Online Versus Offline Rate in Streaming Codes for Variable-Size Messages

Michael Rudow
Carnegie Mellon University
Computer Science Department
Pittsburgh, Pennsylvania
Email: mrudow@andrew.cmu.edu

K.V. Rashmi
Carnegie Mellon University
Computer Science Department
Pittsburgh, Pennsylvania
Email: rvinayak@cs.cmu.edu

***Abstract*—Providing high quality-of-service for live communication is a pervasive challenge which is plagued by packet losses during transmission. Streaming codes are a class of erasure codes specifically designed for such low-latency streaming communication settings. We consider the recently proposed setting of streaming codes under *variable-size messages* which reflects the requirements of applications such as live video streaming. In practice, streaming codes often need to operate in an "online" setting where the sizes of the future messages are unknown. Yet, previously studied upper bounds on the rate apply to "offline" coding schemes with access to all (including future) message sizes.**

In this paper, we evaluate whether the optimal offline rate is a feasible goal for online streaming codes when communicating over a burst-only packet loss channel. We identify two broad parameter regimes where, perhaps surprisingly, online streaming codes can, in fact, match the optimal offline rate. For both of these settings, we present rate-optimal online code constructions. For all remaining parameter settings, we establish that it is impossible for online schemes to attain the optimal offline rate.

An extended version of this paper is accessible at: [1].

I. Introduction

Real-time communication with high quality-of-service is critical to many pervasive streaming applications, including VoIP and video conferencing. These live streaming applications rely on transmitting packets of information and must contend with packet losses during transmission. A standard solution to recover from packet loss is to retransmit lost packets. However, it is infeasible to use the retransmission-based approach in the live communication setting, as the three-way delay of transmission, feedback, and retransmission exceeds the real-time latency constraint [2]. One viable technique to provide robustness to packet loss is forward error correction. Yet using conventional coding schemes while complying with the real-time delay constraint induces a significant bandwidth overhead.

Coding schemes which are designed specifically for live streaming communication can attain significantly higher rate than traditional coding schemes (including the class of maximal distance separable (MDS) codes). This improved performance was demonstrated in [3] in which the authors proposed a new "streaming model" for real-time communication. The authors also presented a coding scheme and an upper bound on rate for the model. Under the streaming model, at each time slot, a "message packet" arrives at a sender who then transmits a "channel packet" to a receiver. The channel packets are transmitted over a burst-only packet loss channel. Due to the real-time latency constraints, each message packet must be decoded by the receiver within the delay of a strict fixed number of time slots. The streaming model is depicted in Figure 1. Code constructions designed specifically for the streaming model can have significantly higher rate than traditional code constructions. This has motivated numerous subsequent works on the streaming model (discussed briefly in Section II).

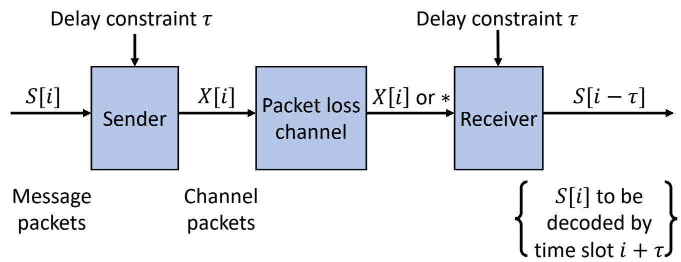

Fig. 1. Overview of the streaming model. At each time slot i, a sender receives a message packet $S[i]$ and transmits a channel packet $X[i]$ over a packet loss channel to a receiver. The message packet $S[i]$ is to be decoded within delay τ, i.e. by time slot $(i + \tau)$.

The streaming model proposed in [3] and studied further in several subsequent works considers a setting where at each time slot a sender receives a message packet comprising some *fixed* constant number of source symbols to be transmitted to a receiver. However, many applications intrinsically require transmitting a stream of variable-size messages. For example, in video conferencing a sender transmits a sequence of compressed video frames of fluctuating sizes. Consequently, a new streaming model incorporating *variable-size* messages was introduced in [4]. In this work, we focus on the setting of communicating variable-size messages over a burst-only packet loss channel.

Under the setting of variable-size messages, the upper bound on rate of the fixed-size regime still applies. However, the variability in the message sizes can induce more stringent rate constraints. Moreover, at each time slot, the optimal number of symbols to transmit can depend on the sizes of future messages, which are inherently variable and unknown. This leads to the distinction between "offline" coding schemes, which have access to the *sizes* of messages of *future* time slots, and "online" schemes, which do not have access to such information. Online constructions are of practical interest, as future message sizes are often unknown in live streaming applications. This leads to the natural question of "whether online coding schemes can match the rate of offline coding schemes?"

In this work, we identify two broad parameter regimes where, perhaps surprisingly, *online coding schemes can match the rate of optimal offline coding schemes*. For both these settings, we present rate-optimal online code constructions. For *all* remaining parameter regimes, we demonstrate that online coding schemes necessarily have strictly lower rate than optimal offline coding schemes.

II. Background, System model and Notation

As discussed in Section I, the streaming model was proposed in [3]. It captures the setting of real-time communication of a sequence of messages of a fixed constant size over a burst-only packet loss channel. The authors also introduced a class of code constructions applicable to the streaming model, called "streaming codes," along with an upper bound on rate (which will be discussed shortly). Later, this bound was met by a construction proposed in [5]. Streaming codes have significantly higher rate than traditional code constructions under the streaming model. This improvement in rate has prompted several works on bounds on rate and code constructions for the streaming model under a variety of settings [6]–[19].

In applications such as video communication, the sizes of the messages to be transmitted fluctuate considerably. To incorporate this, a streaming model for variable-size messages was introduced in [4]. The authors designed streaming codes for this new setting with higher rate than constructions designed for the setting of fixed-size messages. We later present rate-optimal streaming codes for two parameter regimes which outperform the code construction from [4].

This work considers the streaming model from [4] (with a few minor changes in how time slots are indexed). There is a finite stream of t messages for an arbitrary natural number t. At each time slot $i \in \{0, \ldots, t\}$, a sender receives a *message packet* $S[i]$ comprised of k_i symbols from a finite field $\mathbb{F}_q$. The number of symbols is between 0 and m for a natural number m representing the maximum message packet size. The sender then transmits a *channel packet*, $X[i]$, consisting of n_i symbols from $\mathbb{F}_q$ to a receiver. Each channel packet $X[i]$ either arrives at the receiver or is lost. We denote a lost packet by $*$. Each channel packet $X[i]$ depends only on the symbols of previous message packets (i.e. $S[0], \ldots, S[i]$). Due to real-time latency constraints, each message packet $S[i]$ must be decoded by the receiver within a delay of τ time slots (i.e. $S[i]$ is recovered using the channel packets received by time slot $(i + \tau)$). This requirement is called the *worst-case-delay constraint*.

In this setting, the rate R_t is defined as $R_t = \frac{\sum_{i=0}^{t} k_i}{\sum_{i=0}^{t} n_i}$.

The channel packets are transmitted over a burst-only packet loss channel equivalent to the one considered in [3]. This channel is denoted $C(b, w)$ and may introduce a single burst loss of length at most b packets within every sliding window of length w packets. We restrict our attention to $(w > \tau)$ in this work. Under a $C(b, w > \tau)$ channel for any sequence of t message packets, the rate R_t is upper bounded by $\frac{\tau}{\tau+b}$. This upper bound was initially shown for the setting of fixed-size message packets in [5] and was shown to hold for the setting of variable-size message packets in [4]. Depending on the sizes of the message packets, the upper bound may be loose, as will be seen later in this work.

We refer to constructions which at time slot $i \in \{0, \ldots, t\}$ can access all future message packet sizes $(k_{i+1}, \ldots, k_t)$ as "offline." Offline schemes have access to the *sizes* but not the symbols of the future message packets. In contrast, when a code construction cannot access future message packet sizes, we denote it as "online." Thus, at time slot i, for an online construction, the future message sizes $(k_{i+1}, \ldots, k_t)$ are unknown. We distinguish between the feasible rates for offline and online coding schemes. We denote the best possible rate for offline coding schemes as the "offline-optimal-rate" and for online coding schemes as the "online-optimal-rate."

Under the setting of variable-size messages, it was shown in [4] that there is an inherent tradeoff between rate of a code and the decoding delay under lossless transmission (i.e. the number of time slots needed to decode a message packet when all channel packets are received). This tradeoff is captured in [4] via a new delay constraint called the *lossless-delay constraint*: When there are no losses, the receiver must decode each message packet $S[i]$ within a delay of τ_L ($< \tau$) time slots.[1] The lossless-delay constraint is relevant to applications which can infrequently tolerate a worst-case-delay of τ but require faster decoding for most message packets.

The valid value ranges for parameters b, τ, and τ_L are: $1 \leq b \leq \tau$ and $0 \leq \tau_L \leq \tau - b$. A maximum burst length of 0 is not considered, as coding is unnecessary in lossless transmission. Moreover, reliable transmission is impossible when b exceeds τ, as $S[i]$ cannot be decoded by its deadline when $X[i], \ldots, X[i+\tau]$ are all lost in a burst. The restrictions on τ_L hold without loss of generality. τ_L cannot be negative and $S[i]$ is decoded by time slot $(i + \tau - b)$ if there are no losses since a burst can eliminate $X[i+\tau-b+1], \ldots, X[i+\tau]$.

This paper uses the following notation. All vectors are row vectors. The length of a vector V is denoted $|V|$. A vector V is indexed using the notation $V = (V_0, \ldots, V_{|V|-1})$. Let A be a matrix with n columns and $I \subseteq \{0, \ldots, n-1\}$. The quantity A_I refers to restriction of A to the columns in I. The term $[n]$ denotes $\{0, \ldots, n\}$. For message packets $S[0], \ldots, S[t]$, we call $k_0, \ldots, k_t$ the "message size sequence."

The following conventions are used throughout this work. We restrict the parameter t to be at least τ and the final τ message packets to have size 0. This ensures that coding schemes can encode the final message packet of nonzero size using τ extra channel packets. Furthermore, these restrictions can be satisfied by appending τ message packets of size 0 to the stream of messages. This appending does not impact the rate of the code. Furthermore, for convenience of notation of edge cases, $k_{1-b}, \ldots, k_{-1}$ are each defined to be 0. Finally, a

[1] The notation of lossless-delay constraint has been changed from [4].

burst loss of $X[i], \dots, X[i+b-1]$ for $i \in \{1-b, \dots, -1\}$ denotes the burst loss of $X[0], \dots, X[i+b-1]$.

III. Online Code Constructions with Optimal Rate

In this section, we identify two broad parameter regimes where it is possible for online coding schemes to match the offline-optimal-rate. We then present online constructions which do so. Both the settings have unique characteristics that enable the online-optimal-rate to match the offline-optimal-rate. In the first regime, a simple scheme which encodes each message packet separately has optimal rate. Hence, the knowledge of future messages sizes does not provide any leverage. In the second regime, the lossless-delay constraint forces the encoder to send each message packet immediately rather than distributing its symbols over multiple channel packets. This serves to mitigate the potential advantage of offline schemes, enabling online coding schemes to attain the offline-optimal-rate. Later, in Section IV, we show that for all other parameter settings it is impossible for an online code construction to meet the offline-optimal-rate.

The two domains where the online-optimal-rate equals the offline-optimal-rate are: *Regime* 1*:* ($\tau_L = \tau - b$ and $b|\tau$) and *Regime* 2*:* ($\tau_L = 0$).

Under Regime 1, for any parameters (τ, b) in the regime, a simple online coding scheme applied to each message packet separately meets the upper bound on the rate of $\frac{\tau}{\tau+b}$. Each message packet $S[i]$ is partitioned evenly into $\frac{\tau}{b}$ components which are transmitted in channel packets $X[i], X[i+b], \dots, X[i+\tau-b]$ respectively. This satisfies the lossless-delay constraint. The summation of these components is sent in channel packet $X[i+\tau]$. At most one of $X[i], X[i+b], \dots, X[i+\tau-b], X[i+\tau]$ is lost in a burst of length b. Thus, the worst-case-delay constraint is satisfied.[2]

The remainder of this section focuses on Regime 2. Intuitively, in this regime, it is possible for an online coding scheme to match the offline-optimal-rate for the following reason: at each time slot i, for any code construction, at least k_i symbols are sent in channel packet $X[i]$ to meet the lossless-delay constraint. This eliminates the choice of distributing symbols corresponding to $S[i]$ over multiple channel packets.

We next present an online coding scheme for any (τ, b) which meets the offline-optimal-rate. We include a high level description, then present a toy example, and finally provide its details. The scheme can be viewed as extending the Generalized Maximally Short Codes presented in [14] to incorporate variability in the message size sequence. We call the proposed scheme the **(τ, b)-Variable-sized Generalized MS Code**.

Code construction (high level description). During time slot i, each message packet $S[i]$ is partitioned into two pieces $S[i] = (U[i], V[i])$. Redundant parity symbols $P[i] = (U[i-\tau] + P'[i])$ are created where $P'[i]$ consists of linear combinations (taken from a Cauchy matrix) of the symbols of $(V[i-\tau], \dots, V[i-1])$. Channel packet $X[i] = (S[i], P[i])$ is

[2]In a recent work [19], such a coding scheme where each packet is encoded separately was found to be useful in designing a low complexity streaming code with linear field size in the setting of fixed-size messages.

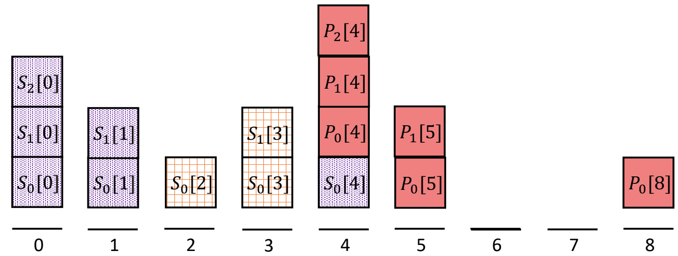

Fig. 2. A toy example of the $(4,2)$-Variable-sized Generalized MS Code. Each message packet $S[i] = (U[i], V[i])$ is transmitted in the corresponding channel packet $X[i]$ along with parity symbols $P[i]$ (when applicable). White boxes with purple dots represent symbols of $U[i]$, white boxes with an orange grid represent symbols of $V[i]$, and solid red boxes represent symbols of $P[i]$. The numbers under the lines indicate the time slots.

then sent. This satisfies the lossless-delay constraint ($\tau_L = 0$). $V[i]$ is defined to contain as many symbols of $S[i]$ possible while meeting the following constraint. For any burst loss of $X[j], \dots, X[j+b-1]$ which includes $X[i]$, the sum of the sizes of $V[j], \dots, V[i]$ is at most the number of parity symbols in $X[j+b], \dots, X[j+\tau-1]$ (i.e. the sum of the sizes of $P[j+b], \dots, P[j+\tau-1]$). The remaining symbols of $S[i]$ are allocated to $U[i]$. The size of $P[i]$ is defined to be $|U[i-\tau]|$.

Loss recovery. A burst loss eliminating $X[i], \dots, X[i+b-1]$ is recovered in two steps. First, for $j \in \{i+b, \dots, i+\tau-1\}$, $U[j-\tau]$ is canceled from $P[j]$ to obtain $P'[j]$. $P'[i+b], \dots, P'[i+\tau-1]$ are used to recover $V[i], \dots, V[i+b-1]$ at time slot $(i+\tau-1)$. Second, at time slot $j \in \{i+\tau, \dots, i+\tau+b-1\}$, $V[i], \dots, V[j-1]$ is used to compute $P'[j]$. Subtracting $P'[j]$ from $P[j]$ decodes $U[j]$.

Code construction (toy example). We present a toy example of the $(4,2)-$Variable-sized Generalized MS Code for message size sequence $k_0 = 3, k_1 = 2, k_2 = 1, k_3 = 2, k_4 = 1$, and $k_j = 0$ for $j \in \{5, \dots, 8\}$ in Figure 2. Each message packet $S[i]$ is sent in the corresponding channel packet $X[i]$ for $i \in [4]$. This satisfies the lossless-delay constraint. For $i \in \{0,1,4\}$, $U[i]$ is defined to equal $S[i]$. For $i \in \{2,3\}$ $V[i]$ is set to be $S[i]$. $P[4] = (S[0] + P'[4])$ is transmitted in $X[4]$ where $P'[4] = (S_0[2], S_0[3], S_1[3])$. $P[5] = (S[1] + P'[5])$ is sent in $X[5]$ for $P'[5] = (S_0[3], S_1[3])$. $P_0[8] = S_0[4]$ is transmitted in $X[8]$. If a burst occurs, within delay 3 of its start all lost symbols of $V[2]$ and or $V[3]$ are decoded. Any lost symbols of $U[0], U[1]$, and $U[4]$ are each decoded with delay exactly 4 using $P[4], P[5]$, and $P[8]$ respectively (and cancelling decoded symbols of $V[2]$ and $V[3]$). Therefore, the worst-case-delay constraint is satisfied.

Code construction (detailed description). At each time slot i, channel packet $X[i] = (S[i], P[i]) = (U[i], V[i], P[i])$ is sent. The scheme is formally described in three parts: initialization, partitioning $S[i]$ into $(U[i], V[i])$, and defining $P[i]$.

Initialization: The size $|P[i]|$ is set to 0 for $i \in [\tau - 1]$ and $k_{i-\tau}$ for $i \in \{\tau, \dots, \tau+b-1\}$. The quantities $U[i] = S[i]$ and $|V[i]| = 0$ are defined for $i \in [b-1]$. Let A be a $\tau m \times \tau m$ Cauchy matrix, where m is the maximum message packet size.

Partitioning $S[i]$*:* For any $i \geq b$, $S[i]$ is partitioned into $S[i] = (U[i], V[i])$ as follows. The auxiliary variable z_i is computed to encapsulate the minimum number of parity symbols available for use in recovering $S[i]$ when $X[i]$ is dropped in a burst (i.e. $z_i = \min_{j\in\{i-b+1,\ldots,i\}} \sum_{l=j+b}^{i+\tau-1} |P[l]| - \sum_{l=j}^{i-1} k_l$.). $V[i]$ is defined to be the first $min(k_i, z_i)$ symbols of $S[i]$. $U[i]$ is set to be the remaining symbols of $S[i]$. $|P[i+\tau]| = |U[i]|$ parity symbols are allocated to be sent in channel packet $X[i+\tau]$, although the actual symbols of $P[i+\tau]$ are not yet identified. This ensures for each burst in $X[j],\ldots,X[j+b-1]$ for $j \in \{i-b+1,\ldots,i\}$, the number of parity symbols sent after the burst by time slot $(i+\tau)$ (i.e. $\sum_{l=j+b}^{i+\tau} |P[l]|$) is enough to recover $S[j],\ldots,S[i]$.

Defining $P[i]$: $P[i]$ is constructed during time slot $(i \geq \tau)$ as follows. $P[i] = (U[i-\tau]+P'[i])$ where the symbols of $P'[i]$ are linear combinations of the symbols of $V[i-\tau],\ldots,V[i-1]$. The linear combinations are defined to ensure for any $j \in \{i-\tau+1,\ldots,i-b\}$, $V[j],\ldots,V[j+b-1]$ can be decoded using $V[j+b-\tau],\ldots,V[j-1], V[j+b],\ldots,V[j+\tau-2], P'[j+b],\ldots,P'[j+\tau-1]$. To meet this objective, the linear combinations are chosen from a Cauchy matrix, as described below. Let $V^*[j]$ be the length m vector obtained by appending $(m-|V[j]|)$ 0's to $V[j]$ for $j \in \{i-\tau,\ldots,i-1\}$. The length τm vector $E[i]$ is defined by placing each $V^*[j]$ into m consecutive positions of $E[i]$ starting with position $(j \mod \tau)m$. The Cauchy matrix A is used to define $P'[i] = E[i]A_{\{(i \mod \tau)m,\ldots,(i \mod \tau)m+|P[i]|-1\}}$.

We observe that the field size requirement is dictated by the Cauchy matrix and is at most $2\tau m$.

In Theorem 1, we verify that the Variable-sized Generalized MS Code meets the requirements of the model.

Theorem 1: For any parameters (τ, b) and message size sequence $k_0,\ldots,k_t$, the (τ, b)-Variable-sized Generalized MS Code satisfies the lossless-delay and worst-case-delay constraints over any $C(b, w > \tau)$ channel.

Proof sketch: The detailed proof is included in [1].

The lossless-delay constraint is satisfied since the scheme transmits $X[i] = (S[i], P[i])$ for $i \in [t]$.

We prove that the worst-case-delay constraint is satisfied by showing for any burst $X[i],\ldots,X[i+b-1]$ that each of $S[i],\ldots,S[i+b-1]$ are recovered within delay τ. First, we show that $V[i],\ldots,V[i+b-1]$ are recovered by time slot $(i+\tau-1)$. Second, we prove that $U[i],\ldots,U[i+b-1]$ are each recovered with delay exactly τ.

First, the construction identifies $P'[j]$ by time slot j by canceling $U[j-\tau]$ from $P[j]$ for $j \in \{i+b,\ldots,i+\tau-1\}$. The total number of parity symbols in $P'[i+b],\ldots,P'[i+\tau-1]$ is at least as many as $V[i],\ldots,V[i+b-1]$ by definition. $P'[i+b],\ldots,P'[i+\tau-1]$ can be used to decode $V[i],\ldots,V[i+b-1]$ by properties of the Cauchy matrix A.

Second, the scheme uses $V[j],\ldots,V[j+\tau-1]$ to compute $P'[j+\tau]$ for $j \in \{i,\ldots,i+b-1\}$. It then cancels $P'[j+\tau]$ from $P[j+\tau]$ to decode $U[j]$ with delay exactly τ.

■

The below Lemma 1 will later be used to prove Theorem 2. It essentially shows that whenever the Variable-sized Generalized MS Code transmits parity symbols in a channel packet, there is some burst loss for which all of these parity symbols are needed to satisfy the worst-case-delay constraint.

Lemma 1: Consider any parameters (τ, b) and message size sequence $k_0,\ldots,k_t$. For the (τ, b)-Variable-sized Generalized MS Code for all $i \geq \tau$ where $|P[i]| > 0$, $\exists j \in \{i-\tau-b+1,\ldots,i-\tau\}$ such that $\sum_{l=j}^{i-\tau} k_l = \sum_{l=j+b}^{i} |P[l]|$.

Proof: This holds for $i \in \{\tau,\ldots,\tau+b-1\}$ due to the initialization and a burst in $X[0],\ldots,X[b-1]$.

For $(i \geq \tau+b)$, if $(|P[i]| = |U[i-\tau]| > 0)$ then $(|V[i-\tau]| < k_{i-\tau})$. By definition of $V[i-\tau]$ there is some $j \in \{i-\tau-b+1,\ldots,i-\tau\}$ for which $|V[i-\tau]| = (\sum_{l=j+b}^{i-1} |P[l]| - \sum_{l=j}^{i-\tau-1} k_l)$. Thus, $(\sum_{l=j}^{i-\tau} k_l = \sum_{l=j+b}^{i} |P[l]|)$.

■

In Theorem 2, we show that the Variable-sized Generalized MS Code meets the offline-optimal-rate by proving that it transmits the minimum necessary number of symbols.

Theorem 2: For any parameters $(\tau, b, \tau_L = 0)$, the (τ, b)-Variable-sized Generalized MS Code attains the offline-optimal-rate over a $C(b, w > \tau)$ channel.

Proof sketch: The detailed proof is included in [1].

For an arbitrary message size sequence $k_0, k_1,\ldots,k_t$, consider any optimal offline construction O. We use a proof by induction on time slot $i = 0, 1, 2,\ldots,t$ to show that the cumulative number of symbols sent by O by time slot i is at least as many as that of the (τ, b)-Variable-sized Generalized MS Code. Thus, the rate, R_t, of the (τ, b)-Variable-sized Generalized MS Code is at least as high as that of O.

In the base case, for each $i \in [\tau-1]$, channel packet $X[i]$ under O must contain at least k_i symbols to satisfy the lossless-delay constraint for message packet $S[i]$. Under the (τ, b)-Variable-sized Generalized MS Code, $|X[i]| = k_i$.

The inductive step for $i \in \{\tau,\ldots,t\}$ has two cases.

First, when $X[i] = S[i]$ is sent under the (τ, b)-Variable-sized Generalized MS Code, at least $|S[i]| = k_i$ symbols are sent in $X[i]$ under O to meet the lossless-delay constraint.

Second, suppose $X[i] = (S[i], P[i])$ is sent under the (τ, b)-Variable-sized Generalized MS Code where $|P[i]| > 0$. Applying Lemma 1 shows that there is a burst loss starting at time slot $j \in \{i-\tau-b+1,\ldots,i-\tau\}$ for which the number of parity symbols received under the (τ, b)-Variable-sized Generalized MS Code in $X[j+b],\ldots,X[i]$ is exactly enough to decode message packet $S[j],\ldots,S[i-\tau]$. Combining this fact with satisfying the lossless-delay constraint for $S[j+b],\ldots,S[i]$ necessitates that at least as many symbols are sent under O between time slots $(j+b)$ and i as are respectively sent under the (τ, b)-Variable-sized Generalized MS Code. Applying the inductive hypothesis for time slot $(j+b-1)$ concludes the proof.

■

IV. Infeasiblity of offline-optimal-rate for Online Schemes

In Section III, we presented online code constructions which match the offline-optimal-rate under the two broad settings of Regime 1 and Regime 2. This motivates us to ask the question of whether there are any other parameter settings where an online coding scheme can attain the offline-optimal-rate. In this section, we show that the online-optimal-rate is strictly less than the offline-optimal-rate for all other parameter settings.

At a high level, the reason for online coding schemes being unable to match the offline-optimal-rate stems from the need to distribute symbols over multiple channel packets. For all parameter settings besides Regime 1 and Regime 2, the optimal approach to spreading symbols from a message packet $S[i]$ over $X[i], \ldots, X[i+\tau_L]$ can depend on the sizes of future message packets (i.e. $k_{i+1}, \ldots, k_t$). This dependency enables offline coding schemes to have higher rate than online coding schemes. This result is formally established in Theorem 3.

Theorem 3: For any parameters (τ, b, τ_L) outside of Regime 1 and Regime 2, the online-optimal-rate is strictly less than offline-optimal-rate.

A. Proof sketch of Theorem 3

The proof is divided into three cases shown in Lemmas 2, 3, and 4. The proof for each of the three lemmas uses the following line of argument. Two distinct message size sequences are introduced which are identical for the first several time slots. We show a lower bound on the offline-optimal-rate for the two message size sequences by presenting an offline coding scheme with rates R_1 and R_2 on the first and second message size sequence respectively. The manner in which symbols are sent to attain rate R_1 on the first message size sequence prohibits a code construction from having rate R_2 on the second. Combining Lemmas 2, 3, and 4 concludes the proof.

We provide the full proof for Lemma 2 below. Proofs for Lemmas 3 and 4 are provided in the extended version [1].

Lemma 2: For parameters $(\tau, b, \tau_L = \tau - b)$ where $(\tau_L \geq b)$, the online-optimal-rate is strictly less than offline-optimal-rate.

Proof: Let $(a = \lfloor \frac{\tau_L}{b} \rfloor)$ and $(e \equiv \tau_L \mod b)$. We note $(e > 0)$ since $(b \not| \tau)$. Let d be an arbitrary multiple of $(a+1)$.

Consider the following two message size sequences:

1) $k_j^{(1)} = d$ for $j \in [e-1]$ and $k_j^{(1)} = 0$ for $j \in \{e, \ldots, t\}$.
2) $k_j^{(2)} = d$ for $j \in [b-2]$, $k_{b-1}^{(2)} = d(\tau_L+1)$, and $k_j^{(2)} = 0$ for $j \in \{b, \ldots, t\}$.

We present an offline coding scheme for message size sequences 1 and 2 which has rates $R_1 = \frac{a+1}{a+2}$ and $R_2 = \frac{\tau}{\tau+b}$ on the two message size sequences respectively. We describe and then validate the scheme for each message size sequence.

On message size sequence 1, each message packet is encoded separately with parameters $(\tau' = \lfloor \frac{\tau}{b} \rfloor b, b' = b, \tau'_L = \tau' - b)$ as described in Section III and detailed below.

- $S[i]$ is evenly split into $S^{(0)}[i], \ldots, S^{(a)}[i]$ for $i \in [e-1]$.
- $X[i+zb] = S^{(z)}[i]$ is sent for $z \in [a]$ and $i \in [e-1]$.
- $X[i+(a+1)b] = \sum_{z=0}^{a} X[i+zb]$ is sent for $i \in [e-1]$.

The lossless-delay and worst-case-delay constraints are met since $S[i]$ is sent by time slot ab and at most one of $X[i], X[i+b], \ldots, X[i+ab], X[i+(a+1)b]$ is lost for $i \in [e-1]$.

Before the scheme for message size sequence 2 is described, we present a rate $\frac{\tau}{\tau+b}$ block code from [9] (or alternatively a block code from [6]–[8]) which the scheme will build on. The block code systematically maps τ input symbols $(s_0, \ldots, s_{\tau-1})$ to $(\tau+b)$ codeword symbols $(s_0, \ldots, s_{\tau-1}, p_0, \ldots, p_{b-1})$. For each $j \in [\tau-1]$ and any burst erasing up to b codeword symbols, the non-erased symbols of $(s_0, \ldots, s_{\tau-1}, p_0, \ldots, p_{min(b-1,j)})$ are sufficient to decode s_j. Hence, each symbol is recovered within τ symbols.

On message size sequence 2, the first $(b-1)$ message packets are sent with no delay and the next message packet is sent evenly over $X[b-1], \ldots, X[\tau-1]$. d symbols are sent in each of $X[\tau], \ldots, X[\tau+b-1]$ to creates d blocks of the code from [9]. The scheme is described in detail below.

- $X[j] = S[j]$ is sent for $j \in [b-2]$.
- $S[b-1]$ is divided evenly into $S^{(0)}[b-1], \ldots, S^{(\tau_L)}[b-1]$.
- $X[b-1+j] = S^{(j)}[b-1]$ is sent for $j \in [\tau_L]$.
- For each $z \in [d-1]$, an instance of the block code from [9] mapping $(X_z[0], \ldots, X_z[\tau-1])$ to $(X_z[0], \ldots, X_z[\tau-1], p_0^{(z)}, \ldots, p_{b-1}^{(z)})$ is created.
- $X[\tau+j] = (p_j^{(0)}, \ldots, p_j^{(d-1)})$ is sent for $j \in [b-1]$.

The lossless-delay constraint is satisfied, as each message packet is transmitted within delay τ_L. Each symbol $X_z[i]$ for $z \in [d-1]$ and $i \in [\tau-1]$ is decoded within delay τ by properties of the block code $(X_z[0], \ldots, X_z[\tau-1], p_0^{(z)}, \ldots, p_{b-1}^{(z)})$. Thus, the worst-case-delay constraint is met.

Due to the offline scheme, the offline-optimal-rate is at least R_1 and R_2 for message size sequences 1 and 2 respectively. Next, we show mutually exclusive conditions for sum of the sizes of $X[0], \ldots, X[e-1]$ to have rates at least R_1 and R_2 on message size sequences 1 and 2 respectively. All online coding schemes, thus, fail the condition for at least one message size sequence, since they are identical until time slot e.

Consider any coding scheme for message size sequence 1. At least de symbols are sent over $X[b], \ldots, X[t]$ since $X[0], \ldots, X[b-1]$ could be lost. At most $d\frac{e}{a+1}$ symbols can be sent over $X[0], \ldots, X[b-1]$ if the rate is at least R_1.

Consider an arbitrary coding scheme for message size sequence 2. At least $d\tau$ symbols are sent in $X[0], \ldots, X[\tau-1]$ to meet the lossless-delay constraint. Let $X^{(+)} \in \{(X[e+ib], \ldots, X[e+(i+1)b-1]) \mid 0 \leq i \leq a\}$. At least $d\tau$ symbols are sent outside of $X^{(+)}$ in case $X^{(+)}$ is lost. Each $|X^{(+)}| \leq db$ and at least $(d\tau - d(a+1)b = de)$ symbols are sent in $X[0], \ldots, X[e-1]$ if the rate is at least R_2.

Thus, for any online scheme with rate at least R_1 on message size sequence 1, at most $d\frac{e}{a+1}$ symbols are sent in $X[0], \ldots, X[b-1]$. Such a scheme necessarily has rate less than R_2 on message size sequence 2 since fewer than de symbols are sent in $X[0], \ldots, X[e-1]$.

■

Lemma 3: For parameters $(\tau, b, \tau_L = \tau - b)$ where $(\tau_L < b)$, the online-optimal-rate is strictly less than offline-optimal-rate.

Lemma 4: For parameters (τ, b, τ_L) where $(\tau_L < \tau - b)$, the online-optimal-rate is strictly less than offline-optimal-rate.

Acknowledgment

This work was funded in part by an NSF grant (CCF-1910813). The authors thank Francisco Maturana for his feedback in editing this work.

REFERENCES

[1] M. Rudow and K. Rashmi, "Online versus offline rate in streaming codes for variable-size messages," *arXiv preprint*, 2020.

[2] A. Badr, A. Khisti, W. Tan, and J. Apostolopoulos, "Perfecting protection for interactive multimedia: A survey of forward error correction for low-delay interactive applications," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 95–113, March 2017.

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

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

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

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

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

[8] D. Dudzicz, S. L. Fong, and A. Khisti, "An explicit construction of optimal streaming codes for channels with burst and arbitrary erasures," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 12–25, 2020.

[9] M. N. Krishnan, D. Shukla, and P. V. Kumar, "A quadratic field-size rate-optimal streaming code for channels with burst and random erasures," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 852–856.

[10] E. Domanovitz, S. L. Fong, and A. Khisti, "An explicit rate-optimal streaming code for channels with burst and arbitrary erasures," in *2019 IEEE Information Theory Workshop (ITW)*, 2019, pp. 1–5.

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

[12] Z. Li, A. Khisti, and B. Girod, "Correcting erasure bursts with minimum decoding delay," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov 2011, pp. 33–39.

[13] A. Badr, A. Khisti, W. Tan, and J. Apostolopoulos, "Streaming codes with partial recovery over channels with burst and isolated erasures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 501–516, April 2015.

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

[15] Y. Wei and T. Ho, "On prioritized coding for real-time streaming under packet erasures," in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*. IEEE, 2013, pp. 327–334.

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

[17] D. Leong, A. Qureshi, and T. Ho, "On coding for real-time streaming under packet erasures," in *2013 IEEE International Symposium on Information Theory*, July 2013, pp. 1012–1016.

[18] M. Haghifam, A. Badr, A. Khisti, X. Zhu, W. Dan, and J. Apostolopoulos, "Streaming codes with unequal error protection against burst losses," in *2018 29th Biennial Symposium on Communications (BSC)*, June 2018, pp. 1–5.

[19] M. N. Krishnan, V. Ramkumar, M. Vajha, and P. V. Kumar, "Simple streaming codes for reliable, low-latency communication," *IEEE Communications Letters*, pp. 1–1, 2019.