

Codes Correcting Burst and Arbitrary Erasures for Reliable and Low-Latency Communication

Serge Kas Hanna, Zhiyuan Tan, Wen Xu, Antonia Wachter-Zeh

▶ **To cite this version:**

Serge Kas Hanna, Zhiyuan Tan, Wen Xu, Antonia Wachter-Zeh. Codes Correcting Burst and Arbitrary Erasures for Reliable and Low-Latency Communication. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Jun 2023, Rhodes Island, France. pp.1-5, 10.1109/ICASSP49357.2023.10095948 . hal-05331002

HAL Id: hal-05331002
https://hal.science/hal-05331002v1

Submitted on 25 Oct 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Codes Correcting Burst and Arbitrary Erasures for Reliable and Low-Latency Communication

Serge Kas Hanna*, Zhiyuan Tan‡, Wen Xu§, and Antonia Wachter-Zeh†
* Department of Mathematics and Systems Analysis, School of Science, Aalto University, Finland
† Institute for Communications Engineering, Technical University of Munich, Germany
‡ Huawei Technologies Co., Ltd., Shanghai, China
§ Munich Research Center, Huawei Technologies Duesseldorf GmbH, Munich, Germany
Emails: serge.kashanna@aalto.fi, tanzhiyuan@huawei.com, wen.dr.xu@huawei.com, antonia.wachter-zeh@tum.de

***Abstract*—Motivated by modern network communication applications which require low latency, we study codes that correct erasures with low decoding delay. We provide a simple explicit construction that yields convolutional codes that can correct both burst and arbitrary erasures under a maximum decoding delay constraint T. Our proposed code has efficient encoding/decoding algorithms and requires a field size that is linear in T. We study the performance of our code over the Gilbert-Elliot channel; our simulation results show significant performance gains over low-delay codes existing in the literature.**

I. Introduction

In network communication, latency is defined as the time elapsed between the transmission of a packet from the source and its recovery at the receiver. Ensuring reliable communication with low latency is the main challenge in designing next-generation communication systems. To this end, we focus on studying error-correcting codes that ensure reliability in the presence of low decoding delay constraints. Our motivation stems from several modern applications such as Extended Reality (XR), Virtual Reality (VR), and Cloud Gaming.

In this work, we focus on forward error correction (FEC). FEC codes for correcting erasures (i.e., packet drops) with low decoding delay were first investigated by Martinian *et al.* in [1], [2]. The works in [1], [2] focused on designing codes that correct a burst of B erasures with a decoding delay that is at most T, and showed that the optimal code rate for this setting is $R = \frac{T}{T+B}$. In [2], Martinian and Trott presented an explicit rate-optimal code that can be constructed for all values of B, T, and R, with a linear field size $q = O(T)$. In [3], the authors generalized the study on codes with low decoding delay by introducing and studying a channel model that considers both burst and arbitrary erasures. This channel model was referred to as the sliding-window erasure channel, denoted by $C(N, B, W)$, where in any sliding window of size W, the channel can introduce either an erasure burst of maximum length B, or else, up to N arbitrary erasures. The authors in [3] showed that, without loss of generality, the channel parameter W could be set to $W = T + 1$, and the optimal code rate for this channel is $R = \frac{T+1-N}{T+1-N+B}$.

The term *streaming codes* is widely used in the literature to refer to codes that can correct *all* erasure patterns generated by the $C(N, B, W)$ sliding-window channel model. A series of works focused on designing rate-optimal streaming codes, e.g., [3]–[13]. These codes have different field size requirements in terms of T, ranging from exponential $q = O(e^T)$ to linear $q = O(T)$. Existing streaming codes with linear field sizes can be constructed only for certain values of N, B, and T, e.g., [7], [10]. The state-of-the-art optimal streaming code that can be constructed for *all* values of the parameters N, B, T, is the one that was recently presented by Domanovitz *et al.* [8], which requires a quadratic field size $q = O(T^2)$. Some works also considered variations of the $C(N, B, W)$ sliding-window channel, the authors in [14] proposed a code with cubic field size $q = O(T^3)$ that can correct most erasures when a window of size W contains a *single* arbitrary erasure *in addition* to a burst erasure of maximum size B.

The goal of introducing sliding-window channels such as the $C(N, B, W)$ channel in [3] was to provide a theoretically tractable approximation of the Gilbert-Elliot (GE) channel [15], [16], which is a random channel with memory that is commonly used to model packet erasures over networks. However, the accuracy of this approximation heavily depends on the values of N, B, T, and the GE channel parameters. Namely, some realizations of the GE channel may include both burst and multiple arbitrary erasures occurring in the same window, and such cases are not covered by the sliding-window erasure channels studied in [3] and [14]. These realizations could have a relatively high probability of occurrence depending on the values of the system parameters.

In our work, we adopt a different approach from the literature for constructing low-delay erasure codes, where we do not optimize the performance of the code over sliding-window channels of any kind. Instead, we focus on constructing low-delay codes with linear field size that can also correct erasure patterns consisting of both burst and multiple arbitrary erasures occurring simultaneously in the same window. Moreover, our code design is focused on minimizing the packet loss probability by allowing the recovery of a maximal number of erased packets in scenarios where the erasure pattern is beyond the error correction capability of the code. The parameters of our code are N, B, and T, where T represents the maximum decoding delay and (N, B) determines the erasure correction capability of the code. Our codes are explicit, systematic and can be constructed for all values of N, B, and T, with a linear field size $q \geq T$. We provide simulation results on the packet loss probability of our code over the GE channel and compare the results to existing constructions in the literature. The comparison shows that our code can outperform existing

streaming codes in terms of both packet loss probability and maximum delay, for the same code rate and lower field size. An important implication of our results is that constructing rate-optimal streaming codes with zero error over theoretical channels, such as sliding-window channels, does not necessarily translate into codes with optimal performance over practical channels, such as the GE channel.

II. Preliminaries

A. System Model

In what follows, we discuss coding approaches for tackling the low-delay packet transmission problem. Convolutional codes are typically used for the low-delay packet transmission setting, and the delay is measured in terms of the time units that one needs to wait for before being able to decode an erased packet. These codes can be constructed in multiple ways but they share the basic idea of introducing redundant symbols or packets through time. The redundancy is generated by encoding the information symbols or packets causally. More specifically, the parity symbols generated at time i can only depend on the information symbols at times $j \leq i$. We will be discussing two main approaches for constructing such convolutional codes that differ in terms of the allocation of redundancy across time instances. We will focus on systematic code constructions since the systematic property ensures that unerased packets are recovered with zero delay.

The first known approach [1] for designing low-delay codes is the following. Consider the setting where at each time i, a source generates an information packet denoted by $\mathbf{u}[i] = (u_1[i], u_2[i], \ldots, u_k[i]) \in \mathbb{F}_q^k$ consisting of k symbols over $\mathbb{F}_q$. At time i, a systematic encoder maps the information packet $\mathbf{u}[i]$ into an n-symbol coded packet denoted by

$$\mathbf{x}[i] = (u_1[i], \ldots, u_k[i], p_1[i], \ldots, p_{n-k}[i]) \in \mathbb{F}_q^n, \quad (1)$$

where the parity symbols $p_1[i], \ldots, p_{n-k}[i]$ are generated as a function of $\mathbf{u}[j]$ with $j \leq i$. The rate of the code is $R = k/n$. Our goal is to design the code such that it allows decoding an erased packet under a maximum delay constraint T, where T is a code parameter. Therefore, in the case where $\mathbf{x}[i]$ is erased, the underlying code should allow recovering the information packet $\mathbf{u}[i]$ by the time the packet $\mathbf{x}[i+T]$ is received.

In the previous description, a delay-constrained code with rate $R = k/n$ is constructed by introducing $n-k$ redundant parity symbols at each time i. Alternatively, a second known approach [17], [18] for constructing a code with the same delay properties and same rate $R = k/n$, is to transmit k information packets, followed by $n-k$ parity packets, in n consecutive time instances. More specifically, the first n transmitted packets would be given by

$$\mathbf{x}[i] = \begin{cases} \mathbf{u}[i] \in \mathbb{F}_q^k, & \text{for } i = 1, \ldots, k, \\ \mathbf{p}[i] \in \mathbb{F}_q^k, & \text{for } i = k+1, \ldots, n, \end{cases} \quad (2)$$

where $\mathbf{p}[k+1], \ldots, \mathbf{p}[n]$ are $n-k$ parity packets of length k that are generated as a function of the k preceding information packets $\mathbf{u}[1], \ldots, \mathbf{u}[k]$. This process can be repeated for every group of k information packets.

B. Diagonal and Horizontal Interleaving

Next, we present two known interleaving techniques for transforming a low-delay block code into a convolutional one with equivalent code properties. The importance of these interleaving techniques is that they reduce our problem to designing classical block codes that satisfy certain properties.

Consider an (n, k) block code $\mathcal{C} \subseteq \mathbb{F}_q^n$. Suppose that a codeword $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{C}$ is affected by erasures resulting in a string $\mathbf{y} = (y_1, \ldots, y_n)$, where $y_i \in \mathbb{F}_q \cup \{\star\}$. We say that an erased codeword symbol x_i is decoded with maximum delay T, if this symbol can be recovered using $(y_1, y_2, \ldots, y_{i+T^*})$, where $T^* = \min\{T, n-i\}$.The two interleaving techniques that we explain next, called diagonal and horizontal interleaving, allow transforming any block code $\mathcal{C}$ into a convolutional code that has the same code rate and equivalent decoding delay properties.

The diagonal interleaving technique [2] is applied in the setting corresponding to (1). In diagonal interleaving, the convolutional code is obtained from its block code counterpart $\mathcal{C}$ by positioning the codeword symbols diagonally across time instances, i.e., for all i, the symbols of the transmitted packet $\mathbf{x}[i] = (x_1[i], \ldots, x_n[i]) \in \mathbb{F}_q^n$ satisfy the following property $(x_1[i], x_2[i+1], x_3[i+2], \ldots, x_n[i+n-1]) \in \mathcal{C}$.

The horizontal interleaving technique [17], [18] is applied in the setting corresponding to (2). Let $\mathcal{I} \triangleq \{1 + mn \mid m \in \mathbb{N}\}$. In horizontal interleaving, each transmitted packet $\mathbf{x}[i]$ is of length k, such that for $i \in \mathcal{I}$ we have

$$\mathbf{x}[i+a-1] = \begin{cases} \mathbf{u}[i+a-1] \in \mathbb{F}_q^k, & a = 1, 2, \ldots, k, \\ \mathbf{p}^{a-k+1}[i+a-1] \in \mathbb{F}_q^k, & a = k+1, \ldots, n, \end{cases}$$

where $\mathbf{p}^1[i+k], \mathbf{p}^2[i+k+1], \ldots, \mathbf{p}^{n-k}[i+n-1]$ represent the $n-k$ parity packets generated by horizontally encoding the symbols of $\mathbf{u}[i], \mathbf{u}[i+1], \ldots, \mathbf{u}[i+k-1]$ using the underlying systematic code $\mathcal{C}$. Namely, for $i \in \mathcal{I}$ and $j = 1, \ldots, k$,

$$\left(u_j[i], \ldots, u_j[i+k-1], p_j^1[i+k], \ldots, p_j^{n-k}[i+n-1]\right) \in \mathcal{C}.$$

One can easily show that the aforementioned interleaving techniques result in a convolutional code that has equivalent decoding properties as the block code $\mathcal{C}$ [2], [9], [17], [18]. For more details on these interleaving techniques, we refer interested readers to [2], [18] and references therein.

III. Code Construction

A. Encoding

Suppose that we are given a target maximum delay T, and we choose certain code design parameters $B \geq 0$ and $N \geq 0$. Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be an (n, k) block code with $k = T - N$ and $n = T + B$, such that $B + N \leq T$ and $q \geq T$. Let $\mathbf{G}$ be the $k \times n$ generator matrix of $\mathcal{C}$. Let $\mathbf{u} = (u_1, \ldots, u_k) \in \mathbb{F}_q^k$ be the information message and $\mathbf{x} \in \mathbb{F}_q^n$ be the codeword. Since the code is systematic, we have $\mathbf{x} = \mathbf{uG} = (u_1, \ldots, u_k, p_1, \ldots, p_{n-k})$, where $p_1, p_2, \ldots, p_{n-k}$ represent the $n-k = B+N$ parity symbols. Next, we present our construction by giving the expressions of the parity symbols $p_1, \ldots, p_{B+N}$, as a function of the information symbols $u_1, \ldots, u_k$.

1) The first N parity symbols $p_1, p_2, \ldots, p_N$ are generated using a systematic MDS code over $\mathbb{F}_q$. For explicitness, the systematic Reed-Solomon (RS) encoder based on the Cauchy matrix construction is used.
2) The B parity symbols $p_{N+1}, p_{N+2}, \ldots, p_{N+B}$, are generated as interleaved parity checks with an interleaving factor of B, such that $\forall i \in \{1, 2, \ldots, B\}$,

$$p_{N+i} = u_i + u_{i+B} + \ldots + u_{i+(q-1)B} + \mathbb{1}_{\{i \leq r\}} u_{k-r+i}, \quad (3)$$

where $k = qB + r$ and $r = k \bmod B$. The final step of the construction is to transform this block code into a convolutional one by applying either diagonal or horizontal interleaving, as explained in Section II-B.

B. Decoding

To explain the decoding scheme, we start by examining the simple case when the total number of erasures in $(x_1, \ldots, x_{k+N})$ is $\leq N$. Notice that the punctured code obtained by only considering the first $k+N$ symbols of each codeword, is an $(k+N, k)$ systematic RS code. Hence, if the total number of erasures in the first $k+N$ symbols is $\leq N$, we apply RS decoding to recover these erasures. In this case, it follows from the code design that the erased symbols can be recovered with a delay that is $< T$.

Next, we examine the case when the total number of erasures in $(x_1, \ldots, x_{k+N})$ is $> N$. In this case, we are interested in decoding as many information symbols (packets) as possible while respecting the maximum delay constraint T. It follows from the code design that the first $B-1$ information symbols $u_1, u_2, \ldots, u_{B-1}$ fall within the maximum decoding delay constraint T since these symbols need to be decoded before the end of the code. Whereas the delay constraint for the remaining information symbols $u_B, u_{B+1}, \ldots, u_k$ is inactive because the constraint falls right at the end or past the end of the code. We refer to the symbols $u_1, \ldots, u_{B-1}$ as urgent symbols, and to $u_B, \ldots, u_k$ as non-urgent symbols.

We start our decoding process by decoding the urgent symbols $u_1, \ldots, u_{B-1}$ symbol-by-symbol from left to right. Suppose that u_1 is erased, we attempt to decode u_1 using the interleaved parity check p_{N+1}. If u_1 is the only summand that is erased in the interleaved parity check p_{N+1}, then u_1 can be successfully recovered by a simple addition/subtraction operation. Otherwise, u_1 is considered to be lost because it could not be decoded within its delay constraint. Then, similarly, we proceed to decode the next erased urgent symbol. If at any point of this symbol-by-symbol decoding process, the number of erased symbols in $(x_1, \ldots, x_{k+N})$ becomes $\leq N$, we use the punctured RS code to decode all symbols, and the decoding is subsequently terminated. In some scenarios, an urgent symbol may not be decodable with delay $\leq T$, but could be decoded at later stages of the decoding process with delay $> T$. In such scenarios, we still consider the urgent symbol to be lost since its recovery was not successful within the maximum decoding delay constraint T.

As for the non-urgent symbols $u_B, \ldots, u_k$, the general decoding approach for these symbols consists of two phases. In the first phase, we decode as many symbols as possible using the interleaved parity checks $p_{N+1}, \ldots, p_{N+B}$, given in (3). In the second phase, we decode the remaining symbols that were not recovered in phase 1 by using the RS parities $p_1, \ldots, p_N$. The worst-case overall decoding complexity is $O(T^2)$ [19].

C. Erasure Correction Capability

In this section, we discuss the erasure correction capability of the code in terms of the code parameters N, B, and T. Namely, we detail some of the cases where *all* erasures can be decoded, and we present a general condition under which the partial recovery of *some* erasures is possible.

Case 1: Suppose that the erasure pattern consists of $N' \leq N$ arbitrary erasures and no other erasures are present. In this case, all erased information symbols can be decoded using the punctured RS code obtained by only considering the first $k+N$ codeword symbols. Furthermore, it follows from the construction that the corresponding decoding delay is $< T$.

Case 2: Suppose that the erasure pattern consists of a burst of size $B' \leq B$ and no other erasures are present. In this case, we show that, under certain conditions, all the erased symbols can be decoded using the interleaved parity checks $p_{N+1}, \ldots, p_{N+B}$. We focus our discussion on the case where all the symbols are erased in a given burst since it is the most challenging one. Recall that $k = qB + r$, with $r = k \bmod B$. For $i \in \{1, 2, \ldots, B\}$, if $r = 0$, let $S_i = \emptyset$; otherwise, let $S_i = \{u_{k-r+i}\}$, and define the sets $U_i \triangleq \{u_i, u_{i+B}, u_{i+2B}, \ldots, u_{i+(q-1)B}\} \cup S_i$, which partition the set of information symbols $\{u_1, \ldots, u_k\}$. Notice that U_i contains all the information symbols that appear as summands in the interleaved parity check p_{N+i}. Assume that the burst erases the systematic symbols $u_{i'}, u_{i'+1}, \ldots, u_{i'+B'-1}$, with $i' \in \{1, \ldots, k-B'+1\}$. It follows from the construction of the interleaved parity checks that the $B' \leq B$ consecutive message symbols $u_{i'}, \ldots, u_{i'+B'-1}$ belong to different partitions U_i. Therefore, any burst of size $B' \leq B$ can erase at most one of the summands in $p_{N+1}, p_{N+2}, \ldots, p_{N+B}$. Hence, the erased symbols $u_{i'}, \ldots, u_{i'+B'-1}$ can be easily decoded by a simple subtraction/addition operation applied on the parity checks.

The argument above covers the case where the burst affects only the systematic symbols. Since the code is systematic, the case where the burst affects only the parity symbols is trivial. For this reason, we examine next the case where the burst affects some systematic and parity symbols simultaneously. Suppose that a systematic symbol u_i, $i \in \{1, \ldots, k\}$, appears as a summand in the interleaved parity check p_j, $j \in \{N+1, \ldots, N+B\}$. If $B|k$, i.e., $r = 0$, it follows from the definition of the interleaved parity checks and their positioning in the codeword that any burst of size $B' \leq B$ cannot erase u_i and p_j simultaneously. Namely, if $r = 0$, the construction guarantees the availability of every systematic symbol in at least one coded symbol after any $B' \leq B$ burst erasures. Hence, all erased information symbols can be decoded using $p_{N+1}, \ldots, p_{N+B}$. Also, if $B \nmid k$, i.e., $r > 0$, one can easily show that any B' burst erasures can be decoded if $B' \leq N+r$.

Furthermore, it follows from the construction that the decoding delay is $\leq T$ for all recovered symbols.

Case 3: Suppose that the erasure pattern consists of a burst of size $B' \leq B$ in addition to $1 \leq N' \leq N$ arbitrary erasures. We will focus on the case where $B' = B > N$. Consider the case where the burst affects the non-urgent symbols $u_B, \ldots, u_k$. In this case, the following general rule applies. The code can correct B burst erasures in addition to $N' \leq N$ arbitrary erasures if the number of erasures to be decoded in phase 2 of the decoding process (Section III-B) is $N_2 < N$, where N_2 denotes the number of erasures to be decoded in phase 2. To understand this, recall that any burst of size B can erase only one of the summands in $p_{N+1}, \ldots, p_{N+B}$. Hence, if N' arbitrary erasures occur in addition to the B burst erasures, then at least $B - N'$ out of the B interleaved parity checks $p_{N+1}, \ldots, p_{N+B}$ will have only one erased summand (that can be recovered by simple addition/subtraction operation). Therefore, by the end of decoding phase 1, at least $B - N'$ out of the $B + N'$ erasures can be successfully decoded. Consequently, the number of erasures to be decoded in phase 2 satisfies $N_2 \leq B + N' - (B - N') = 2N'$. Using the $(k + N, k)$ punctured code systematic RS code, up to N erasures can always be decoded in phase 2. Notice that if $N' \leq N/2$, then $N_2 \leq N$, and hence, all erased symbols can be decoded successfully. Whereas if $N' > N/2$, then some erasure patterns, but not all, can be corrected depending on the locations of the N' arbitrary erasures. We omit the discussion of more cases due to space limitations.

Furthermore, a key feature of our code is that it allows the recovery of a large number of erased symbols in many cases where the erasure pattern is not correctable. Note that this is contrary to codes such as (n, k) systematic RS codes, where not a single erasure can be recovered when the total number of erasures is $> n-k$. In practice, this decoding feature translates into a low packet loss probability as we show in Section IV. The general condition under which an information symbol can be recovered with maximum delay T is given in Proposition 1.

Proposition 1. *Let $\boldsymbol{\varepsilon} \in \{0,1\}^n$ denote an erasure pattern, where $\varepsilon_j = 1$, $j = 1, \ldots, n$, indicates that the codeword symbol x_i is erased. Let $\mathbf{e}_i \in \{0,1\}^k$, $i = 1, \ldots, k$, be the i^{th} canonical basis vector, and $\tilde{\mathbf{G}}(\boldsymbol{\varepsilon}_1^j) \in \mathbb{F}_q^{k \times n^*}$ be the generator matrix of the code $\mathcal{C}$ (Section III) punctured column-wise at the indices of $\boldsymbol{\varepsilon}_1^j = (\varepsilon_1, \ldots, \varepsilon_j)$ whose entries are zero, with $n^* = j - \sum_{i=1}^{j} \varepsilon_i$. For a given $\boldsymbol{\varepsilon}$, an information symbol u_i can be decoded with maximum delay T if and only if $\mathbf{e}_i$ is in the column span of $\tilde{G}(\boldsymbol{\varepsilon}_1^{i+T^*})$, with $T^* = \min\{T, n - i\}$.*

IV. Simulation Results

In this section, we compare the packet loss probability (PLP) of our proposed code with other existing constructions over the GE channel (Figure 1). The empirical PLP is computed as the ratio of the total number of lost information packets to the total number of transmitted information packets. The properties of the simulated codes are shown in Table I. Due to space limitations, we only show results for the case of $\alpha = 5 \times 10^{-3}$, $\beta = 0.45$, $\epsilon_1 = 1$, and varying $\epsilon_0 = \epsilon$.

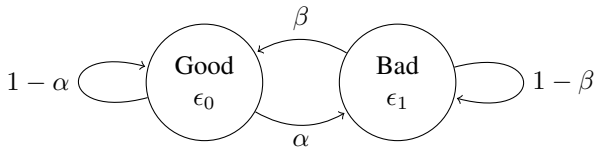


Fig. 1: 2-state Markov chain illustration of the GE channel. The erasure probabilities in the good and bad states are denoted by ϵ_0 and ϵ_1, respectively. The transition probabilities between the good and bad states are denoted by α and β.

	T	(n, k)	Rate	(N, B)
MDS Code	15	$(16, 8)$	0.5	$(8, 8)$
Martinian-Trott Code [2]	15	$(30, 15)$	0.5	$(1, 16)$
Domanovitz et al. Code [8]	15	$(24, 12)$	0.5	$(4, 12)$
New Code 1	15	$(22, 11)$	0.5	$(4, 7)$
New Code 2	14	$(20, 10)$	0.5	$(4, 6)$

TABLE I: The properties of the codes simulated in Figure 2.

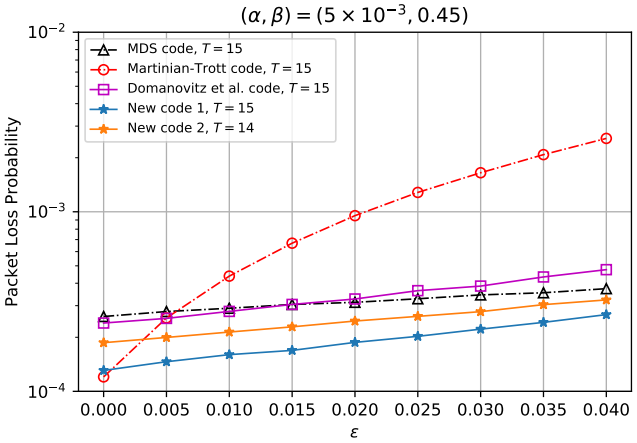


Fig. 2: PLP of the codes in Table I over the GE channel with α =5e-3, $\beta = 0.45$, and varying ϵ. The horizontal interleaving setting is considered (Section II-B) and the channel length is set to 10^7.

The results in Figure 2 show that for $\epsilon = 0$, the Martinian-Trott code, which is optimized for correcting bursts, achieves the best performance. However, its performance rapidly deteriorates as ϵ increases. For $0 < \epsilon \leq 0.04$, our code with $(N = 4, B = 7, T = 15)$ achieves the lowest PLP among all constructions. Also, for $\epsilon \geq 0.005$, our code with $(N = 4, B = 6, T = 14)$ presents a twofold gain since both the PLP and the maximum delay T are lower than that of other constructions. Here it is noteworthy to also mention that these constructions require different field sizes. For $R = 0.5$, the Martinian-Trott code can be constructed in $\mathbb{F}_2$, the MDS code and our code require a linear field size $q \geq T$, and the Domanovitz code requires a quadratic field size $q = O(T^2)$.

In conclusion, the results demonstrate that optimizing code constructions over theoretical sliding-window channels does not necessarily give the best performance over practical channels such as the GE channel. Namely, the results in Figure 2 show that with the same code rate and lower field size, our code can outperform existing rate-optimal streaming codes in terms of both packet loss probability and maximum decoding delay. As part of future work, our goal is to derive a theoretical bound on the PLP of our code over the GE channel.

REFERENCES

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

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

[3] A. Badr, P. Patil, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Layered constructions for low-delay streaming codes," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 111–141, 2016.

[4] 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)*, pp. 1809–1813, IEEE, 2018.

[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 Transactions on Information Theory*, vol. 65, no. 7, pp. 4274–4292, 2019.

[6] 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, 2019.

[7] V. Ramkumar, M. Vajha, M. N. Krishnan, and P. V. Kumar, "Staggered diagonal embedding based linear field size streaming codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 503–508, IEEE, 2020.

[8] 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)*, pp. 1–5, IEEE, 2019.

[9] M. N. Krishnan, D. Shukla, and P. V. Kumar, "Rate-optimal streaming codes for channels with burst and random erasures," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4869–4891, 2020.

[10] M. N. Krishnan, V. Ramkumar, M. Vajha, and P. V. Kumar, "Simple streaming codes for reliable, low-latency communication," *IEEE Communications Letters*, vol. 24, no. 2, pp. 249–253, 2019.

[11] 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 Transactions on Information Theory*, vol. 66, no. 7, pp. 4007–4018, 2020.

[12] A. Badr, A. Khisti, W.-t. Tan, X. Zhu, and J. Apostolopoulos, "Fec for voip using dual-delay streaming codes," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[13] 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, 2011.

[14] A. Badr, A. Khisti, W.-t. 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, 2015.

[15] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[16] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *The Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963.

[17] M. Vajha, V. Ramkumar, M. Jhamtani, and P. V. Kumar, "On the performance analysis of streaming codes over the gilbert-elliott channel," in *2021 IEEE Information Theory Workshop (ITW)*, pp. 1–6, 2021.

[18] M. Vajha, V. Ramkumar, M. Jhamtani, and P. V. Kumar, "On sliding window approximation of gilbert-elliott channel for delay constrained setting," *arXiv preprint arXiv:2005.06921*, 2020.

[19] E. R. Berlekamp, *Algebraic coding theory (revised edition)*. World Scientific, 2015.