Error Rate Analysis for Random Linear Streaming Codes in the Finite Memory Length Regime

Pin-Wen Su*, Yu-Chih Huang†, Shih-Chun Lin‡, I-Hsiang Wang§, and Chih-Chun Wang*
*School of ECE, Purdue University, West Lafayette, USA, {su173, chihw}@purdue.edu
†Department of Communication, National Taipei University, Taipei, Taiwan, ychuang@mail.ntpu.edu.tw
‡Department of ECE, National Taiwan University of Science and Technology, Taipei, Taiwan, sclin@ntust.edu.tw
§Department of EE, National Taiwan University, Taipei, Taiwan, ihwang@ntu.edu.tw

***Abstract*—*Streaming codes* encode a string of source packets and output a string of coded packets in real time, which eliminate the queueing delay of block coding and are thus especially suitable for delay-sensitive applications. This work studies *random linear streaming codes* (RLSCs) and i.i.d. packet erasure channels. While existing works focused on the asymptotic error-exponent analyses, this work characterizes the error rate in the finite memory length regime and the contributions include: (i) A new *information-debt*-based description of the error event; (ii) A matrix-based characterization of the error rate; (iii) A closed-form approximation of the error rate that is provably tight for large memory lengths; and (iv) A new Markov-chain-based analysis framework, which can be of independent research interest. Numerical results show that the approximation, i.e. (iii), closely matches the exact error rate even for small memory length (≈ 20). The results can be viewed as a sequential-coding counterpart of the finite length analysis of block coding [Polyanskiy *et al. 10*] under the specialized setting of RLSCs.**

I. Introduction

Streaming codes are a class of sequential coding for which the encoder receives a string of source packets sequentially and outputs a string of coded packets in real time. Streaming codes can thus be viewed as generalizing the basic encoding unit of the *convolutional codes* from "bits" to "packets" and synchronizing the operation of the shift registers with the actual arrival, encoding, and transmission of the packets. By eliminating the concepts of queueing delay in block coding, streaming codes have significant potential for the delay-sensitive applications such as tele-/video conferencing, online gaming, live TV, and are actively studied as a possible solution to the ultra-reliable and low latency communication (URLLC) services in 5G [1].

The error rate analysis of sequential coding mostly follows the tree-code analysis of [2]. For any given finite memory length α, [2] first derives a genie-aided error-rate lower bound and a union-bound-based achievable error-rate upper bound, and then shows that they share the same error exponent and are thus asymptotically (exponentially) tight when α is sufficiently large, which generalizes the block-coding error-exponent analysis [3]–[5] for sequential coding. Nonetheless, while the genie-aided relaxation and the union bound do not alter the decay rate of the error probability, i.e., the error exponent, they are ill-suited when used to bracket[1] the error probability for arbitrary α.

Motivated by the novel techniques reported in [6] that strengthen the error exponent analysis of block coding with tighter achievability and converse bounds for the finite (codeword) length regime, this work studies the error rate of random linear streaming codes (RLSCs) in the finite memory length regime. The contributions of this work include: (i) A new definition of *information debt* that handles the finite memory setting, a generalization of the infinite-memory-based definition in [7, Chapter 9]; (ii) A matrix-based characterization of the error rate for any $\alpha < \infty$; (iii) A closed-form low-complexity approximation of the error rate that is provably tight for large memory lengths; and (iv) A new Markov-chain-based analysis framework, which can be of independent research interest. Contributions (ii), (iii), and (iv) are brand new developments that have no similar counterparts in [7]. Numerical results show that the proposed approximation, i.e., (iii), closely matches the exact error rate even for small memory length ($\alpha \approx 20$). The results thus represent the first analysis of RLSCs that characterizes the actual error probability of the finite memory length regime, not limited to just the decay rate.

A. Comparison to Other Existing Results

This work follows a probabilistic approach and the goal is to analyze the error rate of RLSCs under the i.i.d. channel model. For comparison, [8]–[15] take a deterministic approach. Namely, given a deterministic set of possible channel error patterns, the goal is to design the optimal streaming codes such that the original message can be perfectly decoded with zero error within a hard deadline constraint Δ, *for all channel realizations in the predefined set.* These two approaches are distinctly different where we use a stochastic channel model and the other one can be viewed as an *adversarial channel model*.

A closely related work is [16], which also uses a stochastic channel model. Specifically, for any fixed ratio of the memory length over the deadline that satisfies $\beta \triangleq \frac{\alpha}{\Delta} \geq 1$, [16] studies the error exponent (decay rate) when α and Δ jointly go to

This work was supported in parts by NSF under Grant CCF-1422997, Grant CCF-1618475 and Grant CCF-1816013, and by MOST Taiwan under Grant 107-2628-E-011-003-MY3.

[1] E.g., the bounds are usually of the form $e^{-\alpha E(R)+o(\alpha)}$ for which the little-o term $o(\alpha)$ can still dramatically change the value of the expression.

infinity while β is fixed. A critical finding is that if β exceeds a certain threshold $\beta^* \geq 1$, the error exponent stops improving and a finite ratio $\frac{\alpha}{\Delta} = \beta^*$ is as good as the infinite ratio $\frac{\alpha}{\Delta} = \infty$. In contrast, this work does not impose any deadline constraint, i.e., $\Delta = \infty$. Therefore, the ratio $\frac{\alpha}{\Delta} = 0$ for any finite α. Then we characterize the exact error probability under any finite α, not just the decay rate.

II. System Model and Basic Notation

Basic notation: The boldface lower and upper letters denote column vectors and matrices, respectively, e.g., $\mathbf{s}(t)$ denotes a column vector indexed by t. We use $\mathbf{s}_a^b$ to represent the *cumulative* column vector $\mathbf{s}_a^b \triangleq \left[\mathbf{s}^\top(a), \mathbf{s}^\top(a+1), \ldots, \mathbf{s}^\top(b)\right]^\top$. The operator $(\cdot)^+ \triangleq \max(0, \cdot)$. Matrix $\mathbf{I}_n$ stands for the identity matrix of size n. $\vec{\delta}_{\text{10-0}}$ (resp. $\vec{\delta}_{\text{0-01}}$) is a column vector with all entries being 0 except for the first one (resp. the last one). $\vec{\mathbf{1}}$ is a column vector of all 1s.

The encoder: Consider a slotted coding system. In every time slot $t \geq 1$, the encoder receives K packets, denoted by $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_K(t)]^\top$ where each packet $s_k(t)$ is of q bits and is drawn from the finite field GF(2^q). The encoder also stores the $\alpha \cdot K$ packets in the previous α slots $\{\mathbf{s}(\tau) : \tau \in [t-\alpha, t)\}$, where α is the *memory length*. Jointly, it uses the $(\alpha+1)K$ packets as input and outputs N coded packets $\mathbf{x}(t) = [x_1(t), \ldots, x_N(t)]^\top$. See Fig. 1 for illustration. Since we focus exclusively on linear codes, define $\mathbf{G}_t$ as the N-by-$(\min(\alpha+1, t) \cdot K)$ *generator matrix* for slot t, and we have

$$\mathbf{x}(t) = \mathbf{G}_t \mathbf{s}_{\max(t-\alpha,1)}^t. \quad (1)$$

The packet erasure channel: In each time slot t, the source transmits all N packets in $\mathbf{x}(t)$. A random subset of these N packets, denoted by $\mathcal{C}_t \subseteq \{1, 2, \cdots, N\}$, will arrive at the decoder perfectly and the complement of which is corrupted heavily and thus considered as erasure. The random set $\mathcal{C}_t$ is i.i.d. across t and we define $C_t \triangleq |\mathcal{C}_t|$ and $P_i \triangleq \Pr(C_t = i)$ as the probability of receiving i packets successfully.

The destination/decoder: The received packets, totally C_t of them, are denoted by $\mathbf{y}(t) = [y_1(t), \ldots, y_{C_t}(t)]^\top$. We write

$$\mathbf{y}(t) = \mathbf{H}_t \mathbf{s}_{\max(t-\alpha,1)}^t \quad (2)$$

where $\mathbf{H}_t$ is the projection of $\mathbf{G}_t$ onto the random (row index) set $\mathcal{C}_t$. The following notation of the *cumulative generator and receiver matrices* turns out to be very useful:

$$\mathbf{x}_1^t = \mathbf{G}^{(t)} \mathbf{s}_1^t \qquad \text{and} \qquad \mathbf{y}_1^t = \mathbf{H}^{(t)} \mathbf{s}_1^t, \quad (3)$$

where we properly shift and stack the instantaneous matrices $\mathbf{G}_t$ and $\mathbf{H}_t$ to create their cumulative representation $\mathbf{G}^{(t)}$ and $\mathbf{H}^{(t)}$, respectively. See Fig. 2 for illustration.

For any $t \geq 1$, $k \in [1, K]$, and $\tau \geq t$, we use $\vec{\delta}_{k,t}$ to denote the *location vector* of packet $s_k(t)$ at time τ, which is a (τK)-dimensional row vector for which the $(k+(t-1)K)$-th entry is one and all other entries are zero. We then have the following self-explanatory lemma.

Lemma 1. *A packet $s_k(t)$ is decodable by time $t + \Delta$ if and only if $\vec{\delta}_{k,t}$ is in the row space of $\mathbf{H}^{(t+\Delta)}$.*

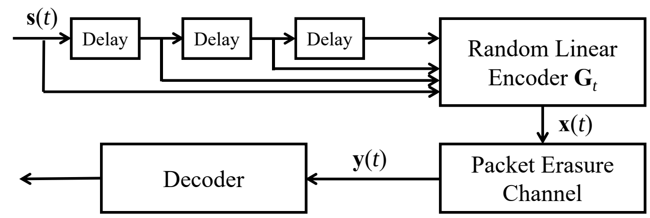

Fig. 1. The block diagram of the random linear streaming codes with $\alpha = 3$.

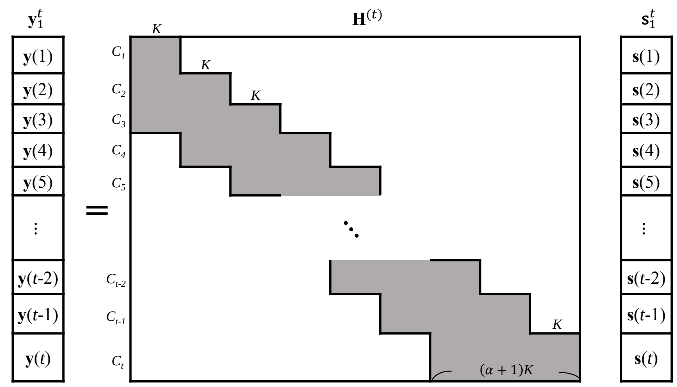

Fig. 2. The illustration of the cumulative receiver matrix in (3) with $\alpha = 2$. The gray area shows the non-zero entries.

Definition 1. *The vector $\mathbf{s}(t)$ is decodable by time $t + \Delta$ if all $\{s_k(t) : k \in [1, K]\}$ are decodable by time $t + \Delta$.*

The objective: Given any finite N, K, α and $\{P_i\}$, we aim to quantify the packet error rate p_e:

$$p_e \triangleq \lim_{T\to\infty} \lim_{\Delta\to\infty} \frac{1}{T} \sum_{t=1}^{T} \Pr\left(\mathbf{s}(t) \text{ is not decodable by time } t + \Delta\right). \quad (4)$$

A. Technical Assumptions

To avoid some corner cases in the analysis, we introduce some technical assumptions.

The Less-than-Capacity (LC) condition: This work assumes $0 < K < \mathbb{E}\{C_t\}$, i.e., we operate within the capacity.

Random linear streaming codes (RLSCs): We assume that each entry of $\mathbf{G}_t$ is chosen uniformly randomly from GF(2^q), excluding 0. RLSCs are known to be *capacity-achieving* when $(\alpha, \Delta) = (\infty, \infty)$, though being strictly suboptimal if $\Delta < \infty$ [7]. In this work, instead of quantifying the probabilistic behavior of RLSCs, we simply assume the RLSC encoder satisfies the following two deterministic conditions.

The Generalized MDS Condition: (i) All $N \cdot (\alpha+1)K$ entries in $\mathbf{G}_t$ are non-zero $\forall t$; and (ii) For any finite t and any finite sequence of pairs $\{(i_l, j_l) : l \in [1, L]\}$, define $S_R = \{i_l : l \in [1, L]\}$ and $S_C = \{j_l : l \in [1, L]\}$ and define $\mathbf{M}$ as the submatrix of $\mathbf{G}^{(t)}$ induced by S_R and S_C. The second half of the **MDS** condition requires that $\mathbf{M}$ is invertible for any t and any $\{(i_l, j_l) : l \in [1, L]\}$ satisfying (ii.a) $i_{l_1} \neq i_{l_2}$ and $j_{l_1} \neq j_{l_2}$ for any $l_1 \neq l_2$ and (ii.b) the (i_l, j_l)-th entry of $\mathbf{G}^{(t)}$ are non-zero for all $l \in [1, L]$.

Non-systematic construction (NS): We assume that for any distinct $k_1, k_2 \in [1, K]$ and any $t < \infty$, $s_{k_1}(t)$ is decodable if and only if $s_{k_2}(t)$ is decodable.

Remark 1: If the transmission only lasts for a bounded duration, then the probability of RLSCs satisfying the **MDS** condition approaches one when q approaches infinity. See the Schwartz-Zippel theorem in [17, Theorems 3 and 4]. Similarly, with close-to-one probability, RLSCs will fully "mix" all K packets in $\mathbf{s}(t)$ to a degree that to decode any one of these K packets requires the decodability of all K packets, i.e., the **NS** condition.

III. Main Results

A. Information-Debt Under Finite Memory

The first main result of this work is to generalize the concept of *information debt* originally defined in the infinite-memory setting [7], denoted by $I_d(t)$, and use it to characterize the error events for the finite memory setting.

Definition 2. *Define a constant $\zeta \triangleq \alpha K + 1$ and initialize $I_d(0) \triangleq 0$. For any $t \geq 1$, we iteratively compute*

$$\hat{I}_d(t) \triangleq (K - C_t + \min\{I_d(t-1), \alpha K\})^+ \tag{5}$$

$$I_d(t) \triangleq \min\left\{\zeta,\ \hat{I}_d(t)\right\}. \tag{6}$$

Clearly, $I_d(t)$ evolves according to the channel realization (linear equations delivered) $\{C_t : t\}$. The intuition behind $I_d(t)$ is as follows. The debt cannot be negative, hence the $(\cdot)^+$ in (5). Also, since the memory length is α, the maximum debt one can "carry forward" is at most αK, thus the $\min(\cdot, \alpha K)$ operation in (5). $\zeta \triangleq \alpha K + 1$ defines the absolute "ceiling" of the information debt, hence the $\min(\zeta, \cdot)$ in (6). The difference between $\zeta = \alpha K + 1$ and the maximum allowable debt αK is that the former represents the event that the information debt exceeds the maximum allowable debt, i.e., go bankrupt, while the latter represents the event of reaching the maximum allowable debt but still maintaining good standing. We introduce two minimum operations, one in (5) and one in (6), to capture the subtle distinction between the two.

Remark 2: When $\alpha = \infty$, the above definition collapses to the original infinite-momory-based definition in [7].

The following propositions show how the information debt can characterize the error event of the RLSCs. Define $t_0 \triangleq 0$ and $\tau_0 \triangleq 0$, and define iteratively

$$t_i \triangleq \inf\{t : t > t_{i-1}, I_d(t) = 0\} \tag{7}$$

$$\tau_i \triangleq \inf\{t : t > \tau_{i-1}, I_d(t) = \zeta\} \tag{8}$$

as the i-th time that $I_d(t)$ hits 0 and ζ, respectively.

Proposition 1. *Assume the* **MDS** *and* **NS** *conditions. For any fixed $i_0 \geq 0$, if there exists no $\tau_j \in (t_{i_0}, t_{i_0+1})$, then $\mathbf{s}(t)$ is decodable by time t_{i_0+1} for all $t \in (t_{i_0}, t_{i_0+1}]$. If there exists a $\tau_j \in (t_{i_0}, t_{i_0+1})$, define τ_{j^*} as the one[2] with the largest j. Then $\mathbf{s}(t)$ is decodable by t_{i_0+1} for all $t \in (\tau_{j^*} - \alpha, t_{i_0+1}]$.*

[2]By definition, $\tau_{j^*} < t_{i_0+1}$. Furthermore, $t_{i_0} + \alpha < \tau_{j^*}$ since by (5) for each time slot $I_d(t)$ can increase by at most K and it takes at least $\alpha + 1$ for $I_d(t)$ to start from $I_d(t_{i_0}) = 0$ to reach $I_d(\tau_{j^*}) = \zeta = \alpha K + 1$.

Proposition 2. *Continue from Proposition 1. None of $\{\mathbf{s}(t) : t \in (t_{i_0}, \tau_j^* - \alpha]\}$ is decodable by time $\tau_j^* - \alpha + \Delta$, regardless how large we set the deadline Δ.*

The intuition behind is quite straightforward. Whenever $I_d(t)$ hits 0 at time t_{i_0+1}, it means that we have observed enough linear equations, i.e., large $\{C_t : t\}$ in (5), and can thus start decoding from $\mathbf{s}(t_{i_0+1}), \mathbf{s}(t_{i_0+1} - 1), \cdots$, in a backward fashion. However, if $I_d(t)$ ever hits the bankrupt ceiling ζ during (t_{i_0}, t_{i_0+1}), say at time τ_{j^*}, then the temporal coupling between the earlier packets $\{\mathbf{s}(t) : t \leq \tau_{j^*} - \alpha\}$ and the later packets $\{\mathbf{s}(t) : t > \tau_{j^*} - \alpha\}$ is severed. The backward decoding thus cannot proceed beyond $\tau_{j^*} - \alpha$, see Propositions 1 and 2. The earlier packets $\{\mathbf{s}(t) : t \leq \tau_{j^*} - \alpha\}$ are forever "stranded" and cannot be decoded.[3]

B. Exact Error Rate Analysis

Note that the iterative definition of $I_d(t)$ in (5) and (6), and the assumption of i.i.d. C_t imply that $I_d(t)$ is a Markov chain with the state space being $\{0, 1, \cdots, \zeta\}$. Propositions 1 and 2 then imply that the packet error rate in (4) can be solved by analyzing the Markov chain $I_d(t)$.

Lemma 2. *Assuming the* **LC**, **MDS** *and* **NS** *conditions, we have*

$$p_e = \frac{\mathbb{E}\left\{\mathbb{1}_{\{\exists \tau_{j^*} \in (t_{i_0}, t_{i_0+1})\}} \cdot (\tau_{j^*} - \alpha - t_{i_0})\right\}}{\mathbb{E}\left\{t_{i_0+1} - t_{i_0}\right\}} \tag{9}$$

for any fixed i_0, where $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

The proof follows from Propositions 1 and 2 and that each round $(t_{i_0}, t_{i_0+1}]$ is a Markov renewal process.

Since the state space is $\{0, 1, \cdots, \zeta\}$, the transition matrix is of dimension $(\zeta+1)$-by-$(\zeta+1)$ and we denote it by $\Gamma = (\gamma_{i,j})$. To slightly abuse the notation, we assume the subscripts $i, j \in [0, \zeta]$, rather than the traditional range of $[1, \zeta + 1]$. The value $\gamma_{i,j}$, the intersection of the i-th row and j-th column of Γ, is the transition probability from state i to state j, i.e., $\gamma_{i,j} = \Pr(I_d(t) = j \mid I_d(t-1) = i)$. The actual value of $\gamma_{i,j}$ can be easily computed by the encoder parameters N, K, the channel distribution $\{P_i\}$, and the iterative update rules of $I_d(t)$ in (5) and (6). In the sequel, we thus assume Γ is known. Define $\boldsymbol{\phi} \triangleq \{1, 2, \ldots, \zeta - 1\}$ as the collection of non-boundary states. We then partition Γ into 9 sub-matrices:

$$\Gamma = \begin{bmatrix} \Gamma_{0,0} & \Gamma_{0,\boldsymbol{\phi}} & \Gamma_{0,\zeta} \\ \Gamma_{\boldsymbol{\phi},0} & \Gamma_{\boldsymbol{\phi},\boldsymbol{\phi}} & \Gamma_{\boldsymbol{\phi},\zeta} \\ \Gamma_{\zeta,0} & \Gamma_{\zeta,\boldsymbol{\phi}} & \Gamma_{\zeta,\zeta} \end{bmatrix}, \tag{10}$$

where $\Gamma_{\mathbf{x},\mathbf{y}} = [\ \gamma_{i,j}\ ], \forall i \in \mathbf{x}$ and $j \in \mathbf{y}$. Subsequently, define two $(\zeta+1)$-by-$(\zeta+1)$ matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ as follows.

$$\mathbf{M}_1 = \begin{bmatrix} 0 & \Gamma_{0,\boldsymbol{\phi}} & \Gamma_{0,\zeta} \\ 0 & \Gamma_{\boldsymbol{\phi},\boldsymbol{\phi}} & \Gamma_{\boldsymbol{\phi},\zeta} \\ 0 & \Gamma_{\zeta,\boldsymbol{\phi}} & \Gamma_{\zeta,\zeta} \end{bmatrix}, \mathbf{M}_2 = \begin{bmatrix} 0 & \Gamma_{0,\boldsymbol{\phi}} & \Gamma_{0,\zeta} \\ 0 & \Gamma_{\boldsymbol{\phi},\boldsymbol{\phi}} & \Gamma_{\boldsymbol{\phi},\zeta} \\ 0 & 0 & 0 \end{bmatrix}. \tag{11}$$

[3]The proof of Proposition 2 is highly nontrivial, though. One has to prove that those packets are undecodable regardless how one designs a decoding algorithm, which may be significantly different from the scheme used in the achievability proof in Proposition 1.

Proposition 3. *Assume the* **MDS**, **NS** *and* **LC** *conditions. The error rate* p_e *in* (9) *equals*

$$p_e = \frac{A_2 - \alpha \cdot A_1}{\mathbb{E}\{t_{i_0+1} - t_{i_0}\}} \tag{12}$$

where

$$\mathbb{E}\{t_{i_0+1} - t_{i_0}\} = \vec{\delta}_{\mathit{I0\text{-}0}}^{\top}(\mathbf{I}_{\zeta+1} - \mathbf{M}_1)^{-1}\vec{\mathbf{1}} \tag{13}$$

$$A_1 \triangleq \vec{\delta}_{\mathit{I0\text{-}0}}^{\top}(\mathbf{I}_{\zeta+1} - \mathbf{M}_2)^{-1}\vec{\delta}_{\mathit{0\text{-}0I}} \tag{14}$$

$$\mathbf{A}_3 \triangleq (\mathbf{I}_{\zeta-1} - \Gamma_{\boldsymbol{\phi},\boldsymbol{\phi}})^{-1} \tag{15}$$

$$A_2 \triangleq (\Gamma_{0,\zeta} + \Gamma_{0,\boldsymbol{\phi}}\mathbf{A}_3\Gamma_{\boldsymbol{\phi},\zeta})\left(\frac{1 + \Gamma_{\zeta,\boldsymbol{\phi}}(\mathbf{A}_3)^2\Gamma_{\boldsymbol{\phi},\zeta}}{1 - \Gamma_{\zeta,\zeta} - \Gamma_{\zeta,\boldsymbol{\phi}}\mathbf{A}_3\Gamma_{\boldsymbol{\phi},\zeta}}\right) + \Gamma_{0,\boldsymbol{\phi}}(\mathbf{A}_3)^2\Gamma_{\boldsymbol{\phi},\zeta}. \tag{16}$$

Since $t_{i_0+1} - t_{i_0}$, the first time $I_d(t)$ goes from state-0 back to state-0, is a stopping time, its expectation formula in (13) is easy to derive and follows directly from [18]. The main challenge of Proposition 3 is that τ_{j^*} in (9) is defined as the $\tau_j \in (t_{i_0}, t_{i_0+1})$ with the largest j^*. As a result, τ_{j^*} is *not* a stopping time, which prevents the use of various well-developed tools in Markov chain analysis. The multi-step computation in (14) to (16) is designed to handle the complication that τ_{j^*} is not a stopping time. We omit the derivation of (12) to (16) due to the limited space.

C. A Provably-Tight Closed-Form Error Rate Approximation

Based on matrix operations, the complexity of using Proposition 3 to compute p_e is $O((\alpha K)^3)$ assuming the computer program has no numerical precision problem when inverting matrices with large α and K. The third goal of this work is to derive an approximation formula of the form

$$p_e = (B_1 \cdot \alpha + B_2) \cdot \exp(-\alpha \cdot B_3) + o(\exp(-\alpha \cdot B_3)) \tag{17}$$

where the values of the constants B_1, B_2, and $B_3 > 0$ depend only on N, K, and $\{P_i\}$ but not on α. Note that the approximation formula (17) is a much stronger result than the classical error exponent analysis [2], also see footnote 1.

To that end, we introduce some conditions and definitions.

Lemma 3. *From the perspective of analyzing the Markov chain* $I_d(t)$*, we can assume* $P_0 > 0$ *and* $P_N > 0$ *without loss of generality.*

Proof. If $P_N = 0$, then we can effectively reduce N to $N-1$ and the Markov chain $I_d(t)$ will follow the same distribution.

The distribution of $I_d(t)$ is decided by two *orthogonal* factors: (i) the distribution of $(K - C_t)$ in (5), and (ii) the thresholds αK and ζ in (5) and (6). If $P_0 = 0$, then $\Pr(C_t \geq 1) = 1$. Since $K > 0$, we can reduce K to $K-1$ and C_t to $C_t - 1$, and the effects of (i) remain unchanged. If we still use the same threshold values in (ii), then the new $I_d(t)$ will follow the same distribution as the old $I_d(t)$. □

In the sequel, we assume exclusively $P_0 > 0$ and $P_N > 0$.

The Irreducible Markov-Chain (IMC) condition: In Markov chain analysis, one has to carefully handle irreducibility. For example, if K is even and $\Pr(C_t \text{ is even}) = 1$, then the Markov chain $I_d(t)$ only hops on even numbers and is thus reducible. To avoid this complications of being reducible, we impose that the values of K and N are coprime.[4]

Definition 3. *The* joint coding and channel characteristic *(JCCC) equation is*

$$x^{N-K} - \sum_{j=0}^{N} P_j x^{N-j} = 0. \tag{18}$$

The expressions of B_1 to B_3 depend on the roots of the JCCC equation. We now characterize the N roots of (18).

Lemma 4. *Assume the* **LC** *and* **IMC** *conditions. The following statements always hold: (i)* $x = 1$ *is a single root of* (18)*; (ii) there exists a positive value* $r > 1$ *such that* $x = r$ *is a single root; (iii) There is no other positive (real-valued) root other than* $x = 1$ *and* $x = r$*; (iv) there are exactly* $K-1$ *complex-valued roots satisfying* $|x| > r$*; (v) there are exactly* $N-K-1$ *complex-valued roots satisfying* $|x| < 1$.

Definition 4. *Continuing from Lemma 4, we say the encoder operates "sufficiently-close-to-capacity" (SCTC) if the following stronger version of statement (v) holds: (vi) there are exactly* $N-K-1$ *complex-valued roots satisfying* $|x| < \frac{1}{r}$.

Remark 3: Whether the **SCTC** condition holds can be easily verified by first using a numerical solver to find all N roots and then checking whether statement (vi) holds.

Example 1: Let $K = 2$, $N = 5$, and C_t be a binomial distribution with $p = \frac{K}{N} + 0.01 = 0.41$, i.e., $P_i = \binom{5}{i}p^i(1-p)^{5-i}$. We solve the JCCC equation numerically and the 5 roots are: $r_1 = -5.3106, r_2 = r = 1.0867, r_3 = 1, r_4 = -0.1253 + i0.1112$, and $r_5 = -0.1253 - i0.1112$ with precision until the fourth decimal point. It is clear that the roots satisfy all five statements in Lemma 4 and statement (vi) in Definition 4.

Remark 4: Suppose we relax the model of using a fixed K and allow the encoder to take a random number of K packets per slot. One can then rigorously prove that when the value of $\mathbb{E}\{C_t\} - \mathbb{E}\{K\}$ is sufficiently small (but still > 0), then statement (vi) in Definition 4 always holds. That is why we call this definition "sufficiently-close-to-capacity".

We now provide the formulas of the B_1 and B_3 in (17). The derivation is omitted due to space limits.

Proposition 4. *When* **LC**, **IMC** *and* **SCTC** *conditions are satisfied, we have*

$$B_3 = K \ln(r). \tag{19}$$

Remark 5: Since (vi) in Definition 4 holds, the unique positive root $r > 1$ becomes the *dominant root*, which in turn determines the error exponent B_3 in Proposition 4. In a broad sense, Definition 4 is in parallel to the classical error exponent result, which states that the error exponent of random block codes has two different expressions, depending on whether

[4] There are other, more complicated conditions that also imply **IMC**. If desired, we can even revise our statements to accommodate for *reducible* Markov chain as well. However, the added complexity will negatively impact readability. We thus use the simplest **IMC** condition herein.

$R < R_0$ or $R_0 < R < C$ where R_0 is the *cutoff rate*. The **SCTC** definition corresponds to the latter, more interesting case of $R_0 < R < C$.

To describe B_1, we notice that Lemma 4 and Definition 4 allow us to partition the N roots into 4 groups, the dominant root r, the unit root 1, and $K-1$ roots denoted by

$$\bar{r}_i : i \in [1, K-1] \text{ and } |\bar{r}_i| > r; \tag{20}$$

and $N-K-1$ roots denoted by

$$\underline{r}_j : j \in [1, N-K-1] \text{ and } |\underline{r}_j| < \frac{1}{r}; \tag{21}$$

Define the following $2 + (N-K-1)$ polynomials

$$\bar{f}(x) = \prod_{i=1}^{K-1} (x - \bar{r}_i), \quad \underline{f}(x) = \prod_{j=1}^{N-K-1} (x - \underline{r}_j), \tag{22}$$

$$\underline{g}_j(x) = \frac{\underline{f}(x)}{(x - \underline{r}_j)}, \ \forall j \in [1, N-K-1]. \tag{23}$$

Proposition 5. *Assume* **LC**, **IMC** *and* **SCTC** *conditions. Define* $\eta \triangleq N-K-1$. *If all* N *roots of the JCCC equation are* single roots, *then the* B_1 *value can be computed by*

$$B_1 = \frac{\bar{f}(1)\bar{f}^{-2}(r)\left(K\bar{f}(0) - \bar{f}(r)\sum_{k=1}^{K}(P_{K-k} \cdot b_k)\right)}{r \cdot B_4},$$

$$B_4 = \frac{\mathbb{E}\left\{(C_t - K)^+\right\} - \sum_{k=1}^{K} P_{K-k} \sum_{j=1}^{\eta} \frac{\underline{r}_j^{\eta}(\underline{r}_j^k - 1)}{(\underline{r}_j - 1)} \frac{\underline{g}_j(1)}{\underline{g}_j(\underline{r}_j)}}{\mathbb{E}\{C_t\} - K},$$

$$b_k = r^k - r^{-\eta}\left(\frac{\underline{f}(r)}{\underline{f}(1)} + \sum_{j=1}^{\eta} \frac{\underline{r}_j^{k+\eta}(r-1)}{(\underline{r}_j - 1)} \frac{\underline{g}_j(r)}{\underline{g}_j(\underline{r}_j)}\right).$$

Remark 6: Proposition 5 holds only when all N roots are single roots, which can be easily verified by a numerical root solver. In all our numerical exploration, we only see single roots. It is not hard to envision that the event of having a double root is of measure zero and the assumption is thus not restrictive in practice.

Remark 7: One must ensure that in the formulas of Proposition 5, the ranges of the summation/product are correctly specified and the denominators of all the fractions are non-zero. That is why a significant amount of efforts is spent on the careful discussion of the locations and multiplicities of the roots in Lemma 4, Definition 4, and Proposition 5.

We have also found a closed-form formula of B_2 in (17). As a second-order term hidden behind the dominant term $B_1 \cdot \alpha$, its expression is the most complicated. We omit the expression of B_2 due to the space limits.

IV. Numerical Verification

We use Example 1 described in Section III and the 5 roots listed there. Numerically plugging in the formulas in Section III-C, we have $B_1 = 0.1393, B_2 = -0.7893$, and $B_3 = 0.1662$. We then compare the results for different α values in Fig. 3. The curve "Approximation 1" plots the expression $(B_1\alpha + B_2)\exp(-\alpha B_3)$ and "Approximation 2" only plots $B_1\alpha\exp(-\alpha B_3)$ while ignoring the B_2 term. The curve "Exact" is obtained by Proposition 3. The curve "Simulation" is plotted by running Monte Carlo simulation on $I_d(t)$ and counting the erroneous packets using Propositions 1 and 2. We deliberately choose the (N, K, α) and $\{P_i\}$ so that the p_e is large ($\geq 10^{-2}$). In this way, simulation can be very accurate and serve as the ground truth. Our analytical results work equally well for small p_e ($< 10^{-6}$) that is beyond the reach of simulation. For example, if the binomial distribution parameter p in Example 1 is changed to $p = 0.45$, our formulas of Exact lead to $p_e = 4.52 \times 10^{-7}$ and 1.80×10^{-10} when $\alpha = 20$ and 30, respectively. The curve of Approximation 1 is indistinguishable from that of Exact, with the *relative gap* less than 0.1% for any $\alpha \geq 9$.

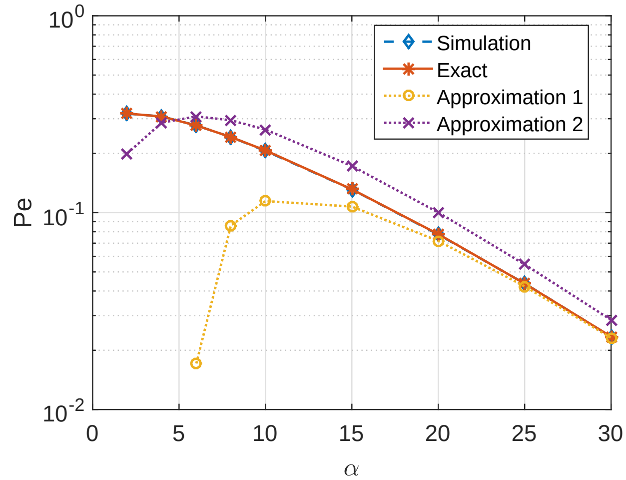

Fig. 3. Packet error rate p_e versus memory length α.

As expected, the simulation curve matches the exact error rate computation. The *absolute gap* between Approximation 1 and Exact is of order $o(\exp(-\alpha B_3))$, which becomes negligible after $\alpha > 20$. The strong characterization power of our approximation follows from the fact that we do not use any union bound or genie-aided techniques. Instead, the approximation is made by carefully quantifying the effects of the dominant root, using a new Markov-chain-based framework. When comparing Approximations 1 and 2, one can see that the seemingly harmless choice of discarding the B_2 term can substantially and negatively impact the accuracy of the approximation for small α, see the gap when $\alpha \in [20, 30]$.

V. Conclusion

We have proposed a new information debt definition to describe the random events of random linear streaming codes with finite memory length. We have derived a matrix-based procedure that computes the exact packet error rate. Additionally, we have provided a closed-form approximation of the error rate that is provably tight for large memory lengths. Numerical results have been used to demonstrate the characterization power of our derivations.

References

[1] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 124–130, Jun. 2018.

[2] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[3] R. G. Gallager, *Information Theory and Reliable Communication*. Springer, 1968, vol. 2.

[4] N. Merhav, "List decoding—random coding exponents and expurgated exponents," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 6749–6759, Nov. 2014.

[5] P. S. Rybin and F. I. Ivanov, "On estimation of the error exponent for finite length regular graph-based ldpc codes," *Journal of Communications Technology and Electronics*, vol. 63, no. 12, pp. 1518–1523, 2018.

[6] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[7] E. Martinian, "Dynamic information and constraints in source and channel coding," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

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

[9] A. Khisti and J. P. Singh, "On multicasting with streaming burst-erasure codes," in *2009 IEEE International Symposium on Information Theory*, Jun. 2009, pp. 2887–2891.

[10] A. Badr, A. Khisti, and E. Martinian, "Diversity embedded streaming erasure codes (de-sco): Constructions and optimality," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 1042–1054, May 2011.

[11] 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, Apr. 2015.

[12] 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.

[13] 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.

[14] 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)*, Jun. 2018, pp. 1809–1813.

[15] 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, Jul. 2019.

[16] S. C. Draper and A. Khisti, "Truncated tree codes for streaming data: Infinite-memory reliability using finite memory," in *2011 8th International Symposium on Wireless Communication Systems*, Nov. 2011, pp. 136–140.

[17] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[18] R. Durrett, *Essentials of Stochastic Processes*. Springer, 1999.