Streaming Codes for Variable-Size Messages

Michael Rudow and K. V. Rashmi, *Member, IEEE*

***Abstract*— Live communication is ubiquitous, and frequently must contend with reliability issues due to packet loss during transmission. The effect of packet losses can be alleviated by using erasure codes, which aid in recovering lost packets. Streaming codes are a class of codes designed for the live communication setting, which encode a stream of message packets arriving sequentially for transmission over a packet-loss channel. The existing study of streaming codes considers settings where the sizes of the message packets to be transmitted are all fixed. However, message packets occur with unpredictable and variable sizes in many applications, such as videoconferencing. In this paper, we present a generalized model for streaming codes that incorporates message packets of variable sizes. We show that the variability in the sizes of message packets induces a new trade-off between the rate and the decoding delay under lossless transmission. Moreover, the variability in the sizes of message packets impacts the optimal rate of transmission. To address this, we introduce algorithms to compute upper and lower bounds on the optimal rate for any given sequence of sizes of message packets. We then design an explicit streaming code for the proposed model. We empirically evaluate the code construction over a live video trace for several representative parameter settings, and show that the rate of the construction is approximately 90% of an upper bound and 5%–48% higher than naively using the existing streaming codes.**

***Index Terms*— Streaming codes, erasure coding, live streaming applications, video conferencing, packet loss.**

I. Introduction

THE information age has heralded widespread demand for live communication with high quality-of-service (QoS). Such demand is evident in the abundance of popular multimedia live streaming applications, including video conferencing, VoIP, and online gaming. These services form an integral part of Internet use. Communication for these live streaming applications involves encoding a sequence of so-called "message packets" for transmission to a receiver over a lossy channel. Despite packet losses, each message packet must be decoded within a strict playback deadline to avoid degrading the QoS. Furthermore, under several settings, the low-latency requirement prohibits using feedback-based retransmission schemes, necessitating a proactive coding-based approach to provide robustness to packet loss.

Manuscript received 23 July 2021; revised 29 December 2021; accepted 11 April 2022. Date of publication 28 April 2022; date of current version 18 August 2022. This work was supported in part by NSF under Grant CIF 1910813 and in part by the Facebook Communications and Networking Research Award. An earlier version of this paper was presented in part at the 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton) [DOI: 10.1109/ALLERTON.2018.8635971]. *(Corresponding author: Michael Rudow.)*

The authors are with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: mhrudow@gmail.com).

Communicated by R. D. Wesel, Associate Editor for Coding and Decoding.
Digital Object Identifier 10.1109/TIT.2022.3170895

One natural coding-based approach uses traditional codes, such as block codes, to recover lost packets. However, such coding schemes are ill-suited for the low-delay communication setting of live streaming applications. The loss patterns faced in streaming applications are correlated (i.e., bursty). Nevertheless, conventional block codes, such as maximum-distance-separable (MDS) codes, are inefficient for recovering from burst losses within a strict decoding delay for the following reason. Using MDS block codes necessitates recovering all packets lost as a burst *simultaneously*. As a result, all lost packets must be decoded by the playback deadline of the first lost packet—an unnecessarily stringent requirement for most lost packets. The redundancy sent by the deadline of the final lost packet is wasted, penalizing the rate. Finally, it is possible to convert an MDS code into a burst-correcting code via the standard technique of interleaving. However, such an approach is inapplicable to the live streaming setting, as it would violate the low-delay requirement.

Convolutional coding schemes tailored to the live communication setting can outperform traditional code constructions. Martinian and Sundberg demonstrated this fact in [2] by formalizing the "streaming model" of live communication and introducing specialized codes —called "streaming codes"—with significantly higher rates than traditional block codes. Under the streaming model, a sender communicates a sequence of message packets, each with a strict fixed decoding delay constraint, to a receiver. The message packets are communicated via transmission of a sequence of "channel packets" over a packet loss channel. Subsequent works provided capacity-achieving streaming codes first in the setting of burst-only losses [3] and second in a setting introduced by [4] with both bursty packet losses and arbitrary packet losses [5]–[9].

The model employed in prior works on streaming codes considers the sizes of message packets and channel packets as *fixed*. However, many live communication settings involve communication with message packets and channel packets of *variable sizes*. Such a situation often occurs in live video communication; wherein video frames are typically compressed prior to transmission to reduce the communication load. The sizes of the compressed frames often exhibit high variability. For example, we demonstrate the variability of compressed frame sizes for a live video trace transmitted by Facebook Live in Figure 1. Variable-sized input data is incompatible with the previously studied streaming model, motivating the need for a new model.

In this paper, we present a generalized model for streaming codes that accommodates message packets of variable sizes. The variability in the sizes of the message packets induces a

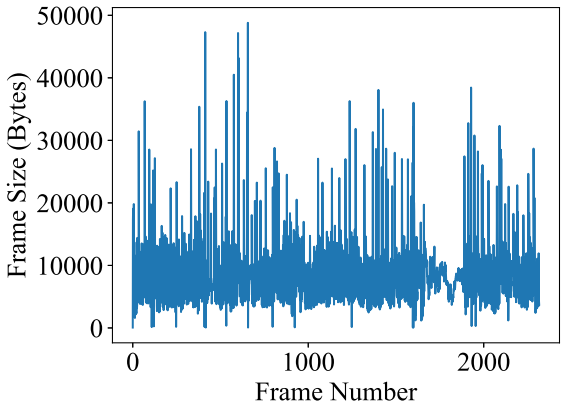

Fig. 1. Frame size variability in a live video trace collected from the Facebook Live application (for a 2000 Kbps live video).

new trade-off between the rate of the code and the minimum decoding delay (i.e., the delay for decoding a message packet when all channel packets are received). Such a trade-off does not exist in the existing model of streaming codes. We capture this trade-off by introducing a new parameter into the model, which we term "lossless-delay."

The variability in the sizes of the message packets impacts the optimal rate. We capture the dependence by explicitly identifying the range of values the optimal rate could take. Specifically, we determine the least upper bound and greatest lower bound on the optimal rate for arbitrary sequences of sizes of message packets. However, the gap between these values is wide in many settings, prompting the need to better characterize the optimal rate for any specific sequence of sizes of message packets. We introduce algorithms to compute tighter upper and lower bounds on the optimal rate for any given sequence of sizes of message packets.

We then present a simple explicit construction of streaming codes for the proposed model with message packets of variable sizes. The construction employs an existing streaming code construction presented in [9] as a building block, along with several techniques to alleviate the adverse effect of variability of sizes of message packets on the rate of the code. We theoretically characterize the rate of the construction when the size of each message packet is drawn independently from any distribution with finite support. Finally, we empirically evaluate the code construction over several representative parameter settings for a live video trace collected from the Facebook live application and show that it performs well. Specifically, the construction exhibits an *average rate of* 90% *of an upper bound and improves the rate by* 5% *to* 48% *over naively using the existing streaming codes*.

II. Background and Related Work

This section provides an overview of the background of streaming codes relevant to this work. First, we will describe the previously studied streaming model in which message packets have the same fixed size. Second, we will detail a sliding-window adversarial channel model which captures the worst-case packet loss patterns which occur in transmissions. The sliding-window adversarial channel model will be used throughout this work. We discuss an upper bound on the rate imposed by such a channel under the previously studied setting where all message packets have the same size. Third, we deconstruct a class of optimal code constructions for the previously studied fixed-size streaming model. We highlight aspects of the construction that we leverage later in this work. Fourth, we discuss alternative formulations of the streaming setting, which can form the basis for potential future studies.

A. Background

The streaming model was first introduced by Martinian and Sundberg in [2]. Under this model, at every time slot, i, the sender receives a message packet, $S[i]$, comprised of k symbols from a finite field $\mathbb{F}_q$ for a natural number k. The sender transmits a channel packet, $X[i]$, consisting of n symbols from $\mathbb{F}_q$ (for a natural number n) over a packet-loss channel to a receiver. Either $X[i]$ is received, or a unique symbol (i.e., $*$) is received, reflecting a packet loss. Packet losses can occur as isolated bursts of some maximal length, b, separated by guardspaces of successful transmissions. Such loss patterns are useful representations of real-world settings where losses occur as occasional bursts, as can be reflected by the Gilbert model [10]. The rate of the code is naturally defined as $\frac{k}{n}$. The encoding is *causal*, meaning that the channel packet $X[i]$ can be any function of $S[0], \ldots, S[i]$ but may not depend on any future message packets. The real-time playback deadline for live communication is incorporated by requiring the receiver to recover each message packet, $S[i]$, within a *worst-case-delay of* τ time slots. In other words, the received channel packets of $X[0], \ldots, X[i+\tau]$ are sufficient to decode $S[i]$.[1] Martinian and Sundberg presented an upper bound on the rate of $\frac{\tau}{\tau+b}$ as well as a rate-optimal code construction for a large class of parameter settings. Later, Martinian and Trott in [3] designed a capacity-achieving code construction for all parameter settings for this streaming model.

In certain real-world settings, burst (correlated) and isolated (uncorrelated) packet losses both occur. These loss patterns are well-approximated by statistical models like the GE channel model [11]. Yet, constructing coding schemes directly for such statistical models is believed to be hard. An analytically tractable sliding-window adversarial channel model approximating the worst-case conditions of models such as the GE model was introduced by Badr et al. in [4]. The channel model is characterized using three parameters a, b, and w and is referred to as $C(a, b, w)$.[2] For every w consecutive channel packets, one burst of no more than b consecutive packets or up to a arbitrary packets may be lost.

A generalized streaming model incorporating a $C(a, b, w)$ sliding-window adversarial channel was introduced by Badr et al. in [4]. The authors designed a near-optimal streaming code construction for this streaming model. Badr et al. also showed that $\frac{\tau-a+1}{\tau+b-a+1}$ and $\frac{w-a}{w+b-a}$ are upper bounds on the rate when the worst-case-delay is $\tau < w$ and $\tau \geq w$ respectively. We will later show that the argument used to prove these bounds extends to the setting where message packets have variable sizes.

[1] In [2], the worst-case-delay parameter was called T rather than τ

[2] In [4], the parameters (a, b, w) were referred to as (N, B, W).

Later, the above upper bound on the rate was attained by streaming code constructions designed in the two independent concurrent works [5] and [6]. An alternative explicit capacity-achieving streaming code for the model was presented by Dudzicz et al. in [7]. These constructions require an exponential field size for certain parameter settings. A capacity-achieving streaming code and an explicit capacity-achieving streaming code with quadratic field size requirements were concurrently designed by Krishnan et al. in [8] and Domanovitz *et al.* in [9]. The design of these streaming codes employs the technique of *diagonal interleaving* to convert the problem of constructing a rate-optimal streaming code into the more tractable challenge of designing a block code of the same rate. To design rate-optimal streaming codes for a worst-case-delay of τ and a $C(a, b, w)$ channel, one can design a block code which decodes each symbol within τ symbols when either a burst of at most b consecutive symbols or up to a arbitrary symbols are lost. The technique of first creating a block code and then applying interleaving has also been employed in several other prior works, including [2]–[4], [12].

Later in this work, we will leverage existing block codes, such as from [5]–[9], as a component of our proposed code construction. Specifically, we shall consider systematic rate-optimal block codes presented in [9], whose field size requirement is quadratic in the delay parameter τ. Any block code designed for the streaming model, including those presented in [5]–[8], could likewise be used by our proposed code construction. We refer to any such code as a Streaming Block Code (SBC). We now highlight a few relevant details for such codes, which we will use later in this work. For any parameter setting, (τ, a, b), we denote any systematic (n, k) SBC where $n = (\tau + b - a + 1)$ and $k = (\tau - a + 1)$ as $\langle s_0, \ldots, s_{\tau-a}, p_0, \ldots, p_{b-1}\rangle$. Specifically, $s_0, \ldots, s_{\tau-a}$ are the $(\tau-a+1)$ systematic symbols and $p_0, \ldots, p_{b-1}$ are the b parity symbols. For these codes, the ith symbol for $i \in \{0, \ldots, n-1\}$ is decoded using the first $\min(i + \tau + 1, n)$ symbols in the presence of a single burst loss of b consecutive symbols or the loss of a arbitrary symbols.

We now illustrate how to use interleaving to convert a block code into a streaming code. An (n, k) systematic block code which maps k systematic code symbols, $(s_0, \ldots, s_{k-1})$, into n code symbols, $(s_0, \ldots, s_{k-1}, p_0, \ldots, p_{n-k-1})$, will be used. In the ith time slot, the sender receives as input the message packet $S[i] = (S_0[i], \ldots, S_{k-1}[i])$ comprising k symbols, and the channel packet $X[i] = (S_0[i], \ldots, S_{k-1}[i], P_0[i-k], \ldots, P_{n-k-1}[i-n+1])$ is sent. The symbol $P_{n-k-1}[i-n+1]$ is the final symbol of a *distinct block code* ("block") consisting of $(S_0[i-n+1], \ldots, S_{k-1}[i-n+k], P_0[i-n+1], \ldots, P_{n-k-1}[i-n+1])$. This block contains a single symbol from each of channel packets $X[i-n+1], \ldots, X[i]$. The channel packets $X[i-n+1], \ldots, X[i-n+k]$ contain $S_0[i-n+1], \ldots, S_{k-1}[i-n+k]$ respectively, and $X[i-n+k+1], \ldots, X[i]$ contain $P_0[i-n+1], \ldots, P_{n-k-1}[i-n+1]$ respectively. Next, we discuss the block which corresponds to message packet $S[i]$. This block comprises (a) symbols of message packets $S[i], \ldots, S[i+k-1]$ sent in channel packets $X[i], \ldots, X[i+k-1]$, and (b) parity symbols, $P_0[i], \ldots, P_{n-k-1}[i]$, sent in channel packets $X[i+k], \ldots, X[i+n-1]$. Specifically, the jth position of the block consists of the jth symbol of the corresponding channel packet for $j \in \{0, \ldots, n-1\}$. Hence, the block comprises

$$\langle S_0[j], S_1[j+1], \ldots, S_{k-1}[j+k-1], P_0[j], \ldots, P_{n-k-1}[j]\rangle.$$

We demonstrate an example of converting a block code into a streaming code with diagonal interleaving for a $(5, 3)$ block code in Figure 2.

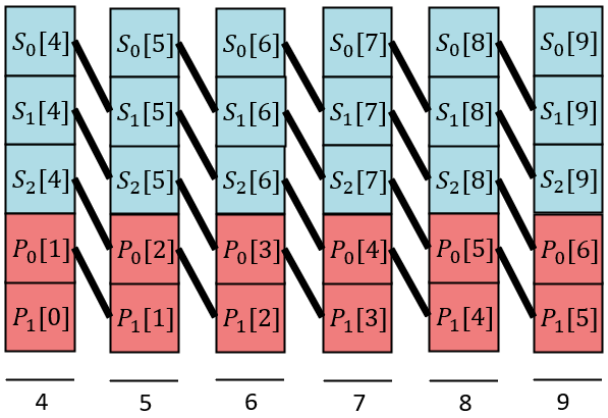

Fig. 2. Interleaving example of a $(5, 3)$ block code. The blue boxes labeled $S_j[i]$ are symbols of message packet $S[i]$, the red boxes labeled $P_j[i]$ are parity symbols, and the black lines connect the boxes which are part of the same block. The numbers under the lines indicate the time slots.

During time slot i, let $i' = (i - k + 1)$. The block code

$$\langle S_0[i'], \ldots, S_{k-1}[i], P_0[i'], \ldots, P_{n-k-1}[i']\rangle$$

is computed. The parity symbols

$$(P_0[i'], \ldots, P_{n-k-1}[i'])$$

are defined before they are sent in channel packets $X[i+1], \ldots, X[i+n-k]$ respectively. Consequently, during time slot i, the each value $P_j[l]$ for $j \in \{0, \ldots, n-k-1\}$ is accessible for $l \leq (i-k+1)$, since it was defined during time slot $(l + k - 1) \leq i$. Finally to handle edge conditions, for any $z < 0$ and $j \in \{0, \ldots, k-1\}$, $S_j[z]$ is defined to be an arbitrary fixed symbol.

B. Other Related Work

Several other variants of the streaming model have been studied in the literature. We briefly discuss them below for the sake of completeness. Most of these models involve message packets having fixed sizes. Under a streaming model with multiplexing, a sender receives two streams of message packets as input with two different decoding delays for transmission over a burst-only channel [13], [14]. Under another model, a sender transmits a stream of message packets to two different receivers over two different burst-only channels subject to two different decoding delays [15], [16]. Another variant of the streaming model includes unequal error protection wherein all symbols from each message packet must be recovered in the event of short bursts, but only certain symbols need to be

recovered in the presence of longer bursts [17]. Another setting considers average rather than worst-case-delay for decoding [18]. Various other streaming models incorporate multiple channel uses between every message packet [4], [19], [20]. Another variation of the streaming model stipulates partial recovery of certain loss patterns wherein only some of the message packets are decoded by their deadlines [21]. The setting of streaming over a three-node relay network is studied in [22], wherein there is a delay in decoding even under lossless transmissions. The notion of two distinct decoding delays has also arisen in the context of VoIP in [23], which introduces codes with a shorter delay to decode a few random packet losses than that of recovering a longer burst of packet losses. A different streaming model formulation considers a channel which can induce multiple burst losses within the worst-case-delay [24]. Diverging from the above models, another streaming model considers (1) high and low priority message packets, each with a (potentially different) fixed size, which occur in a fixed periodic manner, (2) channel packets of a fixed size, and (3) unequal error protection [25]. A formal study of incorporating message packets with arbitrary variable sizes in these models is outside of the scope of this paper and is a potential avenue for future work.

III. A Model for Streaming Codes With Message Packets of Variable Sizes

The streaming model discussed in Section II will now be generalized to incorporate message packets of *variable sizes*. The variability in the sizes of the message packets induces a new trade-off between the optimal rate and the decoding delay under a lossless transmission. A new delay parameter will be introduced to the model to capture this trade-off. A new definition for the rate of a code is included to reflect the varying sizes of the message packets and channel packets.

Under the proposed streaming model, the sender receives a message packet, $S[i] = (S_0[i], \ldots, S_{k_i-1}[i])$, during the ith time slot. The message packet consists of k_i symbols drawn uniformly at random from a finite field, $\mathbb{F}_q$, where k_i is an arbitrary non-negative integer. A channel packet, $X[i] = (X_0[i], \ldots, X_{n_i-1}[i]) \in \mathbb{F}_q^{n_i}$, is transmitted to the receiver, where n_i is an arbitrary non-negative integer. This deviates from the prior models (such as in [2]–[9]) where each $|S[i]| = k$ and $|X[i]| = n$ for some *fixed* positive integers k and n. The channel packet $X[i]$ is a function of the current and previous message packets (i.e., $X[i] = Enc\left(\langle S[j] \mid j \in \{0, \ldots, i)\}\rangle\right)$). Encoding is not a function of the symbols of future message packets (or their sizes), as the sender does not have access to this information.

The channel packet $X[i]$ is transmitted over the $C(a, b, w)$ channel discussed in Section II. The receiver obtains $Y[i] \in \{X[i], *\}$, where $*$ denotes a dropped packet. Under the channel model, for any sliding window of w consecutive packets, up to b consecutive packets may be dropped as a burst, or up to a packets may be dropped in arbitrary locations. In other words, for any time slot i, the packet losses introduced by the $C(a, b, w)$ channel satisfy at least one of the following two conditions for the window $W = \{i, \ldots, i + w - 1\}$:

1) Lose only a burst of length at most b channel packets :
$$\exists j \in W, \forall l \in W \setminus \{j, \ldots, j + b - 1\}, \mathbb{1}(Y[l] = X[l]).$$
2) Lose at most a channel packets :
$$a \geq \sum_{j=i}^{i+w-1} \mathbb{1}(Y[j] = *).$$

Due to the real-time playback deadline, the receiver must decode $S[i]$ within τ time slots. We refer to τ as the "worst-case-delay" parameter and the requirement that $S[i]$ be decoded by time slot $(i + \tau)$ as the "worst-case-delay constraint." More formally, the receiver decodes $S[i]$ as

$$S[i] = Dec\big(\langle S[j] \mid j \in \{i - \tau, \ldots, i - 1\}\rangle, \langle Y[j] \mid j \in \{i, \ldots, i + \tau\}\rangle, \langle k_j \mid j \in \{i, \ldots, i + \tau'\}\rangle\big)$$

where $\tau' \leq \tau$ is the largest value such that $X[i + \tau']$ has been received (i.e., $Y[i + \tau'] = X[i + \tau']$). In other words, the receiver decodes message packet $S[i]$ using the (a) previously-decoded τ message packets, $(S[i - \tau], \ldots, S[i - 1])$, (b) already received channel packets among $(X[i], \ldots, X[i+\tau])$, and (c) sizes of up to $(\tau+1)$ of the message packets $(S[i], \ldots, S[i+\tau])$ which may not have been be decoded. In order to inform the receiver about (c) irrespective of which channel packets are lost, the sender adds the sizes of the current message packet and previous b message packets to a small header of each channel packet.

Under the previously studied model, the rate is $\frac{k}{n}$. But $\frac{k}{n}$ is not well-defined in the proposed model. Accordingly, we introduce a suitable definition of rate for the setting of message packets of variable size. To do so, we limit our attention to finite-length sequences of message packets. For an arbitrary non-negative integer, t, consider an arbitrary sequence of t message packets, $S[0], S[1], \ldots, S[t]$. We refer to the corresponding sequence $k_0, \ldots, k_t$ as the "message packet size sequence." The rate for any code construction is defined as the ratio of the number of symbols of all message packets to the total number of transmitted symbols,

$$R_t = \frac{\sum_{i=0}^{t} k_i}{\sum_{i=0}^{t} n_i}. \tag{1}$$

For convenience of notation, we use the convention that $t \geq 2\tau$ and the sizes of each of the final 2τ message packet is 0 (i.e., $k_{t-2\tau+1} = 0, \ldots, k_t = 0$). This convention can be met by appending 2τ message packets of size 0 to any sequence of message packets to ensure that it meets this convention without altering the rate.

The setting in which message packets have variable sizes differs from where message packets all have the same fixed size in the following critical respect. When the sizes of the message packets and channel packets are fixed, there exists an optimal rate code construction in which each message packet, $S[i]$, is sent as a part of the corresponding channel packet, $X[i]$ ([5]–[8]). In other words, there are rate-optimal coding schemes where each message packet $S[i]$ can be decoded without any delay under lossless transmission. However, this is *no longer true when the sizes of message packets can vary*.

When the message packets have variable sizes, distributing symbols of message packets over multiple channel packets can lead to a higher rate than sending each message packet within its corresponding channel packet. We illustrate this observation with a toy example. Consider the length $(\tau+1)$ sequence of message packets where the first message packet $S[0]$ is of size τ and the next τ message packets have size 0. The $C(a,b,w)$ channel for $(a=1,b=1,w=\tau+1)$ could drop $X[0]$. Therefore, if $S[0]$ were transmitted as part of channel packets $X[0]$, at least τ parity symbols would need to be sent in channel packets $X[1],\ldots,X[\tau]$ to decode $S[0]$ within the worst-case-delay of τ time slots. The rate for such a scheme is at most $\frac{1}{2}$. Alternatively, the symbols of message packet $S[0]$ could be transmitted evenly over $X[0],\ldots,X[\tau-1]$, and a parity of the previous τ channel packets sent in $X[\tau]$ (i.e., $X[\tau]=\sum_{i=0}^{\tau-1}X[i]$). Such a scheme would have a rate of $\frac{\tau}{\tau+1}$ while satisfying the worst-case-delay constraint.

As shown above, under the setting where message packets have variable sizes, distributing the symbols of a message packet over multiple channel packets can lead to a higher rate. However, doing so will delay decoding the message packet when there are no losses. Thus, the variability in the sizes of the message packets *induces a new trade-off between the rate of the code and decoding delay when all channel packets are received*. We incorporate this new trade-off into our model via a new parameter which we call the lossless-delay, τ_L.[3] The receiver must be able to decode every message packet, $S[i]$, using channel packets $X[0],\ldots,X[i+\tau_L]$ if they are all received. In other words,

$$S[i]=Dec^{(L)}\big(\langle X[j],k_j \mid j\in\{0,\ldots,i+\tau_L\}\rangle\big).$$

The newly introduced parameter τ_L represents the tolerable decoding delay under lossless channel conditions, whereas τ reflects the worst-case delay in the presence of packet loss. The two parameters (τ_L,τ) are relevant to settings where the transmission is lossless most of the time, and the rare worst-case channel conditions are captured via the $C(a,b,w)$ channel. In such scenarios, a live streaming application may occasionally tolerate a decoding delay of τ time slots but benefit from the faster decoding of τ_L time slots most of the time.

Due to the worst-case-delay constraint, τ, for transmission over a $C(a,b,w)$ channel, each $S[i]$ must be decoded with $X[0],\ldots,X[i+\tau-b]$ when $X[i+\tau-b+1],\ldots,X[i+\tau]$ are lost. Therefore, under a lossless transmission setting, each $S[i]$ is recoverable from $X[0],\ldots,X[i+\tau-b]$. Consequently, the parameter τ_L is at most $(\tau-b)$, leading to $\tau_L\in\{0,\ldots,\tau-b\}$. Higher values of τ_L enable the symbols of the message packets to be spread over more channel packets, thereby increasing both the rate of the code and decoding delay under lossless transmission.

Finally, the parameters (τ,a,b,w) obey certain restrictions. Because burst losses are a special case of arbitrary losses, $a\le b$. Moreover, if it were the case that $b\ge w$ (or $b>\tau$), for any time slot, i, the channel packets $X[i],\ldots,X[i+\tau]$ could all be lost. As a result, it would be impossible to decode $S[i]$ within a delay of τ, resulting in a capacity of 0. Moreover, if $b=0$, the channel is lossless, and the capacity is trivially 1. Consequently, we restrict our attention to $1\le a\le b<\min(\tau+1,w)$. We consider the setting where $\tau<w$ in this paper to reflect the motivating scenario where each message packet must be decoded in the presence of either a *single* burst of length b (or a arbitrary losses). In contrast, the setting of $\tau\ge w$ requires that each message packet be decoded in the presence of *multiple* bursts of losses. It is, hence, outside of the scope of this paper.

We will refer to input parameters (τ,a,b,w,τ_L) satisfying the above inequalities (along with $0\le\tau_L\le(\tau-b)$) as *valid* throughout this work.

[3] In the conference version of this paper [1], this parameter was called the "good-window-delay" parameter T_G.

IV. General Bounds on Rate for Streaming Codes With Variable-Size Messages

This section discusses general upper and lower bounds on the rate of code constructions for the proposed model. The bounds constitute the least upper bound and greatest lower bound for arbitrary message packet size sequences. Later, Lemma 5 shows that the optimal rate depends on the message packet size sequence and can vary over the entire range between the aforementioned lower and upper bounds on the optimal rate.

A. General Upper Bound on the Rate

As was discussed in Section II, the rate for streaming codes that satisfy the worst-case-delay constraint τ over a $C(a,b,w)$ channel in the setting where all message packets have a fixed size is at most $\frac{\tau-a+1}{\tau+b-a+1}$. Next, we show that this upper bound on the rate applies under the proposed model with message packets of variable sizes by using a simple extension to the proof techniques used by Badr et al. in [4].

Lemma 1: For any valid inputs (τ,a,b,w,τ_L), for any streaming code which satisfies the worst-case-delay constraint over the $C(a,b,w)$ channel, the rate is at most

$$R^{(U)}=\frac{\tau-a+1}{\tau+b-a+1}. \tag{2}$$

Proof Sketch: In [4], Badr et al. show that any streaming code satisfying the worst-case-delay constraint over a $C(a,b,w)$ channel must recover from *any* erasure channel which periodically introduces a burst of length b followed by a guard space of length $(\tau-a+1)$. Let $C_{P,i}$ be such an erasure channel whose bursts each begin in positions $\equiv i \mod (\tau+b-a+1)$, where $i \mod j$ is defined for a non-negative integer i and positive integer j as the remainder of i divided by j. Even when the sizes of channel packets vary, $C_{P,0},C_{P,1},\ldots,C_{P,(\tau+b-a)}$ erase on average $\frac{b}{\tau+b-a+1}$ fraction of the transmitted symbols. Therefore, there is always some C_{P,i_*} that erases at least $\frac{b}{\tau+b-a+1}$ fraction of the transmitted symbols. Consequently, the rate cannot exceed $R^{(U)}$. Thus, the upper bound on the rate provided in [4] for fixed-size message packets continues to hold for the model with variable-size message packets as well. ■

The rate $R^{(U)}$ is attained by the constructions presented in [5]–[9] when the sizes of the message packets are fixed.

Thus, $R^{(U)}$ is the smallest general upper bound on the rate of streaming codes for arbitrary message packet size sequences.

B. General Lower Bound on Rate

Next, we present a general lower bound on the rate. Each message packet can always be encoded separately (i.e., transmitted symbols corresponding to each message packet are kept independent of all other message packets) for any message packet size sequence. The optimal rate of coding schemes encoding message packets separately, thus, serves as a lower bound on the optimal rate. Moreover, this bound is tight for certain message packet size sequences and therefore is the greatest lower bound. For example, it is tight when the worst-case-delay is τ, and each message packet of positive size is followed by at least τ message packets of size 0. For such message packet size sequences, the sender *must* encode each message packet separately.

Next, we will present a simple code construction with the best-possible rate among code constructions that *encode each message packet separately* and identify its rate. For valid inputs (τ, a, b, w, τ_L), the proposed code construction is called the "(τ, a, b, τ_L)-separate encoding scheme." The scheme is presented in two cases.

Case 1: $\tau_L < (a-1)$. In this case, since $(\tau_L+1) < a$, all symbols used to decode a message packet under lossless transmission can be lost under lossy transmission. In addition, either $(a-1-\tau_L)$ arbitrary channel packets may be lost, or the next $(b-\tau_L-1)$ channel packets may be lost. Consequently, the rate is at most 0.5 in this case. We present the scheme first using a toy example and then in detail.

Toy example. An example of the $(7,3,5,1)$-separate encoding scheme is shown in Figure 3 for a message packet $S[i] = (S_0[i], \ldots, S_5[i])$. The blue boxes contain the symbols of $S[i]$, while the red boxes contain parity symbols for a systematic $[14, 6]$ MDS code. The 6 symbols of $S[i]$ are transmitted evenly over $X[i]$ and $X[i+1]$. The 8 parity symbols are transmitted evenly over $X[i+4], \ldots, X[i+7]$. The lossless-delay constraint is satisfied, since $S[i]$ is transmitted over $X[i]$ and $X[i+1]$. The worst-case-delay constraint is met, since for any burst of length 5, or any 3 arbitrary losses, enough symbols are received by time slot 7 to decode $S[i]$ using properties of the MDS code.

Detailed description. The symbols of $S[i]$ are sent evenly over all channel packets within the lossless-delay (i.e., $X[i], \ldots, X[i+\tau_L]$). The k_i symbols corresponding to $S[i]$ are transmitted evenly over the final $(\tau-b+1)$ packets by time slot $(i+\tau)$ to cover the case of a burst of length b starting in time slot i. Parity symbols are sent over the remaining channel packets to ensure at least k_i symbols are received for a arbitrary losses. For convenience of notation, let $a' = (\tau_L + 2 + \tau - b)$. The following terms are used

$$\langle \eta, \eta' \rangle = \begin{cases} \big\langle (\tau_L+1)(\tau-b+1), & \text{if } a \le a' \\ (\tau_L+1)(\tau-b-\tau_L+a) \big\rangle & \\ \big\langle (\tau_L+1)(\tau-b+1)(\tau-a+1), & \\ \big((\tau_L+1)(\tau-b+1)(\tau-a+1)+ & \text{if } a > a' \\ (\tau_L+1)(\tau-b+1)(b-\tau_L-1)\big)\big\rangle. & \end{cases} \tag{3}$$

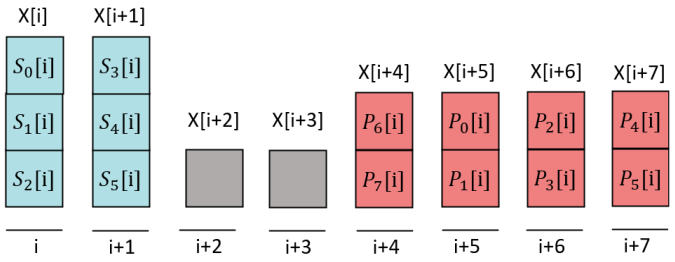

Fig. 3. The $(7,3,5,1)$-separate encoding scheme is shown for a message packet $S[i] = (S_0[i], \ldots, S_5[i])$. The symbols are spread evenly over channel packets $X[i]$ and $X[i+1]$, thereby satisfying the lossless-delay constraint. Additional parity symbols $(P_0[i], \ldots, P_7[i])$ of a $[14, 6]$ systematic MDS code are distributed evenly over channel packets $X[i+4]$, $X[i+5], X[i+6]$, and $X[i+7]$. This ensures that at least 6 out of $(S_0[i], \ldots, S_5[i], P_0[i], \ldots, P_7[i])$ are received in the event of either a burst of length at most 5 losses or the loss of any 3 arbitrary channel packets. Thus, $S[i]$ is decoded within 7 time slots by properties of the MDS code.

We assume that $\eta | k_i$.[4] The message packet is partitioned evenly into sets of η symbols. For each such set

- A $[\eta+\eta', \eta]$ systematic MDS code is applied, leading to symbols $c_0, \ldots, c_{\eta+\eta'-1}$, where the final η' symbols are parity symbols.
- The symbols $c_0, \ldots, c_{\eta-1}$ are evenly transmitted over $X[i], \ldots, X[i+\tau_L]$.
- The symbols $c_\eta, \ldots, c_{2\eta-1}$ are evenly transmitted over $X[i+b], \ldots, X[i+\tau]$.
- The symbols $c_{2\eta}, \ldots, c_{\eta'}$ are sent evenly over $X[i+j], \ldots, X[i+b-1]$, where $j = (\tau_L + b - a + 1)$ if $a \le (\tau_L + 2 + \tau - b)$ and $j = (\tau_L + 1)$ otherwise.

In short, the scheme involves (a) sending $S[i]$ over (τ_L+1) channel packets to satisfy the lossless-delay constraint, (b) sending parity symbols to recover $S[i]$ when $X[i], \ldots, X[i+b-1]$ are lost, and (c) sending parity symbols to recover $S[i]$ when both $X[i], \ldots, X[i+\tau_L]$ and $(a-\tau_L-1)$ additional channel packets of $X[i+\tau_L+1], \ldots, X[i+\tau]$ are lost.

Remark 1: The rate for the (τ, a, b, τ_L)-separate encoding scheme for case 1 is $\frac{\eta}{\eta'+\eta}$. This follows directly from the MDS code employed.

The field size requirement is at most that of a $[\eta'+\eta, \eta]$ Reed-Solomon code. If $a \le (\tau_L + 2 + \tau - b)$, the requirement is at most $(\tau_L+1)(2\tau - 2b - \tau_L + a + 1)$, which is no more than $2\tau a$. Otherwise, the field size requirement is at most

$$(\tau_L+1)(\tau-b+1)\big(2(\tau-a+1)+(b-\tau_L-1)\big),$$

which is no more than $3a^2 b$.

Case 2: $\tau_L \ge (a-1)$. In this case, τ_L is large enough that the symbols of $S[i]$ can be distributed over $(\tau_L+1) \ge a$ channel packets such that at most k_i symbols are lost by making use of a buffer of $(b-a)$ channel packets in which no symbols are sent, as will be described below. This approach leads to a rate of at least 0.5 in this case. We divide the presentation of case 2 into two sub-cases.

[4]It suffices to pad $S[i]$ with strictly fewer than η extra symbols, where $\eta \le \tau^2$ or $\eta \le \tau^3$ depending on whether $a \le (\tau_L + 2 + \tau - b)$. Typically, $\eta \ll k_i$.

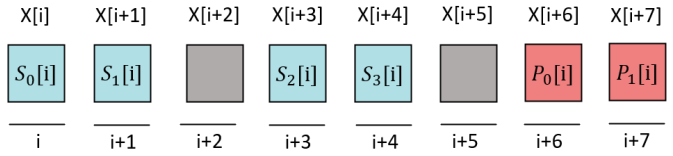

Fig. 4. The $(7, 2, 3, 4)$-separate encoding scheme is shown for a message packet $S[i] = (S_0[i], \ldots, S_3[i])$. The symbols are spread evenly over channel packets $X[i], X[i+1], X[i+3]$, and $X[i+4]$. Parity symbols $(P_0[i], P_1[i])$ of a systematic $[6, 4]$ MDS code are spread evenly over channel packets $X[i+6]$ and $X[i+7]$. The lossless-delay constraint is satisfied, since the symbols of $S[i]$ are sent by time slot $X[i+4]$. At most 2 nonempty channel packets are lost with a burst of length 3 or 2 arbitrary losses. Therefore, at least 4 of $(S_0[i], \ldots, S_3[i], P_0[i], P_1[i])$ are received, so $S[i]$ is decoded by time slot $(i+7)$.

Sub-case 1: either $((\tau_L + 1) \mod b) \in \{0\} \cup \{a, \ldots, b-1\}$ or $((\tau_L + 1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) \geq a$.

Toy example. An example of the $(7, 2, 3, 4)$-separate encoding scheme is shown in Figure 4 for a message packet $S[i] = (S_0[i], S_1[i], S_2[i], S_3[i])$. The parity symbols, $P_0[i]$, and $P_1[i]$, are formed using a $[6, 4]$ systematic MDS code. The 6 symbols, $(S_0[i], S_1[i], S_2[i], S_3[i], P_0[i], P_1[i])$ are then periodically sent over time slots i through $(i + 7)$ by transmitting one symbol for each of two consecutive time slots followed by not transmitting any symbols for one time slot. Specifically, $S_0[i], S_1[i], S_2[i], S_3[i], P_0[i]$, and $P_1[i]$ are sent over $X[i], X[i+1], X[i+3], X[i+4], X[i+6]$, and $X[i+7]$ respectively. The lossless-delay constraint is satisfied, as $S[i]$ is transmitted over $X[i], \ldots, X[i+4]$ where $\tau_L = 4$. At least 4 symbols are received by time slot $(i + 7)$. Hence, $S[i]$ can be decoded by properties of the MDS code.

Detailed description. The symbols of $S[i]$ are periodically spread over a channel packets followed by no symbols being sent in a buffer of $(b - a)$ channel packet until time slot $(i + \tau_L)$. Afterward, a buffer of $(b - a)$ channel packets are sent which do not include any symbols corresponding to $S[i]$. Parity symbols are sent in the next a channel packets. A similar interleaving approach with empty positions (i.e., buffers) was used in [26] and [27], albeit for the streaming model with message packets all having the same fixed size, where each message packet is sent in its entirety as part of the corresponding channel packet, and the parity symbols apply to multiple message packets. For convenience of notation, the following term is used

$$\zeta = \left(\left\lfloor \frac{\tau_L + 1}{b} \right\rfloor a + \min\left((\tau_L + 1) \mod b, a\right) \right). \quad (4)$$

We will assume that $\zeta | k_i$.[5] The message packet is partitioned into sets of ζ symbols. For each such set:

- A $[\zeta + a, \zeta]$ systematic MDS code is applied, leading to symbols $c_0, \ldots, c_{\zeta+a-1}$, where the final a symbols are parity symbols.
- For $J = \{j_0, \ldots, j_{\zeta-1}\} = \{j \mid j \in \{i, \ldots, i + \tau_L\}, j \mod b < (b-a)\}$ and $l \in \{0, \ldots, \zeta-1\}$, c_l is transmitted in $X[j_l]$.
- $c_\zeta, \ldots, c_{\zeta+a-1}$ are transmitted in $X[i + \tau_L + b - a + 1], \ldots, X[i + \tau_L + b]$ respectively.

Remark 2: The rate for the (τ, a, b, τ_L)-separate encoding scheme for case 2 sub-case 1 is $\frac{\zeta}{\zeta+a}$. This follows directly from the MDS code employed.

The field size requirement is that of a $[\zeta + a, \zeta]$ Reed-Solomon code. The requirement is at most $(\zeta + a)$, which is no more than $(\tau + 1 + a)$.

Sub-case 2: $((\tau_L + 1) \mod b) \in \{1, \ldots, a - 1\}$ and $((\tau_L + 1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) < a$. The construction from sub-case 1 applies to this sub-case, but its rate does not attain the greatest lower bound on the rate. The converse proof in sub-case 1 relies on the worst-case losses corresponding to either (a0 a burst of consecutive losses over between $((\tau_L + 1) \mod b)$ and b consecutive channel packets, or (b) a arbitrary losses corresponding to a set of $((\tau_L + 1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right)$ channel packets. However, in sub-case 2, both loss scenarios comprise fewer than a arbitrary losses. Hence, the worst-case a arbitrary losses cannot be limited to just one of these two quantities. In order to design a scheme that leads to the greatest lower bound for this sub-case, we introduce a construction based on a simple integer program (IP) that reflects minimizing the number of symbols sent by a coding scheme while satisfying the lossless-delay and worst-case-delay constraints. The variables of the IP are $n_0^{(c)}, \ldots, n_\tau^{(c)}$, which denote the sizes of the $(\tau + 1)$ channel packets corresponding to the message packet.

IP-based construction 1 Takes as input any valid parameters and k_i and uses integer programming to compute the number of symbols to be sent in the next $(\tau + 1)$ channel packets.

Input: Valid values for (τ, a, b, w, τ_L) and k_i.
Minimize $\sum_{j=0}^{\tau} n_j^{(c)}$ subject to:

1) $\forall j \in \{0, \ldots, \tau\}, n_j^{(c)} \geq 0$
2) $\left(\sum_{j=0}^{\tau_L} n_j^{(c)}\right) - k_i \geq 0.$
3) $\forall l \in \{0, \ldots, \tau\} \left(\sum_{j=0}^{l-1} n_j^{(c)}\right) + \left(\sum_{j=l+b}^{\tau} n_j^{(c)}\right) - k_i \geq 0.$
4) $\forall I \subseteq \{0, \ldots, \tau\}$ such that $|I| = a$, $\left(\sum_{j \in \{0,\ldots,\tau\} \setminus I} n_j^{(c)}\right) - k_i \geq 0$

Output: $n_i^{(*)} = \left(\sum_{j=0}^{\tau} n_i^{(c)}\right)$.

The constraints of the integer program reflect the requirements that (1) the size of each channel packet is non-negative, (2) the lossless-delay constraint is met, (3) the worst-case-delay constraint is met for bursts of at most b consecutive channel packets, and (4) the worst-case-delay constraint is met for a arbitrary losses. The objective function reflects minimizing the total number of symbols which are sent. Observe that $n_i^{(*)}$ is the total number of symbols sent according to the IP subroutine of IP-based construction 1. For a message packet size sequence $k_0, \ldots, k_t$, let $n_0^{(*)}, \ldots, n_t^{(*)}$ be the outputs of IP-based construction 1 applied to each of $k_0, \ldots, k_t$. For each i where $k_i > 0$, a systematic $[n_i^{(*)}, k_i]$ MDS code is applied to encode $S[i]$ into $c_0, \ldots, c_{n^{(*)}-1}$. The symbols are

[5]It suffices to pad $S[i]$ with up to $(\zeta - 1 \leq \tau)$ extra symbols—a quantity typically negligible compared to k_i.

distributed over channel packets $X[i], \ldots, X[i+\tau]$ so that the number sent for each channel packet $X[j]$ is $n_j^{(c)}$. The construction is systematic, as the first k_i symbols (i.e., $S[i]$) are sent over $X[i], \ldots, X[i+\tau_L]$. The following terms will be used to express the rate of IP-based construction 1

$$\langle k^{(*)}, n^{(*)} \rangle = \left\langle \sum_{i=0}^{t} k_i, \sum_{i=0}^{t} n_i^{(*)} \right\rangle. \tag{5}$$

Remark 3: The rate for the (τ, a, b, τ_L)-separate encoding scheme for message packet size sequence $k_0, \ldots, k_t$ for case 2 sub-case 2 is $\frac{k^{(*)}}{n^{(*)}}$. This follows directly from the MDS code employed.

Finally, we note that IP-based construction 1 can be used for any parameter settings. We provide explicit constructions for case 1 and case 2 sub-case 1 because IP-based construction 1 is not explicit.

Next, we use the (τ, a, b, τ_L)-separate encoding scheme described above to provide a general lower bound on the rate. Before doing so, we must verify that the (τ, a, b, τ_L)-separate encoding scheme satisfies the lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel for any valid parameters (τ, a, b, τ_L) and sequence of message packets. This is done below.

Lemma 2: For any valid inputs (τ, a, b, w, τ_L) and any sequence of message packets $S[0], \ldots, S[t]$, the (τ, a, b, τ_L)-separate encoding scheme satisfies the lossless-delay constraint τ_L and the worst-case-delay constraint τ over the $C(a, b, w)$ channel.

Proof Sketch: The proof of Lemma 2 is included in Appendix B. ■

Consequently, the rate of the (τ, a, b, τ_L)-separate encoding scheme constitutes a lower bound on the optimal rate. This quantity is summarized in Lemma 3.

Lemma 3: For any valid inputs (τ, a, b, w, τ_L), the optimal rate for streaming codes that satisfy the lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel for an arbitrary message packet size sequence is at least $R^{(L)} =$

$$\begin{cases} \frac{\eta}{\eta+\eta'} & \text{if } \tau_L < a-1 \\ \frac{\zeta}{\zeta+a} & \text{if } \tau_L \geq (a-1) \text{ and } ((\tau_L+1) \mod b) \in \\ & \{0\} \cup \{a, \ldots, b-1\} \text{ or } \tau_L \geq (a-1) \text{ and} \\ & ((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) \geq a \\ \frac{k^{(*)}}{n^{(*)}} & \text{if } \tau_L \geq (a-1) \text{ and} \\ & 0 < ((\tau_L+1) \mod b) < a \text{ and} \\ & ((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) < a. \end{cases} \tag{6}$$

Proof: The (τ, a, b, τ_L)-separate encoding scheme exhibits this rate. This follows directly from the parameters of the MDS code used in each case, as is noted in Remark 1, Remark 2, and Remark 3. The lossless-delay constraint and worst-case-delay constraint over the channel model are also satisfied by the code construction, as was shown in Lemma 2. ■

For any valid inputs (τ, a, b, w, τ_L), as τ_L increases, the quantity $R^{(L)}$, is monotonically non-decreasing and approaches the upper bound on the rate of $R^{(U)}$. Whenever $\tau_L = (\tau - b)$ and either $b = a$ or $((\tau+1) \mod b) = a$, the least upper bound on the rate of $R^{(U)}$ is equal to the greatest lower bound on the optimal rate of $R^{(L)}$. For such parameter settings, the rate of the (τ, a, b, τ_L)-separate encoding scheme matches the least upper bound on the rate of $R^{(U)}$. In such settings, the (τ, a, b, τ_L)-separate encoding scheme is also analogous to the scheme presented in [26] for the streaming model with message packets of the same fixed size.

We show in Lemma 4 that $R^{(L)}$ is the greatest lower bound on the optimal rate for arbitrary message packet size sequences.

Lemma 4: For any valid inputs (τ, a, b, w, τ_L), $R^{(L)}$ is the greatest lower bound on the optimal rate for arbitrary message packet size sequences for streaming codes that satisfy the lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel.

Proof Sketch: The proof of Lemma 4 is included in Appendix A. ■

V. Bounds on Rate for Specific Message Packet Size Sequences

In the proposed model for streaming codes with message packets of varying sizes, the optimal rate for any transmission depends on the specific message packet size sequence. The optimal rate can be as large as $R^{(U)}$ and as small as $R^{(L)}$, as was shown in Section IV. These general bounds are agnostic to the sizes of the message packets and apply to an *arbitrary* message packet size sequences. In this section, we develop a deeper understanding for the optimal rate of a streaming code for any *specific* message packet size sequence. We refer to the setting in which the sender and receiver have access to the complete message packet size sequence as the "offline" setting and consider it for the rest of this section. This differs from the setting considered in the rest of this work, which we call the "online" setting, where the sender and receiver do not have access to $k_{i+1}, \ldots, k_t$ during time slot i. The optimal rate for the online setting for any specific message packet size sequence is not well-defined because there exists a coding scheme which attains the best possible rate, which is that of the offline setting. However, the rate of that coding scheme may not be optimal for other message packet size sequences, as is discussed in detail in [28].

First, we show that the optimal rates for various message packet size sequences can take values over the entire range of $[R^{(L)}, R^{(U)}]$. Naturally, the general upper and lower bounds on the rate, i.e., $R^{(U)}$ and $R^{(L)}$, are inherently loose for many message packet size sequences, motivating the need for tighter bounds. We then present an algorithm to compute an upper bound on the rate for linear encoding schemes by imposing the lossless-delay constraint and worst-case-delay constraint over the channel model for each message packet. We then present an algorithm to compute the best possible rate for a coding scheme that combines block codes such as those presented in [5]–[9] with the separate encoding scheme presented in Section IV-B. The so-computed rate serves as a lower bound on the optimal rate. Finally, we empirically evaluate these upper and lower bounds on the optimal rate. The empirical evaluation demonstrates that the gap between the lower and

upper bounds computed by the two aforementioned algorithms is a significant improvement over the gap between the bounds agnostic to the size sequence.

Lemma 5: For any valid inputs (τ, a, b, w, τ_L), the set of optimal rates for coding schemes that satisfy the lossless-delay constraint and worst-case-delay constraint over any $C(a, b, w)$ channel over all possible message packet size sequences are dense in $\left[R^{(L)}, R^{(U)}\right]$.

Proof: Let $v \in \left[R^{(L)}, R^{(U)}\right]$, and $\epsilon > 0$ arbitrarily. We will show that there is a message packet size sequence for which the optimal rate is within ϵ of v. Let $p, r \in \mathbb{Z}^+ \cup \{0\}$ be chosen so that the quantity $R^{(p,r)} = \frac{p+r}{\frac{p}{R^{(L)}} + \frac{r}{R^{(U)}}}$ satisfies $|R^{(p,r)} - v| < \epsilon$. When $v \in \left(R^{(L)}, R^{(U)}\right)$, the existence of such p and r follows from the fact that $R^{(p,r)} \to v$ in the limit as $\frac{p}{r} \to \frac{R^{(L)}v - R^{(L)}R^{(U)}}{R^{(L)}R^{(U)} - R^{(U)}v}$. When $v = R^{(L)}$ or $v = R^{(U)}$ it suffices to choose $(r = 0, p > 0)$ and $(p = 0, r > 0)$ respectively. Let d be the smallest positive integer for which $\frac{d}{R^{(L)}}$ and $\frac{d}{R^{(U)}}$ are both integers. Consider the following length $(3\tau - a + 2)$ message packet size sequence: $k_0 = pd$, $k_j = \frac{rd}{\tau - a + 1}$ for $j \in \{\tau + 1, \ldots, 2\tau - a + 1\}$, and $k_j = 0$ for $j \in \{1, \ldots, \tau\} \cup \{2\tau - a + 2, \ldots, 3\tau - a + 1\}$.

The proof follows from verifying that the optimal rate for this message packet size sequence is at most $R^{(p,r)}$ and presenting a coding scheme with rate $R^{(p,r)}$, which we will show below.

Upper bound. The lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel must be satisfied for message packet $S[0]$. This necessitates that As such, at least $\frac{pd}{R^{(L)}}$ symbols are sent by time slot τ (Lemma 4). The lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel must be met for the rd symbols corresponding to the remaining $(\tau - a + 1)$ message packets. Thus, at least $\frac{rd}{R^{(U)}}$ additional symbols must be sent due to the upper bound on the rate of $R^{(U)}$. A total of at least $\frac{pd}{R^{(L)}} + \frac{rd}{R^{(U)}}$ symbols are sent, leading to an upper bound on the rate of $R^{(p,r)}$.

Achievability. Applying the (τ, a, b, τ_L)-separate encoding scheme to message packet $S[0]$ involves transmitting $\frac{pd}{R^{(L)}}$ symbol. The systematic $[\tau + b - a + 1, \tau - a + 1]$ block code, presented in [9] (or alternatively the block codes from [5]–[8]), can be applied to message packets $S[\tau + 1], \ldots, S[2\tau - a + 1]$ by sending each message packet in the corresponding channel packet. Afterward, the channel packets $X[2\tau - a + 2], \ldots, X[2\tau + b - a + 1]$ are defined to each contain $\frac{rd}{\tau - a + 1}$ parity symbols of the block code. The lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel are met by the definition of the (τ, a, b, τ_L)-separate encoding scheme and block codes. This code construction has a rate of $R^{(p,r)}$. ∎

Hence, the quantities $R^{(L)}$ and $R^{(U)}$ are insufficient for understanding the best possible rate for a specific message packet size sequence. As such, Lemma 5 motivates the need to compute upper and lower bounds on the optimal rate for any specific message packet size sequence that can more tightly bound the optimal rate. A desirable property for doing so is that the upper and lower bounds on rate can likewise range from $R^{(L)}$ to $R^{(U)}$. We introduce algorithms to compute upper and lower bounds on the optimal rate for any specific message packet size sequence in Sections V-A and V-B to capture this property.

A. An Upper Bound on the Optimal Rate for Specific message packet size sequences

We now present Algorithm 1, which computes an upper bound on the rate for *linear* encoding schemes for any given message packet size sequence by imposing the lossless-delay constraint and worst-case-delay constraint over the channel model for each message packet. To do so, Algorithm 1 will make use of an integer program by converting the lossless-delay and worst-case-delay constraints into constraints for the IP. In order to avoid confusion over the term "constraint," we refer to constraints of the IP as "constraints" and the lossless-delay and worst-case-delay constraints as "requirements" in this section and Section V-B. Under Algorithm 1, (1) the lossless-delay and worst-case-delay requirements are converted into constraints for an integer program (IP) with a simple minimization objective function, (2) its solution is computed, and (3) its solution is converted into an upper bound on the optimal rate. In Section V-C, we will show empirically that the upper bound on the optimal rate determined by Algorithm 1 can be significantly lower than $R^{(U)}$.

Consider any message packet size sequence of an arbitrary length t. Consider any valid inputs (τ, a, b, w, τ_L). We first model the sizes of the message and channel packets, and the associated parameters will serve as the variables for the IP. Each channel packet, $X[i]$, for $i \in \{0, \ldots, t\}$ comprises $(X^{(0)}[i], X^{(1)}[i])$. Under a lossless transmission, message packets $S[0], \ldots, S[i]$ are decoded using $X^{(0)}[0], \ldots, X^{(0)}[i + \tau_L]$. In contrast, $X^{(1)}[0], \ldots, X^{(1)}[i + \tau_L]$ are used for decoding only under a lossy transmission. The linear equations corresponding to the symbols of $X^{(1)}[i]$ are in the span of the linear equations corresponding to the symbols of $\langle X^{(0)}[j] \mid j \leq i \rangle$. Each quantity $|X^{(1)}[i]|$ will be a variable of the IP, whereas there will be $(\tau_L + 1)$ variables corresponding to $X^{(0)}[i]$ defined shortly. The details of how $(X^{(0)}[i], X^{(1)}[i])$ are defined are only used in in the proof of Theorem 1 and can be found in Appendix C.

The symbols of $X^{(0)}[i]$ are partitioned into $X_l^{(0)}[i]$ for $l \in \{i, \ldots, i - \tau_L\}$ for convenience of notation, where each quantity $|X_l^{(0)}[i]|$ will be a variable of the IP. Under a lossless transmission, the symbols sent in channel packet $X[j]$ for $j \in \{0, \ldots, \tau_L\}$ used to decode $S[0]$ are called $X_0^{(0)}[j]$. Similarly, for $i = 1, \ldots, t$, the symbols sent in channel packet $X[j]$ for $j \in \{i, \ldots, i + \tau_L\}$ that are used to decode $S[i]$ under lossless transmission are labeled as $X_i^{(0)}[j]$. Thus, $X^{(0)}[i] = \langle X_j^{(0)}[i] j \in \{i - \tau_L, \ldots, i\} \rangle$, and hence, $\sum_{j=i-\tau_L}^{i} |X_j^{(0)}[i]| = |X^{(0)}[i]|$ for any $i \in \{0, \ldots, t\}$.[6]

We next outline how the constraints for the IP reflect the worst-case-delay and lossless-delay requirements of the

[6]For convenience of notation, the edge conditions are handled by modeling $S[-\tau_L], \ldots, S[-1], S[t + 1], \ldots, S[t + \tau]$ as message packets of size 0. Furthermore, variables $|X_j^{(0)}[i]| = 0$ whenever at least one of i, j is either negative or i exceeds t. Similarly, $|X^{(1)}[i]| = 0$ whenever i is negative or exceeds t.

streaming model. For ease of presentation, in this paragraph, we assume that the coding scheme is systematic. Under a systematic coding scheme, the quantity $X_j^{(0)}[i]$ for $i \in \{0,\ldots,t\}, j \in \{i-\tau_L,\ldots,i\}$ corresponds to $|X_j^{(0)}[i]|$ distinct symbols of $S[j]$. Each of $|X_j^{(0)}[i]|$ and $|X^{(1)}[i]|$ are non-negative integers to reflect that each channel packet consists of some non-negative quantity of symbols corresponding to message packet $S[j]$ for $j \in \{i-\tau_L,\ldots,i\}$, along with some non-negative number of parity symbols (constraints #1 and #2 in Algorithm 1). The lossless-delay requirement is imposed through requiring that k_i symbols for message packet $S[i]$, for each $i \in \{0,\ldots,t-\tau\}$, be transmitted over $X[i],\ldots,X[i+\tau_L]$ (constraint #3).[7] In the proof of Lemma 1, it was shown that satisfying the worst-case-delay requirement over any $C(a,b,w)$ channel necessitates satisfying the worst-case-delay requirement over all channels which periodically drop b channel packets and allow $(\tau-a+1)$ successful transmissions. This implies that for each burst of length b starting in time slot $i \in \{0,\ldots,t\}$, $S[j]$, for $j \in \{i-\tau_L,\ldots,i+b-1\}$, must be decoded by time slot $(i+\tau+b-a)$, while the worst-case-delay requirement necessitates that $S[j]$ be decoded by $(j+\tau)$ (constraint #4). The worst-case-delay requirement is imposed for all possible patterns of a losses on every sliding window of length $(\tau+1)$. Specifically, under constraint #4 (respectively #5), for any considered burst of length b (respectively a arbitrary losses) beginning in time slot $i \in \{0,\ldots,t-b\}$ (respectively $i \in \{0,\ldots,t-\tau\}$) and terminating in time slot $i' \in \{i,\ldots,t\}$, the following relaxation of the worst-case-delay requirement is imposed. For each message packet $S[j] \in \{S[i-\tau_L],\ldots,S[i']\}$, $S[i-\tau_L],\ldots,S[j]$ must be decoded by time slot $(j+\tau)$. This relaxation is more restrictive than the relaxation which allows all lost message packets to be decoded within τ time slots of the final lost message packet. Finally, we consider the relaxation that each $X_{j'}^{(0)}[i']$ is received even if $X[i']$ is lost for $j' > j$. A toy example of constraint #4 is shown in Figure 5. An analogous figure could be constructed for constraint #5.

The objective function is to minimize the sum of all variables. The summation constitutes a lower bound on the number of transmitted symbols. The solution is easily converted into an upper bound on the rate since the total number of symbols of the message packets is fixed.

We now present Algorithm 1.

In Theorem 1, we verify that the output of Algorithm 1 is an upper bound on the rate.

Theorem 1: For any valid inputs (τ,a,b,w,τ_L) and any message packet size sequence $k_0 \ldots k_t$, the value computed by Algorithm 1 is an upper bound on the rate of streaming codes that satisfy the lossless-delay requirement and worst-case-delay requirement over the $C(a,b,w)$ channel while employing linear encoding.

Proof Sketch: Follows from the high-level description presented above. The full details are shown in Appendix C. ■

[7]The final τ message packets are of size 0 and, therefore, no lossless-delay requirement needs to be imposed. For $i<0$ or $j<0$, as well as $i>t$ and $j \in \{0,\ldots,\tau_L\}$, $|X_j^{(0)}[i]|=0$ is defined only for edge conditions.

Algorithm 1 Takes as input any valid parameters and message packet size sequence and uses integer programming to compute an upper bound on the rate of streaming codes with linear encoding schemes for the input message packet size sequence.

Input: Valid values for (τ,a,b,w,τ_L) and message packet size sequence $k_0,\ldots,k_t$.

Minimize $\sum_{i=0}^{t+\tau}\left(|X^{(1)}[i]|+\sum_{j=i-\tau_L}^{i}|X_j^{(0)}[i]|\right)$ subject to:

1) $\forall i \in \{0,\ldots,t-\tau\}, j \in \{i-\tau_L,\ldots,i\}, |X_j^{(0)}[i]| \geq 0$ and $|X_{j'}^{(0)}[i']|=0$ when $i'<0, i'>t$, or $j'<0$.
2) $\forall i \in \{0,\ldots,t+\tau\}, |X^{(1)}[i]| \geq 0$.
3) $\forall i \in \{0,\ldots,t-\tau\}, \sum_{j=0}^{\tau_L}|X_i^{(0)}[i+j]| \geq k_i$ and $\sum_{j=0}^{\tau_L}|X_j^{(0)}[i+j]| \leq k_i$.
4) $\forall i \in \{0,\ldots,t-b+1\}, \forall j \in \{i-\tau_L,\ldots,i+b-1\}$
$$\sum_{z=i+b}^{\min(j+\tau,i+b+\tau-a)}|X^{(1)}[z]| - \sum_{l=i-\tau_L}^{j}k_l + \sum_{l=i-\tau_L}^{j}\sum_{z\in\{l,\ldots,l+\tau_L\}\setminus\{i,\ldots,i+b-1\}}|X_l^{(0)}[z]| \geq 0.$$
5) $\forall i \in \{0,\ldots,t-\tau\} \forall I \subseteq \{i,\ldots,i+\tau\}$ of size $|I| = a, \forall j \in \{\min(I)-\tau_L,\ldots,\max(I)\}$,
$$\sum_{z\in\{\min(I)+1,\ldots,j+\tau\}\setminus I}|X^{(1)}[z]| - \sum_{l=i-\tau_L}^{j}k_l + \sum_{l=i-\tau_L}^{j}\sum_{z\in\{l,\ldots,l+\tau_L\}\setminus I}|X_l^{(0)}[z]| \geq 0.$$

Output: $\frac{\sum_{i=0}^{t}k_i}{\sum_{i=0}^{t+\tau}\left(|X^{(1)}[i]|+\sum_{j=i-\tau_L}^{i}|X_j^{(0)}[i]|\right)}$.

Remark 4: The value computed by Algorithm 1 is also an upper bound on the rate for streaming code constructions in an online setting since the offline setting involves providing the sender and receiver additional information not available in the online setting.

Next, we show that the outputs of Algorithm 1 can tightly bound the optimal rate for various message packet size sequences with values ranging from $R^{(L)}$ to $R^{(U)}$. Recall from Lemma 5 that this is a desired property because the optimal rate can likewise range from $R^{(L)}$ to $R^{(U)}$. For any message packet size sequence, $(k_0,\ldots,k_t)$, let $Alg^{(1)}_{\tau,a,b,w,\tau_L}(k_0,\ldots,k_t)$ and $Opt_{\tau,a,b,w,\tau_L}(k_0,\ldots,k_t)$ denote the output of Algorithm 1 and the optimal rate respectively.

Lemma 6: For any valid parameters (τ,a,b,w,τ_L), for all $\epsilon>0$ and $v \in [R^{(L)},R^{(U)}]$, there exists a sequence of message packet sizes $(k_0,\ldots,k_t)$ such that $Alg^{(1)}_{\tau,a,b,w,\tau_L}(k_0,\ldots,k_t) = Opt_{\tau,a,b,w,\tau_L}(k_0,\ldots,k_t)$ and $|Alg^{(1)}_{\tau,a,b,w,\tau_L}(k_0,\ldots,k_t) - v| < \epsilon$.

Proof: We now introduce a message packet size sequence for which the optimal is within ϵ of v. Let $p,r \in \mathbb{Z}^+ \cup \{0\}$ be chosen so that the quantity $R^{(p,r)} = \frac{p+r}{\frac{p}{R^{(L)}}+\frac{r}{R^{(U)}}}$ obeys the inequality $|R^{(p,r)} - v| < \epsilon$. Let d be the smallest positive

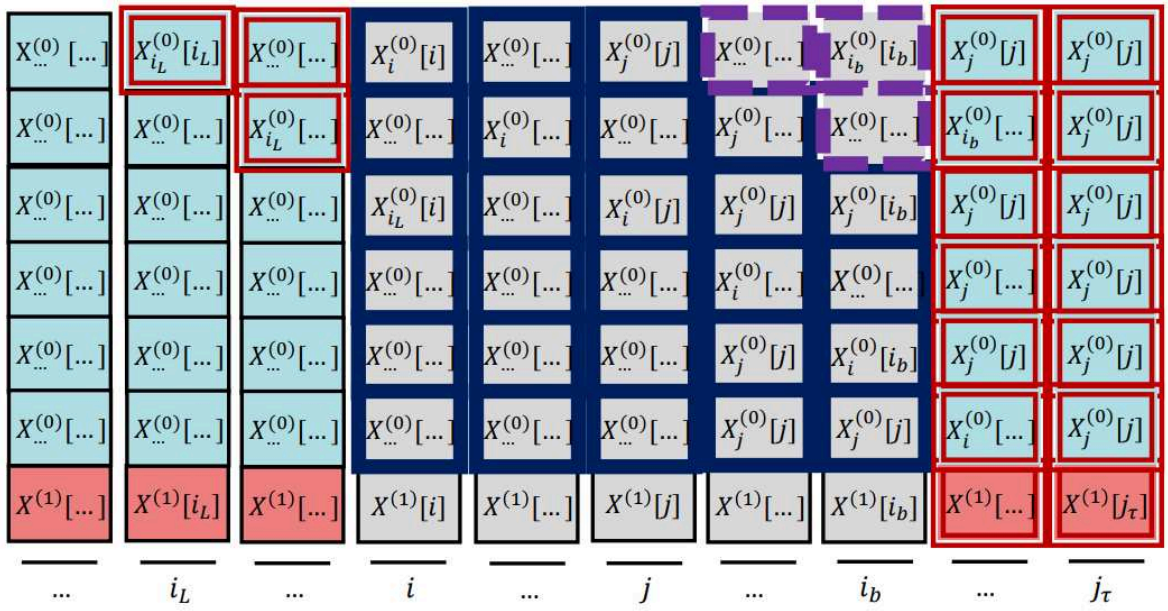

Fig. 5. An example of imposing constraint #4 (in Algorithm 1) for $j \in \{i, \ldots, i+b-2\}$. The quantities i_L, i_b, and j_τ represent $(i-\tau_L), (i+b-1)$, and $(j+\tau)$ respectively. The gray boxes (time slots $i, \ldots, i_b$) are lost in a burst of channel packets $X[i], \ldots, X[i_b]$. The symbols in the gray boxes with thick blue outlines must be recoverable using the symbols inside boxes with double red outlines. This requirement allows for the relaxation that the symbols inside boxes with purple dashed outlines are treated as received.

integer such that $\frac{d}{R^{(L)}}$ and $\frac{d}{R^{(U)}}$ are integers. Consider the message packet size sequence $k_0 = pd, k_j = \frac{rd}{\tau-a+1}$ for $j \in \{\tau+1, \ldots, 2\tau-a+1\}$, and $k_j = 0$ for $j \in \{1, \ldots, \tau\} \cup \{2\tau-a+2, \ldots, 3\tau-a+1\}$.

The code construction presented in the proof of Lemma 5 satisfies lossless-delay requirement and worst-case-delay requirement over the $C(a,b,w)$ channel and has rate $R^{(p,r)}$. Hence, the optimal rate is at least $R^{(p,r)}$.

The lossless-delay requirement and worst-case-delay requirement over the $C(a,b,w)$ channel are both imposed under Algorithm 1. As shown in the proof of Lemma 4, these requirements are sufficient to show that at least $\frac{pd}{R^{(L)}}$ symbols must be sent by time slot τ due to message packet $S[0]$. Recall from Section IV that the rate is upper bounded by $R^{(U)} = \frac{\tau-a+1}{\tau+b-a+1}$. This holds because all message packets are decoded for any lossy channels $C_{P,i}$ for $i \in \{0, \ldots, \tau+b-a\}$ consisting of bursts of length b starting in positions $\equiv i \mod (\tau+b-a+1)$. Furthermore, at least one such channel drops at least $\frac{b}{\tau-a+1}$ fraction of the transmitted symbols. Similarly, one can show that at least one such channel drops at least $\frac{b}{\tau-a+1}$ fraction of the symbols sent strictly after time slot τ. All such periodic packet loss channels $C_{P,i}$ are accounted for under Algorithm 1, due to constraint #4. Hence, the output of Algorithm 1 reflects that at least $\frac{rd}{R^{(U)}}$ additional symbols are sent strictly after time slot τ. The output of Algorithm 1 is, thus, at most $R^{(p,r)}$.

The value computed by Algorithm 1 is an upper bound on the optimal rate, which is at least $R^{(p,r)}$. Therefore, Algorithm 1 must output $R^{(p,r)}$; this is a tight upper bound on the rate for the message packet size sequence, and it is within ϵ of v. ■

The value computed by Algorithm 1 is an upper bound on the rate, but the algorithm can be computationally intensive. We now discuss modifications to the algorithm that trade off tightness for runtime.

There is a simple linear program (LP) relaxation of Algorithm 1 which uses non-negative real-valued variables $|X_i^{(0)}[j]|$ and $|X^{(1)}[i]|$ rather than integral ones. A solution to this LP can be converted into an upper bound on the rate by setting each variable to be the floor of its previous value. The conversion changes the size of each channel packet by at most (τ_L+2) which, in practice, is several orders of magnitude less than the average size of the message packets. Finally, Lemma 6 would likewise apply to the LP relaxation of Algorithm 1.

Remark 5: Modifying Algorithm 1 to solve an LP relaxation of the underlying IP has a negligible impact on its output.

It is simple to analyze the runtime of the modified version of Algorithm 1 that uses an LP relaxation of the IP. For any valid inputs (τ, a, b, w, τ_L) and message packet size sequence $k_0 \ldots k_t$, the total number of combined constraints in Algorithm 1 is at most

$$\begin{aligned}&(\tau_L+2)t + (b+\tau_L+1)t + \binom{\tau}{a}(\tau_L+\tau+1)t \\ \leq &(b+2\tau_L+3+2\tau^{a+1})t.\end{aligned}$$

Consequently, Algorithm 1 with the LP relaxation of the IP runs in $poly(\tau^a, t)$ time.

One might want a polynomial-time algorithm to compute a tighter upper bound on the rate than $R^{(U)}$. Therefore, one may run the LP relaxation of Algorithm 1 without constraint #5. In this case, there are at most $(b+2\tau_L+3)t$ constraints, and the run time is $poly(\tau t)$. Despite the relaxation, the algorithm

imposes the lossless-delay requirement and the worst-case-delay requirement over a burst-only channel model which introduces bursts of length b separated by guardspaces of at least $(\tau - a + 1)$.

Remark 6: Modifying Algorithm 1 to use an LP relaxation of the underlying IP and removing constraint #5 results in a polynomial-time algorithm. Its output is less than or equal to $R^{(U)}$. The modified algorithm imposes more stringent constraints than computing Algorithm 1 for inputs $(\tau, 1, b, w, \tau_L)$.

Algorithm 2 Takes as input any valid parameters and a message packet size sequence and uses integer programming to compute a lower bound on the optimal rate for the input message packet size sequence.

Input: Valid (τ, a, b, w, τ_L) and message packet size sequence $k_0 \dots k_{t-\tau}$.

Minimize $\left(\sum_{i=0}^{t-\tau} \frac{e_i}{R^{(L)}} + bp_i\right) + \left(\sum_{i=0}^{t-\tau} \sum_{j=i}^{i+\tau_L} k_{j,i}\right)$ subject to:

1) $\forall i \in \{0, \dots, t-\tau\}, e_i \geq 0$.
2) $\forall i \in \{0, \dots, t-\tau\}, j \in \{i, \dots, i+\tau_L\}, k_{i,j} \geq 0$.
3) $\forall i \in \{0, \dots, t-\tau\}, p_i \geq 0$.
4) $\forall i \in \{0, \dots t-\tau\}, e_i - k_i + \sum_{j=i}^{i+\tau_L} k_{i,j} = 0$.
5) $\forall i \in \{0, \dots, t-\tau\}, j = \tau_L, \dots, j = 1$, $\sum_{z=\max(i-\tau+a,0)}^{i+a-b-j} p_z - \sum_{z=j}^{\tau_L} k_{i-z,i} \geq 0$
6) $\forall i \in \{0, \dots, t-\tau\} \sum_{z=\max(i-\tau+a,0)}^{i} p_z - \sum_{z=0}^{\tau_L} k_{i-z,i} \geq 0$.

Output: $\frac{\sum_{i=0}^{t-\tau} k_i}{\left(\sum_{i=0}^{t-\tau} \frac{e_i}{R^{(L)}} + bp_i\right) + \left(\sum_{i=0}^{t-\tau} \sum_{j=i}^{i+\tau_L} k_{j,i}\right)}$.

B. A Lower Bound on the Optimal Rate for Specific Message Packet Size Sequences

We now present Algorithm 2, which computes a lower bound on the optimal rate for *offline* streaming codes that satisfy the lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ channel. Specifically, under Algorithm 2, an integer program with a simple minimization function is used to determine the minimum number of symbols which need to be transmitted using a combination of two schemes. The solution to this integer program is then converted into a lower bound on the optimal rate. The values computed by Algorithm 2 over various message packet size sequences can vary over $[R^{(L)}, R^{(U)}]$. We will later see in Section V-C that the empirically computed lower bound on the optimal rate determined by Algorithm 2 can be significantly tighter than that of $R^{(L)}$. Specifically, the gap between the output of Algorithm 2 and Algorithm 1 is shown to be small in Section V-C, highlighting the utility of Algorithm 1 empirically. A high-rate offline construction (e.g., Algorithm 2) is of interest because it lays the groundwork for designing high-rate online constructions. For example, in a recent work [29], the authors convert an offline rate-optimal construction into an online approximately rate-optimal online construction. Algorithm 2 also outputs an upper bound on the rate for a class of coding schemes which include the *online* coding scheme that will be presented in Section VI-B. A detailed discussion on the difference between the best possible rate for online and offline streaming codes is presented in [28]. We first provide an overview of Algorithm 2 before discussing its technical details. Finally, we consider the accuracy-runtime trade-off for the LP relaxation of the algorithm.

Similar to Section V-A, we refer to constraints of an IP as "constraints" and the lossless-delay and worst-case-delay constraints as "requirements" for convenience of notation. Each symbol of each message packet, $S[i]$, is encoded either as part of a block of the SBC or using the (τ, a, b, τ_L)-separate encoding scheme. To reflect this, the number of symbols corresponding to $S[i]$ encoded using the (τ, a, b, τ_L)-separate encoding scheme is denoted e_i. For $j \in \{i, \dots, i+\tau_L\}$, the variable $k_{i,j}$ will represent the number symbols corresponding to $S[i]$ sent in $X[j]$ within blocks of the SBC. Finally, p_i will reflect the number of blocks of the SBC whose first position occurs in channel packet $X[i]$.

The lossless-delay and worst-case-delay requirements are satisfied for symbols of message packets encoded using the (τ, a, b, τ_L)-separate encoding scheme via the properties of the (τ, a, b, τ_L)-separate encoding scheme detailed earlier. There must be a non-negative quantity of symbols encoded in this manner (constraint #1 in Algorithm 2). The number of symbols corresponding to message packet $S[i]$ sent in channel packet j is non-negative (constraint #2). Similarly, the number of blocks corresponding to each message packet is non-negative (constraint #3). All symbols of message packets not encoded using the (τ, a, b, τ_L)-separate encoding scheme must be decoded within delay τ_L under lossless transmission (constraint #4). Finally, under the two considered code constructions, all symbols of message packets which are not encoded using the (τ, a, b, τ_L)-separate encoding scheme must be encoded via a block of SBC which ensures decoding within the worst-case-delay requirement under lossy conditions (constraints #5 and #6). Specifically, all symbols of message packet $S[i]$ for i transmitted in a time slot later than i (not as part of the (τ, a, b, τ_L)-separate encoding scheme) must be encoded as part of blocks whose final parity symbol is transmitted by time slot $(i + \tau)$ (constraint #5). Furthermore, all symbols for message packet $S[i]$ sent in channel packet $X[i]$ are encoded via blocks of the SBC which have an open slot in position i (constraint #6). Figure 6 depicts a toy example of how these constraints may be applied for a single time slot, i.

The objective function of the IP used under Algorithm 2 is the total number of symbols sent via the solution to the IP. Minimizing this quantity ensures that the fewest number of symbols possible are transmitted, thereby maximizing the rate. The combined number of symbols of all message packets is divided by total number of transmitted symbols to output the rate of the corresponding coding scheme.

In Theorem 2, we show that the output of Algorithm 2 is a lower bound on the optimal rate.[8]

Theorem 2: For any valid inputs (τ, a, b, w, τ_L) and any message packet size sequence $k_0 \dots k_t$, Algorithm 2 outputs

[8]The extra padding symbols needed to employ the (τ, a, b, τ_L)-separate encoding scheme is ignored. The padding only negligibly impacts the value computed when the number of extra padding symbols is small compared to the size of each message packet, as is typical.

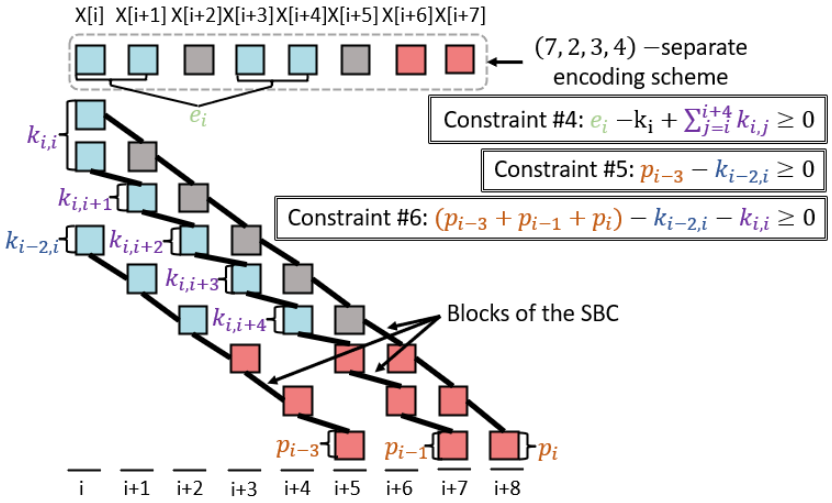

Fig. 6. An example imposing constraints #4, #5, and #6 for time slot i for parameters $(\tau, a, b, \tau_L) = (7, 2, 3, 4)$. Blue boxes can hold symbols of message packets. Red boxes hold parity symbols. Gray boxes contain no symbols. Boxes above time slot j correspond to symbols sent in channel packet $X[j]$. At least k_i symbols are sent for $S[i]$ consisting of (a) e_i symbols sent as part of the $(7, 2, 3, 4)$-separate encoding scheme (shown at the top), and (b) $k_{i,j}$ symbols sent in channel packet $X[j]$ for $j \in \{i, \ldots, i+4\}$ (constraint #4). There are p_{i-3} blocks of the SBC for which the final parity symbols are sent during time slot $(i+5)$. The total number of symbols sent in channel packet $X[i]$ corresponding to message packet $S[i-2]$ (i.e., $k_{i-2,i}$) is at most p_{i-3} (constraint #5). In addition, there are p_{i-1} and p_i blocks of the SBC for which the final parity symbols are sent during time slots $(i+7)$ and $(i+8)$ respectively. The number of symbols of all message packets sent in channel packet $X[i]$ within blocks of the SBC (i.e., $(k_{i-2,i} + k_{i,i})$) is at most $(p_{i-3} + p_{i-1} + p_i)$ (constraint # 6).

a lower bound on the optimal rate for streaming codes that satisfy the lossless-delay requirement and worst-case-delay requirement over the $C(a, b, w)$ channel.

Proof Sketch: Follows from the ideas presented above. A complete proof is included in Appendix D. ■

We now demonstrate that outputs of Algorithm 2 range from $R^{(L)}$ to $R^{(U)}$. Having outputs vary over all possible values of the optimal rate for various message packet size sequences is a useful property because the optimal rate can likewise range from $R^{(L)}$ to $R^{(U)}$, as was shown in Lemma 5. For any message packet size sequence, $(k_0, \ldots, k_t)$, let $Alg^{(2)}_{\tau,a,b,w,\tau_L}(k_0, \ldots, k_t)$ and $Opt_{\tau,a,b,w,\tau_L}(k_0, \ldots, k_t)$ denote the output of Algorithm 1 and the optimal rate respectively.

Lemma 7: For any valid parameters (τ, a, b, w, τ_L), for all $\epsilon > 0$ and $v \in [R^{(L)}, R^{(U)}]$, there exists a sequence of message packet sizes, $(k_0, \ldots, k_t)$, such that $Alg^{(2)}_{\tau,a,b,w,\tau_L}(k_0, \ldots, k_t) = Opt_{\tau,a,b,w,\tau_L}(k_0, \ldots, k_t)$ and $|Alg^{(2)}_{\tau,a,b,w,\tau_L}(k_0, \ldots, k_t) - v| < \epsilon$.

Proof: We will introduce a message packet size sequence whose optimal rate is within ϵ of v. Let $p, r \in \mathbb{Z}^+ \cup \{0\}$ be chosen and the quantity $R^{(p+r)} = \frac{p+r}{\frac{p}{R^{(L)}} + \frac{r}{R^{(U)}}}$ defined so that $|R^{(p+r)} - v| < \epsilon$. Let d be the smallest positive integer such that $\frac{d}{R^{(L)}}$ and $\frac{d}{R^{(U)}}$ are integers. Consider the message packet size sequence $k_0 = pd, k_j = \frac{rd}{\tau - a + 1}$ for $j \in \{\tau + 1, \ldots, 2\tau - a + 1\}$, and $k_j = 0$ for $j \in \{1, \ldots, \tau\} \cup \{2\tau - a + 2, \ldots, 3\tau - a + 1\}$.

The code construction presented in the proof of Lemma 5 satisfies lossless-delay requirement and worst-case-delay requirement over the $C(a, b, w)$ channel and has rate $R^{(p+r)}$. Moreover, the scheme follows from applying the (τ, a, b, τ_L)-separate encoding scheme to message packet $S[0]$ and blocks of the SBC to message packets $S[\tau + 1], \ldots, S[2\tau - a + 1]$. Thus, the variables of the IP computed by Algorithm 2 could represent this scheme while satisfying all constraints. Therefore, Algorithm 2 will output a value of at least $R^{(p+r)}$.

As was shown in Lemma 5, $R^{(p+r)}$ is also an upper bound on the rate. Hence, the value computed by Algorithm 2 is a tight lower bound on the optimal rate. It is also within ϵ of v. ■

Algorithm 2 computes a lower bound on the rate, but it can be computationally intensive. For any valid inputs (τ, a, b, w, τ_L) and message packet size sequence, consider the LP relaxation of Algorithm 2 which uses non-negative real-valued variables $e_i, k_{i,j}, p_i$ for $i \in \{\tau + b, \ldots, t - (\tau + b)\}$. It is possible to transform a real-valued solution into an integral one by setting each variable to be the ceiling of its previous value. The number of symbols transmitted corresponding to each message packet increases by at most $(\tau_L + 3 + b)$. In practice, $(\tau_L + 3 + b)$ is several orders of magnitude less than the average size of the message packets and leads to a negligible impact on the tightness of the bound. The total number of constraints for the LP is at most $t(2\tau_L + 5)$. Hence, the number of constraints is quadratic in the input parameters (and linear in the length of the message packet size sequence).

Remark 7: Modifying Algorithm 2 to use an LP relaxation of the underlying IP results in a polynomial-time algorithm while changing the output only negligibly.

C. Empirical Evaluation of the Bounds on Rate

The general upper and lower bounds on the optimal rate, $R^{(U)}$ and $R^{(L)}$, are tight for certain message packet size sequences. Yet the optimal rate for a specific message packet size sequence varies over the entire range of $[R^{(L)}, R^{(U)}]$. Thus, $R^{(U)}$ and $R^{(L)}$ can be loose depending on the message

TABLE I

PARAMETER SETTINGS USED IN THE EMPIRICAL EVALUATION OF THE BOUNDS ON THE OPTIMAL RATE

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
τ	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
a	1	1	1	1	2	2	1	1	2	2	3	3	1	1	2	2	3	3	4	4	1	2	3	4	5
b	1	1	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	5
τ_L	0	4	0	3	0	3	0	2	0	2	0	2	0	1	0	1	0	1	0	1	0	0	0	0	0

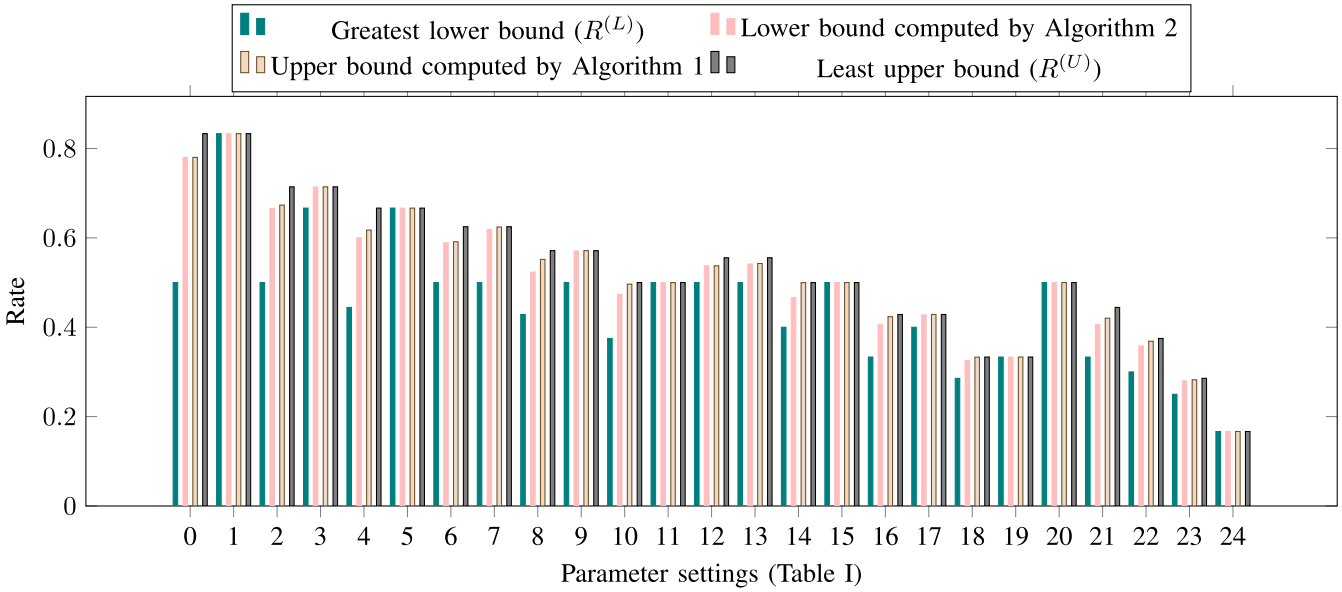

Fig. 7. Comparison over the parameter settings listed in Table I for the live video trace shown in Figure 1 of the four bounds on the optimal rate for the offline setting: the greatest lower bound ($R^{(L)}$)), the lower bound computed by Algorithm 2, the upper bound computed by Algorithm 1, and the least upper bound ($R^{(U)}$).

packet size sequences. In contrast, the upper and lower bounds on optimal rate computed by Algorithms 1 and 2 can range over all feasible values of the optimal rate, $[R^{(L)}, R^{(U)}]$. We evaluate the usefulness of the latter two bounds by empirically evaluating them over the live video trace shown in Figure 1. We show that the gap is *small in magnitude and a significant improvement over the gap between* $R^{(U)}$ *and* $R^{(L)}$.

We consider the setting of a small worst-case-delay (i.e., $\tau = 5$) and all parameter settings $(\tau = 5, a, b, \tau_L)$ where τ_L takes on its minimum and maximum values of 0 and $(\tau - b)$. Algorithms 2 and 1 bound the optimal rate significantly more tightly than $R^{(L)}$ and $R^{(U)}$ for many parameter settings. In the remaining settings, one or both of the lower and upper bounds on the optimal rate of $R^{(L)}$ and $R^{(U)}$ is nearly tight. Specifically, the size of the gap between Algorithms 2 and 1 is, on average, 65.4% smaller than the gap between $R^{(U)}$ and $R^{(L)}$ over the parameter settings from Table I. Furthermore, the gap between the bounds computed by the two algorithms is less than 0.034 (on average $.007$) over the parameter settings from Table I, as is shown in Figure 7. The results demonstrate the effectiveness of the algorithms in bounding the optimal rate for the offline setting.

VI. EXPLICIT CONSTRUCTION OF STREAMING CODES FOR VARIABLE MESSAGE-SIZES

In this section, we present a streaming code construction for any valid inputs (τ, a, b, w, τ_L) and any message packet size sequence, referred to as the (τ, a, b, τ_L)-Variable-size Streaming Code (or (τ, a, b, τ_L)-VSC). We include a toy example to illustrate the details of the construction concretely in Section VI-A. We then present the general construction in Section VI-B. Finally, we determine the field size requirement of the construction in Section VI-C.

The construction is simple and intuitive. At a high level, the SBC is leveraged as an underlying component. The adverse effects of the variability of the sizes of message packets is alleviated via the following two-step process:

1) The symbols from each message packet are distributed over the $(\tau_L + 1)$ channel packets within τ_L time slots. Blocks of the SBC (existing block codes presented in Section II) are created to satisfy the worst-case-delay constraint over a $C(a, b, w)$ channel for $\frac{\tau_L+1}{\tau-a+1}$ fraction of the symbols of each message packet. The objective of doing so is to (a) reduce the variability of the sizes of the channel packets and (b) minimize the number of empty positions in blocks of the SBC.
2) Additional parity packets are sent to ensure all symbols not included in any block of SBC are recovered within the worst-case-delay, τ, over the $C(a, b, w)$ channel.

A. Toy Example

We start by illustrating the details of the code construction for a simple toy setting with input parameters $(\tau = 3, a = 1, b = 2, w \geq 4, \tau_L = 1)$ and the specific sequence of message packets shown in Figure 8. Each message packet

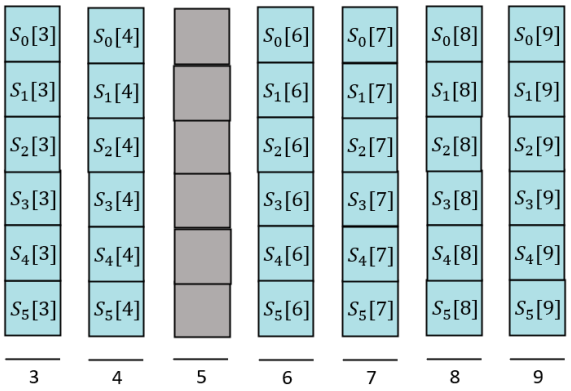

Fig. 8. An example sequence of message packets. Each message packet, $S[i]$, has 6 total symbols, $S_0[i], \ldots, S_5[i]$, which are contained in the blue boxes. The unlabeled gray boxes are empty and contain no symbols. The numbers under the lines at the bottom indicate the time slots.

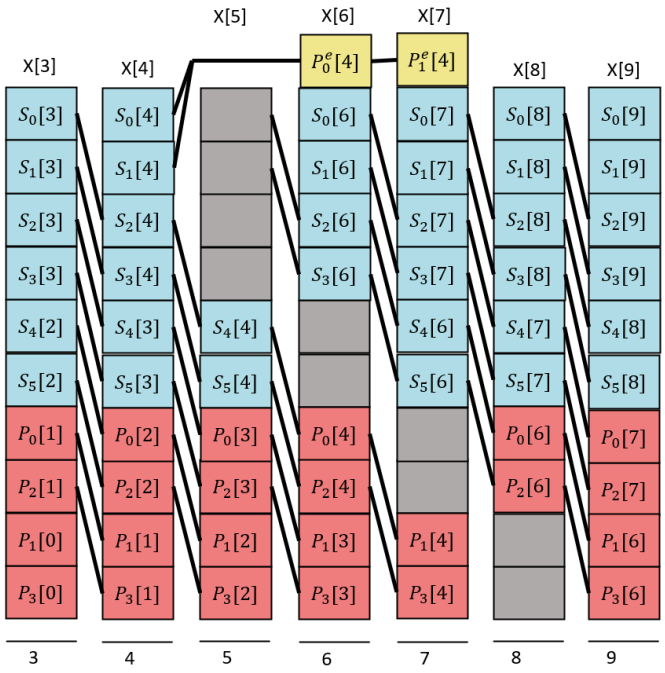

Fig. 9. The channel packets sent under the $(3, 1, 2, 1)$-VSC for the sequence of message packets from Figure 8. The blue boxes labeled $S_j[i]$ each contain one symbol of the message packet $S[i]$. The red boxes labeled $P_j[i]$ each contain one parity symbol from the interleaved SBC. The yellow boxes labeled $P_j^e[4]$ each contain one parity symbol. The unlabeled gray boxes are empty. Black lines connect boxes that are part of the same block. The numbers under the lines at the bottom indicate the time slots.

consists of 6 symbols except for $S[5]$, which contains 0 symbols. The channel packets sent under the $(3, 1, 2, 1)$-VSC for this sequence of message packets are shown in Figure 9. During each time slot, the sender does not have access to the sizes of the future message packets.

In the toy example, for $i \in \{3, \ldots, 9\}$, the first 4 symbols of each message packet $S[i]$, that is $(S_0[i], S_1[i], S_2[i], S_3[i])$, are sent in $X[i]$. The remaining 2 symbols, $(S_4[i], S_5[i])$, are sent in $X[i+1]$. Next, two blocks of the SBC are created: (a) $\langle S_0[i-1], S_2[i], S_4[i], P_0[i], P_1[i]\rangle$, and (b) $\langle S_1[i-1], S_3[i], S_5[i], P_2[i], P_3[i]\rangle$. The parity symbols $P_0[i]$ and $P_1[i]$ are sent in $X[i+2]$ and $X[i+3]$ respectively. Similarly, the parity symbols $P_2[i]$ and $P_3[i]$) are sent in $X[i+2]$ and $X[i+3]$ respectively. Each of these blocks is interleaved over channel packets $\langle X[i-1], X[i], X[i+1], X[i+2], X[i+3]\rangle$. Message packet $S[5]$ had 0 symbols, so the blocks $\langle empty, S_2[6], S_4[6], P_0[6], P_1[6]\rangle$ and $\langle empty, S_3[6], S_5[6], P_2[6], P_3[6]\rangle$ treat $empty$ as a unique finite field symbol used for padding. Moreover, no blocks of the SBC are created corresponding to message packet $S[5]$, since it is of size 0. As a result, $S_0[4]$ and $S_1[4]$ are not placed in any block of the SBC. To ensure that $S_0[4]$ and $S_1[4]$ are decoded within the worst-case-delay under lossy conditions, two additional parity symbols, $P_0^e[4]$ and $P_1^e[4]$, are sent in channel packets $X[6]$ and $X[7]$ respectively. The quantities $P_0^e[4]$ and $P_1^e[4]$ are defined to be parity symbols of a $[4, 2]$ systematic MDS code applied to $(S_0[4], S_1[4])$.

The lossless-delay constraint is satisfied for $i \in \{3, \ldots, 9\}$, since each symbol, $S_j[i]$ for $j \in \{0, \ldots, 5\}$, is transmitted as part of either $X[i]$ or $X[i+1]$. Furthermore, in the presence of packet loss, all symbols of all message packets besides $(S_0[4], S_1[4])$ are recovered within 3 time slots based on the properties of the underlying SBC. Since at most one of $((S_0[4], S_1[4]), (P_0^e[4], P_1^e[4]))$ is lost, $S_0[4]$ and $S_1[4]$ are decoded by $X[7]$. Hence, the construction in the toy example satisfies the lossless-delay constraint and the worst-case-delay constraint over the $C(a, b, w)$ channel.

B. General Construction

We now discuss the details of the proposed construction for valid inputs (τ, a, b, w, τ_L) and an arbitrary sequence of message packets $S[0], \ldots, S[t]$. The encoding scheme is systematic, so we will divide the description into two parts: First, we will detail how the symbols of the message packets are distributed over the channel packets. Second, we will explain how parity symbols are created and transmitted.

During time slot $i \in \{0, \ldots, t-\tau\}$, let $d_i = \frac{k_i}{\tau - a + 1}$. As in the existing streaming code constructions for message packets of a fixed size, we assume that each message packet has a size which is divisible by $(\tau - a + 1)$. The value $(\tau - a + 1)$ is typically negligible compared to the sizes of the message packets.[9] Hence, it has negligible effect on the rate of the construction. The first $d_i(\tau - a - \tau_L + 1)$ symbols of $S[i]$ are sent in channel packet $X[i]$. For each $j \in \{\tau - a - \tau_L + 1, \ldots, \tau - a\}$, the next d_i symbols, namely $(S_{jd_i}[i], \ldots, S_{(j+1)d_i - 1}[i])$, are sent in channel packet $X[i + (j - \tau + a + \tau_L)]$. During time slots $i \in \{t - \tau + 1, \ldots, t\}$, the message packet $S[i]$ is known to consist of 0 symbols by the receiver. Thus, no additional symbols are sent in these packets.

Parity symbols will be generated as part of blocks of the SBC and a $[2\tau - 2b + a + 1, \tau - b + 1]$ systematic MDS code. We now explain how blocks of the SBC corresponding to message packet $S[i]$ are created for $i \in \{0, \ldots, t - \tau\}$.

[9]For example, consider a 2000 kbps video call recorded at 60 frames per second with an RTT of 150ms and a maximum tolerable latency of 150 ms. The maximum possible setting for τ is 6. As such, $(\tau - a - 1) \leq 6$. Consequently, $(\tau - a + 1)$ is 0.14% of the average size of a message packet.

In total, d_i blocks of the SBC are created. These blocks will encode systematic symbols of channel packets $X[i-(b-a)],\ldots,X[i+\tau-b]$ into parity symbols, which are then sent in channel packets $X[i+\tau-b+1],\ldots,X[i+\tau]$. Finally, for the remaining values of i, namely $i>(t-\tau)$, no blocks of the SBC are created.

Systematic symbols of various message packets (or unique padding symbols) occupy the first $(\tau-a+1)$ positions of the blocks of the SBC which are placed in channel packets $X[i-(b-a)],\ldots,X[i+\tau-b]$. Parity symbols are sent in the final b positions (i.e., in channel packets $X[i+\tau-b+1],\ldots,X[i+\tau]$). The symbols $S_{(\tau-a-\tau_L)d_i}[i],\ldots,S_{k_i-1}[i]$ of $S[i]$ are distributed evenly over $X[i],\ldots,X[i+\tau_L]$ and will occupy the positions of the blocks of the SBC for the respective channel packets. There are $(\tau-a-\tau_L)$ remaining positions of each block of the SBC, namely in $X[j]$ for $j\in\{i-(b-a),\ldots,i-1,i+\tau_L+1,\ldots,i+\tau-b\}$. These positions of the SBC are filled by symbols of the corresponding message packets (i.e., block position j is filled with the next symbol of $S[j]$ sent in channel packet $X[j]$ which has not already been allocated to blocks of the SBC). If no such symbols are available to fill any position of the SBC, the empty position is treated as a unique padding symbol.

Specifically, for each $l\in\{0,\ldots,d_i-1\}$, there are b parity symbols of the block of the SBC: $(P_{lb}[i],\ldots,P_{(l+1)b-1}[i])$. They will be transmitted over channel packets $X[i+\tau-b+1],\ldots,X[i+\tau]$ respectively. The lth block consists of

$$\begin{aligned}&\langle S_{z_{i,l,i-(b-a)}}[i-(b-a)],\ldots,S_{z_{i,l,i-1}}[i-1],\\&S_{(\tau-a-\tau_L)d_i+l}[i],\ldots,S_{(\tau-a)d_i+l}[i],\\&S_{z_{i,l,i+\tau_L+1}}[i+\tau_L+1],\ldots,S_{z_{i,l,i+\tau-b}}[i+\tau-b],\\&P_{lb}[i],\ldots,P_{(l+1)b-1}[i]\rangle,\end{aligned}$$

where $S_{z_{i,l,h}}[h]$ for $h\in\{i-(b-a),\ldots,i-1,i+\tau_L+1,\ldots,i+\tau-b\}$ represents the first symbol from $S[h]$ which has not yet been placed in a block of the SBC. If no such symbol exists, then $S_{z_{i,l,h}]}[h]$ is defined to be a unique padding symbol (which is not transmitted). For $j\in\{0,\ldots,b-1\}$, $P_{lb+j}[i]$ is computed during time slot $(i+\tau-b)$ and sent during time slot $(i+\tau-b+j)$.

Finally, we explain how the remaining symbols of the message packets which are not placed in blocks of the SBC are encoded as part of a $[2\tau-2b+a+1,\tau-b+1]$ systematic MDS code. These symbols are $S_{r_i+1}[i],\ldots,S_{(\tau-a-\tau_L)d_i-1}[i]$, where $r_i=\min\left((\tau-a-\tau_L)d_i,\sum_{j\in J_i}d_j\right)$, for $J_i=\{i-\tau+b,\ldots,i-\tau_L-1,i+1,\ldots,i+(b-a)\}$ (i.e., the set of time slots for which there is at least one open position of blocks of the SBC in channel packet $X[i]$). Then $S_{r_i+1}[i],\ldots,S_{(\tau-a-\tau_L)d_i-1}[i]$ are referred to as the "excess" symbols $S_e[i]$. Excess symbols arise if and only if $(r_i+1)<(\tau-a-\tau_L)d_i$. The quantity of excess symbols is defined as $e_i=|S_e[i]|$. We zero pad $S_e[i]$ to ensure divisibility by $(\tau-b+1)$ (although the padding symbols are not transmitted). The extra symbols are referred to as $pad[i]$. Hence, $(\tau-b+1)|(e_i+pad[i])$. Symbols of $S[i]$ are only included in blocks of the SBC corresponding to time slots at or before $(i+b-a)$. Hence, $S_e[i]$ is determined during time slot $(i+b-a)$. For each $j\in\{0,\ldots,\frac{e_i-1}{\tau-b+1}\}$, the symbols $S^e_{j(\tau-b+1)}[i],\ldots,S^e_{(j+1)(\tau-b+1)-1}[i]$ are encoded as part of a $[2\tau-2b+a+1,\tau-b+1]$ systematic MDS code to create parity symbols $P^e_{j(\tau-b+a)j}[i],\ldots,P^e_{(j+1)(\tau-b+a)-1}[i]$. Then these parity symbols are sent in channel packets $X[i+b-a+1],\ldots,X[i+\tau]$ respectively. Finally, for the remaining values of i, namely $i>(t-\tau)$, there are no symbols of $S[i]$, so there are no symbols of $S^e[i]$.

In short, the symbols $(S_{(\tau-a-\tau_L)d_i}[i],\ldots,S_{k_i-1}[i])$ are encoded as part of the d_i blocks of the SBC corresponding to time slot i. In total, there are r_i open positions in these blocks. Of the remaining symbols, $S_0[i]\ldots,S_{(\tau-a-\tau_L)d_i-1}[i]$, the first r_i are encoded as part of the blocks corresponding to time slots $j\in J_i$. The remaining symbols of $S[i]$ are encoded using as part of blocks of a $[2\tau-2b+a+1,\tau-b+1]$ MDS code.

We now verify that the (τ,a,b,τ_L)-VSC meets the constraints of the proposed streaming model.

Theorem 3: For any valid inputs (τ,a,b,w,τ_L) and sequence of message packets $S[0],\ldots,S[t]$, the (τ,a,b,τ_L)-VSC satisfies the lossless-delay constraint and the worst-case-delay constraint over a $C(a,b,w)$ channel.

Proof: We first show that the (τ,a,b,τ_L)-VSC meets the lossless-delay constraint. Recall that the symbols of $S_0[i],\ldots,S_{(\tau-a-\tau_L)d_i-1}[i]$ are sent in $X[i]$, and the symbols of $S_{(\tau-a-\tau_L)d_i}[i],\ldots,S_{k_i-1}[i]$ are transmitted evenly over $X[i+1],\ldots,X[i+\tau_L]$. Thus, under a lossless transmission, $S[i]$ is received within delay τ_L.

In order to show that the (τ,a,b,τ_L)-VSC meets the worst-case-delay constraint over a $C(a,b,w)$ channel, we show that every symbol s' of $S[i]$ is decoded within delay τ over the channel.

Case 1: s' is a part of $S_{(\tau-a-\tau_L)d_i}[i],\ldots,S_{k_i-1}[i]$. Therefore, it is placed in a block of the SBC in which the final parity symbol is sent in $X[i+\tau]$. Hence, by the properties of SBC, s' is recovered over a $C(a,b,w)$ channel within τ time slots.

Case 2: s' is a part of $S_0[i],\ldots,S_{(\tau-a-\tau_L)d_i-1}[i]$ and is placed in a block of the SBC corresponding to a different message packet. Hence, s' is transmitted within $X[i]$ and, by the properties of SBC, s' is recovered over the $C(a,b,w)$ channel within τ time slots.

Case 3: s' is part of $S_e[i]$. Either $X[i]$ is received or at least $(\tau-b+1)$ of $X[i+b-a+1],\ldots,X[i+\tau]$ are received. Therefore, either s' is received as part of $X[i]$, or sufficiently many code symbols are received from a $[2\tau-2b+a+1,\tau-b+1]$ systematic MDS codeword to recover s' within τ time slots. ■

Next, in Section VII, we analyze the rate of the (τ,a,b,τ_L)-VSC. Later, in Section VIII, we empirically evaluate the rate of the (τ,a,b,τ_L)-VSC when the sizes of message packets correspond to a video trace.

C. Field Size Requirement

The field size requirement for the (τ,a,b,τ_L)-VSC must meet two constraints: (1) the field size requirement for the underlying SBC, and (2) the field size requirement for the

$[2\tau - 2b + a + 1, \tau - b + 1]$ systematic MDS code. The field size requirement for the SBC from [9] is $O(\tau^2)$. The field size requirement for a $[2\tau - 2b + a + 1, \tau - b + 1]$ systematic MDS is at most $(2\tau - 2b + a + 1)$. Overall, the former field size requirement dominates the latter, leading to a field size requirement of $O(\tau^2)$.

VII. Analysis of the Rate of the (τ, a, b, τ_L)-VSC

As with any code construction which adapts to the specific message packet size sequence, the rate for the construction proposed in Section VI depends on the message packet size sequence. For concreteness, we will analyze the class of inputs where the size of each message packet is drawn independently from any discrete distribution D with finite support. For any valid inputs (τ, a, b, w, τ_L), we characterize the asymptotic rate of the (τ, a, b, τ_L)-VSC as the length of the message packet size sequence becomes sufficiently long in terms of the mean of D in Theorem 4. We then provide an easily-computable lower bound on the rate of the (τ, a, b, τ_L)-VSC in terms of the mean and variance of D in Lemma 9.

We begin with some notation. A distribution D with mean μ and maximum value d is called an *admissible distribution* if it (1) has finite support, (2) consists only of non-negative elements, and (3) consists solely of elements divisible by $(\tau - a + 1)$. Recall (from Section IV-A) that the rate is never more than $R^{(U)} = \frac{\tau - a + 1}{\tau + b - a + 1}$. Furthermore, recall from Section VI-B that $S_e[i]$ is used to denote the excess symbols in the VSC construction for $S[i]$ for $i \in \{0, \dots, t - \tau\}$. The excess symbols are the symbols that are not placed in blocks of the SBC. Recall that e_i denotes the number of such symbols, and $pad[i]$ refers to the number of padding symbols to ensure that $(\tau - b + 1) | (e_i + pad[i])$. Let $e_i^c = (e_i + pad[i])$ denote the "padded excess," and $e^c = \mathbb{E}[e_i^c]$ be called the expected padded excess. The value e_i^c is uniquely determined by $(k_{\max(0, i-\tau)}, \dots, k_{i+\tau})$. Therefore, by the independence of the sizes of the message packets, the e_i^c are identically distributed for $i \in \{2\tau, \dots, t - 2\tau\}$.

Next, we identify the expected number of symbols in a channel packet under the $(\tau, a, b, w, \tau_L)-$VSC.

Lemma 8: For any valid inputs (τ, a, b, w, τ_L), and sequence of $t > 4\tau$ message packets whose sizes are drawn independently from an admissible distribution D with mean μ and maximum value d, for all $i \in \{2\tau, \dots, t - 2\tau\}$, the expected number of symbols in channel packet $X[i]$ is given by

$$\mu_c = \mathbb{E}[n_i] = \left(\frac{\mu}{R^{(U)}} + e^c \frac{\tau - b + a}{\tau - b + 1} \right). \tag{7}$$

Proof Sketch: Follows from the definition of the (τ, a, b, τ_L)-VSC. The proof involves counting the number of symbols of each $X[i]$, for $i \in [t]$, corresponding to the (a) symbols of $S[i]$, (b) symbols of the previous τ_L message packets, (c) parity symbols corresponding to blocks of the SBC, and (d) parity symbols corresponding to the padded excess. A complete proof is included in Appendix E. ■

Using Lemma 8, we identify the asymptotic rate in terms of Eq. (7) as follows.

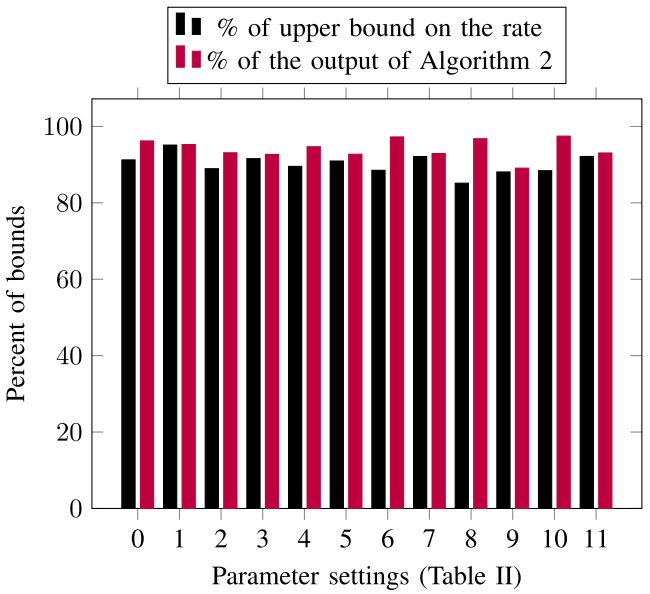

Fig. 10. The rate for the (τ, a, b, τ_L)-VSC as a percent of the upper bound on the rate for the online setting computed by Algorithm 1 and the value computed by Algorithm 2 for a live video trace over the parameter settings included in Table II.

Theorem 4: Consider any valid inputs (τ, a, b, w, τ_L), δ, $\epsilon > 0$, and sequence of $t \geq \left(\frac{36\tau bd\sqrt{2}}{\epsilon \mu_c} \sqrt{\ln\left(\frac{4\tau}{\delta}\right)} + \frac{180db\tau}{\mu_c \epsilon} \right)$ message packets whose sizes are drawn independently from an admissible distribution D with mean μ and maximum value d. With probability at least $(1 - \delta)$, the rate of the (τ, a, b, τ_L)-VSC construction, R_t, satisfies $|R_t - \frac{\mu}{\mu_c}| < \epsilon$.

Proof Sketch: Recall from Eq. (1) that, the rate, R_t, equals $\frac{\sum_{i=0}^{t} k_i}{\sum_{i=0}^{t} n_i} = \frac{\frac{1}{t+1}\sum_{i=0}^{t} k_i}{\frac{1}{t+1}\sum_{i=0}^{t} n_i}$. At a high level, the proof follows by analyzing the rate of convergence of (a) the mean number of symbols of the message packets and (b) the mean number of symbols of the channel packets and then applying the union bound. We show (a) and (b) with simple applications of concentration bounds. A complete proof is included in Appendix F. ■

Theorem 4 establishes the rate of the (τ, a, b, τ_L)-VSC for a randomly generated message packet size sequence as a function of the expected padded excess, e^c, and the mean size of message packets, μ. In turn, e^c is a function of the distribution D. For certain distributions, computing the exact value of e^c may require a high computational complexity. In Lemma 9, we provide an upper bound on expected excess. Furthermore, the expected padding is upper bounded by $(\tau - b + a - 1)$—a term which is typically several orders of magnitude smaller than the average size of a message packet. Combining these two bounds provides an easily-computable upper bound on e^c and a corresponding lower bound on the rate for (τ, a, b, τ_L)-VSC.

Lemma 9: Consider valid inputs (τ, a, b, w, τ_L), $\epsilon, \delta > 0$, and any sequence of $\left(\frac{18bd\sqrt{2}}{\epsilon \mu_c} \sqrt{\ln\left(\frac{4\tau}{\delta}\right)} + \frac{180db\tau}{\mu_c \epsilon} \right)$ message packets whose sizes are drawn independently from any admissible distribution D with maximum size d, mean μ, and variance σ^2. With probability at least $(1 - \delta)$ the rate, R_t,

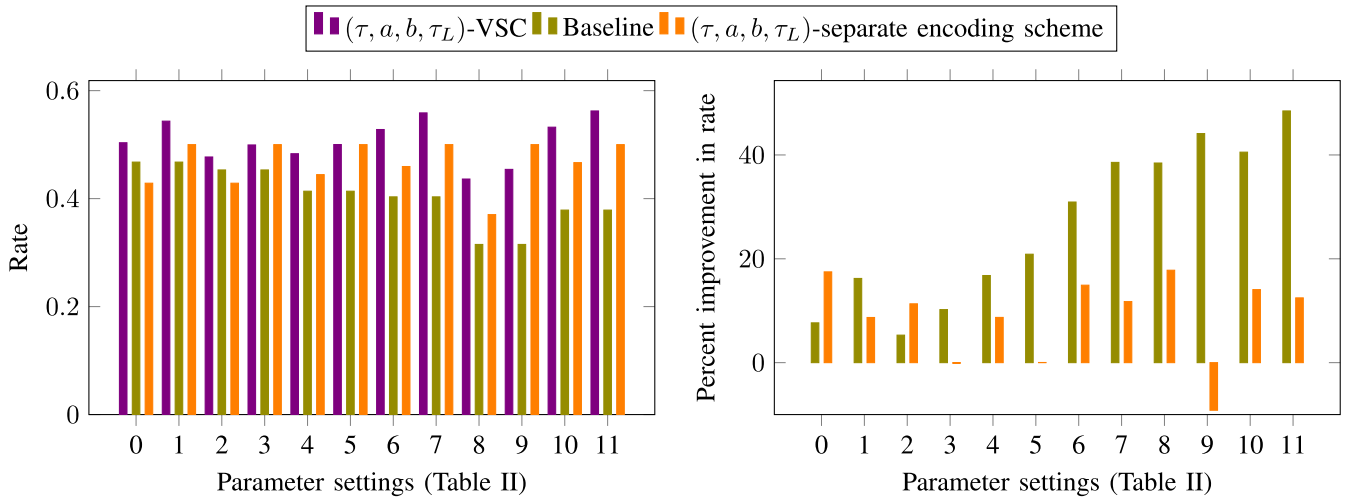

Fig. 11. Comparison of the (τ, a, b, τ_L)-VSC to a baseline streaming code and the (τ, a, b, τ_L)-separate encoding scheme. The baseline is a streaming code designed for the setting where message packets have fixed sizes (with necessary adjustments to account for the variability of the sizes of message packets). The comparison ranges over the parameter settings listed in Table II and represents: (a) rate and (b) percent improvement in rate provided by VSC over the other two schemes.

TABLE II
THE PARAMETER SETTINGS USED IN THE EMPIRICAL EVALUATION OF THE (τ, a, b, τ_L)-VSC

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
τ	5	5	7	7	12	12	40	40	40	40	50	50
a	2	2	2	2	2	2	4	4	8	8	4	4
b	3	3	5	5	9	9	24	24	31	31	30	30
τ_L	0	2	0	2	0	3	0	16	0	9	0	20

of the (τ, a, b, τ_L)-VSC construction is at least

$$R_t \geq \frac{R^{(U)}}{1 + \frac{\tau-b+a}{\tau-b+1} R^{(U)} \left(\frac{\sigma}{\sqrt{2}\mu} + \frac{\tau-b}{\mu} \right)} - \epsilon. \tag{8}$$

Proof Sketch: Follows from using Jensen's inequality to show an upper bound on the expected excess. The upper bound is then applied to Theorem 4. A complete proof is included in Appendix G. ■

As expected, as $\frac{\sigma}{\mu}$ decreases, the above quantity approaches the upper bound on the rate of $R^{(U)}$.

VIII. EMPIRICAL EVALUATION OF THE (τ, a, b, τ_L)-VSC

The optimal rate under the proposed streaming model depends on the specific message packet size sequence. Similarly, the rate of a streaming code construction can depend on the specific message packet size sequence. In order to fairly evaluate the effectiveness of the (τ, a, b, τ_L)-VSC, it is therefore necessary to empirically evaluate the code over a realistic message packet size sequence. We do so in this section by assessing the rate of the (τ, a, b, τ_L)-VSC over several representative parameter settings for a real-world live video trace.

In the simulation, the proposed (τ, a, b, τ_L)-VSC construction and a baseline are evaluated over a live video trace uploaded to Facebook Live through Open Broadcast Software (OBS) [30] at a bitrate of 2000 Kbps. Table II shows the parameter settings used in simulations. The parameter settings included are either considered by existing streaming code literature or are used to reflect the low-latency constraints of live communication. For all parameter settings, the maximal and minimal values of the lossless-delay, namely $\tau_L = (\tau - b)$ and $\tau_L = 0$, are included to illustrate the trade-off between the rate and the decoding delay under lossless transmission.

The empirical comparison of the performance of (τ, a, b, τ_L)-VSC to the upper bound on the rate computed by Algorithm 1 and to the value computed by Algorithm 2 is demonstrated in Figure 10.[10] Recall that Algorithm 2 outputs the best possible rate for coding schemes in the *offline* setting which encode symbols using a combination of (a) blocks of the SBC and (b) the (τ, a, b, τ_L)-separate encoding scheme. Therefore, the output of Algorithm 2 is an upper bound on the rate of the (τ, a, b, τ_L)-VSC, which uses the same combination of encoding schemes but operates in an online setting. Over the considered parameter settings, the *(τ, a, b, τ_L)-VSC exhibits a rate of 85.1% to 95.1% of the upper bound on the rate computed by Algorithm 1, with an average of 90.1%*. Furthermore, the (τ, a, b, τ_L)-VSC does slightly better relative to the upper bound when $\tau_L = (\tau - b)$ than it does when $\tau_L = 0$. This property is in addition to the fact that the rate for the (τ, a, b, τ_L)-VSC improves as τ_L increases, which we will discuss subsequently. Over the considered parameter settings, the (τ, a, b, τ_L)-VSC exhibits a rate of 89.1% to 97.5% of the value computed by Algorithm 2, with an average of 94.3%.

The closeness of the comparisons to Algorithms 1 and 2 reflects the (small) penalty on the rate for considering a restricted class of coding schemes. The gap between the rate of the (τ, a, b, τ_L)-VSC and the value computed by Algorithm 2 reflects an upper bound on the cost of operating in an online setting for the restricted class of coding schemes.

[10] [31] and [32] were used in the computation of the LP solution for the modified version of Algorithm 1 using an LP relaxation.

Next, the (τ, a, b, τ_L)-VSC is compared to the (τ, a, b, τ_L)-separate encoding scheme and a baseline which is an adapted version of existing streaming codes that are rate-optimal when all message packets have the same fixed size. The changes to the baseline are mandatory modifications to account for the variability of the sizes of the message packets. A detailed description of the baseline is provided in Appendix H. Figure 11a compares the rates of the three code constructions directly, while Figure 11b shows the percent improvement of the (τ, a, b, τ_L)-VSC over the baseline and the (τ, a, b, τ_L)-separate encoding scheme. The (τ, a, b, τ_L)-VSC outperforms the (τ, a, b, τ_L)-separate encoding scheme over all parameter settings except for $(\tau = 40, a = 8, b = 31, \tau_L = 9)$, where the rate of the (τ, a, b, τ_L)-separate encoding scheme is 97% of an upper bound on the rate. Moreover, the (τ, a, b, τ_L)-*VSC improves the rate between* 5% *and* 48% *over the baseline.* Increasing τ_L leads to a higher rate for the (τ, a, b, τ_L)-VSC. Such a result is expected, as the distribution of each message packet over multiple channel packets under the (τ, a, b, τ_L)-VSC alleviates the variability in the sizes of the channel packets. Thus, fewer parity symbols are needed to recover from packet losses.

IX. Conclusion

In this work, we present a model for streaming codes that captures the requirements of live streaming applications that send sequences of messages of varying sizes, such as videoconferencing. We show that variability in the sizes of messages leads to several unique challenges for streaming codes. Examples include a new trade-off between the rate and the decoding delay under lossless transmission and the achievable rate being a function of the message size sequence. We present integer programming-based algorithms to compute upper and lower bounds on the rate of streaming codes for any given message size sequence. We show that they significantly improve general bounds that do not take the sequence into account. We also present an explicit construction for streaming codes under varying message sizes. We theoretically prove that the construction meets the latency requirements for any message size sequence under a sliding window channel model with arbitrary and burst losses. We also empirically evaluate the proposed construction on a Facebook Live video trace for a wide variety of channel settings. We show that the proposed construction attains a rate of $85\% - 95\%$ of an upper bound and 5%–48% higher than naively using the existing streaming codes.

Videoconferencing applications are becoming the mainstay of communication over the Internet. Streaming codes that can support varying message sizes well can help such applications improve the quality-of-service. Many questions remain open for streaming codes with varying message sizes. For instance, how to convert high-rate offline streaming codes into online ones. In addition, several additional constraints explored under the fixed-size setting, such as correlations between the sizes of the messages, unequal error protection across messages, and varying latency constraints on different messages, are interesting future directions to explore under the variable-size setting. Several such constraints are equivalent to restricted variants of the proposed model. As such, the fundamental principles introduced in this work are well-suited for designing and evaluating new communication schemes for such restricted models.

Appendix A
Proof of Lemma 4

Proof: The optimal rate of streaming codes that satisfy the lossless-delay and worst-case-delay constraint over the $C(a, b, w)$ model is no more than $R^{(L)}$, as was shown in Lemma 3. In order to show that $R^{(L)}$ is the greatest lower bound on rate, it suffices to show for at least one message packet size sequence that the optimal rate is at most $R^{(L)}$. We do so in this proof.

Consider the following length $(\tau + 2b)$ message packet size sequence: $k_i = 0$ for $i \in \{0, \ldots, b-1, b+1, \ldots, \tau+2b-1\}$ and $k_b = (\tau_L+1)(\tau - b + 1)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min\left((\tau_L+1) \mod b, a\right)\right)(\tau - a + 1)$.

The proof is divided into two cases to match the two cases of the construction. In both cases, we show that meeting the lossless-delay constraint and worst-case-delay constraint over the $C(a, b, w)$ requires sending at least $\frac{k_b}{R^{(L)}}$ symbols. Hence, the rate of any streaming code that satisfies the lossless-delay and worst-case-delay constraint over the $C(a, b, w)$ model is at most $R^{(L)}$ for the considered message packet size sequence. The proof will make use of the fact that $S[b]$ cannot be decoded unless at least k_b symbols are received.

Case 1: $\tau_L < (a-1)$.

At least k_b symbols must be sent over $X[b], \ldots, X[b+\tau_L]$ to meet the lossless-delay constraint. At least k_b symbols must be sent over $X[2b], \ldots, X[b+\tau]$ to meet the worst-case-delay constraint when $X[b], \ldots, X[2b-1]$ are lost.

The average number of symbols per channel packet over $X[b], \ldots, X[b+\tau_L]$ is at least $\frac{k_b}{\tau_L+1}$. The average number of symbols per channel packet over $X[2b], \ldots, X[b+\tau]$ is at least $\frac{k_b}{\tau-b+1}$.

By definition, $(\tau_L + 1) \leq (\tau - b + 1)$.

When $a \leq (\tau_L + 1 + \tau + 1 - b)$, a arbitrary losses can result in a loss of (1) k_b symbols in $X[b], \ldots, X[b+\tau_L]$, and (2) at least $\frac{k_b}{\tau-b+1}(a - \tau_L - 1)$ symbols in $(a - \tau_L - 1)$ adversarially chosen channel packets among $X[2b], \ldots, X[b+\tau]$. Thus, at least $\frac{k_b}{\tau-b+1}(a - \tau_L - 1)$ additional symbols must be sent. Let us combine these $\frac{k_b}{\tau-b+1}(a - \tau_L - 1)$ symbols with the at least k_b symbols sent in $X[b], \ldots, X[b+\tau_L]$ and at least k_b symbols sent in $X[2b], \ldots, X[b+\tau]$. In total, at least $k_b\left(2 + \frac{a-\tau_L-1}{\tau-b+1}\right) = \frac{k_b}{R^{(L)}}$ symbols are transmitted.

When $a > (\tau_L + 1 + \tau + 1 - b)$, $(b - \tau_L - 1) > (\tau - a + 1)$. Hence, due to arbitrary losses, $X[b], \ldots, X[b+\tau_L], X[2b], \ldots, X[b+\tau]$ may all be lost. Thus, it is possible that only $(\tau - a + 1)$ arbitrary packets of $X[b+\tau_L+1], \ldots, X[2b-1]$ to be received.

As such, any $(\tau + 1 - a)$ channel packets of $X[b+\tau_L+1], \ldots, X[2b-1]$ must contain at least k_b symbols. Therefore, the channel packets $X[b+\tau_L+1], \ldots, X[2b-1]$ contain on average at least $\frac{k_b}{\tau-a+1}$ symbols. At least $(b - \tau_L - 1)\frac{k_b}{\tau-a+1}$ symbols are sent over $X[b+\tau_L+1], \ldots, X[2b-1]$. In total, at least $k_b\left(2 + \frac{b-\tau_L-1}{\tau-a+1}\right)$ symbols are transmitted.

Case 2: $\tau_L \geq (a-1)$.

Sub-case 1: either $((\tau_L+1) \mod b) \in \{0\} \cup \{a, \ldots, b-1\}$ or $((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) \geq a$.

At least k_b symbols must be sent over $X[b], \ldots, X[b+\tau_L]$ to satisfy the lossless-delay constraint. We will show in several sub-cases that at least $\frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min((\tau_L+1) \mod b, a)} k_b$ symbols could be lost by time slot $(b+\tau_L)$. At least k_b symbols must be received. Therefore, at least $k_b \frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min((\tau_L+1) \mod b, a)}$ parity symbols are transmitted. In total, $\frac{k_b}{R^{(L)}}$ symbols are sent. A final sub-case will handle the remaining parameter settings and follows from showing the correctness of IP-based construction 1.

Sub-sub-case $((\tau_L+1) \mod b \geq a)$:

All channel packets of one of

$$\begin{aligned}
&\left(X[b], \ldots, X[2b-1]\right), \ldots, \\
&\left(X\left[b + \left(\left\lfloor \frac{\tau_L}{b} \right\rfloor - 1\right)b\right], \ldots, X\left[b + \left\lfloor \frac{\tau_L}{b} \right\rfloor b - 1\right]\right), \\
&\left(X\left[b + \left\lfloor \frac{\tau_L}{b} \right\rfloor b\right], \ldots, X[b+\tau_L]\right)
\end{aligned}$$

could be dropped as part of a single burst. There are $\left(\left\lfloor \frac{\tau_L}{b} \right\rfloor + 1\right)$ quantities, and at least k_b symbols are sent over $X[b], \ldots, X[b+\tau_L]$. By the pigeonhole principle, at least one such quantity contains at least $\frac{k_b}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1} = \frac{a k_b}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min((\tau_L+1) \mod b, a)}$ symbols.

Sub-sub-case $((\tau_L+1) \mod b \equiv 0)$:

All channel packets of one of

$$\begin{aligned}
&\left(X[b], \ldots, X[2b-1]\right), \ldots, \\
&\left(X\left[b + \left\lfloor \frac{\tau_L}{b} \right\rfloor b\right], \ldots, X\left[b + \left(\left\lfloor \frac{\tau_L}{b} \right\rfloor + 1\right)b - 1\right]\right)
\end{aligned}$$

could be dropped as part of a single burst. There are $\left(\left\lfloor \frac{\tau_L}{b} \right\rfloor + 1\right) = \frac{\tau_L+1}{b}$ such quantities. By the pigeonhole principle, at least one contains at least $\frac{k_b}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor} = \frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min((\tau_L+1) \mod b, a)} k_b$ symbols.

Sub-sub-case $0 < (\tau_L+1) \bmod b < a$ and $((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) \geq a$:

Note that $(\tau_L+1 \neq b)$ by the sub case. Also, (τ_L+1) must be strictly greater than b in accordance with $(\tau_L \geq a-1)$.

Let $e = ((\tau_L+1) \mod b)$. If any b consecutive channel packets of $X[b], \ldots, X[b+\tau_L]$ contains at least $k_b \frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e}$ symbols the proof is immediate, since all b of them could be lost. Otherwise, let

$$X' = \bigcup_{i=0}^{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor} \left\{ (X[b+ib], \ldots, X[b+ib+e-1]) \right\}.$$

Consider any $(X[j], \ldots, X[j+e-1]) \in X'$. The remaining channel packets of $X[b], \ldots, X[b+\tau_L]$ can be partitioned into $\left\lfloor \frac{\tau_L+1}{b} \right\rfloor$ groups of b consecutive channel packet. Recall that each group of b consecutive packets contains at most $k_b \frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e}$ symbols. In total, the $\left\lfloor \frac{\tau_L+1}{b} \right\rfloor$ groups contain at most $k_b \frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e} \left\lfloor \frac{\tau_L+1}{b} \right\rfloor$ symbols. In order to satisfy the lossless-delay, at least k_b symbols must be received over $X[b], \ldots, X[b+\tau_L]$. Hence, the total combined number of symbols in $X[j], \ldots, X[j+e-1]$ is at least the following

$$\left(1 - \frac{(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor)a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e}\right) k_b = \frac{e}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e} k_b.$$

There are $e\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) \geq a$ channel packets in X, each of which lies within $X[b], \ldots, X[b+\tau_L]$. In total, these channel packets contain at least $\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) \frac{e}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e} k_b$ symbols. The channel packets in X contain on average at least $\frac{1}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e} k_b$ symbols. In expectation, if a of these channel packets are dropped uniformly at random, at least $\frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e} k_b$ symbols are lost. Thus, at least one choice of a arbitrary channel packet losses results in a total number of lost symbols of at least $\frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + e} k_b$.

In all above sub-cases for $\tau_L \geq (a-1)$, at least $\frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min((\tau_L+1) \mod b, a)} k_b$ symbols could be lost. This necessitates transmitting at least $k_b\left(1 + \frac{a}{\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min((\tau_L+1) \mod b, a)}\right) = \frac{k_b}{R^{(L)}}$ symbols.

Sub-case 2: $((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor + 1\right) < a$.

For this case, recall that the number of symbols sent per channel packet is determined by IP-based construction 1. Each channel packet contains a non-negative number of symbols, as is imposed by constraint #1. At least k_i symbols must be received within the first τ_L channel packets due to the lossless-delay constraint. This requirement is imposed with constraint #2. For any loss pattern under the $C(a,b,w)$ channel, the total number of symbols over the received channel packets must be at least k_i. This requirement is imposed with constraints #3 and #4 of the integer program. All constraints of the integer program must be met by any code construction. The integer program solves for the minimum number of symbols to be sent subject to these three constraints. Hence, the optimal rate is attained. ■

APPENDIX B
PROOF OF LEMMA 2

Proof: Consider the encoding for a message packet $S[i]$ for $i \in \{0, \ldots, t\}$. When $i > (t-\tau)$, $k_i = 0$, and $S[i]$ is automatically known by the receiver. Otherwise, the symbols message packet $S[i]$ are sent over $X[i], \ldots, X[i+\tau_L]$, thereby satisfying the lossless-delay constraint.

The proof that the worst-case-delay constraint is satisfied over the $C(a,b,w)$ channel is divided into two cases depending on whether $\tau_L \geq (a-1)$.

Case 1: $\tau_L < (a-1)$.

If $a \leq (\tau_L + 2 + \tau - b)$, let $\eta = (\tau_L+1)(\tau-b+1)$ and $\eta' = (\tau_L+1)(\tau-b+a-\tau_L)$. Otherwise, let $\eta = (\tau_L+1)(\tau-b+1)(\tau-a+1)$ and $\eta' = \eta + (\tau_L+1)(\tau-b+1)(b-\tau_L-1)$. The message packet, $S[i]$, is partitioned evenly into sets of η symbols. For an arbitrary such set, $\{c_0, \ldots, c_{\eta-1}\}$, we verify all symbols are decoded within τ time slots. The set is encoded as part of a $[\eta+\eta', \eta]$ systematic MDS code. Let $(c'_0, \ldots, c'_{\eta+\eta'})$ be the code symbols corresponding to $\{c_0, \ldots, c_\eta\}$. It suffices to show that at least η symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$ are received by time slot $(i+\tau)$.

We note that $(\tau_L + b \leq \tau) \implies (\tau - b + 1 \geq \tau_L + 1)$. Furthermore, $X[i], \ldots, X[i+\tau_L]$ and $X[i+b], \ldots, X[i+\tau]$ each contain η symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$. The number of symbols per channel packet in $X[i], \ldots, X[i+\tau_L]$ is $\frac{\eta}{\tau_L+1}$. The number of symbols per channel packet in $X[i+b], \ldots, X[i+\tau]$ is $\frac{\eta}{\tau-b+1}$. If $a \leq (\tau_L + 2 + \tau - b)$, the number of symbols per channel packet in $X[i+\tau_L+1], \ldots, X[b-1]$ is either 0 or $\frac{\eta}{\tau-b+1}$. If $a > (\tau_L + 2 + \tau - b)$, the number of symbols per channel packet in $X[i+\tau_L+1], \ldots, X[b-1]$ is $\frac{\eta}{\tau-a+1}$. Finally, $\frac{\eta}{\tau-a+1} \leq \frac{\eta}{\tau-b+1} \leq \frac{\eta}{\tau_L+1}$

Burst losses: For all bursts of length b starting during or after time slot $(i+\tau_L+1)$, η symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$ are received via $X[i], \ldots, X[i+\tau_L]$. Hence, decoding within a delay of τ follows immediately. For any burst starting in channel packet $X[i+j]$ for $j \in \{0, \ldots, \tau_L\}$, $(\tau_L - j + 1)$ channel packets $X[i], \ldots, X[i+j-1]$ are received, each of which contain $\frac{\eta}{\tau_L+1}$ symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$. Moreover, channel packets $X[i+j+b], \ldots, X[i+\tau]$ are received, each of which contain $\frac{\eta}{\tau-b+1}$ symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$. The fewest symbols are received when $j = 0$, in which case exactly η symbols are received. Any η symbols are sufficient for decoding.

Arbitrary losses: The maximum number of symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$ are lost when the a largest channel packets are lost, such as when

$$X[i], \ldots, X[i+\tau_L], X[i+\tau-a+\tau_L+2], \ldots, X[i+\tau]$$

are lost. If $a \leq (\tau_L+2+\tau-b)$, the $(\tau-b+1)$ channel packets $X[i+b-a+\tau_L+1], \ldots, X[i+\tau-a+\tau_L+1]$ are received, each of which contains $\frac{\eta}{\tau-b+1}$ symbols. If $a > (\tau_L+2+\tau-b)$, then the $(\tau-a+1)$ channel packets $X[i+\tau_L+1], \ldots, X[i+\tau-a+\tau_L+1]$ are received, each of which contains $\frac{\eta}{\tau-a+1}$ symbols. Therefore, at least η symbols of $(c'_0, \ldots, c'_{\eta+\eta'})$ are received within a delay of τ, enabling decoding.

Case 2: $\tau_L \geq (a-1)$.

Sub-case 1: $((\tau_L+1) \mod b) \in \{0\} \cup \{a, \ldots, b-1\})$ or $(0 < (\tau_L+1) \mod b < a$ and $((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+}{b} \right\rfloor + 1\right) \geq a)$.

Let $\zeta = \left(\left\lfloor \frac{\tau_L+1}{b} \right\rfloor a + \min\left((\tau_L+1) \mod b, a\right)\right)$. Recall that each $S[i]$, for $i \in \{0, \ldots, t-\tau\}$, is partitioned into sets of ζ symbols. Every such set is encoded separately as part of a $[\zeta+a, \zeta]$ systematic MDS code. We verify for an arbitrary such set, $\{c_0, \ldots, c_{\zeta-1}\}$, with corresponding code symbols, $(c'_0, \ldots, c'_{\zeta+a})$, that at least ζ code symbols are received within τ time slots. Any ζ symbols suffice to decode $\{c_0, \ldots, c_{\zeta-1}\}$.

Every burst of b consecutive channel packets eliminates at least $(b-a)$ channel packets which contain no symbols of $(c'_0, \ldots, c'_{\zeta+a})$. Thus, at most a symbols of $(c'_0, \ldots, c'_{\zeta+a})$ are lost. For any sequence of a arbitrary losses, at least ζ of the symbols of $(c'_0, \ldots, c'_{\zeta+a})$ are received within τ time slots. For either loss pattern, recovery with a delay of τ time slots follows immediately by properties of the $[\zeta+a, \zeta]$ systematic MDS code.

Sub-case 2: $(0 < (\tau_L+1) \mod b < a$ and $((\tau_L+1) \mod b)\left(\left\lfloor \frac{\tau_L+1b}{b} \right\rfloor + 1\right) < a$.

Recall that each $S[i]$ is encoded according to the outputs of IP-based construction 1. Constraints #3 and #4 of IP-based construction 1 ensure that least k_i symbols of an $[n_i^{(*)}, k_i]$ MDS code are received by time slot $(i+\tau)$. Hence, $S[i]$ is recovered within τ time slots by the MDS property. ■

APPENDIX C
PROOF OF THEOREM 1

Proof: We will show that each constraint corresponds to a valid requirement to impose on coding schemes. This ensures that the solution is a lower bound on the number of symbols that must be sent. As there are $\sum_{i=0}^{t} k_i$ symbols of the message packets, the output must be an upper bound on the rate.

Before presenting the proof of correctness for the constraints, we will formally define how a channel packet, $X[i]$ for $i \in \{0, \ldots, t\}$, is split into $(X^{(0)}[i], X^{(1)}[i])$. Recall that each symbol of $X[i]$ comprises a linear combination of the symbols of

$$\langle S_0[0], \ldots, S_{k_0-1}[0], \ldots, S_0[i], \ldots, S_{k_i-1}[i] \rangle . \tag{9}$$

The symbols

$$\left\langle X_0[0], \ldots, X_{|X[0]|-1}[0], \ldots, X_0[i], \ldots, X_{|X[i-1]|-1}[i-1] \right\rangle \tag{10}$$

correspond to linear equations over the symbols of Eq. (9) where the linear equations for each $j \in \{0, \ldots, i-1\}$ of $X[j]$ have 0 in positions corresponding to $\left\langle S_0[j+1], \ldots, S_{|k_i|-1}[i] \right\rangle$ due to causality. Next, the symbols of channel packet $X[i]$ are partitioned into $(X^{(0)}[i], X^{(1)}[i])$. Initially, consider $X^{(0)}[i]$ and $X^{(1)}[i]$ as being empty. The symbols of $X[i]$ are labeled as being in either $X^{(0)}[i]$ or $X^{(1)}[i]$ by iterating over $j \in \{0, \ldots, |X[i]-1|\}$ as follows. If the set of linear equations corresponding $X_j[i]$, the symbols of $X^{(0)}[i]$, and the symbols of Eq. (10) are linearly independent, $X_j[i]$ is added to $X^{(0)}[i]$. Otherwise, $X_j[i]$ is added to $X^{(1)}[i]$.

Constraints #1 and #2: For $i \in \{0, \ldots, t-\tau_L\}, j \in \{i-\tau_L, \ldots, i\}$, $|X_i^{(0)}[i+j]|$ reflects a number of symbols sent in channel packet $X[i+j]$ corresponding to message packet $S[i]$, so it is non-negative. For $i < 0$, $i > t$, or $j < 0$, $|X_j^{(0)}[i]|$ is defined to be 0 to handle edge conditions of indexing. For $i \in \{0, \ldots, t+\tau\}$, $|X^{(1)}[i]|$ corresponds to a number of parity symbols sent in channel packet $X[i]$ and similarly is non-negative.

Constraint #3: We will show that any construction satisfying the lossless-delay requirement, even with the relaxations allowed under Algorithm 1, will satisfy constraint #3. We will prove this by induction on $i \in \{0, \ldots, t\}$. In the base case, $i = 0$ and $X_0^{(0)}[0], \ldots, X_0^{(0)}[\tau_L]$ consist of exactly k_0 symbols used to decode $S[0]$ under lossless transmission. In the inductive step, for $i = 1, \ldots, t$, $S[0], \ldots, S[i]$ can be decoded using channel packets $X[0], \ldots, X[i+\tau_L]$ by solving a system of linear equations. Only the symbols corresponding to $X^{(0)}[0], \ldots, X^{(0)}[i+\tau_L]$ need to be used, since the linear equations corresponding to $X^{(1)}[0], \ldots, X^{(1)}[i+\tau_L]$ are in their span. Moreover, the linear equations corresponding to

the symbols of $X^{(0)}[0],\ldots,X^{(0)}[i+\tau_L]$ are linearly independent by definition. By induction, $\sum_{j=0}^{i-1}\sum_{l=0}^{\tau_L}|X_j^{(0)}[j+l]| = \sum_{j=0}^{i-1}k_j$. The symbols of $X_j^{(0)}[j+l]$ in the first term reflect the symbols sent in channel packet $X[j+l]$ used to decode message packet $S[j]$. When $S[i]$ is decoded, at least $\sum_{j=0}^{i}k_j$ of the symbols of $X^{(0)}[0],\ldots,X^{(0)}[i+\tau_L]$ are required to decode $S[0],\ldots,S[i]$ (along with perhaps some additional symbols of $S[i+1],\ldots,S[i+\tau_L]$). For $j\in\{0,\ldots,i+\tau_L\}$, each symbol of $X^{(0)}[j]$ included reflects adding a linearly independent equation. $S[0],\ldots,S[i]$, along with perhaps some additional symbols of $S[i+1],\ldots,S[i+\tau_L]$, are decoded together. Exactly k_i equations (corresponding to symbols) used in decoding are used to decode symbols of $S[i]$. Since the encoding is causal, each of these k_i equations correspond to symbols are sent in channel packets $X[i],\ldots,X[i+\tau_L]$ and are labeled $X_i^{(0)}[i],\ldots,X_i^{(0)}[i+\tau_L]$.

Constraint #4: Consider any burst starting in time slot $i\in\{0,\ldots,t-b+1\}$. The proof of Lemma 1 showed that satisfaction of the worst-case-delay requirement over the $C(a,b,w)$ channel implies that message packets $S[i-\tau_L],\ldots,S[i+b-1]$ must be decoded by time slot $(i+\tau+b-a)$. Moreover, for any burst loss of length b starting in channel packet i, for each of $j\in\{i-\tau_L,\ldots,i+b-1\}$, $S[i-\tau_L],\ldots,S[j]$ must be recoverable by time slots $(i-\tau_L+\tau),\ldots,(j+\tau)$ respectively. Message packets $S[i-\tau_L],\ldots,S[j]$ are therefore decoded by time slot $(j+\tau)$. We consider the relaxation that symbols of $X_l^{(0)}[z]$ for $l\in\{j+1,\ldots,i+b-1\}, z\in\{l,\ldots,l+\tau_L\}$ are received.

We allow the relaxation that each symbol corresponding to $X_l^{(0)}[z]$ where $l\in\{i-\tau_L,\ldots,j\}, z\in\{l,\ldots,l+\tau_L\}$ which are received can be used to decode one symbol of $S[i-\tau_L],\ldots,S[j]$. These symbols are received during time slots $(i-\tau_L),\ldots,(i-1)$ and $(i+b),\ldots,\min(j+\tau,i+\tau+b-a)$. Furthermore, parity symbols received during time slots $(i+b),\ldots,\min(j+\tau,i+\tau+b-a)$ can be used to decode message packets $S[i-\tau_L],\ldots,S[j]$. However, by definition any parity symbols sent before time slot i are in the span of symbols of $X_l^{(0)}[z]$ for $l\in\{0,\ldots,i-1\}, z\in\{l,\ldots,\min(i-1,l+\tau_L)\}$. Therefore, given access to the symbols of $X_l^{(0)}[z]$, these parity symbols are not used to decode message packets $S[i-\tau_L],\ldots,S[j]$. All symbols received strictly after $\min(j+\tau,i+\tau+b-a)$ are received after message packets $S[i-\tau_L],\ldots,S[j]$ have already been decoded. Thus, the symbols of $S=$

$$\begin{aligned}&\big\{X_l^{(0)}[z]\mid l\in\{0,\ldots,t\}, z\in\{l,\ldots,l+\tau_L\}\backslash\\&\{i,\ldots,i+b-1\}\big\}\cup\\&\big\{X_l^{(0)}[z]\mid l\in\{j+1,\ldots,i+b-1\}, z\in\{l,\ldots,l+\tau_L\}\big\}\cup\\&\big\{X^{(1)}[z]\mid z\in\{i+b,\ldots,\min(j+\tau,i+\tau+b-a)\}\big\}\end{aligned}$$

must be sufficient to decode $(S[0],\ldots,S[t])$. Therefore, $|S|\geq\sum_{i=0}^{t}k_i$. By constraint #3, $\sum_{z=l}^{l+\tau_L}|X_l^{(0)}[z]|=k_l$ for any $l\in\{0,\ldots,t-\tau\}$. Thus, the size of

$$\begin{aligned}&\big\{X_l^{(0)}[z]\mid l\in\{i-\tau_L,\ldots,j\}, z\in\{l,\ldots,l+\tau_L\}\backslash\\&\{i,\ldots,i+b-1\}\big\}\cup\\&\big\{X^{(1)}[z]\mid z\in\{i+b,\ldots,\min(j+\tau,i+\tau+b-a)\}\big\}\end{aligned}$$

is at least $\sum_{l=i-\tau_L}^{j}k_l$.

Constraint #5: Consider any combination of a arbitrary packet losses in a sliding window of length $w=(\tau+1)$ which begin during some time slot $i\in\{0,\ldots,t-a+1\}$. Let the time slots of the packet losses be denoted as I, and let i' be the final time slot in I where $i'\leq t$. For any $j\in\{0,\ldots,i'\}$, each of $S[i-\tau_L],\ldots,S[j]$ must be decoded by time slot $(j+\tau)$ in order to satisfy the worst-case-delay requirement. The relaxation is used that each received symbol of $X_l^{(0)}[z]$ for $l\in\{i-\tau_L,\ldots,j\}, z\in\{l,\ldots,l+\tau_L\}$ can be used to decode one symbol of $S[i-\tau_L],\ldots,S[j]$. Furthermore, the relaxation is taken that each received symbol $X^{(1)}[l]$ for $l\in\{i+1,\ldots,j+\tau\}$ can be used to decode one lost symbol of $S[i-\tau_L],\ldots,S[j]$. We assume $X_l^{(0)}[z]$ is received for $l\in\{0,\ldots,i-1\}, z\in\{l,\ldots,\min(l+\tau_L,i-1)\}$, so the received symbols of $X^{(1)}[0],\ldots,X^{(1)}[i-1]$ are not useful for decoding $S[i-\tau_L],\ldots,S[j]$. Moreover, all symbols received strictly after time slot $(j+\tau)$ are not used in decoding $S[i-\tau_L],\ldots,S[j]$, since they are decoded by time slot $(j+\tau)$. This follows from similar reasoning to that discussed for constraint #4. Hence, the symbols of $S=$

$$\begin{aligned}&\big\{X_l^{(0)}[z]\mid l\in\{i-\tau_L,\ldots,j\}, z\in\{l,\ldots,l+\tau_L\}\setminus I\big\}\cup\\&\big\{X^{(1)}[j]\mid j\in\{i+1,\ldots,j+\tau\}\setminus I\big\}\cup\\&\big\{X_l^{(0)}[z]\mid l\in\{0,\ldots,i-\tau_L-1\}\cup\{i'+1,\ldots,t\},\\&\quad z\in\{l,\ldots,l+\tau_L\}\big\}\end{aligned}$$

are sufficient to decode $S[0],\ldots,S[t]$. Consequently, the size of S must be at least $\sum_{l=0}^{t}k_l$. Similar to the discussion for constraint #4, $\sum_{z=l}^{l+\tau_L}|X_l^{(0)}[z]|=k_l$ for any $l\in\{0,\ldots,t-\tau\}$. Thus, as a relaxation of the worst-case-delay requirement for at most a arbitrary losses, the size of

$$\begin{aligned}&\big\{X_l^{(0)}[z]\mid l\in\{i-\tau_L,\ldots,j\}, z\in\{l,\ldots,l+\tau_L\}\setminus I\big\}\cup\\&\big\{X^{(1)}[j]\mid j\in\{i+1,\ldots,j+\tau\}\setminus I\big\}\end{aligned}$$

must be at least $\sum_{z=i-\tau_L}^{j}k_z$. ■

APPENDIX D
PROOF OF THEOREM 2

Proof: The value computed by Algorithm 2 is the optimal rate for a coding scheme that combines (1) blocks of the SBC with (2) the (τ,a,b,τ_L)-separate encoding scheme. Therefore, this rate is feasible. The total number of symbols of all message packets divided by the total number of symbols transmitted by the scheme is returned. Hence, the rate of the corresponding coding scheme is returned. The objective function is to compute the minimal possible value for the number of symbols transmitted over the considered coding schemes for the message packet size sequence. For a fixed total number of symbols of all message packets, this minimizes the rate.

It remains to verify that the lossless-delay and worst-case-delay requirements are satisfied. We do so for (a) the symbols of message packets encoded as part of the (τ,a,b,τ_L)-separate encoding scheme, and (b) for all remaining symbols. For $i>(t-\tau)$, the message packet $S[i]$ is of size 0 and is

automatically known by the receiver. Hence, the lossless-delay and worst-case-delay requirements are satisfied for such message packets.

First, a non-negative number of symbols of any $S[i]$, for $i \in \{0, \ldots, t-\tau\}$, are modeled as being encoded using the (τ, a, b, τ_L)-separate encoding scheme for each message packet due to constraint #1. The lossless-delay and worst-case-delay are met for such symbols by properties of the (τ, a, b, τ_L)-separate encoding scheme.

Second, we verify that all remaining symbols are accurately modeled as being encoded within blocks of the SBC such that the lossless-delay and worst-case-delay requirements are met.

The lossless-delay requirement: For any $S[i]$, for $i \in \{0, \ldots, t-\tau\}$, a non-negative number of symbols are modeled as sent in each of $X[i], \ldots, X[\tau_L + 1]$, as is reflected by constraint #2. Moreover, the total number of such symbols is sufficient to satisfy the lossless-delay requirement by constraint #4.

The worst-case-delay requirement: A non-negative number of blocks starting in each channel packet is modeled due to constraint #3. Recall that the variables p_i reflect the quantity of blocks whose first position occurs during time slot i. The first position of blocks occur between time slot 0 and $(t-\tau)$. It remains to verify that all symbols not encoded using the (τ, a, b, τ_L)-separate encoding scheme can be modeled as being placed in the corresponding blocks which ensure that they are decoded within τ time slots. Under constraint #5, each $j \in \{\tau_L, \ldots, 1\}$ is sequentially considered for each channel packet $X[i]$. Without loss of generality, each symbol sent in channel packet $X[i]$ corresponding to message packet $(i-j)$ is modeled as being placed in the earliest block ending by time slot $(i-j+\tau)$ with an available position in channel packet $X[i]$. Sequentially doing so ensures that all symbols corresponding to message packets $(i-\tau_L), \ldots, (i-1)$ sent in channel packet $X[i]$ are modeled as being encoded as part of by blocks of the SBC whose final position occur by the time slots $(i-\tau_L+\tau), \ldots, (i+\tau-1)$ respectively. This ensures that the worst-case-delay is satisfied over a $C(a, b, w)$ channel for such symbols. Finally, due to constraint #6, each symbol corresponding to $S[i]$ which is modeled as being transmitted without delay in $X[i]$ is modeled as being placed in an available block of the SBC. This ensures recovery within τ time slots over a $C(a, b, w)$ channel. In so doing, all blocks which have an available position in channel packet $X[i]$ are considered, as the recovery properties of the SBC ensure recovery within τ time slots. Furthermore, all symbols corresponding to message packets $S[i-\tau_L], \ldots, S[i-1]$ sent in channel packet $X[i]$ still must also be modeled as being placed in available blocks of the SBC, as is reflected in constraint #6. ■

APPENDIX E
PROOF OF LEMMA 8

Proof: By definition, a channel packet is comprised of three quantities. First, symbols of message packets. Second, parity symbols corresponding to blocks of the SBC. Third, parity symbols corresponding to the padded excess.

The number of systematic symbols in channel packet $X[i]$ is given by

$$= \left(\sum_{j=i-\tau_L}^{i-1} \frac{k_j}{\tau - a + 1} \right) + \frac{k_i(\tau - a + 1 - \tau_L)}{\tau - a + 1}.$$

In expectation, this is

$$\mathbb{E}\left[\left(\sum_{j=i-\tau_L}^{i-1} \frac{k_j}{\tau - a + 1} \right) + \frac{k_i(\tau - a + 1 - \tau_L)}{\tau - a + 1} \right] \tag{11}$$

$$= \left(\sum_{j=i-\tau_L}^{i-1} \mathbb{E}[k_j] \frac{1}{\tau - a + 1} \right) + \mathbb{E}[k_i] \frac{\tau - a + 1 - \tau_L}{\tau - a + 1} \tag{12}$$

$$= \left(\sum_{j=i-\tau_L}^{i-1} \mu \frac{1}{\tau - a + 1} \right) + \mu \frac{(\tau - a + 1 - \tau_L)}{\tau - a + 1} \tag{13}$$

$$= \mu. \tag{14}$$

The number of parity symbols corresponding to blocks of the SBC in $X[i]$ is given by

$$\sum_{j=i-(\tau+b-a)}^{i-(\tau-a+1)} \frac{k_j}{\tau - a + 1}.$$

In expectation, this equals

$$\sum_{j=i-(\tau+b-a)}^{i-(\tau-a+1)} \frac{\mathbb{E}[k_j]}{\tau - a + 1} = \mu \frac{b}{\tau - a + 1}$$

The number of parity symbols corresponding to padded excess sent in channel packet $X[i]$ is given by

$$\sum_{j=i-\tau}^{i-(b-a+1)} \frac{e_j^c}{\tau - b + 1}.$$

In expectation, this equals

$$\sum_{j=i-\tau}^{i-(b-a+1)} \frac{\mathbb{E}[e_j^c]}{\tau - b + 1} = e^c \frac{\tau - b + a}{\tau - b + 1}.$$

Altogether, the expected number of symbols of a channel packet is

$$\mu \left(1 + \frac{b}{\tau - a + 1} \right) + e^c \frac{\tau - b + a}{\tau - b + 1}.$$

■

APPENDIX F
PROOF OF THEOREM 4

The proof of Theorem 4 is shown in three parts. First, we show convergence of the mean number of symbols of message packets (i.e., $\frac{1}{t+1} \sum_{i=0}^{t} k_i$) to its expected value, μ, using the Hoeffding inequality [33] in Lemma 10. Second, we show convergence of the mean number of transmitted symbols (i.e., $\frac{1}{t+1} \sum_{i=0}^{t} n_i$) to its expected value, μ_c, using the Hoeffding inequality [33] in Lemma 11. Third, show convergence of the rate given convergence of the mean number of symbols of message packets and channel packets.

First, we analyze the rate of convergence of the average number of symbols of message packets in Lemma 10.

Lemma 10: Consider any valid inputs (τ, a, b, w, τ_L), δ, $\epsilon > 0$, and a sequence of

$$\frac{\sqrt{2}d}{\epsilon}\sqrt{\ln\left(\frac{2}{\delta}\right)} + \frac{5\tau d}{\epsilon}$$

message packets whose sizes are drawn independently from an admissible distribution with mean μ and maximum value d. With probability at least $(1-\delta)$, the mean number of symbols of the message packets of the message packet size sequence satisfy the following inequality:

$$\left|\left(\frac{1}{t+1}\sum_{i=0}^{t} k_i\right) - \mu\right| < \epsilon.$$

Proof: This proof will follow from a simple application of the Hoeffding inequality [33] after handling the fact that the final 2τ message packets have size 0.

We rewrite

$$\frac{1}{t+1}\sum_{i=0}^{t} k_i = \frac{1}{t+1}\sum_{i=0}^{t-2\tau} k_i = \frac{t-2\tau}{t+1}\frac{1}{t-2\tau}\sum_{i=0}^{t-2\tau} k_i.$$

When $t > 2\frac{2\tau}{\epsilon}$, it means that $\frac{t-2\tau+1}{t+1} = 1 - \frac{2\tau}{t+1} \geq 1 - \frac{\epsilon}{2}$.

The quantity $\frac{1}{t-2\tau}\sum_{i=0}^{t-2\tau} k_i$ is the sum of random variables drawn independently from D and each $k_i \in \{0, \ldots, d\}$ for some positive integer d for $i \geq \tau$ and $i \leq (t-\tau)$. We apply the Hoeffding inequality [33] to bound the probability that $\left|\left(\frac{1}{t-2\tau+1}\sum_{i=0}^{t-2\tau} k_i\right) - \mu\right| < \frac{\epsilon}{2}$ and find it to be at least $(1 - 2e^{-\frac{2(t+1-2\tau)^2\epsilon^2}{4d^2}})$. To ensure that this value is at least $(1-\delta)$, it suffices to have $t \geq \left(\frac{\sqrt{2}d}{\epsilon}\sqrt{\ln\left(\frac{2}{\delta}\right)} + 2\tau\right)$.

Whenever $t = \left(\frac{\sqrt{2}d}{\epsilon}\sqrt{\ln\left(\frac{2}{\delta}\right)} + \frac{5\tau\mu}{\epsilon}\right)$, with probability at least $(1-\delta)$,

$$\begin{aligned}\left|\left(\frac{1}{t+1}\sum_{i=0}^{t} k_i\right) - \mu\right| &= \left|\left(\frac{t-2\tau}{t+1}\frac{1}{t-2\tau}\sum_{i=0}^{t-2\tau} k_i\right) - \mu\right| \\ &\leq \left|(1-\frac{\epsilon}{2})\frac{\epsilon}{2}\right| = \left|\frac{\epsilon}{2} - \frac{\epsilon^2}{4}\right| \\ &< \epsilon.\end{aligned}$$

Applying the inequality $\mu \leq d$ concludes the proof. ■

Second, we analyze the rate of convergence of the average number of symbols of channel packets in Lemma 11.

Lemma 11: Consider any valid inputs (τ, a, b, w, τ_L), δ, $\epsilon > 0$, and a sequence of

$$\frac{18\tau bd}{\sqrt{2}\epsilon}\sqrt{\ln\left(\frac{4\tau}{\delta}\right)} + \frac{45db\tau}{\epsilon}$$

message packets whose sizes are drawn independently from an admissible distribution with mean μ and maximum value d. Recall from Eq. (7) that μ_c is the expected number of symbols of a channel packet under the $(\tau, a, b, w, \tau_L)-$VSC. With probability at least $(1-\delta)$, the mean number of symbols of channel packets for the $(\tau, a, b, w, \tau_L)-$VSC is given by: $\left|\left(\frac{1}{t+1}\sum_{i=0}^{t} n_i\right) - \mu_c\right| < \epsilon$.

Proof: We break $\frac{1}{t+1}\sum_{i=0}^{t} n_i$ into three components. This is necessary to handle the first τ and final 2τ channel packets having sizes taken from a different distribution based on boundary conditions as compared to the remaining terms. Specifically, we rewrite $\frac{1}{t+1}\sum_{i=0}^{t} n_i$ as

$$\begin{aligned}&\frac{1}{t+1}\left(\sum_{i=0}^{\tau-1} n_i + \sum_{i=t-2\tau+1-p}^{t} n_i\right) + \\ &\frac{t-3\tau+1-p}{t+1}\frac{1}{t-3\tau+1-p}\sum_{i=\tau}^{t-2\tau-p} n_i \qquad (15)\end{aligned}$$

where p is the smallest non-negative integer so that $(2\tau)|(t-3\tau+1-p)$, which is chosen so as to make the number of terms in $\sum_{i=t}^{t-2\tau+1-p} n_i$ divisible by 2τ, which will be used later.

Each channel packet comprises at most d symbols of message packets, $d\frac{\tau-b+a}{\tau-b+1}$ parity symbols from encoding the padded excess, and $d\frac{b}{\tau-a+1}$ parity symbols corresponding to blocks of the SBC. Therefore, the size of a channel packet lies in $\{0, \ldots, d\cdot\left(1 + \frac{\tau-b+a}{\tau-b+1} + \frac{b}{\tau-a+1}\right) \leq d(1+a+b) \leq 3db\}$. When $t \geq \frac{45db\tau}{\epsilon}$,

$$\frac{1}{t+1}\left(\sum_{i=0}^{\tau-1} n_i + \sum_{i=t-2\tau+1-p}^{t} n_i\right) \leq \frac{\epsilon}{3} \qquad (16)$$

.

When $t > 3\frac{3\tau+1}{\epsilon}$, it means that

$$\frac{t-3\tau+1-p}{t+1} \geq 1 - \frac{\epsilon}{3}. \qquad (17)$$

Next, we analyze the last term in Eq. (15), $\frac{1}{t-3\tau+1-p}\sum_{i=\tau}^{t-2\tau-p} n_i$. Recall that n_i is a function of $k_{i-\tau}, \ldots, k_i$, so $n_i \perp n_j$ whenever $|i-j| > 2\tau$. To apply Hoeffding's inequality, we rewrite

$$\begin{aligned}&\frac{1}{t-3\tau+1-p}\sum_{i=\tau}^{t-2\tau-p} n_i = \\ &\sum_{r=0}^{2\tau-1}\frac{1}{2\tau}\left(\frac{2\tau}{t-3\tau+1-p}\sum_{i=0}^{\frac{t-3\tau+1-p}{2\tau}-1} n_{\tau+(2\tau i)+r}\right). \qquad (18)\end{aligned}$$

Observe that $n_{\tau+(i-\tau)2\tau+r} \perp n_{\tau+(i'-\tau)2\tau+r}$ for $i \neq i' \in \left\{\tau, \ldots, \frac{t-2\tau-p}{2\tau}\right\}$. Consider any $r \in \{0, \ldots, 2\tau-1\}$. We apply Hoeffding's inequality [33] to show that $\left|\left(\frac{2\tau}{t-3\tau+1-p}\sum_{i=0}^{\frac{t-3\tau+1-p}{2\tau}-1} n_{\tau+(2\tau i)+r}\right) - \mu_c\right| < \frac{\epsilon}{3}$ with probability at least $\left(1 - 2e^{-\frac{2(t-3\tau+1-p)^2\epsilon^2}{324\tau^2b^2d^2}}\right)$. Taking a union bound over the 2τ values of r and applying it to Eq. (18) leads to

$$\left|\frac{1}{t-3\tau+1-p}\left(\sum_{i=\tau}^{t-2\tau-p} n_i\right) - \mu_c\right| < \frac{\epsilon}{3} \qquad (19)$$

with probability at least $\left(1 - 4\tau e^{-\frac{2(t-3\tau+1-p)^2\epsilon^2}{324\tau^2b^2d^2}}\right)$. In order for this to be at least $(1-\delta)$, it suffices to have

$t \geq \left(\frac{18\tau bd}{\sqrt{2}\epsilon}\sqrt{\ln\left(\frac{4\tau}{\delta}\right)} + 5\tau\right)$. The proof follows from combining Eq. (15) with Eq. (16), Eq. (17), and Eq. (19) as,

$$\begin{aligned}
&\left|\left(\frac{1}{t+1}\sum_{i=0}^{t} n_i\right) - \mu_c\right| \\
&= \left|\frac{1}{t+1}\left(\sum_{i=0}^{\tau-1} n_i + \sum_{i=t-2\tau+1-p}^{t} n_i\right) + \right. \\
&\quad \left.\left(\frac{t-3\tau+1-p}{t+1}\frac{1}{t-3\tau+1-p}\sum_{i=\tau}^{t-2\tau-p} n_i - \mu_c\right)\right| \\
&\leq \left|\frac{\epsilon}{3} + (1-\frac{\epsilon}{3})\frac{\epsilon}{3}\right| \\
&= \left|\frac{2\epsilon}{3} - \frac{\epsilon^2}{9}\right| \\
&< \epsilon.
\end{aligned}$$

Thus, $\left|\left(\frac{1}{t+1}\sum_{i=0}^{t} n_i\right) - \mu_c\right| < \epsilon$ with probability at least $(1-\delta)$ whenever

$$t = \max\left(\frac{18\tau bd}{\sqrt{2}\epsilon}\sqrt{\ln\left(\frac{4\tau}{\delta}\right)} + 5\tau, \frac{45db\tau}{\epsilon}\right).$$

■

Third, we analyze the rate of convergence of the coding rate of the $(\tau, a, b, w, \tau_L)-$VSC to conclude the proof of Theorem 4.

Proof: Using $t \geq \left(\frac{36\tau bd\sqrt{2}}{\epsilon\mu_c}\sqrt{\ln\left(\frac{4\tau}{\delta}\right)} + \frac{180db\tau}{\mu_c\epsilon}\right)$, we apply Lemmas 10 and 11 with $\delta' = \frac{\delta}{2}$ and $\epsilon' = \frac{\epsilon\mu_c}{4}$. With probability $(1 - \frac{\delta}{2} - \frac{\delta}{2}) = (1-\delta)$, there exists $|\epsilon_1|, |\epsilon_2| \leq \epsilon'$ such that

$$\begin{aligned}
\left|\frac{\sum_{i=0}^{t} k_i}{\sum_{i=0}^{t} n_i} - \frac{\mu}{\mu_c}\right| &= \left|\frac{\mu+\epsilon_1}{\mu_c+\epsilon_2} - \frac{\mu}{\mu_c}\right| \\
&= \frac{1}{\mu_c(\mu_c+\epsilon_2)}|\mu\mu_c + \epsilon_1\mu_c - \mu\mu_c - \epsilon_2\mu_c| \\
&= \frac{1}{\mu_c+\epsilon_2}|\epsilon_1 - \epsilon_2| \\
&\leq \frac{4\epsilon'}{\mu_c} \\
&\leq \epsilon.
\end{aligned}$$

■

Appendix G
Proof of Lemma 9

Before presenting the proof of Lemma 9, we include the auxiliary Lemma 12 which will later used in the proof of Lemma 9.

Lemma 12: Let U and V be any two discrete random variables drawn from distributions with finite support such that $\mathbb{E}[U] = \mathbb{E}[V]$. Then $\mathbb{E}[\max(U-V, 0)] = \frac{1}{2}\mathbb{E}[|U-V|]$.

Proof: Let $p_{U,V}(u,v)$ be the joint probability mass function of U and V. By assumption,

$$\begin{aligned}
0 &= \mathbb{E}[U-V] \\
&= \sum_{(u,v)\in U\times V} p_{U,V}(u,v)(u-v) \\
&= \sum_{(u,v)\in U\times V} p_{U,V}(u,v)[\max(u-v,0) + \min(u-v,0)].
\end{aligned}$$

Rearrangement yields

$$\begin{aligned}
\mathbb{E}[\max(U-V,0)] &= \sum_{(u,v)\in U\times V} p_{U,V}(u,v)\max(u-v,0) \\
&= -\sum_{(u,v)\in U\times V} p_{U,V}(u,v)\min(u-v,0).
\end{aligned} \tag{20}$$

The value of $\mathbb{E}[|U-V|]$ can be written as

$$\begin{aligned}
&\mathbb{E}[|U-V|] = \\
&\sum_{(u,v)\in U\times V} p_{U,V}(u,v)\left(\max(u-v,0) - \min(u-v,0)\right).
\end{aligned} \tag{21}$$

Combining Eq. (20) and Eq. (21) along with dividing by 2 yields

$$\mathbb{E}[\max(U-V,0)] = \frac{1}{2}\mathbb{E}[|U-V|].$$

■

Next, we use Lemma 12 to prove Lemma 9 as follows

Proof: We will show for any $i \in \{2\tau, \ldots, t-2\tau\}$ that the expected excess $E[e_i]$ is at most

$$\frac{\sigma\sqrt{(\tau-a-\tau_L)^2 + (\tau-a-\tau_L)}}{2(\tau-a+1)} \leq \frac{\sigma}{\sqrt{2}}.$$

The result follows from combining this inequality with Theorem 4, the definition of μ_c from Eq. (7), and the identity $\mathbb{E}[pad[i]] \leq (\tau - b)$.

For any message packet $S[i]$, the symbols $(S_{\tau-a-\tau_L}[i], \ldots, S_{\tau-a}[i])$ are always included in the blocks of SBC. The remaining symbols $(S_0[i], \ldots, S_{\tau-a-\tau_L-1}[i])$ can be placed in blocks corresponding to message packets

$$\begin{aligned}
I_i = &\{j \mid j \in \{i+1, \ldots, i+b-a\}\} \cup \\
&\{j \mid j \in \{i-\tau+b, \ldots, i-\tau_L-1\}\}.
\end{aligned}$$

Therefore, the expected excess is given by

$$\mathbb{E}[e_i] = \frac{1}{\tau-a+1}\mathbb{E}\left[\max\left((\tau-a-\tau_L)\,k_i - \sum_{j\in I_i} k_j, 0\right)\right] \tag{22}$$

$$= \frac{\mathbb{E}\left[\sqrt{\left((\tau-a-\tau_L)k_i - \sum_{j\in I_i} k_j\right)^2}\right]}{2(\tau-a+1)} \tag{23}$$

$$\leq \frac{\sqrt{\mathbb{E}\left[\left((\tau-a-\tau_L)k_i - \sum_{j\in I_i} k_j\right)^2\right]}}{2(\tau-a+1)} \tag{24}$$

$$= \frac{1}{2(\tau-a+1)}\Bigg((\tau-a-\tau_L)^2\left(\mathbb{E}[k_i^2] - \mathbb{E}[k_i]^2\right) + \tag{25}$$

$$\mathbb{E}\big[\big(\sum_{j\in I_i} k_i\big)^2\big] - \mathbb{E}\big[\sum_{j\in I_i} k_j\big]^2\Bigg)^{1/2}$$

$$= \frac{\sqrt{(\tau - a - \tau_L)^2 \mathrm{Var}(k_i) + \mathrm{Var}\left(\sum_{j \in I_i} k_j\right)}}{2(\tau - a + 1)} \quad (26)$$

$$= \frac{\sigma\sqrt{(\tau - a - \tau_L)^2 + (\tau - a - \tau_L)}}{2(\tau - a + 1)}$$

$$\leq \frac{\sigma}{\sqrt{2}}. \quad (27)$$

Eq. (23) follows from applying Lemma 12 to $\mathbb{E}[e_i]$ along with the identity $|A| = \sqrt{A^2}$. Eq. (24) holds due to Jensen's inequality and the fact that $\sqrt{\cdot}$ is concave. Eq. (25) follows from the facts that $(\tau - a - \tau_L)\mathbb{E}[k_i] = \mathbb{E}[\sum_{j \in l_i} k_j]$ and $k_j \perp k_i$ for $j \in I_i$. Eq. (26) holds by the fact that $k_j \perp k_{j'}$ for $j \neq j' \in I_i$. Finally, Eq. (27) is a consequence of the inequality $(\tau - a + 1) \geq (\tau - a - \tau_L)$. ■

APPENDIX H
DESCRIPTION OF BASELINE

We now provide a detailed explanation of the baseline scheme to which we compare the (τ, a, b, τ_L)-VSC in Section VIII. Recall from Section II that rate-optimal streaming codes for the setting of message packets of the same fixed size, k, such as [5]–[9], involve sending each $S[i]$ as part of $X[i]$. In addition, $\frac{k}{\tau-a+1}$ blocks of the SBC corresponding to $S[i]$ are created. This involves sending $k\frac{b}{\tau-a+1}$ parity symbols evenly over channel packets $X[i+\tau-a+1], \ldots, X[i+\tau-a+b]$. Hence, the first parity symbol corresponding to $S[i]$ is sent during time slot $(i+\tau-a+1)$. Another way of viewing such schemes is that $\frac{\tau-a}{\tau-a+1}k$ symbols of $S[i]$ are encoded using blocks of the SBC corresponding to message packets $X[i-\tau+a], \ldots, X[i-1]$. The remaining $\frac{k}{\tau-a+1}$ symbols of $S[i]$ are encoded by creating the additional $\frac{k}{\tau-a+1}$ blocks of the SBC corresponding to $S[i]$. Thus, during time slot $(i+\tau-a+1)$, the minimum necessary number of additional blocks of the SBC are created to ensure that message packet $S[i]$ is recovered within the worst-case-delay constraint over the $C(a, b, w)$ channel.

For the "baseline" scheme used in our evaluation, we introduce as few adjustments to the aforementioned rate-optimal code construction for message packets of the same fixed size, as are needed to account for the new setting where message packets are of varying sizes. As is done by the aforementioned schemes, the baseline coding scheme will involve sending each message packet $S[i]$ as part of channel packet $X[i]$. This satisfies the lossless-delay constraint. Each symbol of the message packets is protected using a block of the SBC, as is the case under the existing schemes. To do so, d_i blocks of the SBC are created corresponding to message packet $S[i]$ where $d_i = \max\left(0, k_i - \sum_{j=\max(0,i-\tau+a)}^{i-1} d_j\right)$ for $i \geq (\tau - a + 1)$ and $0 = d_0 = \ldots = d_{\tau-a}$. The b parity symbols of each stripe are sent in channel packets $X[i+\tau-a+1], \ldots, X[i+\tau+b-a+1]$. The quantity d_i is therefore defined during time slot $(i+\tau-a+1)$. This is the final time slot during which a block of the SBC can be created which includes a symbol of message packet $S[i]$. Each block includes one not yet encoded symbol of $S[i]$ and one not yet encoded symbol of $S[j]$ for $j \in \{i+1, \ldots, i+\tau-a\}$ (or a unique padding symbol if none are available). This ensures that the worst-case-delay constraint is satisfied for message packet $S[i]$. The number of blocks, d_i, is the minimal value for which all symbols of message packet $S[i]$ can be encoded in a block of the SBC.

ACKNOWLEDGMENT

The authors would like to acknowledge Jack Kosaian for providing the video traces used in this paper. The authors also acknowledge Francisco Maturana for his helpful feedback in the editing process.

REFERENCES

[1] 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.
[2] 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.
[3] E. Martinian and M. Trott, "Delay-optimal burst erasure code construction," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 1006–1010.
[4] A. Badr, P. Patil, A. Khisti, W. Tan, and J. Apostolopoulos, "Layered constructions for low-delay streaming codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 1, pp. 111–141, Jan. 2017.
[5] 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.
[6] 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.
[7] D. Dudzicz, S. L. Fong, and A. Khisti, "An explicit construction of optimal streaming codes for channels with burst and arbitrary erasures," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 12–25, Jan. 2020.
[8] M. N. Krishnan, D. Shukla, and P. V. Kumar, "Rate-optimal streaming codes for channels with burst and random erasures," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4869–4891, Aug. 2020.
[9] 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.
[10] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, 1960. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1960.tb03959.x
[11] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, Sep. 1963.
[12] G. Forney, Jr., "Burst-correcting codes for the classic bursty channel," *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 772–781, Oct. 1971.
[13] A. Badr, D. Lui, A. Khisti, W. 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.
[14] S. L. Fong, A. Khisti, B. Li, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "Optimal multiplexed erasure codes for streaming messages with different decoding delays," *IEEE Trans. Inf. Theory*, vol. 66, no. 7, pp. 4007–4018, Jul. 2020.
[15] A. Badr, A. Khisti, and E. Martinian, "Diversity embedded streaming erasure codes (DE-SCo): Constructions and optimality," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1042–1054, 2011.
[16] A. Badr, D. Lui, and A. Khisti, "Streaming codes for multicast over burst erasure channels," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4181–4208, Aug. 2015.
[17] M. Haghifam, M. N. Krishnan, A. Khisti, X. Zhu, W.-T. Tan, and J. Apostolopoulos, "On streaming codes with unequal error protection," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 4, pp. 1165–1179, Dec. 2021.
[18] N. Adler and Y. Cassuto, "Burst-erasure correcting codes with optimal average delay," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 2848–2865, May 2017.
[19] D. Leong and T. Ho, "Erasure coding for real-time streaming," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2012, pp. 289–293.
[20] D. Leong, A. Qureshi, and T. Ho, "On coding for real-time streaming under packet erasures," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 1012–1016.

[21] A. Badr, A. Khisti, W. T. Tan, and J. Apostolopoulos, "Streaming codes with partial recovery over channels with burst and isolated erasures," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 3, pp. 501–516, Apr. 2015.
[22] 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.
[23] A. Badr, A. Khisti, W.-T. Tan, X. Zhu, and J. Apostolopoulos, "FEC for VoIP using dual-delay streaming codes," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.
[24] Z. Li, A. Khisti, and B. Girod, "Correcting erasure bursts with minimum decoding delay," in *Proc. Conf. Rec. 45th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 33–39.
[25] Y. Wei and T. Ho, "On prioritized coding for real-time streaming under packet erasures," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2013, pp. 327–334.
[26] M. Nikhil Krishnan, V. Ramkumar, M. Vajha, and P. Vijay Kumar, "Simple streaming codes for reliable, low-latency communication," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 249–253, Feb. 2020.
[27] V. Ramkumar, M. Vajha, M. N. Krishnan, and P. Vijay Kumar, "Staggered diagonal embedding based linear field size streaming codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 503–508.
[28] M. Rudow and K. V. Rashmi, "Online versus offline rate in streaming codes for variable-size messages," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 509–514.
[29] M. Rudow and K. V. Rashmi, "Learning-augmented streaming codes are approximately optimal for variable-size messages," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022.
[30] *Open Broadcaster Software*. Accessed: Jun. 19, 2018. [Online]. Available: https://obsproject.com/
[31] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, Apr. 2016.
[32] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *J. Control Decision*, vol. 5, no. 1, pp. 42–60, 2018.
[33] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. New York, NY, USA: Springer, 1994, pp. 409–426.

Michael Rudow received the bachelor's degree in computer science and the master's degree in mathematics from the University of Pennsylvania. He is currently pursuing the Ph.D. degree with the Computer Science Department, Carnegie Mellon University, advised by Prof. Rashmi Vinayak. He is broadly interested in the intersection of coding theory and machine learning.

K. V. Rashmi (Member, IEEE) received the Ph.D. degree from UC Berkeley in 2016. She was a Post-Doctoral Scholar with UC Berkeley from 2016 to 2017. She is an Assistant Professor with the Computer Science Department, Carnegie Mellon University. During her Ph.D. studies, she was a recipient of Facebook Fellowship 2012–2013, the Microsoft Research Ph.D. Fellowship 2013–2015, and the Google Anita Borg Memorial Scholarship 2015–2016. Her research interests broadly lie in information/coding theory and computer/networked systems. She was a recipient of the VMWare Systems Research Award 2021, the NSF CAREER Award 2020–2025, the Tata Institute of Fundamental Research Memorial Lecture Award 2020, the Facebook Distributed Systems Research Award 2019, the Google Faculty Research Award 2018, and the Facebook Communications and Networking Research Award 2017. Her Ph.D. thesis was awarded the UC Berkeley Eli Jury Dissertation Award 2016. Her work has received the USENIX NSDI 2021 Community (Best Paper) Award, and the IEEE Data Storage Best Paper and Best Student Paper Awards for the years 2011/2012.