

Successive Segmentation-based Coding for Broadcasting over Erasure Channels

Louis Tan *Student Member, IEEE*, Yao Li *Member, IEEE*, Ashish Khisti *Member, IEEE* and Emina Soljanin *Fellow, IEEE*.

Abstract—Motivated by error correction coding in multimedia applications, we study the problem of broadcasting a single common source to multiple receivers over heterogeneous erasure channels. Each receiver is required to partially reconstruct the source sequence by decoding a certain fraction of the source symbols. We propose a coding scheme that requires only off-the-shelf erasure codes and can be easily adapted as users join and leave the network. Our scheme involves splitting the source sequence into multiple segments and applying a systematic erasure code to each such segment. We formulate the problem of minimizing the transmission latency at the server as a linear programming problem and explicitly characterize an optimal choice for the code-rates and segment sizes. Through numerical comparisons, we demonstrate that our proposed scheme outperforms both separation-based coding schemes, and degree-optimized rateless codes and performs close to a natural outer bound in certain cases.

We further study *individual* user decoding delays for various orderings of segments in our scheme. We provide closed-form expressions for each individual user's excess latency when parity checks are successively transmitted in both increasing and decreasing order of their segment's coded rate and also qualitatively discuss the merits of each order.

Index Terms—Application-Layer Error Correction Coding, Broadcast Channels, Joint Source-Channel Coding, Linear Programming, Multimedia broadcast/multicast services (MBMS), Rateless Codes, Unequal Error Protection.

I. INTRODUCTION

Consumers of video and other content in today's networks have very diverse video and computing equipment ranging from mobile phones and handheld devices to desktops and HDTVs. When serving multiple diverse users, the most straightforward approach is to establish independent unicast sessions. However, when a large number of users require the same small content, (e.g., video clips at stadiums), or when a small number of users require the same large content, (e.g., a large movie), the multiple unicast approach clearly results in highly inefficient use of overall network resources. In such applications, broadcast techniques can lead to significant gains.

One important difference between point-to-point and broadcast/multicast applications lies in the way packet losses are handled. In packet-based data networks, large files are usually segmented into smaller blocks that are put into transport

packets. Packet losses occur because of the physical channel and other limitations such as processing power and buffer space. In point-to-point scenarios, the sender can adjust its transmission/coding rate to avoid packet losses and retransmit lost packets according to the feedback from the receiver through very efficient physical-layer schemes such as HARQ. In contrast, in broadcast/multicast applications, it is costly for the sender to collect and respond to individual receiver feedbacks, and thus HARQ schemes are disabled and packet losses are inevitable. Forward error correction coding provides a natural solution in such applications. A number of these schemes have already been standardized and are being implemented.

Rateless codes are a popular class of codes that enable efficient communications over multiple unknown erasure channels at the packet level by simultaneously approaching the channel capacity at all erasure rates. Raptor codes, a special class of rateless codes, also have very low encoding and decoding complexity [3]. Because of these properties, Raptor codes have been standardized for Multimedia Broadcast/Multicast Service (MBMS) and are being deployed in applications such as LTE eMBMS. Raptor codes are essentially optimal for multicast over erasure channels where all receivers require identical content.

In certain applications however, the receivers may not require all the source packets and may not have identical demands. For example, in emerging eMBMS systems, there are two distinct phases of transmission. The first phase is a fixed-rate broadcast transmission, after which, each user is left with only a subset of source packets. Each user then recovers the remaining source packets through individual unicast from a dedicated repair server. Thus, during the broadcast phase, the server is required to only deliver a fraction of source packets to each user. As another example, consider a system that applies a multiple description code (MDC) [4]–[6] to an analog source sequence to generate a large number of MDC coded packets. The reconstruction quality depends on the number of MDC packets available to the destination. Thus, each user can have a different *demand* based on its screen resolution and channel conditions. In such scenarios where the user demands are not identical, both the fundamental limits and practical coding schemes remain a fertile area of research to the best of our knowledge.

In this paper, we propose a coding scheme for transmitting to multiple receivers with heterogeneous channels and demands. Our scheme relies only on off-the-shelf erasure codes. The key idea in our scheme is to partition the source sequence into

L. Tan and A. Khisti are with the Dept. of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. Y. Li is with the Dept. of Electrical Engineering, UCLA, Box 951594, Los Angeles, CA 90095. E. Soljanin is with Bell Labs, Alcatel-Lucent, Murray Hill, NJ 07974, USA. Part of this work was presented at the 2013 Information Theory Workshop in Seville, Spain [1] and will be presented at the 2014 International Symposium on Information Theory [2] in Honolulu, Hawaii.

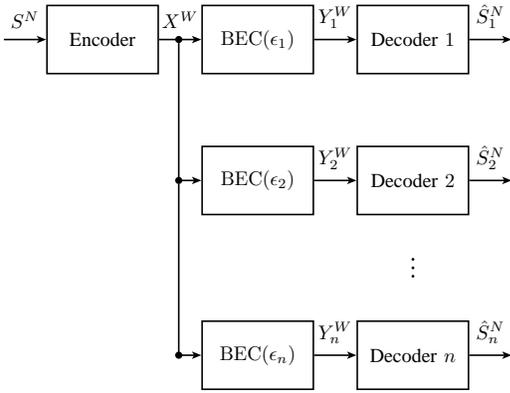


Fig. 1: Broadcasting an equiprobable binary source over an erasure broadcast channel.

multiple non-overlapping segments and to apply a systematic erasure code to each segment. We formulate the problem of selecting the segment lengths and code rates that minimize the transmission latency as a linear programming problem and characterize an explicit solution. We discuss how the solution naturally evolves as users join or leave the network. We further compare our scheme numerically with separation-based schemes, and degree-optimized rateless codes and demonstrate that significant performance gains are possible. We also discuss how a tradeoff between the latencies of individual users can be attained by selecting various transmission orders for the parity checks.

Throughout this paper, we adhere to the notation defined herein. The sample space of a random variable is written in calligraphic font, e.g., \mathcal{S} and we let \mathcal{S}^N be the set of all N -vectors with components in \mathcal{S} . We use t when referring to the symbol-index of a vector, which is enclosed in round brackets when actually referring to a vector component. Thus, the t^{th} component of a vector $S^N \in \mathcal{S}^N$ is denoted by $S(t)$ so that S^N in fact denotes $(S(1), S(2), \dots, S(N))$. In general, a variable's subscript is reserved for user indices and indicates a correspondence between a user and the variable in question. For example, when the symbol d is used for distortion, d_i denotes the distortion of user i . Finally, for convenience, we also denote the set $\{1, 2, \dots, N\}$ as $[N]$.

II. SYSTEM MODEL

The problem is illustrated in Fig. 1. We consider binary memoryless source $\{S(t)\}_{t=1,2,\dots}$ which produces equiprobable symbols in the alphabet $\mathcal{S} = \{0, 1\}$ and which we wish to communicate to n users over an erasure broadcast channel. The source is communicated by a block-encoding function that maps a length- N source sequence, S^N , to a length- W channel input sequence, $X^W = (X(1), X(2), \dots, X(W))$, where $X(t)$ denotes the t^{th} channel input taken from the alphabet $\mathcal{X} = \{0, 1\}$.

Let $Y_i(t)$ be the channel output observed by user i on the t^{th} channel use for $i \in [n]$ and $t \in [W]$. Our channel model is a binary erasure broadcast channel as shown in Fig. 1. In particular, let ϵ_i denote the erasure rate of the channel corresponding to user i , where we assume that $0 < \epsilon_1 < \epsilon_2 < \dots < \epsilon_n < 1$.

This is without loss of generality since we can address users that experience identical erasure rates by serving the user with the most stringent distortion requirement. Our model specifies that $Y_i(t)$ exactly reproduces the channel input $X(t)$ with probability $(1 - \epsilon_i)$ and otherwise indicates an erasure event, which happens with probability ϵ_i . We let $Y_i(t)$ take on values in the alphabet $\mathcal{Y} = \{0, 1, \star\}$ so that an erasure event is represented by \star , the erasure symbol.

Having observed his channel output, user i then uses it to reconstruct the source as a length- N sequence, denoted as \hat{S}_i^N . We will be interested in a fractional recovery requirement so that each symbol in \hat{S}_i^N either faithfully recovers the corresponding symbol in S^N or otherwise a failure is indicated with an erasure symbol, i.e., we do not allow for any bit flips.

More precisely, we choose the reconstruction alphabet $\hat{\mathcal{S}}$ to be an augmented version of the source alphabet so that $\hat{\mathcal{S}} = \{0, 1, \star\}$, where the additional \star symbol indicates an erasure symbol. We then express the constraint that an achievable code ensures that each user $i \in [n]$ achieves a fractional recovery of $1 - d_i$, where $d_i \in [0, 1]$, with the following definition.

Definition 1. An $(N, W, d_1, d_2, \dots, d_n)$ code for source S on the erasure broadcast channel consists of

- 1) an encoding function $f_N : \mathcal{S}^N \rightarrow \mathcal{X}^W$ such that $X^W = f_N(S^N)$, and
- 2) n decoding functions $g_{i,N} : \mathcal{Y}^W \rightarrow \hat{\mathcal{S}}^N$ such that $\hat{S}_i^N = g_{i,N}(Y_i^W)$ and for each $i \in [n]$,
 - a) \hat{S}_i^N is such that for $t \in [N]$, if $\hat{S}_i(t) \neq S(t)$, then $\hat{S}_i(t) = \star$,
 - b) $\mathbb{E} \left[\left| \{t \in [N] \mid \hat{S}_i(t) = \star\} \right| \right] \leq Nd_i$,

where $\mathbb{E}(\cdot)$ is the expectation operation and $|A|$ denotes the cardinality of set A .

For a given code, we next define the *latency* that the code requires before all users can recover their desired fraction of the source. Finally, we then state our problem as characterizing the achievable latencies under a prescribed distortion vector as per the following definitions.

Definition 2. The latency, w , of an $(N, W, d_1, d_2, \dots, d_n)$ code is the number of channel uses per source symbol that the code requires to meet all distortion demands, i.e., $w = W/N$.

Definition 3. Latency w is said to be (d_1, d_2, \dots, d_n) -achievable over the erasure broadcast channel if for every $\delta > 0$, there exists for sufficiently large N , an $(N, wN, \hat{d}_1, \hat{d}_2, \dots, \hat{d}_n)$ code such that for $i \in [n]$, $d_i + \delta \geq \hat{d}_i$.

Remark 1. Throughout this paper we will assume that for each user $i \in [n]$, we have that $d_i < \epsilon_i$. Any user with $d_i \geq \epsilon_i$ will be trivially satisfied by the systematic portion of our segmentation-based coding scheme. Furthermore, we will show in Lemma 4 that within our class of coding schemes, such a systematic portion can be transmitted without loss of optimality when at least one user satisfies $d_i < \epsilon_i$. Finally, if every user satisfies $d_i \geq \epsilon_i$, a simple uncoded transmission scheme is easily shown to be optimal.

Remark 2. While our system model has assumed binary alphabets for both the source and channel input sequences,

our results can be easily extended to larger alphabet sizes for the purpose of applying our results to packet erasure networks.

A code that satisfies the content demands of a set of users may in fact afford different users the ability to finish receiving their content at intervals so that some users require only a short latency while others require longer ones (e.g., see [7]). In particular, we can also define what we will call a *discretized code* that accounts for users' separate decoding latencies as follows.

Definition 4. An $(N, W_1, W_2, \dots, W_n, d_1, d_2, \dots, d_n)$ discretized code for source S on the erasure broadcast channel consists of

- 1) an encoding function $f_N : \mathcal{S}^N \rightarrow \mathcal{X}^W$ such that $X^W = f_N(S^N)$, and $W = \max_{i \in [N]} W_i$,
- 2) n decoding functions $g_{i,N} : \mathcal{Y}^{W_i} \rightarrow \hat{\mathcal{S}}^N$ such that $\hat{S}_i^N = g_{i,N}(Y_i^{W_i})$ and for each $i \in [n]$,
 - a) \hat{S}_i^N is such that for $t \in [N]$, if $\hat{S}_i(t) \neq S(t)$, then $\hat{S}_i(t) = \star$,
 - b) $\mathbb{E} \left| \{t \in [N] \mid \hat{S}_i(t) = \star\} \right| \leq Nd_i$,

Using Definition 4, we can similarly define what it means when latency tuple (w_1, w_2, \dots, w_n) is (d_1, d_2, \dots, d_n) -achievable as in Definition 3.

Clearly, if we let $W = \max_{i \in [N]} W_i$, we see that an $(N, W_1, W_2, \dots, W_n, d_1, d_2, \dots, d_n)$ discretized code is also an $(N, W, d_1, d_2, \dots, d_n)$ code. Definition 4 is of interest from the perspective of content *consumers* as it concerns both the latencies that they will each have to endure for their content requirements and also the possible tradeoffs amongst themselves. Alternatively, Definition 3 is unconcerned with individual latencies and instead provides us with the *minmax latency metric* by taking the maximum over all user latencies. In this way, the minmax latency metric is of interest from a content *provider's* perspective as it will allow the provider to compare codes based on which ones minimize the overall transmission time that is required from it.

The focus in this paper will primarily be the minmax latency metric, and the solution that we propose is a code that is (minmax) latency-optimal within the class of segmentation-based codes. Our discussion of individual latencies will be limited to Section V where given a segmentation-based code, we consider different orderings of the segments for individual latency considerations.

In a related work, the minmax latency problem that we study was also treated in [8] where a set of *predetermined* messages were required by each user such that the stronger users had to decode all the messages intended for the weaker users. Such a formulation is essentially a degraded message sets problem for which superposition coding is optimal for degraded broadcast channels. For the special case of packet erasure broadcast channels, the capacity can be achieved using optimal erasure codes. In contrast, we allow for flexibility in *which* symbols are recovered so long as this number exceeds a certain threshold. Our formulation can be viewed as a *joint source-channel coding* problem involving an equiprobable binary source and an erasure distortion measure. For the case of two users, such an approach was studied in [1], [9], and coding

schemes that involved adaptations of techniques used in the Gaussian models (see e.g., [10]–[14] and references therein) were considered. To the best of our knowledge, such schemes do not attain smaller latencies than the scheme proposed in the present paper. Furthermore, such schemes involve joint source-channel code designs and do not have the practical advantages of the proposed scheme that were discussed previously. It is also worth mentioning that techniques developed for deriving outer bounds for the Gaussian model [11] have so far not resulted in non-trivial bounds in the present setup. Nevertheless, for a closely related problem involving the erasure broadcast channel and a *Hamming* distortion, non-trivial outer bounds can be obtained [15].

III. SEGMENTATION-BASED CODING

A. The Main Idea

Let v denote the user with the highest erasure rate, and consider the case when this user is the only one in our system. The optimal latency of $(1 - d_v)/(1 - \epsilon_v)$ can be achieved by e.g., first compressing the source with distortion d_v and then losslessly transmitting the compressed version of the source with a channel code of rate $(1 - \epsilon_v)$. The compression process is particularly simple in our case; we simply retain the first $N(1 - d_v)$ source sequence symbols and discard the remaining symbols. Note that this (separation) scheme can also be decoded by any user s with erasure rate $\epsilon_s \leq \epsilon_v$ and results in the same distortion d_v . Thus, if $d_s \geq d_v$, the introduction of user s into the system does not modify the code since user s does not require any dedicated coding.

Consider however, when $d_s < d_v$. We accommodate user s by incrementally modifying our coding; in addition to transmitting the first $N(1 - d_v)$ source symbols as before, we also transmit the *following* $N(d_v - d_s)$ source symbols with a channel code of rate $(1 - \epsilon_s)$. Thus, if $d_s < d_v$, the addition of user s *does* modify the code since user s *does* require dedicated coding.

It is not hard to generalize this type of coding for n users. We simply identify the users that require dedicated coding, and code for only these users by following the procedure mentioned above. In general, we see that for $1 \leq i < j \leq n$, user i is able to decode whatever was channel coded for user j . Therefore, user i requires dedicated coding only if whatever was already sent to users with worse channel qualities is not sufficient for his own distortion requirement, i.e., if $d_i < d_j$ for $j \in \{i+1, i+2, \dots, n\}$. For future reference, we will call this a *layered* coding scheme.

We observe that whenever a user does not require dedicated coding, he achieves the same distortion as some user j that has a worse channel quality and who *did* require dedicated coding. Thus, this coding does not allow for graceful improvements in distortion for increasingly favourable channel qualities. We circumvent this by modifying our coding. Consider again, the case when user v is the only user in the system. Instead of the separation-based scheme, we now split the source sequence into *two* segments. The first segment consists of a fraction of a_0 source symbols and is transmitted uncoded, while the second segment consists of another fraction of a_v source

symbols and is transmitted using a systematic channel code of rate $(1 - \epsilon_v)$. Note that the latency in this scheme is $a_0 + a_v/(1 - \epsilon_v)$, while the fraction of symbols received is $a_0(1 - \epsilon_v) + a_v$. By setting $a_0 = d_v/\epsilon_v$ and $a_v = 1 - a_0$, we achieve the same latency as the (optimal) separation-based scheme while satisfying the distortion constraint.

Fundamentally, this approach functions by first ensuring that user v losslessly recovers all but a fraction of d_v/ϵ_v source symbols via a channel code. By construction, the positions of the missing Nd_v/ϵ_v symbols are known. Therefore, if they are transmitted uncoded in a second step, we expect that a reduced number of only $N(d_v/\epsilon_v) \cdot \epsilon_v = Nd_v$ symbols will be missing afterwards.

In what follows, we will extend this approach to the case of n receivers. For $i \in [n]$, instead of guaranteeing user i 's recovery of all but the last Nd_i source symbols as in the layered approach, we will instead guarantee his recovery of all but the last Nd_i/ϵ_i symbols. Each user can then recover what he additionally requires by listening to uncoded transmissions or the systematic portions of the channel codes used. For the layered scheme, we saw that if user i recovered all but the last Nd_i symbols, he required dedicated coding if $d_i < d_j$ for all $j > i$. Since we guarantee the recovery of all but the last Nd_i/ϵ_i symbols in our *new* coding, we will analogously see in Section III-B, when defining *active* users, that a user requires dedicated coding in our proposed code if $d_i/\epsilon_i < d_j/\epsilon_j$ for all $j > i$.

B. Scheme Description

In this section, we formally discuss the class of segmentation-based schemes and formulate the problem of selecting optimal segment sizes. We then present an analytical solution and discuss connections with the scheme presented in the previous subsection.

The source sequence S^N is divided into $K + 1$ non-overlapping subsequences, $\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_K$, where for $k = 0, 1, \dots, K$, \mathbf{S}_k carries a_k fraction of source bits and $\sum_{k=0}^K a_k \leq 1$. For each k , the segmentation encoder maps subsequence \mathbf{S}_k into channel input \mathbf{X}_k by using a rate- r_k *systematic* erasure code. We take $r_0 = 1$ so that $\mathbf{X}_0 = \mathbf{S}_0$, i.e., \mathbf{S}_0 is sent uncoded. The broadcast channel input sequence X^W is obtained by concatenating the segments $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_K$.

User i observes the channel input through a channel with erasure probability ϵ_i and can therefore completely recover all source segments that are coded at rates $r_k \leq 1 - \epsilon_i$, as well as a fraction $(1 - \epsilon_i)$ of the systematic (uncoded) part of all source segments that are coded at rates $r_k > 1 - \epsilon_i$. This is formally stated in the following claim, which directly follows from Definition 3 and by construction of the scheme.

Claim 1. *The above segmentation-based coding scheme has latency*

$$a_0 + \frac{a_1}{r_1} + \dots + \frac{a_K}{r_K}, \quad (1)$$

and the fraction of source symbols recovered at user i is

$$\left\{ (1 - \epsilon_i) \sum_{\substack{0 \leq j \leq K \\ r_j > 1 - \epsilon_i}} a_j + \sum_{\substack{0 \leq k \leq K \\ r_k \leq 1 - \epsilon_i}} a_k \right\}. \quad (2)$$

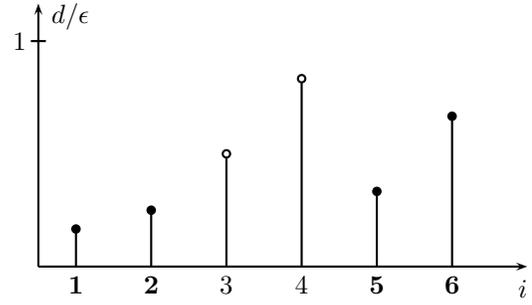


Fig. 2: Distortion ratios plotted by user for $n = 6$ users, where user indices increase with user erasure rates. A user i is active if $d_i/\epsilon_i < d_j/\epsilon_j$ for all $j > i$. Active users are shown in bold.

Note that in our formulation so far, the segment sizes a_i , the associated code-rates r_i , as well as the number of segments K need to be specified. Our optimization problem involves selecting these parameters such that the latency in (1) is minimized, and for each user i , the received fraction of symbols in (2) is at least equal to $1 - d_i$. We first show that the choice of optimal rates r_i admits a natural solution that significantly simplifies our optimization problem.

Claim 2. *The latency of a segmentation-based scheme can be reduced with no penalty in achievable distortion by modifying its segment lengths, a_0, a_1, \dots, a_K , and code rates, r_1, \dots, r_K , s.t. the rates belong to the set $\mathcal{R} = \{1\} \cup \{1 - \epsilon_i, i \in [n]\}$.*

Proof: The proof is given in Appendix -A. ■

With Claims 1 and 2 in hand, we can formulate an optimization problem to minimize the system latency over the segment lengths $\mathbf{a} = (a_0, a_1, \dots, a_n)$, given the distortion constraints as follows.

$$\begin{aligned} \min_{\mathbf{a}} \quad & a_0 + \frac{a_1}{1 - \epsilon_1} + \dots + \frac{a_n}{1 - \epsilon_n} \\ \text{subject to} \quad & a_0 + a_1 + \dots + a_n \leq 1, \\ & (1 - \epsilon_i) \sum_{j=0}^{i-1} a_j + \sum_{j=i}^n a_j \geq 1 - d_i, \quad \text{for } i \in [n] \\ & a_j \geq 0, \quad \text{for } j = 0, 1, \dots, n. \end{aligned} \quad (3)$$

We provide an explicit solution to (3) below. We first define the set of *active* users as those whose distortion-erasure ratio is smaller than that of each user with a higher erasure rate (see Fig. 2 for an illustration).

$$\mathcal{J} = \{j_1, j_2, \dots, j_l\} = \{j \in [n] : \frac{d_i}{\epsilon_i} > \frac{d_j}{\epsilon_j}, \forall i > j\}. \quad (4)$$

Note that from the above definition, it immediately follows that if $\mathcal{J} = \{j_1, j_2, \dots, j_l\}$ and $j_1 < j_2 < \dots < j_l$, then $d_{j_1}/\epsilon_{j_1} < d_{j_2}/\epsilon_{j_2} < \dots < d_{j_l}/\epsilon_{j_l}$.

Theorem 3. *Let $\epsilon_1 < \epsilon_2 < \dots < \epsilon_n$, (d_1, d_2, \dots, d_n) be a distortion vector, and \mathcal{J} be as defined above. Then the optimal solution to (3) gives a latency of*

$$\frac{d_{j_1}}{\epsilon_{j_1}} + \sum_{m=1}^{l-1} \frac{1}{1 - \epsilon_{j_m}} \left(\frac{d_{j_{m+1}}}{\epsilon_{j_{m+1}}} - \frac{d_{j_m}}{\epsilon_{j_m}} \right) + \frac{1}{1 - \epsilon_{j_l}} \left(1 - \frac{d_{j_l}}{\epsilon_{j_l}} \right),$$

which is (d_1, d_2, \dots, d_n) -achievable by a segmentation-based coding scheme with $|\mathcal{J}| + 1 = l + 1$ segments of normalized segment lengths

$$\begin{aligned} a_0 &= \frac{d_{j_1}}{\epsilon_{j_1}}, & a_{j_i} &= 1 - \frac{d_{j_i}}{\epsilon_{j_i}}, \\ a_{j_m} &= \frac{d_{j_{m+1}}}{\epsilon_{j_{m+1}}} - \frac{d_{j_m}}{\epsilon_{j_m}} & \text{for } 1 \leq m < l, \end{aligned} \quad (5)$$

and corresponding code rates

$$r_0 = 1 \quad \text{and} \quad r_{j_m} = 1 - \epsilon_{j_m} \quad \text{for } 1 \leq m \leq l.$$

Proof: The proof is given in Appendix -B. ■

It is interesting to ask if and how the scheme has to be redesigned if another user t joins the system. Clearly, the scheme will be affected only if the user's parameters place him in \mathcal{J} . If the channel erasure rate the user experiences is between the channel erasure rates experienced by two adjacent users in \mathcal{J} , the user is placed in \mathcal{J} if and only if its distortion-erasure value is also between the distortion-erasure values of those two adjacent users. Suppose t falls between users ℓ and m that are adjacent in \mathcal{J} , with $\epsilon_\ell < \epsilon_m$. Without user t , the segment corresponding to the fraction $d_m/\epsilon_m - d_\ell/\epsilon_\ell$ is protected by a channel code of rate $1 - \epsilon_\ell$. This segment will now be split into one of size $d_m/\epsilon_m - d_t/\epsilon_t$, protected by a channel code of rate $1 - \epsilon_t$, and the other of size $d_t/\epsilon_t - d_\ell/\epsilon_\ell$, protected by a channel code of rate $1 - \epsilon_\ell$. Departure of user t reverses the process. Note that the scheme scales easily with the number of users.

C. Special Cases

We now consider several interesting erasure rates and distortion vector setups and interpret the segmentation-based coding scheme in these special cases.

1) *Uniform Channel Condition:* When all users are subject to the same channel erasure rate ϵ_1 , we effectively have $n = 1$. As in Section III-A, we simply set $a_0 = \frac{d_1}{\epsilon_1}$ and $a_1 = 1 - \frac{d_1}{\epsilon_1}$, where d_1 is the minimum distortion of all users. The latency achieved equals $\frac{1-d_1}{1-\epsilon_1}$, which is easily seen as optimal.

2) *Uniform Distortion:* When all users have the same distortion constraint, d , but experience different channel erasure rates, we have that $\mathcal{J} = \{n\}$ so that we encode for the weakest user by setting $a_0 = \frac{d}{\epsilon_n}$ and $a_n = 1 - \frac{d}{\epsilon_n}$. Expectedly, all stronger users achieve the same distortion, and the latency achieved is $\frac{1-d}{1-\epsilon_n}$, which is optimal.

3) *Constant $\frac{d_i}{\epsilon_i}$:* If $\frac{d_i}{\epsilon_i} = c < 1$ for each user $i \in [n]$, we again have that $\mathcal{J} = \{n\}$. Thus, $a_0 = c$, $a_n = 1 - c$, and we achieve a latency of $w = \frac{1-c\epsilon_n}{1-\epsilon_n} = \frac{1-d_n}{1-\epsilon_n}$, which is again optimal.

4) $d_i = \epsilon_i^2$: When user distortions are quadratic in their erasure rates, we have $\frac{d_i}{\epsilon_i} = \epsilon_i$, and hence $\mathcal{J} = [n]$. Thus, $a_0 = \epsilon_1$, $a_i = \epsilon_{i+1} - \epsilon_i$ for $i \in [n-1]$, and $a_n = 1 - \epsilon_n$. We refer to this as the ‘‘proportional allocation scheme.’’ The amount of bits allocated to the segment protected with an erasure code of rate $(1 - \epsilon_i)$ is the difference in the channel capacity between user i and the next weakest user, user $i+1$. The latency achieved in this case is $w = 1 + \epsilon_1 + \sum_{i=1}^{n-1} \frac{\epsilon_{i+1} - \epsilon_i}{1 - \epsilon_i}$.

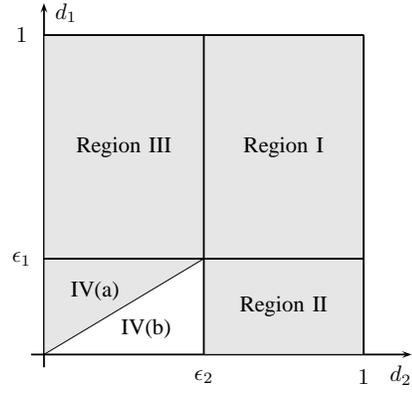


Fig. 3: For $n = 2$ users, we show the demarcation of regions requiring distinct coding in the (d_1, d_2) -plane. A region is shaded if its corresponding code is optimal.

5) *Two Users:* When there are only two users in the system, we can partition the (d_1, d_2) -plane into distinct regions that each have a separate encoding scheme (see Fig. 3). Region I is where $d_i \geq \epsilon_i$ for both $i = 1, 2$. Clearly, an uncoded transmission strategy is optimal in this case, and so we shade this region in Fig. 3 to indicate that we have matching inner and outer bounds. Similarly, in Region II where $d_2 \geq \epsilon_2$, but $d_1 < \epsilon_1$, we can also be optimal, albeit this time with a segmentation-based code. The segmentation is done as if user 1 was the only user in the network, and the systematic portion of the code is sufficient for user 2 as each source symbol is eventually sent uncoded over the channel (see Remark 1 and Lemma 4). An analogous argument can be made for Region III, where we would code as if user 2 was the only user in the network. Next, Region IV(a) illustrates the final region where we obtain optimality, which happens when $d_2/\epsilon_2 \leq d_1/\epsilon_1 \leq 1$. In this case, only user 2 is active (see (4)), and the coded/uncoded transmissions for user 2 is also sufficient for user 1 (see Section III-A). Region IV(b) is the final region, and shows where both users are active, i.e., $d_1/\epsilon_1 < d_2/\epsilon_2$.

D. Numerical Comparisons

We compare the latency achievable by our segmentation scheme of Theorem 3 against some baseline coding schemes. The comparison is done in a way that parallels the discussion in Section III-A. We first consider a single user and successively add additional users to see how the overall latency changes as a function of the number of users in the network. The users are added so that we begin with the one having the highest erasure rate and successively add users with the next highest erasure rate.

The first coding scheme we compare Theorem 3 to is a separation-based approach which, for example, may be implemented with a random linear network code (RLNC) [16]. Here we satisfy all user demands by sending a common message that is intended for everyone to decode. The common message is a compressed version of the source at a distortion equal to the minimum of all user distortion constraints. It is channel coded at a rate that the weakest user can decode. This

scheme achieves an overall latency of $w_{\text{RLNC}} = \frac{1 - \min_{i \in [n]} d_i}{1 - \epsilon_n}$. The reader may verify that the RLNC scheme is also optimal in cases (1) and (2) in Section III-C, but will lead to higher latencies in the remaining cases. The other coding scheme we consider is a simplified version of the optimization problem in (3) where all the non-zero segment sizes are forced to be identical. In particular, each segment a_i , for $i \in \{0, 1, \dots, n\}$, can be either zero or take a fixed value. We note that the RLNC scheme is a special case of this scheme, when only a_n is non-zero.

The numerical comparisons are shown in Fig. 4 where we have taken $n = 5$. Let $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_5)$ and $\mathbf{d}_1 = (d_1, d_2, \dots, d_5)$. In the first plot of Fig. 4, we set ϵ and \mathbf{d}_1 so that for $i \in \{1, 2, \dots, 5\}$, $\epsilon_i = 0.1 \times i$ and $d_i = \epsilon_i^2$. For the sake of clarity, $\epsilon = (0.1, 0.2, 0.3, 0.4, 0.5)$ and $\mathbf{d}_1 = (0.01, 0.04, 0.09, 0.16, 0.25)$. In this case, it can be seen that the addition of each user expands the set \mathcal{J} and thus leads to an increase in the latency. In the second plot of Fig. 4, we have slightly modified only the third component of the distortion vector so that now the third user requires a higher distortion of 0.13 instead of 0.09. For clarity, we now have that $\mathbf{d}_2 = (0.01, 0.04, 0.13, 0.16, 0.25)$. In this case, we see that for our proposed scheme, when the third user is added to the network, his distortion is sufficiently high so that he may simply meet his distortion constraint by virtue of his better channel quality, and from what is already sent over the channel (cf. Section III-A). The latency does not increase in this step. In all cases, we see that our proposed coding scheme performs much better than the other baseline schemes.

Finally, we further highlight the potential benefits of Theorem 3 by plotting a larger example with 80 users. In Fig. 5, we take $\epsilon_i = c(i + 1)$ and $d_i = ci$ for $c = 0.01$ and $i = 1, 2, \dots, 80$. Note that Fig. 5 again adds users in order of decreasing erasure rates so that in fact, user 80 is added first. For this example, we see that all users require dedicated coding for both Theorem 3 and the layered scheme. As described in Section III-A, the layered scheme channel codes a fraction of $d_{i+1} - d_i = c$ source symbols for user i , which is constant among all users. In contrast, Theorem 3 channel codes a fraction $d_{i+1}/\epsilon_{i+1} - d_i/\epsilon_i = 1/(i+1)(i+2)$ for user i . Thus, we see that the longest segments are sent to the *better* users for Theorem 3. In addition, what must be coded for user i decreases quickly with i for Theorem 3 but stays constant for the layered scheme, and Fig. 5 reflects this advantage.

IV. A COMPARISON TO RATELESS CODES

In this section, we compare our segmentation-based scheme with rateless codes optimized for unequal user demands. As discussed earlier, rateless codes provide near-optimal, low-complexity performance when the users are interested in identical content.

A rateless code maps N binary source symbols $\{u_1, \dots, u_N\}$ into a potentially infinite sequence of binary code symbols $\{v_l\}_1^\infty$, where v_l are linear combinations of $\{u_1, \dots, u_N\}$, i.e., $v_l = \theta_1^l u_1 + \dots + \theta_N^l u_N$, $\theta_j^l \in \{0, 1\}$. The coefficients θ_j^l are generated in the following way: (1) we select a degree distribution $\{p_1, \dots, p_N\}$ for the code,

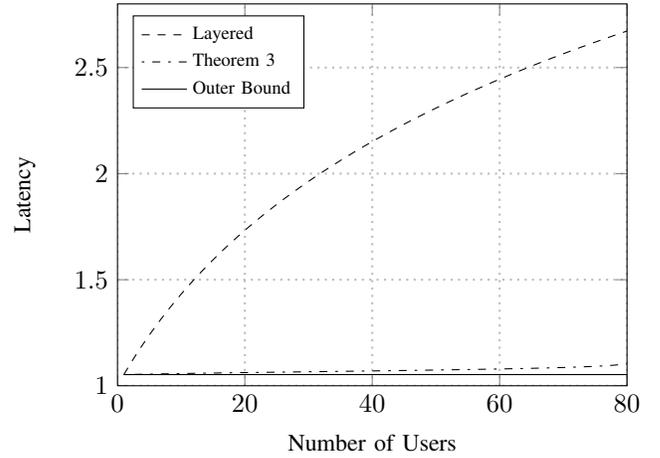


Fig. 5: The source latency plotted as more users are added to the system. The users are added in order of decreasing erasure rates. We take $\epsilon_i = c(i + 1)$ and $d_i = ci$ for $c = 0.01$ and $i = 1, 2, \dots, 80$.

and for each v_l we sample the associated degree M from this distribution; (2) we randomly and uniformly select M elements from the set $\{\theta_1^l, \dots, \theta_N^l\}$ to be non-zero and let the remaining entries be zero. In the classical rateless code design [17], [18], the degree distribution is selected such that the overhead when recovering all N source symbols is kept as small as possible. However in our present setup, where each receiver requires different demands, such a degree distribution will not be suitable. Building upon the approach taken in [19], [20], we briefly discuss how a suitable degree distribution can be obtained for our setup and then compare the performance with our segmentation-based scheme.

We note in advance that codes designed this way do not include the segmentation-based scheme of Theorem 3 as a special case. This is because in the rateless code construction, the choice of non-zero source symbol coefficients is done uniformly over the *entire* source sequence. In contrast, each parity bit in the segmentation-based scheme is generated from source bits restricted to a certain segment.

A. Rateless Coding Approach

In this subsection, we describe the main difference in our present approach compared to [19], [20], which is the way we handle degree-1 symbols. In previous works, the degree-1 symbols were sampled uniformly at random. This resulted in many repetitions, where the same source symbol was transmitted multiple times and others were not transmitted. Thus, our current work proposes an alternative that chooses these symbols deterministically in a round-robin fashion. Note that if a source symbol is sent uncoded T times over a channel with erasure rate ϵ , the probability that it is recoverable after these transmissions is $(1 - \epsilon^T)$. In a *round-robin* scheme, we have that after Nz transmissions of source symbols, a fraction of $(z - \lfloor z \rfloor)$ source symbols were transmitted $(\lfloor z \rfloor + 1)$ times, while the remaining $(1 - (z - \lfloor z \rfloor))$ fraction was transmitted only $\lfloor z \rfloor$ times. The average fraction of recovered symbols is therefore given by $\phi(z, \epsilon)$, where

$$\phi(z, \epsilon) = (1 - (z - \lfloor z \rfloor))(1 - \epsilon^{\lfloor z \rfloor}) + (z - \lfloor z \rfloor)(1 - \epsilon^{\lfloor z \rfloor + 1}). \quad (6)$$

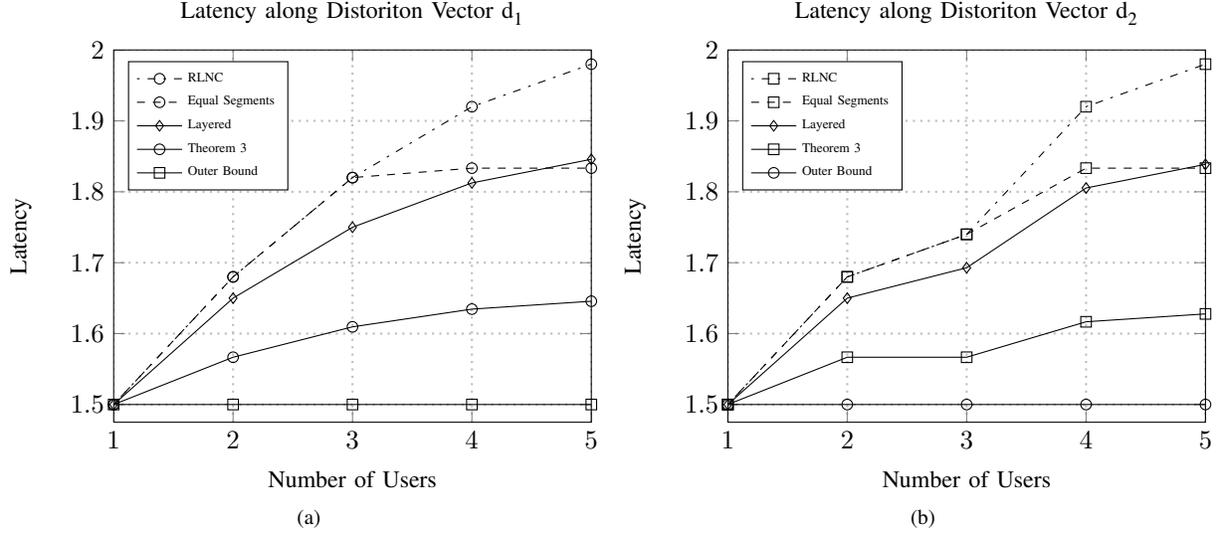


Fig. 4: The source latency plotted as more users are added to the system. The users are added in order of *decreasing erasure rates* for two different distortion constraints. We set $\epsilon = (0.1, 0.2, 0.3, 0.4, 0.5)$ and take $\mathbf{d}_1 = (0.01, 0.04, 0.09, 0.16, 0.25)$ in (a) while $\mathbf{d}_2 = (0.01, 0.04, 0.13, 0.16, 0.25)$ in (b).

Following [20], we can express the optimal degree distribution that minimizes the maximum latency as follows.

$$\begin{aligned} \min_{w, p_1, \dots, p_N} \quad & w \\ \text{subject to} \quad & \log(1-x) - \log(1-\phi(wp_1, \epsilon_i)) \\ & + (1-\epsilon_i)w \sum_{j>1} j p_j x^{j-1} > 0, \\ & \forall x \in (0, 1-d_i), \quad i = 1, 2, \end{aligned}$$

where the probabilities satisfy $\sum_j p_j = 1$, and $p_j \geq 0$, and recall that d_i , and ϵ_i denote the distortion and erasure probabilities for the two users. To interpret the above expression, note that the left-hand-side, when multiplied by $1-x$, is proportional to the size of the *ripple* [21] induced in the belief propagation decoding process when a fraction of x source symbols have been recovered. Hence, the constraint ensures that the ripple remains non-empty until a fraction of $1-d_i$ source symbols have been recovered, which in turn ensures a distortion smaller than d_i . Using the approach in [20], we can numerically compute the optimal degree distribution by using a linear programming approach. We omit the details due to space constraints.

B. Numerical Results

Fig. 6 plots the latency vs. d_2 with the rest of the parameters, i.e., d_1 , ϵ_1 , and ϵ_2 , fixed. We plot the outer bound $w_M = \max\{\frac{1-d_1}{1-\epsilon_1}, \frac{1-d_2}{1-\epsilon_2}\}$ together with the latency achieved by the segmentation-based scheme in Theorem 3 and the optimal latency achievable by a code designed through (7). We refer to this plot as LT-based scheme due to the similarities with LT codes [17]. Alongside these curves, we plot the convex hull of the latencies achieved with the LT-based scheme and denote this as the “timesharing” curve in Fig. 6.

We observe that there are two regions where Theorem 3 meets the outer bound. The first is where $d_2 \geq \epsilon_2 = 0.4$, and

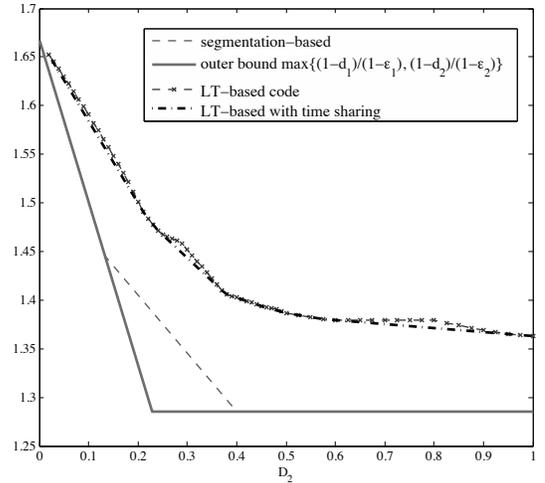


Fig. 6: Latency versus d_2 : $d_1 = 1/10$, $\epsilon_1 = 0.3$, $\epsilon_2 = 0.4$.

the other is where $d_2 \leq d_1 \epsilon_2 / \epsilon_1 \approx 0.13$ (see Section III-C5 for a more detailed discussion of these regions). Note that there is a considerable gap between the degree-optimized rateless codes and the segmentation-based scheme. The LT-based scheme forces the code to have a single degree distribution from which each coded bit is sampled. The segmentation-based scheme applies a different code to each of the segments and hence provides greater flexibility to simultaneously satisfy each user’s demand. Note that in Fig. 6, the LT-based scheme is optimal as $d_2 \rightarrow 0$, but the gap increases as the distortion increases. We also observe in numerical experiments that for small d_2 (up to around 0.2 in Fig. 6), the optimal latency of (7) is achieved when the degree distribution is designed for user 2 only, oblivious of user 1. This, to some extent, echoes the segmentation-based scheme when $d_2 \leq d_1 \epsilon_2 / \epsilon_1 \approx 0.13$, as discussed above.

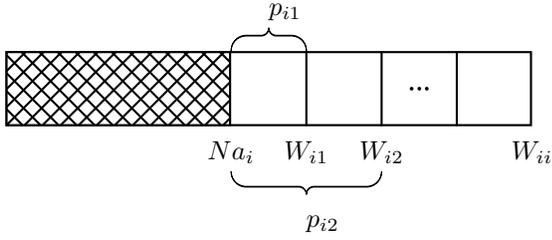


Fig. 7: The systematic (cross-hatched) and parity components of the length- W_{ii} sequence \mathbf{X}_i , which is the channel encoder output of the length- Na_i source segment \mathbf{S}_i .

V. INDIVIDUAL DECODING DELAYS

In this section, we consider possible orderings for the transmission of the (minmax) latency-optimal segments given in Theorem 3. In doing so, we will observe the subsequent effect this has on individual decoding delays. For clarity of exposition, we do this by revisiting the numerical example given in Sections III-C4, and III-D and comparing two possible segment orderings. We mention however, that the procedure we follow for our derivation is not dependent on this example and is easily generalizable. Now, recall that for this example, we have that $\mathcal{J} = [n]$. In turn, this implies that each user's distortion constraint in (3) is tight since in general, any user in \mathcal{J} will have their distortion constraint met with equality. This fact can be verified by combining (3) and (5). The consequence of this is that each user will need to receive a portion of every segment, a fact which will be taken into account when considering possible segment orderings.

Before we begin discussing some possible orderings however, let us first consider the process involved in transmitting a length- Na_i source segment \mathbf{S}_i for $i \in [n]$. Given that this segment is channel coded with a rate- $(1-\epsilon_i)$ code to obtain the channel input \mathbf{X}_i , we see that W_{ii} channel uses are required to transmit the segment where

$$Na_i = W_{ii}(1 - \epsilon_i). \quad (7)$$

Since the channel code is systematic, the W_{ii} channel uses consists of a length- Na_i portion of the original source symbols in \mathbf{S}_i , followed by a length- p_{ii} portion of parity symbols where

$$p_{ii} = W_{ii} - Na_i = \frac{N\epsilon_i}{1 - \epsilon_i}a_i. \quad (8)$$

We denote the length- p_{ii} portion of parity symbols in \mathbf{X}_i as \mathbf{P}_i . This partitioning into systematic and parity components is depicted in Fig. 7.

Notice however, that user i is the only user who must listen for the entire W_{ii} channel uses. For $j \in \{i+1, i+2, \dots, n\}$, user j in fact cannot decode for the entire segment \mathbf{S}_i and instead relies only on what he can obtain from the systematic portion. He can therefore stop listening after Na_i channel uses. On the other hand, for $k \in \{1, 2, \dots, i-1\}$, since $\epsilon_k < \epsilon_i$, user k can decode segment \mathbf{S}_i by listening to only $W_{ik} < W_{ii}$ channel uses where

$$W_{ik}(1 - \epsilon_k) = W_{ii}(1 - \epsilon_i). \quad (9)$$

The earlier decoding times W_{i1} and W_{i2} for users 1 and 2 are

also shown in Fig. 7.

In light of these facts, we will treat the systematic portion of each channel coded segment as a common requirement for all users. In the next two subsections, we will therefore consider orderings that begin with uncoded transmissions. That is, we will first send the length- Na_0 segment \mathbf{S}_0 uncoded and subsequently isolate and transmit the systematic component of \mathbf{X}_i for $i \in [n]$. This requires a total of $N(a_0 + a_1 + \dots + a_n) = N$ transmissions where we have used the fact that the source segments partition the entire source sequence (see (5) and Lemma 4).

The entire source sequence is therefore sent over the first N channel uses, and the only remaining task is to determine the subsequent ordering of the n parity components \mathbf{P}_i for $i \in [n]$. This option of ordering parity components provides much flexibility to a content provider. For example, he can make any user $k \in [n]$ able to decode at a latency that is point-to-point optimal. We again note that for $i \in \{k+1, k+2, \dots, n\}$, user k does not have to receive the entire p_{ii} parity symbols of \mathbf{X}_i . He can instead listen to only p_{ik} symbols where

$$p_{ik} = W_{ik} - Na_i = \frac{N\epsilon_k}{1 - \epsilon_k}a_i, \quad (10)$$

and W_{ik} is given by (9) (see Fig. 7). Since the systematic portion of segment \mathbf{S}_j , $j \in \{1, 2, \dots, k-1\}$, has already been sent within the first N transmissions, user k has therefore decoded as much as he can for this segment and therefore does not have to listen to the parities for it. Hence, if the content provider follows the uncoded transmissions by successively transmitting the first p_{ik} parity symbols of \mathbf{P}_i for $i \in \{k, k+1, \dots, n\}$, it is not hard to see that user k can meet his optimal latency.

Given such latitude in our problem, an exhaustive approach to considering possible segment orderings is therefore out of the scope of this article. In the following two subsections, we will instead consider two simple orderings. They will consist of transmitting the \mathbf{P}_i in either increasing or decreasing order of i . A numerical comparison of these two approaches will be given in Section V-C.

A. Parity Segments Sent in Decreasing Order

In this subsection, we consider the case when \mathbf{P}_i , the parity for segment \mathbf{S}_i , is sent in decreasing order of i . That is, we first transmit \mathbf{P}_n followed by \mathbf{P}_{n-1} to \mathbf{P}_1 (see Fig. 8a). We will calculate the *excess latency* each user experiences with this ordering. The excess latency is defined relative to the point-to-point optimal latency, w_k^* , which is given for user k by

$$w_k^* = \frac{1 - d_k}{1 - \epsilon_k}. \quad (11)$$

Given that user k achieves a latency of w_k , we then define his excess latency δ_k to be

$$\delta_k = w_k - w_k^*. \quad (12)$$

To calculate δ_k , we first remind the reader that for the example we are considering, user k requires parities from \mathbf{P}_i for $i \in \{k, k+1, \dots, n\}$, but does not require any parities from \mathbf{P}_j , $j \in \{1, 2, \dots, k-1\}$, since they are intended for

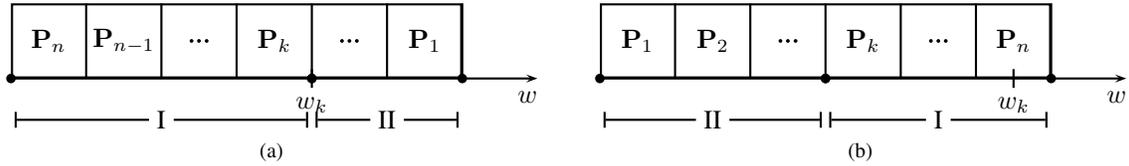


Fig. 8: The ordering of \mathbf{P}_i for (a) decreasing and (b) increasing i . In both cases, user k decodes the parities in Region I and ignores those in Region II. The latency of user k is w_k .

users with better channel qualities. He can therefore meet his distortion constraint after \mathbf{P}_k is sent (see Fig. 8a). We recall from the previous section that user k needs to listen to only p_{ik} of the p_{ii} symbols in \mathbf{P}_i . By combining (8) and (10), we see that the excess latency incurred by listening to the full p_{ii} parity symbols is therefore cumulatively given by

$$\delta_k = \frac{1}{N} \sum_{i=k+1}^n (p_{ii} - p_{ik}) \quad (13)$$

$$= \sum_{i=k+1}^n \left(\frac{\epsilon_i}{1 - \epsilon_i} - \frac{\epsilon_k}{1 - \epsilon_k} \right) a_i. \quad (14)$$

Hence, the latency tuple $(w_1^* + \delta_1, w_2^* + \delta_2, \dots, w_n^* + \delta_n)$ is (d_1, d_2, \dots, d_n) -achievable, where the a_i 's that appear in (14) are given by Theorem 3 for $i \in \{0, 1, \dots, n\}$

B. Parity Segments Sent in Increasing Order

In this subsection, we consider the case when \mathbf{P}_i , the parity for segment \mathbf{S}_i , is sent in increasing order of i . That is, we first transmit \mathbf{P}_1 followed by \mathbf{P}_2 to \mathbf{P}_n (see Fig. 8b). We will again calculate the excess latency user k experiences with this ordering, which we will denote this time by Δ_k .

In calculating Δ_k , we again observe that the first $k-1$ parities, $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{k-1}$, are useless to user k since they are intended for users with better channel qualities. For $j \in \{1, 2, \dots, k-1\}$, the excess latency for each of these segments is thus p_{jj} .

In contrast, user k *does* require parities from \mathbf{P}_i for $i \in \{k, k+1, \dots, n\}$. For these parities, we can again derive the excess latency incurred as being $(p_{ii} - p_{ik})$. Notice however, that user k is not forced to listen to the full amount of parity symbols for \mathbf{P}_n . Since this is the last parity segment sent, he can actually decode after listening to p_{nk} of these symbols, and so there is no excess latency incurred from \mathbf{P}_n (see Fig. 8b). The cumulative excess latency is therefore given by

$$\Delta_k = \frac{1}{N} \left(\sum_{i=1}^{k-1} p_{ii} + \sum_{i=k+1}^{n-1} (p_{ii} - p_{ik}) \right) \quad (15)$$

$$= \sum_{i=1}^{k-1} \frac{\epsilon_i}{1 - \epsilon_i} a_i + \sum_{i=k+1}^{n-1} \left(\frac{\epsilon_i}{1 - \epsilon_i} - \frac{\epsilon_k}{1 - \epsilon_k} \right) a_i. \quad (16)$$

Again, the latency tuple $(w_1^* + \Delta_1, w_2^* + \Delta_2, \dots, w_n^* + \Delta_n)$ is therefore (d_1, d_2, \dots, d_n) -achievable, where the a_i 's that appear in (16) are given by Theorem 3 for $i \in \{0, 1, \dots, n\}$

C. A Numerical Comparison of Orderings

We now compare the individual latencies achieved with the orderings proposed in Sections V-A and V-B. We do

the comparison for the example discussed in Sections III-C4 and III-D where each user i 's distortion is quadratic in his erasure rate, i.e., $d_i = \epsilon_i^2$ for $i \in \{1, 2, \dots, 5\}$.

Let $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_5)$ and $\mathbf{d}_1 = (d_1, d_2, \dots, d_5)$. In the first example of Fig. 9a, we again take $\epsilon = (0.1, 0.2, 0.3, 0.4, 0.5)$ and $\mathbf{d}_1 = (0.01, 0.04, 0.09, 0.16, 0.25)$. In this figure, each user is shown on the horizontal axis and the individual latency he achieves is plotted on the vertical axis. Each user's point-to-point optimal latency, as given by (11), is also shown so that the excess latency can easily be inferred.

From this figure, we see that the sum excess latency is lower when the parities are sent in *increasing* order. At first, this may seem counterintuitive since when the parities are transmitted in increasing order, a user k has no use for parities \mathbf{P}_j for $j \in \{1, 2, \dots, k-1\}$ and essentially postpones the decoding process until the transmission of these parities is completed (see Fig. 8b). On the other hand, when the parities are sent in *decreasing* order, user k has already finished decoding by the time any parities \mathbf{P}_j , $j \in [k-1]$, are sent (see Fig. 8a). The lengths of the parities in Figures 8a, and 8b were drawn only for convenience however, as p_{ii} , the number of parity symbols in \mathbf{P}_i , will generally vary depending on i (see (8)). As discussed in Section V-B, the ability for certain users to avoid receiving the entire p_{55} parities of $\mathbf{P}_5 = \mathbf{P}_n$ is the other important benefit in this example as \mathbf{P}_5 happens to be the longest of all parity segments.

In contrast, Figure 9b plots when the distortions and erasure rates have been chosen such that the lengths of all coded segments are equal. Specifically, we set $\epsilon = (0.31, 0.32, 0.33, 0.34, 0.35)$, and $\mathbf{d}_3 = (0.155, 0.192, 0.231, 0.272, 0.315)$, so that $(a_1, a_2, a_3, a_4, a_5) = (0.1, 0.1, 0.1, 0.1, 0.1)$. The erasure rates were chosen within a short interval so that users experience similar channel qualities. In turn, the excess latency stronger users incur when listening to parities of weaker users is small. Thus, each term in (13) is small and the excess latency for sending parities in decreasing order is minimal. On the other hand, differing channel qualities does not account for the entire excess latency when sending parities in *increasing* order. There is also the excess latency incurred by beginning transmission with parities that are not decodable for certain users, which is represented by the first summation in (15). We see then that in Figure 9b, the sum excess latency is lower when sending parities in *decreasing* order.

VI. CONCLUSIONS

In this paper, we proposed a successive segmentation-based coding scheme for broadcasting a binary source over a multi-

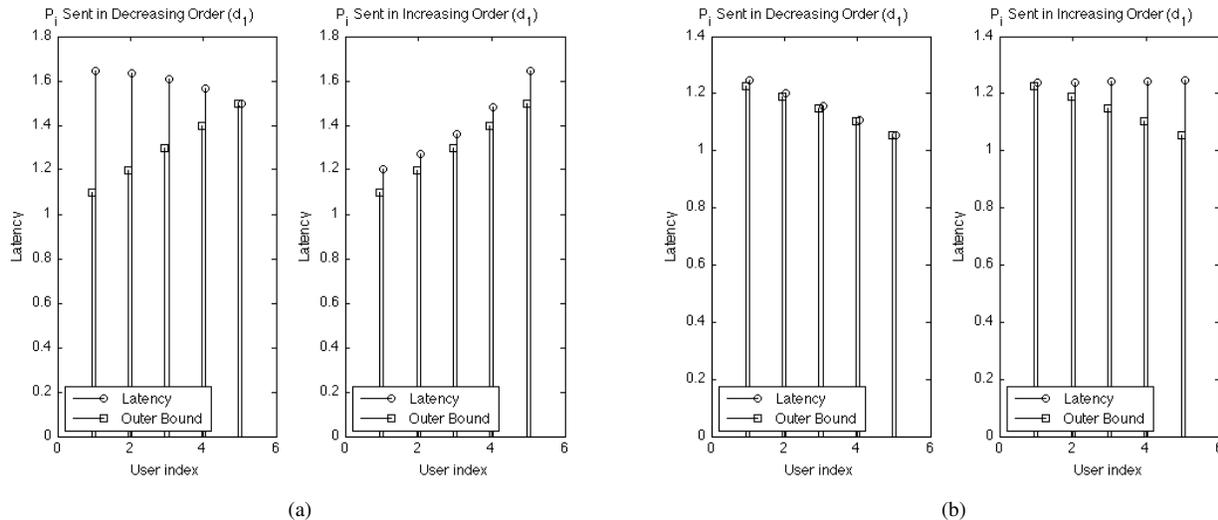


Fig. 9: The individual latency of each user for the segment orderings of Sections V-A and V-B. In (a), we set $\epsilon = (0.1, 0.2, 0.3, 0.4, 0.5)$ and $\mathbf{d} = \mathbf{d}_1 = (0.01, 0.04, 0.09, 0.16, 0.25)$. In (b), we take $\epsilon = (0.31, 0.32, 0.33, 0.34, 0.35)$ and $\mathbf{d}_3 = (0.155, 0.192, 0.231, 0.272, 0.315)$. The point-to-point optimal latency for each user, as given in (11), is also shown.

receiver erasure broadcast channel. Each receiver has individual distortion constraints and experiences distinct channel erasure rates. The proposed scheme partitions the source sequence into multiple segments and applies a systematic erasure code to each segment. We provided optimal choices for segment sizes, and code-rates for each segment, which were based on the users' channel erasure rates, and distortion constraints.

Not only does this proposed scheme outperform Raptor and network coding, it also has two other practical advantages, namely simplicity, and scalability. Firstly, it uses only off-the-shelf systematic erasure codes rather than a joint source-channel code, which would otherwise be required for optimality. Secondly, it can easily be adjusted as users are added or deleted from the system and thus scales to an arbitrary number of users while retaining optimality.

We also discussed the effects of transmission orderings on the decoding latencies of individual users. We provided closed-form expressions for each individual user's excess latency when parity check bits are successively transmitted in both increasing and decreasing order of their segment's coded rate. We then demonstrated user distortion and channel condition setups in which one of the two orderings incurred a smaller individual latency than the other.

For future work, it is our interest to conduct a thorough analysis of individual latencies achieved by users in our segmentation-based scheme. We would also like to analyze the segmentation-based scheme for finite block-lengths and extend the scheme for multiple-descriptions-coded Gaussian sources.

APPENDIX

A. Proof of Claim 2

By way of contradiction, suppose that the optimal rates do not belong to the set $\mathcal{R} = \{1\} \cup \{1 - \epsilon_i, i \in [n]\}$. Then in the optimal solution $(K^*, \mathbf{a}^*, \mathbf{r}^*)$, there exists some $j, l \in [K^*]$, $j \leq l$, and $i' \in \{0\} \cup [n]$, such that $1 - \epsilon_{i'} > r_j^* > r_{j+1}^* >$

$\dots > r_l^* > 1 - \epsilon_{i'+1}$ where we have defined $\epsilon_0 = 0$. Let $j' = \min\{j : 1 - \epsilon_{i'} \geq r_j^*\}$. Then, consider $(K', \mathbf{a}', \mathbf{r}')$ where $K' = K^* - (l - j')$,

$$a'_k = \begin{cases} a_k^*, & k = 0, 1, \dots, j' - 1, \\ \sum_{k=j'}^l a_k^*, & k = j', \\ a_{k+l-j'}^*, & k = j' + 1, \dots, K', \end{cases}$$

and

$$r'_k = \begin{cases} r_k^*, & k = 0, 1, \dots, j' - 1, \\ 1 - \epsilon_{i'}, & k = j', \\ r_{k+l-j'}^*, & k = j' + 1, \dots, K'. \end{cases}$$

It is not hard to verify that $(K', \mathbf{a}', \mathbf{r}')$ satisfies all the distortion constraints, while the latency (1) is strictly reduced. This contradicts the optimality assumption.

B. Proof of Theorem 3

We first reformulate the optimization problem in (3) by introducing a change of variables. If we let $b_i = \sum_{j=0}^i a_j$ for $i = 0, 1, \dots, n$ (and hence $a_0 = b_0$ and $a_i = b_i - b_{i-1}$ for $i = 1, 2, \dots, n$), we can rearrange terms so that (3) becomes

$$\min_{b_0, \dots, b_n} \frac{b_n}{1 - \epsilon_n} - b_0 \left(\frac{1}{1 - \epsilon_1} - 1 \right) - \sum_{i=1}^{n-1} b_i \left(\frac{1}{1 - \epsilon_{i+1}} - \frac{1}{1 - \epsilon_i} \right) \quad (17a)$$

subject to

$$0 \leq b_0 \leq b_1 \leq \dots \leq b_n \leq 1 \quad (17b)$$

$$(1 - \epsilon_{i+1})b_i + (b_n - b_i) \geq 1 - d_{i+1} \quad (17c)$$

Our problem is therefore reduced to finding the optimal solution for Problem (17), and it is not hard to see that this will in turn allow us to construct the optimal solution for Problem (3). We proceed along these lines by first giving a lemma that states that in our search for a segmentation-based code that minimizes latency, we do not sacrifice any

optimality by restricting our search to those codes whose segments partition the entire source sequence, i.e., those with $b_n = 1$.

Lemma 4. *Let $b^* = (b_0^*, b_1^*, \dots, b_n^*)$ be an optimal solution to (17) where $b_n^* < 1$. Then $\beta^* = (b_0^* + \Delta, b_1^* + \Delta, \dots, b_{n-1}^* + \Delta, 1)$ is also an optimal solution where $\Delta = (1 - b_n^*)/\epsilon_n$.*

Proof: It is readily verified that in addition to being feasible, β^* also does not change the objective function in comparison to b^* . The verification requires $d_n < \epsilon_n$ which is assumed in our setup. ■

We now use Lemma 4 in order to show that Theorem 3 gives the optimal segmentation-based scheme.

Theorem 5. *For the optimization problem in (17), there is an optimal solution with $b_n = 1$ and $b_i = \min_{j=i+1}^n \left\{ \frac{d_j}{\epsilon_j} \right\}$ for $i = 0, 1, \dots, n-1$.*

Proof: Using Lemma 4 it is sufficient to consider segmentation-based codes with $b_n = 1$. From the feasibility constraints of (17b) and (17c), which when evaluated with $b_n = 1$, we have

$$b_{i-1} \leq \min \left\{ b_i, \frac{d_i}{\epsilon_i} \right\} \text{ for } i \in [n]. \quad (18)$$

Upon inspection of (17a), we see that in order to minimize the objective function, we would like to *maximize* b_{i-1} for $i \in [n]$. Consider first, b_{n-1} , which is upper-bounded as $b_{n-1} \leq d_n/\epsilon_n$. Continuing, we have that $b_{n-2} \leq \min\{d_{n-1}/\epsilon_{n-1}, d_n/\epsilon_n\}$ and in general

$$b_i \leq \min_{j \in \{i+1, \dots, n\}} \left\{ \frac{d_j}{\epsilon_j} \right\} \text{ for } i = 0, 1, \dots, n-1. \quad (19)$$

We can therefore individually maximize each b_i by choosing equality in (19). This completes the claim. ■

Finally to complete the justification of Theorem 3 we note that the expression for b_i in (19) is just an alternate representation for the variables (a_0, a_1, \dots, a_n) stated in Theorem 3.

REFERENCES

- [1] L. Tan, Y. Li, A. Khisti, and E. Soljanin, "Source broadcasting over erasure channels: Distortion bounds and code design," in *Proc. IEEE Information Theory Workshop (ITW)*, Seville, Spain, Aug. 2013.
- [2] Y. Li, L. Tan, A. Khisti, and E. Soljanin, "Successive segmentation-based coding for broadcasting over erasure channels," in *Proc. IEEE International Symposium on Information Theory*, Honolulu, Hawaii, Jun. 2014.
- [3] M. A. Shokrollahi and M. Luby, "Raptor codes," *Foundations and Trends in Communications and Information Theory*, vol. 6, no. 3-4, 2009.
- [4] V. Goyal, "Multiple description coding: compression meets the network," *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 74-93, Sep 2001.
- [5] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 57-70, Jan 2005.
- [6] F. H. P. Fitzek, B. Can, R. Prasad, and M. Katz, "Overhead and quality measurements for multiple description coding for video services," in *Wireless Personal Multimedia Communications (WPMC)*, 2004, pp. 524-528.
- [7] L. Tan, A. Khisti, and E. Soljanin, "Quadratic Gaussian source broadcast with individual bandwidth mismatches," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Jul. 2012, pp. 204-208.
- [8] R. L. Urbanke and A. D. Wyner, "Packetizing for the erasure broadcast channel with an internet application," in *Proc. Int. Conf. Combinatorics, Information Theory and Statistics*, Portland, ME, 1997, p. 93.
- [9] Y. Huang and K. R. Narayanan, "Distortion bounds for binary erasure broadcast channels," Texas A&M University, College Station, Tech. Rep.
- [10] U. Mittal and N. Phamdo, "Hybrid digital-analog (HDA) joint source-channel codes for broadcasting and robust communications," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1082-1102, May 2002.
- [11] Z. Reznic, M. Feder, and R. Zamir, "Distortion bounds for broadcasting with bandwidth expansion," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3778-3788, Aug. 2006.
- [12] V. M. Prabhakaran, R. Puri, and K. Ramchandran, "Hybrid digital-analog codes for source-channel broadcast of Gaussian sources over Gaussian channels," *IEEE Trans. Inf. Theory*, no. 7, pp. 4573-4588, Jul. 2011.
- [13] K. R. Narayanan, G. Caire, and M. Wilson, "Duality between broadcasting with bandwidth expansion and bandwidth compression," in *Proc. IEEE Int. Symp. Information Theory*, Jun. 2007, pp. 1161-1165.
- [14] C. Tian, S. N. Diggavi, and S. Shamai, "Approximate characterizations for the gaussian source broadcast distortion region," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 124-136, Jan. 2011.
- [15] L. Tan, A. Khisti, and E. Soljanin, "Distortion bounds for broadcasting a binary source over binary erasure channels," in *Proc. 13th Canadian Workshop on Information Theory CWIT*, Toronto, ON, Jun. 2013.
- [16] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3-20, 2008.
- [17] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, Nov. 2002.
- [18] M. A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2551-2567, 2006.
- [19] Y. Li and E. Soljanin, "Rateless codes for single-server streaming to diverse users," in *Proc. 47th Annual Allerton Conference on Communication, Control, and Computing*, Montecello, IL, Sep. 2009.
- [20] Y. Li, E. Soljanin, and P. Spasojević, "Three schemes for wireless coded broadcast to heterogeneous users," *Physical Communication*, vol. 6, no. 0, pp. 114 - 123, 2013, network Coding and its Applications to Wireless Communications. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1874490712000420>
- [21] G. Maatouk and A. Shokrollahi, "Analysis of the second moment of the LT decoder," *IEEE Trans. Inf. Theory*, Jan. 2012.