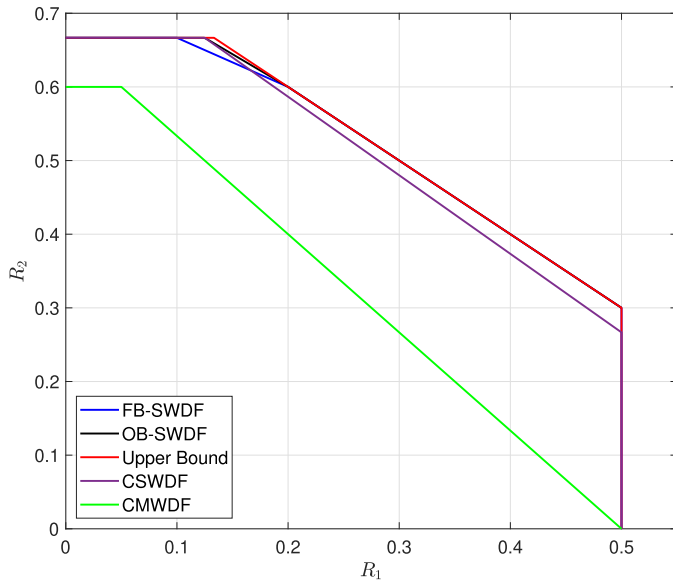


Fig. 5. Rate region for the weak relay-destination bottleneck regime with parameters $N_1 = 3, N_2 = 2, N_3 = 1, T = 6$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward.

10 to 20% compared to CSWDF. Nonetheless, it suggests that the “Time Sharing” scheme represents a good achievable rate, for which we have closed form expressions. In fact, in Fig. 5 or Fig. 4, it achieves the same performance as OB-SWDF.

Although our schemes are unable to always achieve the sumrate, and unable to achieve the whole rate region, comparing them to the alternatives—CSWDF and CMWDF—should show that the proposed schemes are significantly superior, as they are able to achieve the sumrate under some set of conditions that occur frequently, while the baselines are always unable to achieve it in any non-trivial setting, and even in cases where no scheme achieves the upper bound, ours offer rate gains in the order of 10 to 20%. Furthermore, OB-SWDF presents a single scheme that is able to achieve the highest rates among known schemes, with no need for considering different schemes.

VII. EXTENSION TO GENERAL PARAMETERS

In this section, we extend the previous achievability results provided for the weak relay-destination bottleneck to the other three regimes of operation. Thus, we provide achievable results for any parameters N_1, N_2, N_3 and T , as any choice of these parameters results in one of the four defined regimes.

A. Strong Source-Relay Bottleneck

In this case, the converse region is entirely defined by

$$R_1 \leq C(T - N_3, N_1) \quad (54)$$

$$R_2 \leq C(T - N_3, N_2) \quad (55)$$

since the sumrate bound is never active. Furthermore, it is easy to see that $N_2 \geq N_3$ is required in such scenario, otherwise,

it is impossible for the condition for strong source-relay bottleneck to hold (that is, otherwise we would be in the strong relay-destination bottleneck).

In order to show that the entire rate region is achievable, we state the following Lemma

Lemma 11: For the strong source-relay bottleneck regime, there exists an (N_1, N_2, N_3) -achievable code with the following rate pair

$$R_1 = C(T - N_3, N_1) \quad (56)$$

$$R_2 = C(T - N_3, N_2) \quad (57)$$

Proof: This follows from Lemma 8 (Concatenated Symbol-wise Decode-and-forward scheme), using $A = B = 1$. This will give us an (N_1, N_2, N_3) -achievable code with the following parameters: $k_1 = (T + 1 - N_1 - N_3)$, $k_2 = (T + 1 - N_2 - N_3)$, $n_1 = n_2 = (T + 1 - N_3)$, $n_3 = (T + 1 - N_1) + (T + 1 - N_2)$. Then, it remains to show that $n = n_1 = n_2 \geq n_3$.

The proof is as follows: since $N_1 \geq N_2$, we have

$$\frac{T + 1 - N_1 - N_3}{T + 1 - N_1} \leq \frac{T + 1 - N_2 - N_3}{T + 1 - N_2}. \quad (58)$$

From (58), it follows that⁹

$$\begin{aligned} \frac{(T + 1 - N_1 - N_3) + (T + 1 - N_2 - N_3)}{(T + 1 - N_1) + (T + 1 - N_2)} \\ \geq \frac{T + 1 - N_1 - N_3}{T + 1 - N_1}. \end{aligned} \quad (59)$$

Further, from the strong source-relay bottleneck assumption, we have

$$\begin{aligned} \frac{T + 1 - N_1 - N_3}{T + 1 - N_1} \\ \geq \frac{(T + 1 - N_3 - N_1) + (T + 1 - N_3 - N_2)}{T + 1 - N_3}. \end{aligned} \quad (60)$$

It follows directly then that

$$(T + 1 - N_1) + (T + 1 - N_2) \leq (T + 1 - N_3). \quad (61)$$

Now recall that, for $A = B = 1$, we have $n_1 = n_2 = T + 1 - N_3$ and $n_3 = (T + 1 - N_1) + (T + 1 - N_2)$, therefore, we completed the proof that $n = n_1 = n_2 \geq n_3$. ■

Furthermore, since each single-user capacity point is achievable, by applying Lemma 7, it is easy to see that the entire capacity rate region is achievable using CSWDF.

An example of such rate region is presented in Fig. 6, using the parameters $N_1 = 9, N_2 = 8, N_3 = 1, T = 12$. Note that, while the proof we used for achievability uses the concatenated symbol-wise scheme for simplicity, our proposed schemes are also able to achieve the entire rate region, while MSWDF is not guaranteed to.

⁹In (58), we have $\frac{a}{b} \geq \frac{c}{d}$. Then, we know that $\frac{a}{b} \geq \frac{a+c}{b+d} \geq \frac{c}{d}$, where the second inequality is used in order to derive (59). Note that this holds because all terms are non-negative from assumption.

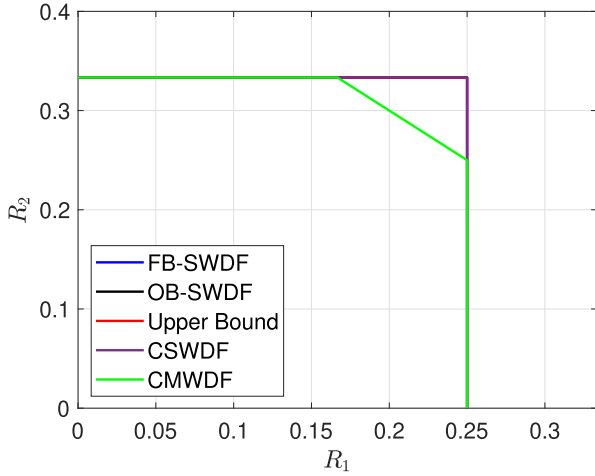


Fig. 6. Capacity Rate Region for a strong source-relay bottleneck regime with parameters $N_1 = 9$, $N_2 = 8$, $N_3 = 1$, $T = 12$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward. All schemes other than CMWDF are able to achieve the upper bound.

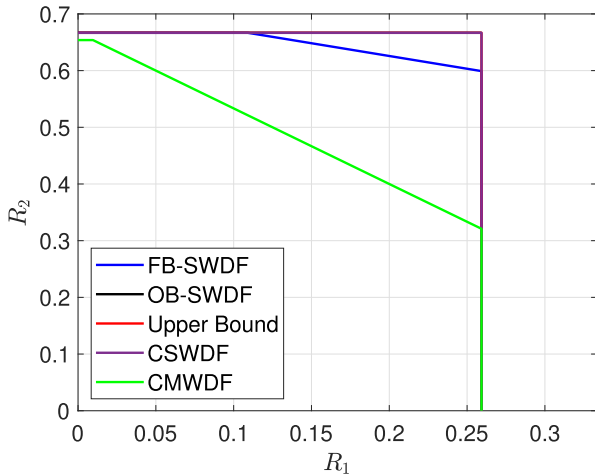


Fig. 7. Capacity Rate Region for a weak source-relay bottleneck regime with parameters $N_1 = 20$, $N_2 = 9$, $N_3 = 1$, $T = 27$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward.

B. Weak Source-Relay Bottleneck

In this case, similar to before, the converse is reduced to

$$R_1 \leq C(T - N_3, N_1) \quad (62)$$

$$R_2 \leq C(T - N_3, N_2). \quad (63)$$

However, we may not be able to achieve the corner point (i.e., achieve this bound with equality). Nonetheless, all schemes presented in our paper can be employed in this regime, although the FB-SWDF scheme is certainly going to underperform if we choose to keep the chosen R_{bn} , as it is certainly not achievable in this regime. We take this opportunity to highlight the effectiveness of OB-SWDF, which again presents the best achievability in Fig. 7 and Fig. 8.

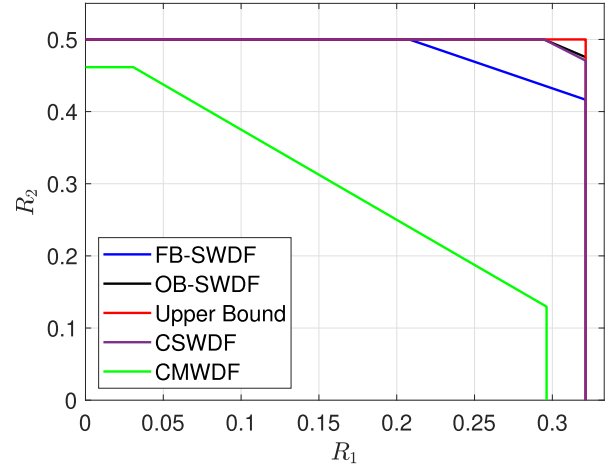


Fig. 8. Capacity Rate Region for a weak source-relay bottleneck regime with parameters $N_1 = 19$, $N_2 = 14$, $N_3 = 3$, $T = 30$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward.

Nonetheless, we can guarantee achievability of the entire region under the following sufficient condition. If the condition is not satisfied, it is unknown whether there exists a scheme that achieves the upper bound or not.

Lemma 12: For a network in the weak source-relay bottleneck regime, if

$$T \leq N_1 + N_2 - N_3 - 1$$

then the entire rate region is achievable.

Proof: This follows directly from Lemma 8 with $A = B = 1$. If the condition stated in the lemma holds, we are able to achieve the corner point $R_1 = C(T - N_3, N_1)$ and $R_2 = C(T - N_3, N_2)$, and, again, since the single-user points are achievable, we can achieve the entire rate region. ■

Remark 9: The Lemma can be seen in the example figures. In the first case, we have $T = 27 \leq 27 = N_1 + N_2 - N_3 - 1$, and we can achieve the capacity region. In the second example, we have $T = 30 > 29 = N_1 + N_2 - N_3 - 1$, and we can not achieve the capacity region.

C. Strong Relay-Destination Bottleneck

The condition for the network to be in this regime of operation is that $N_3 \geq N_2$. In this case, we have that the converse is given by

$$R_1 \leq C(T - N_3, N_1) \quad (64)$$

$$R_1 + R_2 \leq C(T - N_2, N_3). \quad (65)$$

We now show that both points are achieved with equality.

Lemma 13: In the strong relay-destination bottleneck regime, the following rate pairs are achievable:

$$R_1 = 0 \quad (66)$$

$$R_2 = C(T - N_2, N_3) \quad (67)$$

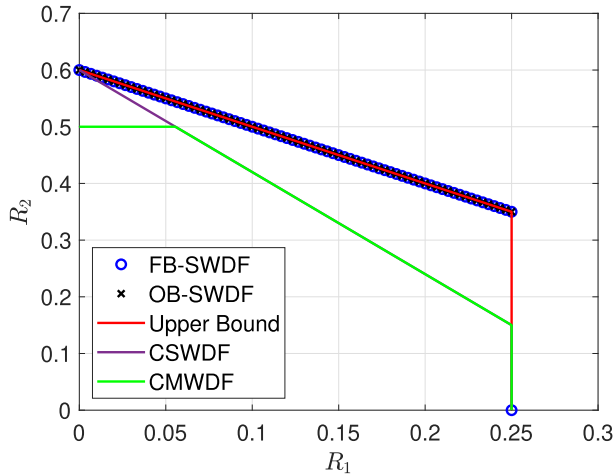


Fig. 9. Capacity Rate Region for a strong relay-destination bottleneck regime with parameters $N_1 = 3$, $N_2 = 1$, $N_3 = 2$, $T = 5$, where R_1 is the rate of the first user and R_2 is the rate of the second user. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward. Note that the line-style used for FB-SWDF and OB-SWDF is different from earlier figures to improve clarity.

and

$$R_1 = C(T - N_3, N_1) \quad (68)$$

$$R_2 = C(T - N_2, N_3) - C(T - N_3, N_1). \quad (69)$$

Proof: The rate pair $R_1 = 0, R_2 = C(T - N_2, N_3)$ is achievable using the three-node network capacity-achieving codes presented in [2], or, in other words, a single-user capacity achieving code. The second rate pair can be easily achieved using our FB-SWDF scheme. In order to see that, recall that, from Corollary 3, if

$$(T+1-N_3)(T+1-N_1-N_2) \geq (T+1-N_1-N_3)(N_2-N_3). \quad (70)$$

then we have $C(T - N_2, N_3) - C(T - N_3, N_1) \leq R'_2$ in Lemma 5. Furthermore, note that, from the strong relay-destination bottleneck assumption, we have $N_3 \geq N_2$, therefore $N_2 - N_3 \leq 0$, thus the right-hand side of the inequality is always negative. On the other hand, the left-hand side is always positive, as $T \geq N_1 + N_3$ (otherwise we have $R_1 = 0$ and this degenerates to a single-user setting). ■

It then easily follows that the entire capacity region is achievable from Lemma 7.

An example of such rate region is presented in Fig. 9. Note that neither the concatenated symbol-wise decode-and-forward or the concatenated message-wise decode-and-forward are able to achieve the entire capacity region, while our proposed scheme is able to, and further note that the rate improvement in the maximal R_1 point is in the order of 100% increase in the rate of the second user.

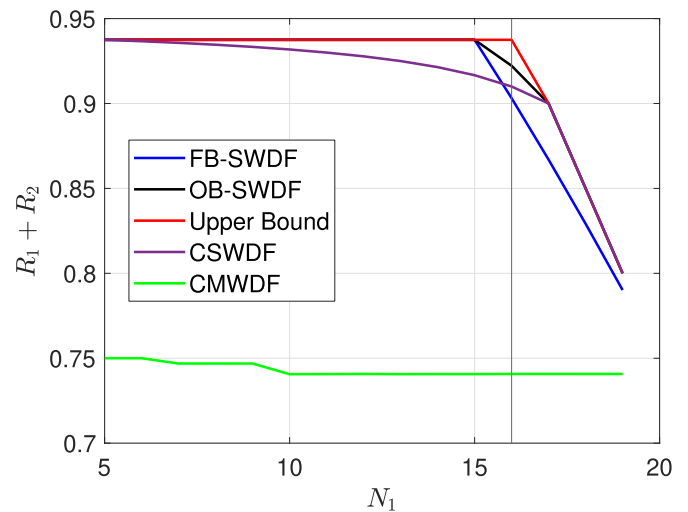


Fig. 10. Maximal sumrate as a function of N_1 for $N_2 = 5$, $N_3 = 1$, $T = 20$. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward. The vertical black line represents changing the regime from weak relay-destination bottleneck (left) to weak source-relay bottleneck (right).

D. Trends

Now that all regimes of operation have been presented, and achievable results have been proposed, we wish to understand how each scheme behaves as the parameters change. In order to do so, let us define $W \triangleq T + 1$, and let us assume $N_1 = \alpha W$, $N_2 = \beta W$, $N_3 = \gamma W$. Then, all the results presented in this paper are functions only¹⁰ of α , β and γ , which represent the fractions of erasures that occur in the window of length W in each link. Therefore, it suffices to fix a large enough W , and vary N_1 , N_2 and N_3 .

In order to present the information, we consider the maximal sum rate achieved by each scheme. In all experiments, we consider $T = 20$.

We first set $N_2 = 5$, $N_3 = 1$, and vary N_1 from 5 up to 19. In Fig. 10, it can be noted that, while α (or N_1) is sufficiently small (or, relatively, T is sufficiently large), FB-SWDF can achieve the sumrate in the weak relay-destination, and so can OB-SWDF. On the other hand, CSWDF is only able to achieve it for $N_1 = N_2$, and then the gap from CSWDF to the sumrate increases. When the regime changes to source-relay bottleneck (indicated by a vertical black line), CSWDF is eventually able to achieve the sumrate, and FB-SWDF underperforms, since it is not designed to handle this regime. In all cases, OB-SWDF achieves at least the best performance between the two. We note that, because the sumrate bound is only defined by N_2 and N_3 in the relay-destination bottleneck, increasing N_1 does not decrease it. We repeat the same experiment for $N_2 = 10$, $N_3 = 7$. In Fig. 11, it can be seen that no scheme is able to achieve the upper bound after the trivial case $N_1 = N_2$. Nonetheless, both of our schemes achieve higher rates than the baseline.

¹⁰Note that $\alpha \geq \beta$, since $N_1 \geq N_2$, and the values of α , β and γ must be such that the number of erasures is integer.

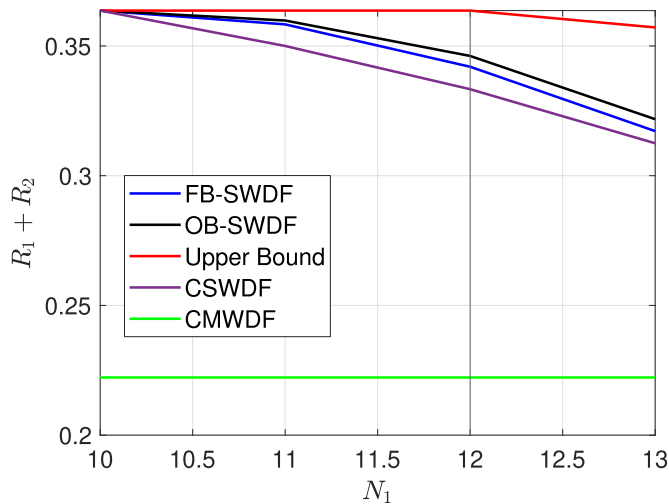


Fig. 11. Maximal sumrate as a function of N_1 for $N_2 = 10$, $N_3 = 7$, $T = 20$. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward. The vertical black line represents changing the regime from weak relay-destination bottleneck (left) to weak source-relay bottleneck (right).

We then wish to consider the effect of β , or N_2 . However, recall that we assumed $N_1 \geq N_2$ throughout the paper. We consider $N_1 = 15$ and $N_3 = 4$. This choice allows us to demonstrate all regimes of operation. In Fig. 12, for low values of N_2 , we are in the strong relay-destination bottleneck, that is, we have $N_3 \geq N_2$. In this case, achieving the sumrate is trivial, as it can be achieved by setting $R_1 = 0$ and using a capacity-achieving single-user code for the second user. We then move into the weak relay-destination bottleneck, and CSWDF is unable to achieve the sumrate, while our schemes are able to achieve it for some range of N_2 . In the weak source-relay bottleneck regime, again FB-SWDF performance degrades quickly, as it is not designed for this regime, but OB-SWDF still outperforms all other schemes. Then, as N_2 increases, or, in other words, T decreases relative to the number of erasures, the sumrate becomes again achievable by both CSWDF and OB-SWDF, as prescribed by Lemma 12. Finally, in the strong source-relay bottleneck, all schemes are again able to achieve the sumrate.

The results as function of N_3 are similar, and mostly depend on the choice of N_1 and N_2 and in which regime of operation we are, and therefore are not presented.

Remark 10: Although we focus on the maximum sumrate analysis for simplicity and presentation, the same analysis can be performed for different points of operation. For example, if one is concerned about “fairness”, intuitively the point $R_1 = 0$ is “unfair”, and this is the point for maximal sumrate in the strong relay-destination bottleneck regime for CSWDF in Fig. 12 (see Fig. 9 for comparison), and we would be interested in points where the rates are closer to each other.

VIII. COMMENTS AND LIMITATIONS

In this section we wish to make some comments about our results, acknowledge a few limitations and discuss future work.

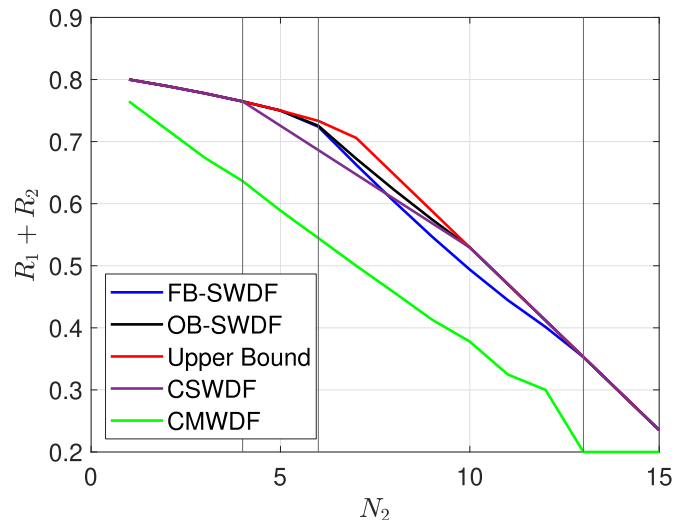


Fig. 12. Maximal sumrate as a function of N_2 for $N_1 = 15$, $N_3 = 4$, $T = 20$. In the figure, FB-SWDF refers to the Fixed-Bottleneck Symbol-wise Decode-and-forward scheme, OB-SWDF to Optimized-Bottleneck Symbol-wise Decode-and-forward, CSWDF to Concatenated Symbol-wise Decode-and-forward, and CMWDF to Concatenated Message-Wise Decode-and-forward. The first vertical black line represents the transition from strong relay-destination bottleneck (left) to weak relay-destination bottleneck (right). Then, the second vertical line represents the transition from weak relay-destination bottleneck to weak source-relay bottleneck. The last vertical line represents the transition from weak source-relay bottleneck to strong source-relay bottleneck.

One comment to be made about our results is about the asymptotic optimality. For a sufficiently large T , or equivalently, a small fraction of erasures within a window, it is easy to see that the condition on Corollary 3 is always met¹¹, as the left hand side grows with T^2 , while the right hand side grows with T . Furthermore, the upper corner point tends to degenerate to the single-user scenario with $T \rightarrow \infty$, as can be seen from the upper bound and from Lemma 9. Therefore, with $T \rightarrow \infty$, the capacity region is entirely achievable.

We also should mention that the proposed upper bound is optimistic, and implicitly assumes the first user is subject to N_2 erasures instead of $N_1 \geq N_2$ erasures when computing the sumrate bound. This is a possible reason for the gap between our achievable result and the bound, especially in the region where the first user transmits at a rate close to its single-user rate. Improving the bound by considering both N_1 and N_2 when computing the sumrate is an interesting and challenging work. Nonetheless, it is surprising that even such an optimistic bound can be achieved for a large set of parameters and rate pairs of interest, as shown by our results.

In terms of the presented schemes, our OB-SWDF presents two limitations: one is that the results are numerical, thus no performance guarantees are given. However, one may construct the optimization in such a way that it certainly passes through the rates achieved by FB-SWDF and CSWDF, guaranteeing an outperformance of these schemes. The second

¹¹In practice, for small values of N_i , which are desired in order to achieve low latency, the condition is met rather quickly (See, e.g. Fig. 5), even for a T slightly larger than the minimum $N_1 + N_3$. We present the asymptotic analysis for completeness.

limitation is that the choice of n in order to obtain the best possible performance might be very large (e.g. $n \geq 2000$) and impractical for the applications which are the target of low latency streaming codes. This is highly dependent on the channel parameters. However, experimental results show that one may achieve rates close to the presented in our results even with a significantly smaller choice of n .

Another limitation is that we focus on time-invariant codes, and, more specifically, our relaying strategy does not exploit the knowledge of the erasures from source to relay. This constraint on the streaming codes reduces the capacity. However, this is justified by practical limitations on the relay's processing capabilities, as well as by allowing no-overhead transmission. Nonetheless, it is possible that the use of adaptive schemes may allow for better performance in the multi-access setting and it is considered as future work.

Finally, it should be noted that while we focus our analysis on the uplink, our framework is fairly symmetrical and can be simply generalized to the downlink, in which a source wishes to transmit different messages to different destinations through a relay (assuming wired connections between nodes). This is especially useful since many applications, such as cloud gaming, involve two-way communication.

IX. CONCLUSION

In this paper, we have presented an in-depth analysis of achievability using symbol-wise decode-and-forward. We have derived the condition for which the sumrate is achievable, and, when it is, what part of the whole rate region is achievable. Furthermore, we have compared the proposed schemes to concatenated message-wise decode-and-forward, which is the naive solution to the problem, and the performance of our scheme is clearly superior. It should also be noted that, while the analysis in this paper is restricted to 2 users, the tools and schemes presented can be easily extended to any number of users, and the achievable rates can easily be solved numerically, although analysis of the closed-form expressions become increasingly hard to derive.

For future works, the gap between achievability and converse should be tightened. It is likely that the current converse is optimistic, and new converse-deriving techniques are necessary in order to obtain better results. It is also unclear whether employing a different (than symbol-wise decode-and-forward) strategy in the relay can improve the achievable results, and an analysis based on information rates could be helpful.

APPENDIX

Proof of Theorem 1: The bounds on R_1 and R_2 are direct extensions from [2] and do not require the channel state independent constraint. We present the proof for user 1. The same steps can be applied to user 2. For user 1, because each source packet must be recovered by the destination with delay T , and a burst of N_3 erasures may occur from relay to destination, for every packet the following holds

$$H\left(s_{t,1} | (\{x_{t,1}^{(1)}, \dots, x_{t+T-N_3,1}^{(1)}\} \setminus \{x_{\theta_1,1}^{(1)}, \dots, x_{\theta_{N_1},1}^{(1)}\}), \{s_{t',1}\}_{t'=0}^{t-1}\right) = 0 \quad (71)$$

for any N_1 non-negative integers denoted by $\theta_1, \dots, \theta_{N_1}$. It then follows that

$$H\left(\{s_{t',1}\}_{t'=0}^{T-N_3+(j-1)(T+1-N_3)} | \left\{x_{i(T+1-N_3),1}^{(1)}, x_{1+i(T+1-N_3),1}^{(1)}, \dots, x_{T-N_1-N_3+i(T+1-N_3),1}^{(1)}\right\}_{i=0}^j\right) = 0 \quad (72)$$

where the conditional entropy involves $j(T+1-N_3)$ source packets and $(j+1)(T+1-N_1-N_3)$ encoded packets. Now, note that (72) implies

$$n_1(j+1)(T+1-N_1-N_3) \quad (73)$$

$$\geq H\left(\{x_{i(T+1-N_3),1}^{(1)}, x_{1+i(T+1-N_3),1}^{(1)}, \dots, x_{T-N_1-N_3+i(T+1-N_3),1}^{(1)}\}_{i=0}^j\right) \quad (74)$$

$$\geq H(\{s_{t',1}\}_{t'=0}^{T-N_3+(j-1)(T+1-N_3)}) \quad (75)$$

$$= k_1 j(T+1-N_3) \quad (76)$$

which, by making $j \rightarrow \infty$ (recall that it must hold for any j), results in $\frac{k_1}{n_1} \leq \frac{T+1-N_1-N_3}{T+1-N_3} = C(T-N_3, N_1)$. Finally, since $n \geq n_1$, from definition, it follows that $R_1 = \frac{k_1}{n} \leq \frac{k_1}{n_1}$, completing the proof of the bound on R_1 .

As mentioned previously, the same argument holds for user 2, and therefore for R_2 , changing N_1 to N_2 .

Now, we consider a similar argument for the relay-destination link. Recall that both $s_{t,1}$ and $s_{t,2}$ must be recovered by the destination at time $t+T$. Further, recall that we assume $N_1 \geq N_2$ without loss of generality, therefore, a burst of N_2 erasures may occur from time t up to time $t+N_2-1$ simultaneously in the links from both users to the relay. Therefore, the following holds

$$H\left(s_{t,1}, s_{t,2} | (\{x_{t+N_2}^{(2)}, \dots, x_{t+T}^{(2)}\} \setminus \{x_{\phi_1}^{(2)}, \dots, x_{\phi_{N_3}}^{(2)}\}), \{s_{t',i}\}_{t'=0}^{t-1}\right) = 0. \quad (77)$$

Now, let us understand why the argument fails if the relaying scheme is allowed to change depending on the erasure patterns, which was previously missed in [2]. Let us consider another time instant t' . Similarly, $s_{t',1}$ and $s_{t',2}$ must be recovered by the destination at time $t'+T$, and, as before, N_2 erasures may occur in both links from time t' up to $t'+N_2-1$. However, the relay can observe this different erasure pattern (before it was from t up to $t+N_2-1$, now it starts at t'), and adapt its relaying strategy accordingly. Therefore, the output of the relaying function will differ, and we can only write

$$H\left(s_{t',1}, s_{t',2} | (\{x_{t'+N_2}^{(2)}, \dots, x_{t'+T}^{(2)}\} \setminus \{x_{\phi_1}^{(2)}, \dots, x_{\phi_{N_3}}^{(2)}\}), \{s_{t'',i}\}_{t''=0}^{t'-1}\right) = 0. \quad (78)$$

However, because $x_t^{(2)} \neq x_{t'}^{(2)}$, (77) and (78) can not, in general, be combined. That is, unlike in the source-relay case, we can not get an equation similar to (72), because (77) does not hold **simultaneously** for all t , only for one t for each erasure pattern. Therefore, we must now use the assumption that the code is channel state invariant, i.e., Definition 4,

and therefore $x_t'^{(2)} = x_t^{(2)}$ for all t , because the relay is not allowed to adapt to the different erasure patterns. It then follows similarly that

$$H\left(\left\{\left\{s_{t',i}\right\}_{t'=0}^{T-N_2+(j-1)(T+1-N_2)}\right\}_{i=1}^2 \middle| \left\{x_{i(T+1-N_2),1}^{(2)}\right\}_{i=0}^j\right) = 0. \quad (79)$$

From (79), it then follows that

$$n_3(j+1)(T+1-N_2-N_3) \quad (80)$$

$$\geq H\left(\left\{x_{i(T+1-N_2),1}^{(2)}\right\}_{i=0}^j\right) \quad (81)$$

$$\geq H\left(\left\{\left\{s_{t',i}\right\}_{t'=0}^{T-N_2+(j-1)(T+1-N_2)}\right\}_{i=1}^2\right) \quad (82)$$

$$=(k_1+k_2)j(T+1-N_2). \quad (83)$$

As before, by making $j \rightarrow \infty$ and noting that $n \geq n_3$ from definition, we get

$$R_1 + R_2 = \frac{k_1 + k_2}{n} \quad (84)$$

$$\leq \frac{k_1 + k_2}{n_3} \quad (85)$$

$$\leq \frac{T+1-N_2-N_3}{T+1-N_2} \quad (86)$$

$$=C(T-N_2, N_3) \quad (87)$$

which completes the proof. \blacksquare

Proof (Delay Spectrum of Symbol-Wise Decode-and-Forward:) Let

$$\mathbf{s}_t'' = [s_{t-(T^{(1)}[1])}[1], s_{t-(T^{(1)}[2])}[2], \dots, s_{t-(T^{(1)}[k])}[k]] \quad (88)$$

that is, \mathbf{s}_t'' is the vector of symbols that are decoded at time t . From assumption, the relay has access to this vector at time t . The relay can then apply the permutation

$$\mathbf{s}'_t = \pi \mathbf{s}_t'' \quad (89)$$

and encode such sequence. Now, consider the j th symbol. It has been recovered with a delay $T^{(1)}[j]$ by the relay, then it has been permuted into the j' symbol and recovered with delay $T^{(2)}[j']$ by the destination. For all j and all permutations, the destination has recovered the symbol by time $T^{(1)}[j] + T^{(2)}[j']$, where j' is defined by the permutation.

That is, the delay spectrum of symbol-wise decode-and-forward is given by $\mathbf{T}^{(1)} + \pi \mathbf{T}^{(2)}$ where π is some permutation. \blacksquare

Proof of Lemma 1: Consider the concatenation code of both codes. Since the first code is able to achieve delay spectrum \mathbf{T}' under N erasures, the first k' symbols can be recovered certainly at times \mathbf{T}' . Similarly, since the second code achieves delay spectrum \mathbf{T}'' under N erasures, the last k'' symbols can be recovered certainly at times \mathbf{T}'' . Thus, the concatenation code achieves the delay spectrum $[\mathbf{T}', \mathbf{T}'']$ under N erasures. \blacksquare

Proof of Lemma 2: This follows directly from the choice of encoder. Let us denote by $f_t(\{s_j\}_{j=0}^t)$ the encoding function

of a code that achieves delay spectrum \mathbf{T} under N erasures. Without loss of generality, let us analyze the first two source symbols. This code is able to recover them by times $T[1]$ and $T[2]$, respectively. Now, consider the code generated by applying the function to \mathbf{s}' , i.e., $f_t(\{s'_j\}_{j=0}^t)$, where $\mathbf{s}' = \pi \mathbf{s}$. Again without loss of generality, assume the permutation swaps the first two positions. Then, $\mathbf{s}'[1]$ is now recovered at time $T[1]$, and $\mathbf{s}'[2]$ is recovered at time $T[2]$. However, $\mathbf{s}'[1] = \mathbf{s}[2]$, therefore, $\mathbf{s}[2]$ is recovered at time $T[1]$, and similarly for $\mathbf{s}[1]$.

It is easy to see that, in general, applying the desired permutation over the source symbols before encoding results in the desired permuted delay spectrum. Thus, if a code is able to achieve delay spectrum \mathbf{T} under N erasures, permuting the source symbols and using the same code suffices to achieve delay spectrum $\pi \mathbf{T}$ under N erasures. \blacksquare

The following lemma has been proven in [2] and is useful in the proof of Lemma 3.

Lemma 14: A systematic $(N+k, k, \mathbf{T})_{\mathbb{F}}$ point-to-point diagonally-interleaved MDS code [5] achieves the delay spectrum $\mathbf{T} = [N, \dots, N+k-1]$ under N erasures.

Proof: Recall that we wish to construct a systematic (n, k) code such that, under N erasures, $k^{(g)}[j]$ symbols are recovered at time $T^{(g)}[j]$ and

$$k^{(g)}[j] = \begin{cases} n - \frac{T^{(g)}[1]}{N}(n-k), & j = 1 \\ \frac{n-k}{N}, & j \geq 2 \end{cases}. \quad (90)$$

For the remaining of the proof, we denote $k^{(g)}[j] = \frac{n-k}{N}$, i.e., we assume $j \geq 2$.

Consider the following coding scheme: denote by $m = (T^{(g)}[1] + 1 - N)$ the number of possible delays. We concatenate $k^{(g)}[1] = n - \frac{T^{(g)}[1]}{N}(n-k)$ diagonally-interleaved MDS codes with parameters $(N+m, m)$ and $k^{(g)}[j] - k^{(g)}[1] = \frac{T^{(g)}[1]+1}{N}(n-k) - n$ interleaving MDS codes with parameters $(N+m-1, m-1)$. Note that, since we are concatenating systematic codes, the resulting code is systematic.

It follows directly from Lemma 14 and Lemma 1 that $k^{(g)}[1]$ symbols are recovered with delay $T^{(g)}[1]$ and that $k^{(g)}[j]$ symbols are recovered with every other delay (i.e., $T^{(g)}[j]$). From definition, we have $k = k^{(g)}[1] + \sum_{j=2}^m k^{(g)}[j]$, since the number of possible delays is exactly m .

Therefore, it remains to show that the number of channel uses this proposed code uses, which we will briefly denote as n' , is the designed number of channel uses n .

The number of channel uses of this concatenation is, from definition of concatenation, given by

$$n' = k^{(g)}[1](N+m) + (N+m-1)(k^{(g)}[j] - k^{(g)}[1]) \quad (91)$$

$$= (N+m-1)k^{(g)}[j] + k^{(g)}[1] \stackrel{(a)}{=} Nk^{(g)}[j] + k \quad (92)$$

$$\stackrel{(b)}{=} (n-k) + k = n \quad (93)$$

where (a) follows from $k^{(g)}[1] + (m-1)k^{(g)}[j] = k$ and (b) follows from $k^{(g)}[j] = \frac{1}{N}(n-k)$.

Now, note that this requires $n - \frac{T[1]}{N}(n-k)$ and $\frac{T[1]+1}{N}(n-k)$ to be integer. Both are true from assumption. This is intuitive from the construction: we are concatenating codes of the form

$(N + k', k')$, that is, all the codes that composite the code have exactly N parity symbols, thus the number of parity symbols has to be some multiple of N . ■

Proof of Corollary 1: Let us build a code using Lemma 3 using a code with a k that satisfies the conditions on the corollary, that is, a k such that

$$k \leq n - n \cdot N \cdot \frac{\left(1 - \sum_{\ell=1}^{j-1} \frac{k^{\text{con}}[\ell]}{n}\right)}{T^{(g)}[j] + 1} \forall j \in \{1, 2, \dots, \ell^{(g)}\} \quad (94)$$

$$k \leq \sum_{\ell=1}^{\ell^{(g)}} k^{\text{con}}[\ell]. \quad (95)$$

Then, let us analyze $k^{(g)}[1]$. From Lemma 3, we have

$$k^{(g)}[1] = n - \frac{T^{(g)}[1]}{N}(n - k) \quad (96)$$

$$\stackrel{(a)}{\leq} n - \frac{T^{(g)}[1]}{N} \left(n - n + n \cdot N \cdot \frac{\left(1 - \sum_{\ell=1}^1 \frac{k^{\text{con}}[\ell]}{n}\right)}{T^{(g)}[2] + 1} \right) \quad (97)$$

$$\stackrel{(b)}{=} n - \frac{T^{(g)}[1]}{N} \left(n \cdot N \cdot \frac{\left(1 - \frac{k^{\text{con}}[1]}{n}\right)}{T^{(g)}[1]} \right) \quad (98)$$

$$= k^{\text{con}}[1] \quad (99)$$

where (a) follows from the fact that (94) holds for all j , in particular, $j = 2$; (b) follows from $T^{(g)}[2] = T^{(g)}[1] - 1$ from construction in Lemma 3, and the final step is just arithmetic simplification. Now, by repeating the same steps for each j , we obtain that

$$\sum_{\ell=1}^j k^{(g)}[\ell] \leq \sum_{\ell=1}^j k^{\text{con}}[\ell] \forall j. \quad (100)$$

This is still different from what we desire, which is $k^{(g)}[\ell] \leq k^{\text{con}}[\ell]$ for all ℓ . However, note that, as ℓ increases, the delay is decreasing. Thus, it is trivial that one could “redistribute” the delays by introducing artificial delay through buffering. That is, we may, without affecting achievability, assume that $k^{(g)}[\ell] = k^{\text{con}}[\ell]$ for any “small” ℓ .

More formally, let us show that (100) implies $k^{(g)}[\ell] \leq k^{\text{con}}[\ell] \forall \ell$. Let ℓ' be the first ℓ such that $k^{(g)}[\ell'] > k^{\text{con}}[\ell']$. Further, let us denote by $\delta(\ell') \triangleq k^{(g)}[\ell'] - k^{\text{con}}[\ell'] > 0$. However, from assumption (from (100)), we have $\sum_{\ell=1}^{\ell'} k^{(g)}[\ell] \leq \sum_{\ell=1}^{\ell'} k^{\text{con}}[\ell]$, therefore, $\sum_{\ell=1}^{\ell'-1} k^{\text{con}}[\ell] - k^{(g)}[\ell] \geq \delta(\ell')$. We can, then, take $\delta(\ell')$ symbols which are transmitted with delay $T^{(g)}[\ell']$, and “buffer” them, increasing the delay. We do this in a way such that $k^{(g)}[\ell] \leq k^{\text{con}}[\ell] \forall \ell < \ell'$ still holds. Note, again, that this is certainly possible, because $\sum_{\ell=1}^{\ell'-1} k^{\text{con}}[\ell] - k^{(g)}[\ell] \geq \delta(\ell')$, that is, there are enough “free” symbols at higher delays. We now have $k^{(g)}[\ell'] = k^{\text{con}}[\ell']$.

Therefore, if the conditions on the corollary statement hold, then there exists a code such that $\mathbf{k}^{(g)} \leq \mathbf{k}^{\text{con}}$ according to Lemma 3. ■

Proof of Lemma 4: First, recall that, in order to a delay spectrum to be achievable under N erasures, the decoder must

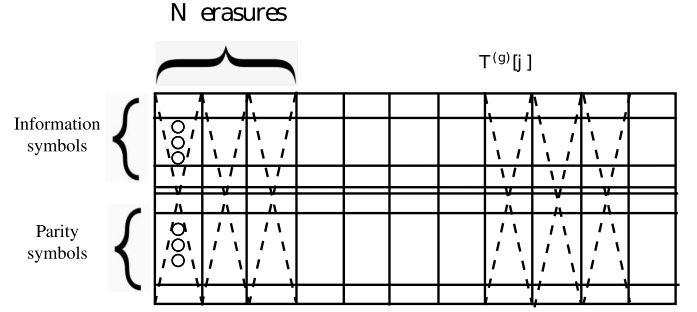


Fig. 13. Periodic erasure pattern.

be able to recover the $k^{(g)}[j]$ information symbols at time $T^{(g)}[j]$ with no ambiguity as long as the number of erasures is at most N .

In order to prove the lemma, we make a counting argument similar to [32] and [2]. The argument can be formalized in terms of entropy, as seen in the mentioned papers. Consider N erasures at the first N positions, and other N erasures after time $T^{(g)}[j]$, as in Fig. 13.

More precisely, due to the systematic assumption, we have the following condition

$$H(\mathbf{s}_{N:T^{(g)}[j]} | \mathbf{x}_{N:T^{(g)}[j]}) = 0$$

that is, the source packets at non-erased times must be fully recoverable.

Without loss of generality, let us assume the symbols are ordered with increasing delay, and let us denote by j' the first symbol such that $T[j'] = T^{(g)}[j]$, that is, symbols from 1 up to $j' - 1$ are allowed a delay higher than $T^{(g)}[j]$. Then, since all symbols with delays smaller than or equal to $T^{(g)}[j]$ must be recoverable with delay at most $T^{(g)}[j]$, we also must have that

$$H(\mathbf{s}_{0:N-1} | \mathbf{x}_{N:T^{(g)}[j]}, \mathbf{s}_{0:N-1}[1 : j' - 1]) = 0$$

that is, if we are given the symbols with higher delay for the erased packets, then we must be able to fully recover the erased packets using only the available information from \mathbf{x} up to time $T^{(g)}[j]$.

Then, we can write

$$\sum_{t=N}^{T^{(g)}[j]} H(\mathbf{x}_t) + \sum_{t=0}^{N-1} H(\mathbf{s}_t[1 : j' - 1]) \quad (101)$$

$$\geq H(\mathbf{x}_{N:T^{(g)}[j]}, \mathbf{s}_{0:N-1}[1 : j' - 1]) \quad (102)$$

$$= H(\mathbf{s}_{0:T^{(g)}[j]}) \quad (103)$$

$$= \sum_{t=0}^{T^{(g)}[j]} H(\mathbf{s}_t). \quad (104)$$

Finally, by noting that $H(\mathbf{s}_t) = k$ and $H(\mathbf{x}_t) = n$, and further that $k = \sum_{j=1}^{\ell^{(g)}} k^{(g)}[j]$, we can write

$$\begin{aligned} & (T^{(g)}[j] + 1 - N)n \\ & \geq (T^{(g)}[j] + 1) \sum_{\ell=j}^{\ell^{(g)}} k^{(g)}[\ell] + (T^{(g)}[j] + 1 - N) \sum_{\ell=1}^{j-1} k^{(g)}[\ell] \end{aligned} \quad (105)$$

where $\sum_{t=0}^{N-1} H(\mathbf{s}_t[1 : j' - 1]) = N \sum_{\ell=1}^{j'-1} k^{(g)}[\ell]$ follows because the symbols are i.i.d. and the number of symbols from 1 to $j' - 1$ is exactly $\sum_{\ell=1}^{j'-1} k^{(g)}[\ell]$ from definition.

If this condition does not hold for some $T^{(g)}[j]$, there is ambiguity in the recovery of the symbols and the information of the $k^{(g)}[j]$ symbols can not be recovered by the deadline, therefore, this delay spectrum is not achievable under N erasures.

This condition can then be rewritten as

$$T^{(g)}[j] \geq N \frac{\left(1 - \sum_{\ell=1}^{j-1} \frac{k^{(g)}[\ell]}{n}\right)}{\left(1 - \sum_{\ell=1}^{\ell^{(g)}} \frac{k^{(g)}[\ell]}{n}\right)} - 1. \quad (106)$$

Now, recall that $k = \sum_{\ell=1}^{\ell^{(g)}} k^{(g)}[\ell]$, then we have

$$T^{(g)}[j] \geq \frac{Nn}{n-k} \left(1 - \sum_{\ell=1}^{j-1} \frac{k^{(g)}[\ell]}{n}\right) - 1. \quad (107)$$

The implicit assumption in the erasure pattern and the deduction is that $T^{(g)}[j] \geq T^{(g)}[j+1]$ and $T^{(g)}[j+1] + N \geq T^{(g)}[j]$. However, we can always set $T^{(g)}[j+1] = T^{(g)} - 1$, and simply have some $k^g[j] = 0$, thus such assumption is not a problem. ■

Proof of Corollary 2: This follows immediately from rearranging Lemma 4 and solving it for $T^{(g)}[2] = T^{(g)}[1] - 1$ and $T^{(g)}[\ell^{(g)}] = N$, with $\sum_{\ell=1}^{\ell^{(g)}} k^{(g)}[\ell] = k - k^{(g)}[\ell^{(g)}]$. ■

Proof of Theorem 3: First, note that the delay spectrum described is the same as the one in Lemma 3, therefore, the achievable side of the statement (i.e., there exists a code with this delay spectrum that achieves it under N erasures) is completed. We now must show that this code is better than any other code, i.e., (17) holds for any other code. More precisely, we will use Proposition 1, which is a sufficient condition for optimality.

Let us consider an arbitrary achievable delay spectrum $\mathbf{k}'^{(g)}$. Recall that $\sum_{\ell=1}^{\ell^{(g)}} k'^{(g)}[\ell] = k$. Then, let us rearrange (16) as follows

$$\mathbf{T}^{(g)}[j] \geq \frac{Nn}{n-k} \left(1 - \sum_{\ell=1}^{j-1} \frac{k'^{(g)}[\ell]}{n}\right) - 1 \quad (108)$$

$$(\mathbf{T}^{(g)}[j] + 1) \frac{n-k}{N} \geq n-k + \sum_{\ell=j}^{\ell^{(g)}} k'^{(g)}[\ell] \quad (109)$$

$$\sum_{\ell=j}^{\ell^{(g)}} k'^{(g)}[\ell] \leq (\mathbf{T}^{(g)}[j] + 1 - N) \frac{n-k}{N}. \quad (110)$$

Now, let us solve this for $j = \ell^{(g)}, \ell^{(g)} - 1, \dots, 1$. We get

$$k'^{(g)}[\ell^{(g)}] \leq \frac{n-k}{N} = k^{(g)}[\ell^{(g)}] \quad (111)$$

$$\begin{aligned} k'^{(g)}[\ell^{(g)}] + k'^{(g)}[\ell^{(g)} - 1] &\leq 2 \frac{n-k}{N} \\ &= k^{(g)}[\ell^{(g)}] + k^{(g)}[\ell^{(g)} - 1] \end{aligned} \quad (112)$$

$$\vdots \quad (113)$$

$$\begin{aligned} \sum_{\ell=2}^{\ell^{(g)}} k'^{(g)}[\ell] &\leq (\ell^{(g)} - 1) \frac{n-k}{N} \\ &= \sum_{\ell=2}^{\ell^{(g)}} k^{(g)}[\ell]. \end{aligned} \quad (114)$$

This can be interpreted as follows: there is a maximum number of symbols that can be transmitted with the best delay, that is, there is a direct upper bound on $k'^{(g)}[\ell^{(g)}]$, which is the number of symbols transmitted with the lowest delay (i.e., N). Furthermore, note that transmitting less symbols with the lowest delay only allows (at best) exactly that same number of symbols (i.e., the difference between the bound and the number of symbols transmitted) to be transmitted with worse delays, therefore, if equality can be achieved, then it is optimal, as the only choice is to transmit those symbols with a better delay or not, and there is no reason to transmit with a worse delay. Finally, we note that we stopped one short in (114). This is because $\sum_{\ell=1}^{\ell^{(g)}} k'^{(g)}[\ell] = \sum_{\ell=1}^{\ell^{(g)}} k^{(g)}[\ell] = k$, thus the equality holds trivially from definition. Therefore, from Lemma 3, delay spectrum $\mathbf{k}^{(g)}$ is achievable, and, as seen above, it is optimal. ■

Proof of Lemma 5: Let us consider using $(T+1-N_3)c$ concatenations of a $\frac{T+1-N_3-N_2}{T+1-N_2}$ interleaving MDS code in the second hop and $(T+1-N_2)c$ concatenations of a $\frac{T+1-N_3-N_1}{T+1-N_3}$ code in the first link. Finally, let us fix $n = (T+1-N_2)(T+1-N_3)c$. Note that, under such conditions, we achieve the desired R_1 from the assumption of the Lemma. The constant c is an auxiliary constant that should satisfy that the number of symbols in each timeslot is integer for all links. These choices of code parameters are taken from the upper bound. Examples of such codes can be found in Section V-D.2.

Under such conditions, it follows from Lemma 3 that the delay spectrum of the second hop is uniform with $(T+1-N_3)c$ symbols from N_3 to $T-N_2$, and the delay spectrum of the first link is $(T+1-N_2)c$ symbols from N_1 to $T-N_3$. Therefore, we have the following constraint on the second link

$$\mathbf{G}^{\text{con}} = \begin{bmatrix} (T-N_3, (N_2-N_3)c) \\ (T-N_3-1, (N_2-N_3)c) \\ \vdots \\ (N_1, (N_2-N_3)c) \\ (N_1+1, (T+1-N_3)c) \\ \vdots \\ (N_2, (T+1-N_3)c) \end{bmatrix} \quad (115)$$

which follows from computing the number of symbols still allowed to be transmitted with each delay from relay to destination (i.e., that were not already used up by the first user).

Under this constraint, applying Corollary 1, we have the following conditions for achievability on the second link

$$\begin{aligned} k_2 &\leq n - \frac{nN_2}{T-N_3+1} \\ k_2 &\leq n - \frac{nN_2(1 - \frac{(N_2-N_3)c}{n})}{(T-N_3-1)+1} \end{aligned} \quad (116)$$

$$\begin{aligned}
k_2 &\leq n - \frac{nN_2(1 - 2\frac{(N_2-N_3)c}{n})}{(T - N_3 - 2) + 1} \\
&\vdots \\
k_2 &\leq n - \frac{nN_2(1 - (T + 1 - N_3 - N_1)\frac{(N_2-N_3)c}{n})}{N_1} \quad (117) \\
k_2 &\leq n - \left(\frac{nN_2(1 - (T + 1 - N_3 - N_1)\frac{(N_2-N_3)c}{n})}{N_1 - 1} \right. \\
&\quad \left. - \frac{(T+1-N_3)c}{N_1 - 1} \right) \\
k_2 &\leq n - \left(\frac{nN_2(1 - (T + 1 - N_3 - N_1)\frac{(N_2-N_3)c}{n})}{N_1 - 2} \right. \\
&\quad \left. - \frac{2\frac{(T+1-N_3)c}{n}}{N_1 - 2} \right) \\
&\vdots \\
k_2 &\leq n - \left(\frac{nN_2(1 - (T + 1 - N_3 - N_1)\frac{(N_2-N_3)c}{n})}{N_2} \right. \\
&\quad \left. - \frac{(N_1 - N_2)\frac{(T+1-N_3)c}{n}}{N_2} \right) \quad (118)
\end{aligned}$$

We note that equation (116) represents the single user upper bound, that is, $R_2 \leq C(T - N_3, N_2)$. Equation (118) represents the sumrate bound, i.e., $R_1 + R_2 \leq C(T - N_2, N_3)$. While there are many other constraints to be considered, we now show that they are dominated by (117) and (118) when $R_1 = C(T - N_3, N_1)$. In order to do that, let us first consider the function

$$f(x) = n - n \frac{N_2(1 - x\frac{(N_2-N_3)c}{n})}{(T - N_3 - x) + 1} \quad (119)$$

with derivative

$$\begin{aligned}
f'(x) &= -nN_2 \left[-\frac{\frac{(N_2-N_3)c}{n}}{(T - N_3 - x) + 1} \right. \\
&\quad \left. + \left(1 - x\frac{(N_2 - N_3)c}{n} \right) \frac{1}{(T + 1 - N_3 - x)^2} \right]. \quad (120)
\end{aligned}$$

It can be shown that, for $T \geq 2N_2 - N_3 - 1 = c_1$, this derivative is always negative. Further, recall that, in order for the second hop to be the bottleneck (i.e., the scenario we are interested on), we require $T \geq \frac{1}{2} \left(\sqrt{N_1^2 - 4N_3(N_2 - N_3)} + N_1 + 2N_2 - 2 \right) = c_2$. It can be shown that, for any $N_1 \geq N_2 \geq N_3$, we have $c_2 \geq c_1$, that is, in our regime of operation, this derivative is always negative. From this, it follows that (117) is the minimum among all constraints from (116) to (117).

Similarly, we consider the function

$$\begin{aligned}
g(x) &= n \\
&\quad - nN_2 \frac{1 - (T + 1 - N_3 - N_1)\frac{(N_2-N_3)c}{n} - x\frac{(T+1-N_3)c}{n}}{N_1 - x} \quad (121)
\end{aligned}$$

with derivative

$$\begin{aligned}
g'(x) &= -nN_2 \left[-\frac{\frac{(T+1-N_3)c}{n}}{N_1 - x} \right. \\
&\quad \left. + \frac{1 - (T + 1 - N_3 - N_1)\frac{(N_2-N_3)c}{n} - x\frac{(T+1-N_3)c}{n}}{(N_1 - x)^2} \right]. \quad (122)
\end{aligned}$$

It can be shown that the derivative is negative if

$$(T+1-N_3)(T+1-N_2-N_1) - (T+1-N_3-N_1)(N_2-N_3) \geq 0 \quad (123)$$

and positive otherwise. That is, either the derivative is always negative and (118) is the minimum, or the derivative is always positive and (117) is the minimum. Dividing both sides by n (in order to obtain $R_2 = k_2/n$) and then substituting $n = (T + 1 - N_2)(T + 1 - N_3)c$ in both equations completes the proof. ■

Proof of Corollary 3: This follows directly from the condition for $R'_2 \geq C(T - N_2, N_3) - C(T - N_3, N_1)$ presented in the proof for Lemma 5. ■

Proof of Lemma 6: Consider the following scheme: in the second hop, we use $(T + 1 - N_3)c$ concatenations of a $\frac{T+1-N_3-N_2}{T+1-N_2}$ code and the second link (of the first hop) uses $(T + 1 - N_3)c$ concatenations of a $\frac{T+1-N_2-N_2}{T+1-N_2}$ code. Note that we have $n_2 = n_3 = (T + 1 - N_2)(T + 1 - N_3)c$. Then, we use $c(T + 1 - N_3)(N_2 - N_3)/(T + 1 - N_3 - N_1)$ codes of rate $\frac{T+1-N_1-N_3}{T+1-N_3}$ in the first link. Again, c is an auxiliary constant that should satisfy that the number of concatenations is integer, e.g., $c = (T + 1 - N_1 - N_3)$. Now, note that we have

$$k_2 = c(T + 1 - N_2 - N_2)(T + 1 - N_3) \quad (124)$$

$$k_1 = c(T + 1 - N_3)(N_2 - N_3) \quad (125)$$

$$k = c(T + 1 - N_3 - N_2)(T + 1 - N_3) \quad (126)$$

and we have exactly k symbols being transmitted in the bottleneck. Furthermore, it is easy to see that the delay spectrum of these codes achieve the desired overall delay under N_1, N_2 and N_3 erasures. This is because the code used in the second hop allows for $(T + 1 - N_3)c$ symbols to be transmitted with delays $\{N_2, N_2 + 1, \dots, T - N_3\}$ by the users in the first hop. The second user transmits $(T + 1 - N_3)c$ symbols with each delay $\{N_2, N_2 + 1, \dots, T - N_2\}$. Thus, the first user can transmit $(T + 1 - N_3)c$ symbols in each delay $\{T - N_2 + 1, T - N_2 + 2, \dots, T - N_3\}$, which is satisfied by the delay spectrum of the code employed by the first user. Then, it remains to find $n = \max(n_1, n_2)$. Note that if $n = n_2$, we achieve the sum capacity. Thus, the condition for achieving the sum capacity is

$$c(T + 1 - N_2)(T + 1 - N_3 - N_1) \geq c(T + 1 - N_3)(N_2 - N_3) \quad (127)$$

Note that this condition is equivalent to the condition that we achieve the sum capacity at some point (from Corollary 3).

Further, note that, if this condition holds, we have

$$R_1 = \frac{c(T+1-N_3) * (N_2 - N_3)}{c(T+1-N_2)(T+1-N_3)} \quad (128)$$

$$= \frac{N_2 - N_3}{T+1-N_2} \quad (129)$$

which completes the proof. \blacksquare

Remark 11: Note that, if the condition for achievability of the sum capacity holds, we have

$$\frac{N_2 - N_3}{T+1-N_2} \leq C(T-N_3, N_1) \quad (130)$$

and

$$C(T-N_2, N_2) \geq C(T-N_2, N_3) - C(T-N_3, N_1). \quad (131)$$

That is, the point that achieves the sum capacity in Lemma 6 is to the “left” and “up” in the rate region than the point that achieves it in Lemma 5. In fact, this is the point with highest R_2 that can achieve the sum capacity using the family of schemes we propose.

Proof of Theorem 2: For simplicity, let us consider a code construction similar to the one used in Lemma 5. As before, the relay employs $(T+1-N_3)c$ concatenations of a $\frac{T+1-N_3-N_2}{T+1-N_2}$ diagonally-interleaved MDS code in the second hop. Similarly, let us also fix $n = (T+1-N_2)(T+1-N_3)c$. The code construction for the first user is as follows: as in the proof of Lemma 5, the first user employs multiple concatenations of a $\frac{T+1-N_3-N_1}{T+1-N_3}$ diagonally-interleaved MDS code. However, instead of employing $(T+1-N_2)c$ concatenations, the user only employs $\frac{R_1 n}{T+1-N_1-N_3}$ concatenations. Note that, if $R_1 = \frac{T+1-N_1-N_3}{T+1-N_3}$, as in Lemma 5, both codes coincide. The delay spectrum of this code is uniform, with $\frac{R_1 n}{T+1-N_1-N_3}$ symbols transmitted with each delay $\{N_1, N_1+1, \dots, T-N_3\}$. We denote by $R_{bn} = C(T-N_3, N_2) = \frac{T+1-N_3-N_2}{T+1-N_2}$. When the first user employs this code, as opposed to the one in Lemma 5, the constraints to the code employed by the second user change to

$$k_2 \leq n - \frac{nN_2}{T-N_3+1} \quad (132)$$

$$k_2 \leq n - \frac{nN_2(1 - R_{bn}\frac{1}{T+1-N_2-N_3} + R_1\frac{1}{T+1-N_1-N_3})}{(T-N_3-1)+1}$$

$$k_2 \leq n - \frac{nN_2(1 - R_{bn}\frac{2}{T+1-N_2-N_3} + R_1\frac{2}{T+1-N_1-N_3})}{(T-N_3-2)+1}$$

\vdots

$$k_2 \leq n - \frac{nN_2(1 - R_{bn}\frac{T+1-N_1-N_3}{T+1-N_2-N_3} + R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3})}{N_1} \quad (133)$$

$$k_2 \leq n - \left(\frac{nN_2(1 - R_{bn}\frac{T+1-N_1-N_3}{T+1-N_2-N_3})}{N_1-1} + \frac{R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3} - \frac{(T+1-N_3)c}{n}}{N_1-1} \right)$$

$$k_2 \leq n - \left(\frac{nN_2(1 - R_{bn}\frac{T+1-N_1-N_3}{T+1-N_2-N_3})}{N_1-2} + \frac{R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3} - 2\frac{(T+1-N_3)c}{n}}{N_1-2} \right)$$

\vdots

$$k_2 \leq n - \left(\frac{nN_2(1 - R_{bn}\frac{T+1-N_1-N_3}{T+1-N_2-N_3})}{N_2} + \frac{R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3} - (N_1-N_2)\frac{(T+1-N_3)c}{n}}{N_2} \right) \quad (134)$$

Similar to before, let us consider the function

$$f(x) = n - nN_2 \frac{1 - R_{bn}\frac{x}{T+1-N_2-N_3} + R_1\frac{x}{T+1-N_1-N_3}}{(T-N_3-x+1)} \quad (135)$$

with derivative

$$f'(x) = -nN_2 \left[\frac{\frac{R_1}{T+1-N_1-N_3} - \frac{R_{bn}}{T+1-N_2-N_3}}{T+1-N_3-x} + \frac{\left(1 - \frac{xR_{bn}}{T+1-N_2-N_3} + \frac{xR_1}{T+1-N_1-N_3}\right)}{(T+1-N_3-x)^2} \right]. \quad (136)$$

Then, note that

$$f'(x) \leq 0 \Rightarrow \left[\left(\frac{R_1(T+1-N_3)}{T+1-N_1-N_3} - \frac{(T+1-N_3)}{T+1-N_2} \right) + 1 \right] \geq 0. \quad (137)$$

That is, for any set of parameters and R_1 , this derivative is always positive or always negative, thus, the minimum of all expressions between (132) and (133) is always in the extremes, i.e., one of these two expressions.

Similarly, let us consider the function

$$g(x) = n - \left(\frac{nN_2(1 - R_{bn}\frac{T+1-N_1-N_3}{T+1-N_2-N_3})}{N_1-x} + \frac{R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3} - x\frac{(T+1-N_3)c}{n}}{N_1-x} \right) \quad (138)$$

with derivative

$$g'(x) = -nN_2 \left[-\frac{\frac{(T+1-N_3)c}{n}}{N_1-x} + \frac{\left(1 - R_{bn}\frac{T+1-N_1-N_3}{T+1-N_2-N_3}\right)}{(N_1-x)^2} + \frac{R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3} - x\frac{(T+1-N_3)c}{n}}{(N_1-x)^2} \right]. \quad (139)$$

And again, note that

$$g'(x) \leq 0 \Rightarrow \left[1 - \frac{T+1-2N_1-N_3}{T+1-N_2} + R_1\frac{T+1-N_1-N_3}{T+1-N_1-N_3} \right] \geq 0. \quad (140)$$

That is, again, the derivative is always negative or always positive depending on the set of parameters and R_1 , which means the minimum of all expressions between (133) and (134) lies in the extreme points, i.e., one of these two expressions. Therefore, it suffices to take only these three points into consideration. Finding $R_2 = k_2/n$ and substituting $n = (T + 1 - N_3)(T + 1 - N_2)c$ completes the proof. ■

Proof of Lemma 7: The proof follows directly from concatenating the two streaming codes used. More specifically, the first user concatenates A copies of the (n_1, k_1) code with B copies of the (n'_1, k'_1) code, the second users concatenates A copies of the (n_2, k_2) code with B copies of the (n'_2, k'_2) code, and the relay concatenates A copies of the $(n_3, k_1 + k_2)$ code with B copies of the $(n'_3, k'_1 + k'_2)$ code. Then, from the fact that these codes are able to achieve delay T under (N_1, N_2, N_3) erasures, it follows from Lemma 1 that the concatenation of these codes is also able to achieve delay T under (N_1, N_2, N_3) erasures. ■

Proof of Lemma 9: This follows directly from applying Lemma 7 to the streaming codes described for the CSWDF, that is

$$(T + 1 - N_3, 0, T + 1 - N_1, T + 1 - N_1 - N_3, 0)_{\mathbb{F}} \quad (141)$$

$$(0, T + 1 - N_3, T + 1 - N_2, 0, T + 1 - N_2 - N_3)_{\mathbb{F}} \quad (142)$$

Specifically, we use $A = N_2 - N_3$ and $B = T + 1 - N_1$, therefore, we have a streaming code certain to achieve the required delay, with parameters

$$k_1 = (N_2 - N_3)(T + 1 - N_1 - N_3) \quad (143)$$

$$n_1 = (N_2 - N_3)(T + 1 - N_3) \quad (144)$$

$$k_2 = (T + 1 - N_1)(T + 1 - N_2 - N_3) \quad (145)$$

$$n_2 = (T + 1 - N_1)(T + 1 - N_3) \quad (146)$$

$$n_3 = (N_2 - N_3)(T + 1 - N_1) + (T + 1 - N_1)(T + 1 - N_2). \quad (147)$$

Then, it remains to find n and the respective rates. Note that

$$n_3 = (T + 1 - N_1)(T + 1 - N_3) = n_2. \quad (148)$$

Finally, we need to show that

$$T + 1 - N_1 \geq N_2 - N_3 \quad (149)$$

which implies $n_3 \geq n_1$. The proof is as follows: recall that we assume

$$T \geq \frac{1}{2} \left(\sqrt{N_1^2 - 4N_3(N_2 - N_3)} + N_1 + 2N_2 - 2 \right)$$

thus, it suffices to show that this assumption implies that the above condition holds. We start by computing

$$\begin{aligned} & \frac{1}{2} \left(\sqrt{N_1^2 - 4N_3(N_2 - N_3)} + N_1 + 2N_2 - 2 \right) \\ & - (N_1 + N_2 - N_3 - 1) \\ & = \frac{1}{2} \left(\sqrt{N_1^2 - 4N_3(N_2 - N_3)} - N_1 + 2N_3 \right). \end{aligned} \quad (150)$$

In order to have $T + 1 - N_1 \geq N_2 - N_3$ guaranteed, we need this expression to be greater than zero, or

$$\left(\sqrt{N_1^2 - 4N_3(N_2 - N_3)} \right) \geq N_1 - 2N_3. \quad (151)$$

Now, note that, if $N_1 - 2N_3 \leq 0$, then this condition holds. If $N_1 - 2N_3 \geq 0$, then we can rewrite the condition as

$$N_1^2 - 4N_3(N_2 - N_3) \geq N_1^2 - 4N_1N_3 + 4N_3^2 \quad (152)$$

$$-4N_3N_2 \geq -4N_1N_3 \quad (153)$$

$$N_1N_3 \geq N_2N_3 \quad (154)$$

$$N_1 \geq N_2. \quad (155)$$

Finally, recall that this is also an assumption, therefore, in this regime of operation, we always have $n = n_3 = n_2 \geq n_1$. Finally, we can compute the rates

$$R_1 = \frac{(N_2 - N_3)(T + 1 - N_1 - N_3)}{(T + 1 - N_1)(T + 1 - N_3)} \quad (156)$$

$$\begin{aligned} R_2 &= \frac{(T + 1 - N_1)(T + 1 - N_2 - N_3)}{(T + 1 - N_1)(T + 1 - N_3)} \\ &= \frac{T + 1 - N_2 - N_3}{T + 1 - N_3} = C(T - N_3, N_2) \end{aligned} \quad (157)$$

which are the rates described in the lemma. ■

Proof (More Details on Proof of Lemma 12): This follows directly from the fact that, under such condition, we can use the same scheme used for the strong source-relay bottleneck, that is, a simple CSWDF with one concatenation of each single-user capacity-achieving code (i.e. $A = B = 1$). In that case, we have, as before, $n_3 = (T + 1 - N_1) + (T + 1 - N_2)$, and $n_1 = n_2 = T + 1 - N_3$. Then, if $n_1 \geq n_3$, we achieve the capacity, since the rates will be $R_1 = k_1/n_1 = C(T - N_3, N_1)$ and $R_2 = k_2/n_2 = C(T - N_3, N_2)$. To complete the proof, it is sufficient to notice that the condition in the Lemma is exactly the condition for $n_1 \geq n_3$. ■

REFERENCES

- [1] G. K. Facenda, E. Domanovitz, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Streaming erasure codes over multi-access relay networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 724–729.
- [2] 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 Trans. Inf. Theory*, vol. 66, no. 5, pp. 2696–2712, May 2020.
- [3] E. Martinian and C. E. W. Sundberg, "Burst erasure correction codes with low decoding delay," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2494–2502, Oct. 2004.
- [4] D. Leong and T. Ho, "Erasure coding for real-time streaming," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 289–293.
- [5] A. Badr, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Streaming codes for channels with burst and isolated erasures," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2850–2858.
- [6] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, vol. 41, no. 1, 2003, pp. 11–20.
- [7] G. Joshi, Y. Kochman, and G. W. Wornell, "On playback delay in streaming communication," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2856–2860.
- [8] M. Karzand, D. J. Leith, J. Cloud, and M. Médard, "Design of FEC for low delay in 5G," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 8, pp. 1783–1793, Aug. 2017.
- [9] A. Badr, P. Patil, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Layered constructions for low-delay streaming codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 1, pp. 111–141, Jan. 2017.
- [10] A. Badr, A. Khisti, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "FEC for VoIP using dual-delay streaming codes," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.
- [11] M. Rudow and K. V. Rashmi, "Streaming codes for variable-size arrivals," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2018, pp. 733–740.

- [12] M. N. Krishnan and P. V. Kumar, "Rate-optimal streaming codes for channels with burst and isolated erasures," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1809–1813.
- [13] 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 Trans. Inf. Theory*, vol. 65, no. 7, pp. 4274–4292, Jul. 2019.
- [14] E. Domanovitz, S. L. Fong, and A. Khisti, "An explicit rate-optimal streaming code for channels with burst and arbitrary erasures," *IEEE Trans. Inf. Theory*, vol. 68, no. 1, pp. 47–65, Jan. 2022.
- [15] M. N. Krishnan, D. Shukla, and P. V. Kumar, "Low field-size, rate-optimal streaming codes for channels with burst and random erasures," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4869–4891, Aug. 2020.
- [16] E. Domanovitz, A. Khisti, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Streaming erasure codes over multi-hop relay network," 2020, *arXiv:2006.05951*.
- [17] M. N. Krishnan, G. K. Facenda, E. Domanovitz, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "High rate streaming codes over the three-node relay network," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2021, pp. 1–6.
- [18] A. Cohen, G. Thiran, V. Bar Bracha, and M. Médard, "Adaptive causal network coding with feedback for multipath multi-hop communications," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 766–785, Feb. 2021.
- [19] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Math. Comput. Model.*, vol. 57, nos. 11–12, pp. 2883–2894, Jun. 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0895717711007771>
- [20] P. Quax, A. Beznosyik, W. Vanmontfort, R. Marx, and W. Lamotte, "An evaluation of the impact of game genre on user experience in cloud gaming," in *Proc. IEEE Int. Games Innov. Conf. (IGIC)*, Sep. 2013, pp. 216–221.
- [21] V. Clincy and B. Wilgor, "Subjective evaluation of latency and packet loss in a cloud-based game," in *Proc. 10th Int. Conf. Inf. Technol., New Generat.*, Apr. 2013, pp. 473–476.
- [22] M. Claypool and D. Finkel, "The effects of latency on player performance in cloud-based games," in *Proc. 13th Annu. Workshop Netw. Syst. Support Games*, Dec. 2014, pp. 1–6.
- [23] I. Slivar, M. Suznjevic, L. Skorin-Kapov, and M. Matijasevic, "Empirical QoE study of in-home streaming of online games," in *Proc. 13th Annu. Workshop Netw. Syst. Support Games*, Dec. 2014, pp. 1–6.
- [24] Z.-Y. Wen and H.-F. Hsiao, "QoE-driven performance analysis of cloud gaming services," in *Proc. IEEE 16th Int. Workshop Multimedia Signal Process. (MMSp)*, Sep. 2014, pp. 1–6.
- [25] S. Schmidt, S. Zadtootaghaj, and S. Moller, "Towards the delay sensitivity of games: There is more than genres," in *Proc. 9th Int. Conf. Quality Multimedia Exper. (QoMEX)*, May 2017, pp. 1–6.
- [26] P. Maynard-Koran. (Feb. 2016). *Fixing the Internet for Real Time Applications*. [Online]. Available: <https://technology.riotgames.com/news/fixing-internet-real-time-applications-part-ii>
- [27] G. Hains, C. Mazur, J. Ayers, J. Humphrey, Y. Khmelevsky, and T. Sutherland, "The WTFast's gamers private network (GPN) performance evaluation results," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Aug. 2020, pp. 1–6.
- [28] K. Lee et al., "Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming," in *Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*. New York, NY, USA: Association for Computing Machinery, 2015, pp. 151–165, doi: [10.1145/2742647.2742656](https://doi.org/10.1145/2742647.2742656).
- [29] 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 *Proc. 7th Int. Workshop Quality Multimedia Exper. (QoMEX)*, May 2015, pp. 1–6.
- [30] M. Carrascosa and B. Bellalta, "Cloud-gaming: Analysis of Google stadia traffic," *Comput. Commun.*, vol. 188, pp. 99–116, Apr. 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366422000810>
- [31] I. Slivar, L. Skorin-Kapov, and M. Suznjevic, "QoE-aware resource allocation for multiple cloud gaming users sharing a bottleneck link," in *Proc. 22nd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2019, pp. 118–123.
- [32] A. Badr, D. Lui, A. Khisti, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Multiplexed coding for multiple streams with different decoding delays," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4365–4378, Jun. 2018.

Gustavo Kasper Facenda (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the University of Toronto, Toronto, ON, 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.

Elad Domanovitz (Member, IEEE) received the B.Sc. (*cum laude*), M.Sc., and Ph.D. degrees in electrical engineering from Tel Aviv University, Israel, in 2005, 2011, and 2020, respectively. 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.

Ashish Khisti (Member, IEEE) 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 with the Electrical and Computer Engineering (ECE) Department, University of Toronto, where he was an Assistant Professor from 2009 to 2015, and an Associate Professor from 2015 to 2019, where he 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 the B.S. degree in electrical engineering from Brown University, the M.S. degree in electrical engineering from Stanford University, and the Ph.D. degree in electrical engineering from UC Berkeley. He works in the general area of multimedia networking, systems for video communications, and wireless networking. He was a Principal Engineer in Enterprise Networking Business with Cisco, and worked with Hewlett Packard Laboratories, Palo Alto, before joining Cisco.

John Apostolopoulos (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees from the MIT. He was a VP/CTO of Cisco's Intent Based Networking Group (Cisco's largest business), and also founded Cisco's Innovation Labs. He was a Consulting Associate Professor of EE at Stanford University. His coverage included wireless (Wi-Fi 6, 5G, OpenRoaming), the Internet of Things, multimedia networking, software-defined WAN, and ML/AI applied to the above. Previously, he was a Distinguished Technologist, and then the Laboratory Director of the Mobile & Immersive Experience (MIX) Laboratory, HP Labs. The MIX Laboratory conducted research on novel mobile devices and sensing, mobile client/cloud multimedia computing, immersive environments, video and audio signal processing, computer vision and graphics, multimedia networking, glasses-free 3D, wireless, and user experience design. He published over 100 articles, receiving five best paper awards, and over 100 granted U.S. patents. He is an IEEE SPS Distinguished Lecturer, named "one of the world's top 100 young innovators" by MIT Technology Review, contributed to the U.S. 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.