Streaming Codes for Multiplicative Matrix Channels with Burst Rank Loss

Rafid Mahmood, Ahmed Badr, and Ashish Khisti

Abstract

The burst rank loss channel is an extension of the burst erasure channel where the channel matrix between the sender and receiver becomes rank-deficient for a certain period of consecutive time-slots. We study streaming communication over the burst rank loss channel, propose a new class of codes, ROBIN codes, and establish their optimality. Our construction uses the Maximum Rank Distance (MRD) and Maximum Sum Rank (MSR) codes from previous works as constituent codes, and combines them in a layered fashion. Our results generalize previous work on both the single-link and multiple-parallel-link streaming setups over burst erasure channels. We perform simulations over statistical network models to show that ROBIN codes attain low packet loss rates in comparison to existing codes.

I. Introduction

Streaming communication is characterized by two factors: causality and delay. Source packets arrive sequentially at a transmitter, which generates channel packets causally. The receiver collecting the transmitted packets must reconstruct the source within a decoding deadline, using only what it has observed up to that point. A receiving user is interested in sequential playback, meaning if a packet has not been recovered within the deadline, the decoder considers it lost and moves to the next packet. Error correcting codes capable of low delay decoding have previously been designed in the context of a single-link erasure channel connecting the transmitter and receiver [1]–[5]. In Internet communication, packet erasures primarily occur in bursts [6], and consequently the works of [1], [3]–[5] focus primarily on low-delay recovery from a burst of consecutive erasures. Alternatively, [2], [3], [5] also consider arbitrary loss patterns and guarantee low-delay recovery when there are fewer than the maximum tolerable number of erasures in a window.

The above mentioned works consider the case of a single communication link between the sender and receiver. As communication methods increase in sophistication, it is natural to consider streaming over a network where there are multiple links and paths connecting the source and destination nodes [7], [8]. The simplest extension of a single-link setting is the case when there are multiple parallel links. This extension has been previously studied for streaming in [7, Chapter 8], [9]. In the network setting. multiple parallel links correspond to separate paths chosen by a naive routing algorithm. Link failures in the network then lead to packet erasures over the associated path. It was shown that joint coding across packets transmitted over different paths, can outperform separate coding applied to individual paths only. An alternative to path routing is generation-based linear network coding, where the received packets are a linear transformation of the transmitted packets [10]–[12]. Links failures in the network now can potentially reduce the channel rank [13]. Rank metric codes such as Gabidulin codes can be used as end-to-end codes for protecting packets in such rank-deficient channels [14]–[17].

In this work, we extend streaming under the burst packet loss model, see e.g., [18], to a rank-deficient matrix channel. In the context of network coding, the rank is equal to the min-cut of the network, decreasing to a minimum tolerable rank when links fail in a burst [10], [13], [19]. For such a burst rank loss model, we design an end-to-end code with a layered structure and establish its optimality. Our approach generalizes the prior work in [7, Chapter 8], which addressed a network with naive routing and permitted at most one link to fail during a burst. While there, the problem simplified to multiple parallel links with one behaving as a periodic burst erasure channel, we permit both multiple bursts and linear transformations of the channel packets that can ensue as a result of network coding. Furthermore, their code construction involved diagonally interleaving

The authors are with the University of Toronto, Toronto, ON, Canada {rmahmood, abadr, akhisti}@comm.utoronto.ca. This work was supported by an Early Research Award from the Ministry of Ontario and an NSERC Discovery Grant. This work was presented in part at the IEEE International Symposium on Information Theory (ISIT), 2016, Barcelona, Spain.

MDS block codes. This technique seems specific to the parallel link case and does not easily generalize to the matrix channel. In reference [8], we introduced a new class of rank metric convolutional codes known as Maximum Sum Rank (MSR) codes. These codes maximize the column sum rank, which determines the maximum tolerable rank loss in a window and are network streaming counterparts of the m-MDS codes introduced in [20]. Our construction in this work uses both MSR codes and Gabidulin codes in a layering technique that can be seen as a generalization of that used for the burst correcting codes in [4], [5]. In comparison to the previous works, our constructions take advantage of the partial information recoverable in a rank-deficient channel in order to achieve a higher rate.

This paper is outlined as follows. We set up the network streaming problem by defining the properties of the encoder, network model, and decoder in Section II. Our main result is also stated there. In Section III, we review rank metric codes, including Gabidulin codes and MSR codes. Both of these can be applied to the network streaming problem, although they are not generally capacity achieving. We introduce a family of codes known as Recovery Of Bursts In Networks (ROBIN) codes in Section IV. These codes are constructed by layering Gabidulin and MSR codes as constituents. The encoding and decoding steps are detailed, proving that this code indeed achieves the maximum rate. A converse proof to the capacity is given in Section V. Finally, simulation results are provided in Section VI, where we evaluate the performances of the various codes over statistical network models. We conclude this paper with a discussion on the necessary field size, further improvements, and more robust network models to be addressed in future works.

The following notation is used throughout this work. We count all indices of vectors and matrices starting from the 0-th entry. A vector $\mathbf{x}_t = (x_{t,0}, \dots, x_{t,n-1}) \in \mathbb{F}_{q^M}^n$ is denoted in lower-case bold type, whereas a matrix $\mathbf{X}_t$ is identified using upper-case bold type. The first subscript refers to the time index of the vector. A sequence over multiple time indices $\mathbf{x}_{[t,t+j]} = (\mathbf{x}_t, \dots, \mathbf{x}_{t+j})$ is denoted using bracketed subscripts. This notation naturally extends to matrices, i.e., $\mathbf{A}_{[t,t+j]} = \operatorname{diag}(\mathbf{A}_t, \dots, \mathbf{A}_{t+j})$ is a block diagonal matrix. Moreover, block diagonal matrices constructed from copies of one base matrix use a further simplified notation. We write $\operatorname{diag}(\mathbf{A}_t; \nu) = \operatorname{diag}(\mathbf{A}_t, \dots, \mathbf{A}_t)$ to refer to a block diagonal construction of ν copies of $\mathbf{A}_t$.

II. Problem Setup and Main Result

Let q be a prime power and $\mathbb{F}_q$ the ground field with q elements. For $M \geq 0$, let $\mathbb{F}_{q^M}$ be an extension field of $\mathbb{F}_q$. Throughout this paper, we assume a network with the max-flow min-cut capacity r achieved using a random linear network code [12], [21], [22]. The channel matrix $\mathbf{A}_t \in \mathbb{F}_q^{r \times r}$ lies in the ground field and is always known to the decoder. In practice, this is achievable with the use of header bits that contain the coefficients for previous linear combinations. Thus, $\mathbf{A}_t$ is full rank and invertible when no failures occur, but decreases in rank when links deactivate. Additional redundancy in the form of an end-to-end code is considered in order to permit recovery from the rank-deficient matrix. This end-to-end code however is constructed over an extension field $\mathbb{F}_{q^M}$ with a sufficiently large M. The problem is defined in three steps: the encoder, network model, and decoder.

A. Encoder

At each time $t \geq 0$, a source packet $\mathbf{s}_t \in \mathbb{F}_{q^M}^k$ arrives causally at a transmitter node. Each source packet is assumed to be uniformly sampled and drawn independent of all other packets. A codeword $\mathbf{x}_t \in \mathbb{F}_{q^M}^n$ is a function of the prior source packets, i.e., $\mathbf{x}_t = \gamma_t(\mathbf{s}_0, \dots, \mathbf{s}_t)$, where $\gamma_t(\cdot)$ is the encoding function. We consider the class of linear time invariant encoders and focus on convolutional code constructions. A rate $R = \frac{k}{n}$ encoder with memory m generates the codeword

$$\mathbf{x}_t = \sum_{i=0}^{m} \mathbf{s}_{t-i} \mathbf{G}_i \tag{1}$$

using a set of generator matrices $\mathbf{G}_i \in \mathbb{F}_{q^M}^{k \times n}$ for $0 \leq i \leq m$. Furthermore, recall that due to the max-flow min-cut capacity, the network cannot transmit more than r channel packets in one time instance. We assume that $n = r\nu$ for some $\nu > 0$. The codeword $\mathbf{x}_t$ can be rearranged to a matrix $\mathbf{X}_t = \left(\begin{array}{c|c|c} \mathbf{x}_{t,0} & \dots & \mathbf{x}_{t,r-1} \end{array} \right) \in \mathbb{F}_{q^M}^{\nu \times r}$. In the matrix notation, the codeword is referred to as a channel macro-packet or a generation. We interchange between matrix and vector notation when one is

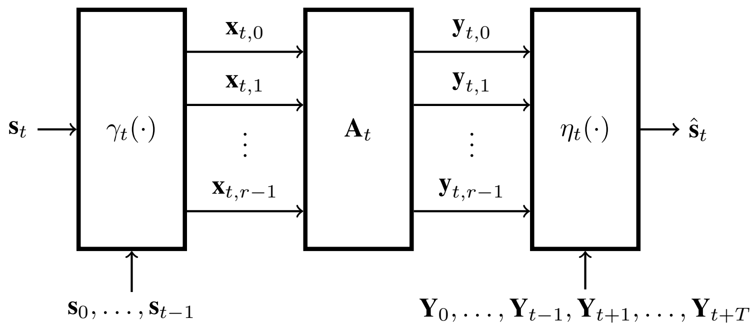


Fig. 1: The Encoder and Decoder are connected by r disjoint paths, each of which is responsible for a channel packet, or column of $\mathbf{X}_t$, of ν symbols at each time instance. The received packets are all linear combinations of the transmitted packets.

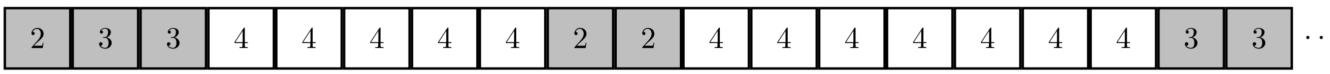


Fig. 2: An example of a Burst Rank Loss Network $\mathcal{CH}(4, 2, 3, 5)$. Every burst consists of at most $B = 3$ shots, with gaps lasting at least $G = 5$ shots. The rank of each channel matrix is labelled inside its associated box and does not fall below $r - \rho = 2$ during a burst.

preferable over the other in analysis. For $0 \leq i \leq r$, the i-th column of $\mathbf{X}_t$ is a vector $\mathbf{x}_{t,i}$ of length ν. This vector is referred to as a channel packet and there are r such packets that comprise a macro-packet.

B. Network Model

At each time instance, a macro-packet is transmitted over the network with assumed zero delay. Although link delays naturally affect the end-to-end transmission, we assume that a single time instance contains the encoding, transmission, and decoding of one channel packet[1]. The channel transfer matrix $\mathbf{A}_t \in \mathbb{F}_q^{r \times r}$ is determined by a random linear network code and is known to the decoder [12], [19], [23]. Thus for $t \geq 0$, the receiver observes $\mathbf{Y}_t = \mathbf{X}_t \mathbf{A}_t$. The network model is represented in Fig. 1 as the collection of r disjoint paths connecting the transmitter and receiver. All channel packets are transmitted simultaneously and the receiver observes r different linear combinations of the transmitted channel packets. We can write this relationship in the vector notation as $\mathbf{y}_t = \mathbf{x}_t \operatorname{diag}(\mathbf{A}_t; \nu)$, where $\mathbf{x}_t$ is the codeword in (1) and $\mathbf{y}_t \in \mathbb{F}_{q^M}^n$ is a received vector. Here, $\operatorname{diag}(\mathbf{A}_t; \nu)$ is the effective transfer matrix, a block diagonal matrix consisting of ν copies of $\mathbf{A}_t$.

A single network use is referred to as a shot. Each shot is independent of all others. For $j \geq 0$, the transfer matrix over a window $[t, t+j]$ of $j+1$ shots is given by $\mathbf{A}_{[t,t+j]} = \operatorname{diag}(\mathbf{A}_t, \ldots, \mathbf{A}_{t+j})$. The channel matrix in the window possesses a block diagonal structure consisting of the channel matrices at each time instance [19].

Let $\rho_t \triangleq \operatorname{rank}(\mathbf{A}_t)$. The rank of the channel in a window is equal to the sum rank of the blocks, i.e., $\sum_{i=t}^{t+j} \rho_i = \operatorname{rank}(\mathbf{A}_{[t,t+j]})$. At any time t, if the network operates perfectly, we have $\rho_t = r$ and the decoder simply inverts the channel matrix to retrieve the codeword. Unreliable links can modify the channel matrix to a rank-deficient form. The total rank deficiency of $\mathbf{A}_{[t,t+j]}$ is equal to $r(j+1) - \sum_{i=t}^{t+j} \rho_i$. We define below a model for a burst rank-deficient network that generalizes the classic burst erasure channel introduced in [18].

Definition 1. Consider a network where for all $t \geq 0$, the receiver observes $\mathbf{Y}_t = \mathbf{X}_t \mathbf{A}_t$. In the **Burst Rank Loss Network** $\mathcal{CH}(r, p, B, G)$, the channel matrix may have $\operatorname{rank}(\mathbf{A}_t) = \rho_t \geq r - p$ for a burst of at most B consecutive shots but must subsequently guarantee $\operatorname{rank}(\mathbf{A}_t) = r$ for a gap of at least $G \geq B$ consecutive shots.

An example of a Burst Rank Loss Network is provided in Fig. 2. Analysis of streaming video traffic suggests that internet streaming is susceptible primarily to burst losses [6]. Furthermore, these bursts often occur separated from each other by a significant margin. This motivates the practical constraint $G \geq B$. We use an adversarial model rather than a statistical model, as it allows for tractable analysis in terms of producing perfect low delay recovery. However, we perform simulations of our

[1]For example in audio streaming, coded speech packets are generated once every 20 ms. If the propagation delays are sufficinetly smaller than this value, they can be ignored.

codes over Markov channel models in Section VI and can show that the Burst Rank Loss Network satisfactorily approximates the more practical Markov models.

As the received channel packets are each linear combinations of the transmitted packets, the rank loss in $\mathbf{A}_t$ forces a number of received packets to become redundant. In single-link streaming, the number of erased or erroneous received packets is counted to determine the condition of a channel; we similarly count the number of redundant linear combinations in the network model. The rank deficiency of the channel matrix is equivalent in discussion to the number of redundant received packets.

The redundant received packets are not useful to the decoder. In analysis, the linearly dependent columns of the transfer matrix and the associated columns of the received macro-packet are discarded. The receiver effectively observes $\mathbf{Y}_t^* = \mathbf{X}_t\mathbf{A}_t^*$, where $\mathbf{A}_t^* \in \mathbb{F}_q^{r\times\rho_t}$ is the reduced channel matrix, containing only the linearly independent columns.

Remark 1. One motivation for modelling rank deficiencies in the channel matrix is that a network with fewer nodes or links can be represented as a channel matrix with a lower rank. According to [13], one deactivated link can damage at most one of the disjoint paths connecting the transmitter and receiver, thereby reducing the rank of $\mathbf{A}_t$ by at most 1.

C. Decoder

Let T be the maximum delay permissible by the decoder. A packet received at time t must be recovered by $t+T$, i.e., there should exist a sequence of functions that solves $\hat{\mathbf{s}}_t = \eta_t(\mathbf{Y}_0, \ldots, \mathbf{Y}_{t+T})$, where $\eta(\cdot)$ is the decoding function. If $\hat{\mathbf{s}}_t = \mathbf{s}_t$, then the source packet is perfectly recovered by the deadline; otherwise, it is declared lost. A code capable of decoding every source packet with delay T over $\mathcal{CH}(r, p, B, G)$ is defined as feasible over the network. A rate $R = \frac{k}{n}$ is achievable for delay T if there exists a feasible code with that rate. The supremum of all achievable rates for a network is the **streaming capacity**. The streaming capacity of the single-link burst erasure channel was derived in [5], [18]. We propose below the streaming capacity for the network setup, which is proved in later sections.

Theorem 1. The streaming capacity for the Burst Loss Network $\mathcal{CH}(r, p, B, G)$ with $G \geq B$ is given by

$$C = \begin{cases} \frac{T_{\text{eff}}r + B(r-p)}{(T_{\text{eff}}+B)r} & T \geq B \\ \frac{r-p}{r} & T < B \end{cases} \tag{2}$$

where $T_{\text{eff}} = \min(T, G)$ is the effective delay.

We make several remarks about Theorem 1.

Remark 2. When $T < B$, the recovery of the source packets must be initiated before the burst-loss terminates. In this case we show that applying a MRD code separately to each packet suffices to achieve the capacity. When $T \geq B$, we must use both the partial information available during the burst-loss, as well as the subsequent channel packets in the recovery process. This is accomplished using a layered coding scheme.

Remark 3. Setting $r = p$ in (2) recovers the streaming capacity for the single-link burst erasure channel [5]. Under this condition, the problem consists of all r paths between transmitter and receiver simultaneously disappearing, similar to an erasure of the entire macro-packet.

Remark 4. Upon setting $p = 1$ in (2), we recover the streaming capacity of the parallel link burst loss model introduced in [7, Chapter 8]. This earlier work featured r parallel paths between transmitter and receiver, with the restriction that only $p = 1$ path is erased in a burst within a window. In contrast, we relax $p \leq r$ as the maximum number of paths that may fail. Our proposed codes are significantly different from the construction given in [7, Chapter 8]. Our constructions involved a layered approach along the lines of [5] for the burst erasure channel. In contrast to [5], our choice of code parameters have been carefully selected to incorporate partial information available to the decoder during bursts, which does not arise in the erasure setting.

Remark 5. The upper bound above remains valid even when $G < B$. However, we do not address the achievability of this region in this work. The precursory work on single-link burst erasure streaming did not include a general construction for this scenario either. One consideration is that for large T, the $G < B$ region introduces the problem of low-delay recovery under multiple bursts rather than a single burst. We refer the reader to [24] for some specific code constructions that are provided for this problem for burst erasure channels.

III. Rank Metric Codes

A. Maximum Rank Distance (MRD) Block Codes

There exists an isomorphism between the extension field $\mathbb{F}_{q^M}$ and the vector space $\mathbb{F}_q^M$. Similarly for a vector space $\mathbb{F}_{q^M}^n$, there is an isomorphic matrix space $\mathbb{F}_q^{M\times n}$. Let $\phi_n : \mathbb{F}_{q^M}^n \to \mathbb{F}_q^{M\times n}$ be a bijection between the two. We then denote the column rank of a vector $\mathbf{x} \in \mathbb{F}_{q^M}$ as the rank of its isomorphic matrix $\phi_n(\mathbf{x})$. The rank distance between two vectors $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{F}_{q^M}^n$ is defined

$$d_R(\mathbf{x}, \hat{\mathbf{x}}) = \text{rank}\,(\phi_n(\mathbf{x}) - \phi_n(\hat{\mathbf{x}})).$$

Rank distance was proven to be a metric in [14]. For a linear block code $\mathcal{C}[n, k]$ over $\mathbb{F}_{q^M}$, the minimum rank is an analogue of minimum Hamming distance and must meet a Singleton-like bound given by $d_R \leq \min\left\{1, \frac{M}{n}\right\}(n-k)+1$. Maximum Rank Distance (MRD) codes achieve this bound with equality [14]. In addition, we assume $M \geq n$; an MRD code is then also MDS. These MRD codes possess the following property.

Theorem 2 (Gabidulin, [14])**.** Let $\mathbf{G} \in \mathbb{F}_{q^M}^{k\times n}$ be the generator matrix of an MRD code. The product of $\mathbf{G}$ with any full-rank matrix $\mathbf{A} \in \mathbb{F}_q^{n\times k}$ achieves $\text{rank}\,\mathbf{GA} = k$.

A complementary theorem was proved in [14] for the parity-check matrix of an MRD code. We use an equivalent property for the generator matrix, which arises from the fact that the dual of an MRD code is also an MRD code.

Gabidulin codes are the most common family of MRD codes in the $M \geq n$ regime [14]. Although they are naturally constructed via non-systematic generator matrices, systematic Gabidulin codes can be constructed by applying Gaussian elimination on a Gabidulin generator matrix. For the systematic version, we write each codeword as $\mathbf{x}_t = (\mathbf{s}_t, \mathbf{p}_t)$ where $\mathbf{p}_t$ consists of the parity-check symbols [23].

B. Maximum Sum Rank (MSR) Convolutional Codes

Let $\mathcal{C}[n, k, m]$ be a linear time invariant convolutional code with memory m over $\mathbb{F}_{q^M}$. For a source sequence $\mathbf{s}_{[0,j]} = (\mathbf{s}_0, \dots, \mathbf{s}_j)$, the codeword sequence $\mathbf{x}_{[0,j]} = \mathbf{s}_{[0,j]}\mathbf{G}_j^{\text{EX}}$ is calculated using the extended form generator matrix

$$\mathbf{G}_j^{\text{EX}} = \begin{pmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_j \\ & \mathbf{G}_0 & \dots & \mathbf{G}_{j-1} \\ & & \ddots & \vdots \\ & & & \mathbf{G}_0 \end{pmatrix}, \tag{3}$$

where $\mathbf{G}_j \in \mathbb{F}_{q^M}^{k\times n}$ and $\mathbf{G}_j = \mathbf{0}$ for $j > m$ [25].

The sum rank distance between codeword sequences $\mathbf{x}_{[0,j]}$ and $\hat{\mathbf{x}}_{[0,j]}$ is defined as the sum of the rank distance between each $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$. The j-th column sum rank of a code is defined

$$d_R(j) = \min_{\mathbf{x}_{[0,j]} \in \mathcal{C}, \mathbf{s}_0 \neq 0} \sum_{t=0}^{j} \text{rank}\,(\phi_n(\mathbf{x}_t))$$

as a convolutional extension to the minimum rank of a block code.

The column sum rank is bounded by $d_R(j) \leq (n-k)(j+1)+1$. It was shown in [8] that if $d_R(j)$ achieves the upper limit, then $d_R(i)$ does as well for all $i \leq j$. This naturally implies that $\mathbf{G}_0$ is also an MRD code generator matrix. The j-th column sum rank determines the maximum tolerable rank deficiency of the channel matrix in any window of j consecutive shots. MSR codes are convolutional codes with memory m that achieve the maximum $d_R(m)$. They possess the following relevant properties.

Lemma 1. Consider an MSR code $\mathcal{C}[n,k,T]$ with rate $R = \frac{k}{n}$ and $n = r\nu$ for some $\nu > 0$, that is being used to encode packets in a network for the channel $\mathbf{A}_{[t,t+T]}$. Suppose that all source packets before $\mathbf{s}_t$ are known to the decoder by some time $t+j$ for $j \leq T$.

1) If $\mathrm{rank}\left(\mathbf{A}_{[t,t+j]}\right) \geq R(j+1)r$, then the *first source packet* $\mathbf{s}_t$ is entirely recoverable by time $t+j$.
2) If $\mathrm{rank}\left(\mathbf{A}_{[t,t+T]}\right) \geq R(T+1)r$ with the additional constraint that

$$\mathrm{rank}\left(\mathbf{A}_i\right) = \begin{cases} r-p & i \in [t, t+B-1] \\ r & i \in [t+B, t+T] \end{cases}, \tag{4}$$

then *every source packet* $\mathbf{s}_t, \ldots, \mathbf{s}_{t+T}$ *in the window* is entirely recoverable by time $t+T$.

Proof: The proof for Statement 1 was given in [8] for $\nu = 1$. The extension to arbitrary ν is obvious when considering the channel in the vector notation. For completeness, we provide the proof in Appendix A.

For Statement 2, (4) in conjunction with the constraint that $\mathrm{rank}\left(\mathbf{A}_{[t,t+T]}\right) \geq R(T+1)r$ implies

$$Bp \leq (1-R)(T+1)r. \tag{5}$$

The left hand side is the rank deficiency of $\mathbf{A}_{[t,t+T]}$. The proof for Statement 2 is divided into two cases for when $p \leq (1-R)r$ and when $p > (1-R)r$.

Case 1: If $p \leq (1-R)r$, each source packet is recovered immediately after the channel macro-packet is received. The condition on p implies that $\mathrm{rank}\left(\mathbf{A}_t\right) = r - p \geq Rr$ for the first time slot. Using Statement 1 for $j = 0$, $\mathbf{s}_t$ is guaranteed recoverable at time t. We repeat the argument for each subsequent packet.

Case 2: If $p > (1-R)r$, each source packet is recovered at time $t+T$. Because $\mathrm{rank}\left(\mathbf{A}_{[t,t+T]}\right) \geq R(T+1)r$, $\mathbf{s}_t$ is recoverable at time $t+T$ using Statement 1 and letting $j = T$. In order to recover $\mathbf{s}_{t+1}$, we shrink the window of interest to $[t+1, t+T]$. Due to the block diagonal structure, the rank of the channel matrix is bounded

$$\begin{aligned} \mathrm{rank}\left(\mathbf{A}_{[t+1,t+T]}\right) &= \mathrm{rank}\left(\mathbf{A}_{[t,t+T]}\right) - \mathrm{rank}\left(\mathbf{A}_t\right) \\ &\geq R(T+1)r - r + p \\ &> RTr. \end{aligned}$$

The third line follows from the lower bound on p. Using Statement 1 for $j = T-1$ ensures that $\mathbf{s}_{t+1}$ is recoverable at time $t+T$. We then repeat the argument for the subsequent source packets $\mathbf{s}_{t+i}$ by showing that $\mathrm{rank}\left(\mathbf{A}_{[t+i,t+T]}\right) > R(T+1-i)r$ for all $0 \leq i \leq T$ and applying Statement 1 for $j = T - i$. ■

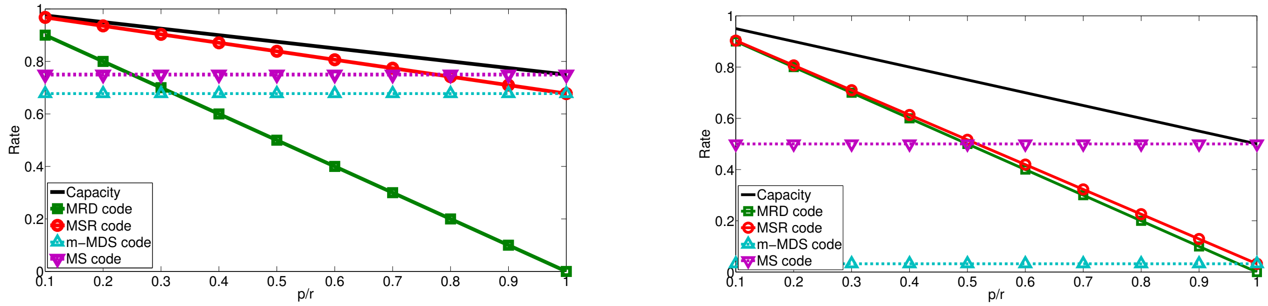
C. Numerical Comparison

Gabidulin codes can be directly used as end-to-end codes in network streaming. These codes are treated as convolutional codes with zero memory. Each source packet is encoded and decoded independently of the others. As a result, neither the burst length, gap length, nor the decoding deadline affect the code performance. The decoder successfully recovers the packet immediately if $\mathrm{rank}\left(\mathbf{A}_t\right)$ is sufficiently large; otherwise the packet is considered unrecoverable and discarded entirely. Thus, the only parameters of interest are r and p. In general, a Gabidulin code achieves perfect recovery of every packet in $\mathcal{CH}(r,p,B,G)$ when the rate is bounded

$$R \leq \frac{r-p}{r}. \tag{6}$$

Comparing with (2), we conclude that the streaming capacity of $\mathcal{CH}(r,p,B,G)$ is achievable with a Gabidulin code when $T < B$. However, for $T \geq B$, these codes generally provide sub-optimal rates.

MSR codes can also be used directly for network streaming. Lemma 1 considers two different channel loss patterns, the former being arbitrary rank losses in a window and the latter being a burst where several paths deactivate for consecutive shots. The total tolerable rank deficiency of the channel matrix is the same for both cases and the decoder does not perform



(a) Varying p for $\mathcal{CH}(10, p, 10, 30)$

(b) Varying p for $\mathcal{CH}(10, p, 30, 30)$

Fig. 3: A comparison of achievable rates using existing codes against the streaming capacity given in Theorem 1. The decoding deadline $T = 30$ is fixed. The burst length is $B = 10$ and $B = 30$ for the two setups, while the tolerable rank loss per time instance p is varied. The solid black line marking the capacity is achievable using our proposed code.

better for any specific pattern. An MSR code with memory $m = T_{\text{eff}}$ is feasible over $\mathcal{CH}(r, p, B, G)$ for

$$R \leq \max\left(\frac{(T_{\text{eff}}+1)r - Bp}{(T_{\text{eff}}+1)r}, \frac{r-p}{r}\right). \tag{7}$$

The two rates mark the different approaches an MSR code can take when decoding. The left value is drawn from rearranging (5) and is dominant when $T_{\text{eff}} \geq B$. If the burst length exceeds the code memory, i.e., $T_{\text{eff}} < B$, then the code must recover portion of the packets before the burst ends. Given that $\mathbf{G}_0$ acts as an MRD code, simply adopting the MRD strategy and recovering according to Theorem 2 yields a higher rate. Furthermore, $T_{\text{eff}} < B$ directly corresponds to the capacity when $T < B$. The streaming capacity of $\mathcal{CH}(r, p, B, G)$ is not achievable by MSR codes for $T \geq B$ in general. It is achievable with an MSR code only for $T < B$ and $B = 1$ when $T \geq B$.

Fig. 3 compares achievable rates for MRD and MSR codes in (6) and (7) respectively with the channel capacity in (2). Baseline streaming codes used in the single-link scenario are also compared. The m-MDS code is designed to maximize the column Hamming distance and acts effectively as a single-link version of the MSR code. The Maximally-Short (MS) code achieves the streaming capacity for the single-link burst erasure channel. Both of these codes treat rank-deficient channel matrices as erasure channels where the entire channel macro-packet is discarded. When $\text{rank}\,(\mathbf{A}_t) = r$, the decoders invert the channel matrix and apply the relevant single-link decoding technique. As a result, neither of these single-link codes vary their rate as a function of p.

Two different streaming setups are considered in Fig. 3, comparing a smaller $B = 10$ versus larger $B = 30$ for a fixed $r = 10$ and $T = 30$. The tolerable rank deficiency p is varied from 1 to r. When $p = r$, the channel is effectively an erasure channel where $\mathbf{A}_t$ is either a full rank matrix or simply the zero matrix. At this point, the single-link codes achieve their best rates relative to the capacity. This is also the area of highest disparity between the achievable rates of the MRD and MSR codes and the channel capacity, as the codes are designed to take advantage of the partial information available in a rank-deficient channel. While the MSR code excels in comparison to the other codes for small p, the code is equivalent to an m-MDS code when $p = r$. Moreover, the MRD code fails absolutely at this point, as the receiver does not observe any useful linear combinations of the transmitted channel packets.

IV. Recovery Of Bursts In Networks Codes

We prove the achievability of Theorem 1 by using a new family of codes designed to recover from a burst of rank-deficient channel matrices. These codes are referred to as ROBIN codes and achieve the streaming capacity over $\mathcal{CH}(r, p, B, G)$.

A. Encoder

Recall the prior assumption for the codeword length $n = \nu r$. For our construction, we further assume that $k = \kappa r$ for some $\kappa > 0$. Consequently, the code rate simplifies to $R = \frac{\kappa}{\nu}$ and the encoder can be designed for any pair of κ and ν. When

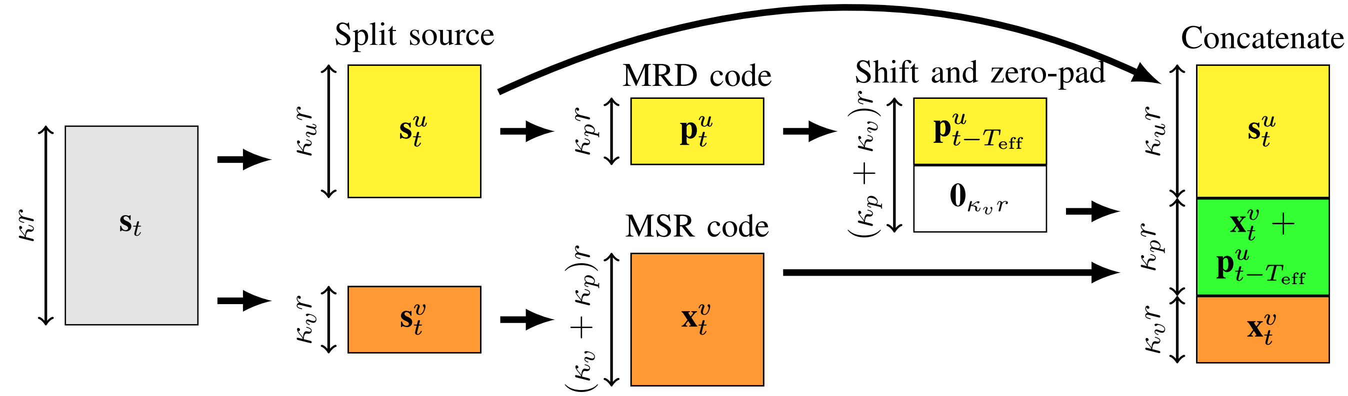


Fig. 4: A block diagram illustrating the encoding steps of a ROBIN code. The source packet is first split into two sub-packets and a different code is applied to each sub-packet. The resulting parity-checks are then combined to form the overall parity-check packet. Finally, the parity-check packet and the source packet are concatenated to generate the channel packet.

introducing the decoder, the values of κ and ν are fixed as a function of the channel parameters and decoding delay. The encoding steps are provided below and summarized in Fig. 4.

1) **Split source:** Let $\kappa_u, \kappa_v \geq 0$ be code parameters that satisfy $\kappa_u + \kappa_v = \kappa$. Split the source packet $\mathbf{s}_t \in \mathbb{F}_{q^M}^{\kappa r}$ into two groups of $\kappa_u r$ urgent ($\mathbf{s}_t^u$) and $\kappa_v r$ non-urgent ($\mathbf{s}_t^v$) source symbols, i.e.,

$$\mathbf{s}_t = (\underbrace{s_{t,0}, \ldots, s_{t,\kappa_u r-1}}_{\mathbf{s}_t^u}, \underbrace{s_{t,\kappa_u r}, \ldots, s_{t,\kappa_u r+\kappa_v r-1}}_{\mathbf{s}_t^v}). \tag{8}$$

The terminology of urgent and non-urgent symbols is inherited from prior layered streaming code constructions [1], [5]. The intuition is as follows. When decoding, all non-urgent source symbols affected by a burst are recovered simultaneously once the decoder receives sufficient information. In contrast, the urgent source symbols are recovered immediately as the associated parity-check symbols arrive.

2) **MSR code:** Let $\kappa_p > 0$ be a code parameter, to which we assign a value later. Apply an MSR code $\mathcal{C}[(\kappa_v+\kappa_p)r, \kappa_v r, T_{\text{eff}}]$ over the non-urgent source component $\mathbf{s}_t^v$ in order to generate the non-urgent codeword

$$\mathbf{x}_t^v = \sum_{i=0}^{T_{\text{eff}}} \mathbf{s}_{t-i}^v \mathbf{G}_i.$$

Here, $\mathbf{G}_i$ follow (1) for $m = T_{\text{eff}}$.

3) **MRD code:** Apply a systematic MRD code $\mathcal{C}[(\kappa_u+\kappa_p)r, \kappa_u r]$ over the urgent source sub-packet $\mathbf{s}_t^u$ in order to generate the urgent codeword

$$(\mathbf{s}_t^u, \mathbf{p}_t^u) = \mathbf{s}_t^u \mathbf{G}. \tag{9}$$

4) **Shift and zero-pad:** Shift the parity-check symbols $\mathbf{p}_t^u$ in time with delay T_{eff} for later use. At time t, we receive the the delayed parity-check symbols $\mathbf{p}_{t-T_{\text{eff}}}^u$. Zero-pad this vector with $\kappa_v r$ symbols in order to construct $(\mathbf{p}_{t-T_{\text{eff}}}^u, \mathbf{0}_{\kappa_v r})$. The zero-padded vector now has the same length as $\mathbf{x}_t^v$.

5) **Concatenate:** Concatenate the urgent source sub-packet with the summation of the non-urgent and parity-check symbols to construct the codeword

$$\mathbf{x}_t = (\underbrace{\mathbf{s}_{t,0}^u, \ldots, \mathbf{s}_{t,\kappa_u r-1}^u}_{\text{urgent}}, \underbrace{\mathbf{x}_{t,0}^v + \mathbf{p}_{t-T_{\text{eff}},0}^u, \ldots, \mathbf{x}_{t,\kappa_p r-1}^v + \mathbf{p}_{t-T_{\text{eff}},\kappa_p r-1}^u}_{\text{overlapped}}, \underbrace{\mathbf{x}_{t,\kappa_p r}^v, \ldots, \mathbf{x}_{t,(\kappa_p+\kappa_v)r-1}^v}_{\text{non-urgent}}). \tag{10}$$

We simplify the notation to $\mathbf{x}_t = (\mathbf{s}_t^u, \mathbf{x}_t^v + (\mathbf{p}_{t-T_{\text{eff}}}^u, \mathbf{0}_{\kappa_v r}))$.

The codeword is effectively divided into three partitions: the urgent source symbols $\mathbf{s}_t^u$, the non-urgent codeword symbols from $\mathbf{x}_t^v$, and the overlapped parity-check symbols from the summation of $\mathbf{x}_t^v$ and $\mathbf{p}_{t-T_{\text{eff}}}^u$. The three partitions contain $\kappa_u r$, $\kappa_v r$, and $\kappa_p r$ symbols respectively. The channel macro-packet is formed by sequencing the codeword into vectors of length r. The codeword can be separated into κ_u, κ_v, and κ_p vectors comprised solely of urgent, non-urgent, and overlapped parity-check symbols respectively. Each of these vectors becomes a row of the macro-packet

$$\mathbf{X}_t = \left(\begin{array}{ccc} \mathbf{s}_{t,0}^u & \cdots & \mathbf{s}_{t,r-1}^u \\ \mathbf{s}_{t,r}^u & \cdots & \mathbf{s}_{t,2r-1}^u \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{t,(\kappa_u-1)r}^u & \cdots & \mathbf{s}_{t,\kappa_u r-1}^u \\ \hline \mathbf{x}_{t,0}^v + \mathbf{p}_{t-T_{\text{eff}},0}^u & \cdots & \mathbf{x}_{t,r-1}^v + \mathbf{p}_{t-T_{\text{eff}},r-1}^u \\ \mathbf{x}_{t,r}^v + \mathbf{p}_{t-T_{\text{eff}},r}^u & \cdots & \mathbf{x}_{t,2r-1}^v + \mathbf{p}_{t-T_{\text{eff}},2r-1}^u \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{t,(\kappa_p-1)r}^v + \mathbf{p}_{t-T_{\text{eff}},(\kappa_p-1)r}^u & \cdots & \mathbf{x}_{t,\kappa_p r-1}^v + \mathbf{p}_{t-T_{\text{eff}},\kappa_p r-1}^u \\ \hline \mathbf{x}_{t,\kappa_p r}^v & \cdots & \mathbf{x}_{t,(\kappa_p+1)r-1}^v \\ \mathbf{x}_{t,(\kappa_p+1)r}^v & \cdots & \mathbf{x}_{t,(\kappa_p+2)r-1}^v \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{t,(\kappa_p+\kappa_v-1)r}^v & \cdots & \mathbf{x}_{t,(\kappa_p+\kappa_v)r-1}^v \end{array}\right).$$

A single column of $\mathbf{X}_t$ contains κ_u, κ_v, and κ_p urgent, non-urgent, and overlapped symbols respectively. The columns are each channel packets. The rank of $\mathbf{A}_t$ decreasing by one implies that a column of $\mathbf{X}_t$, or $\nu = \kappa_u + \kappa_v + \kappa_p$ linear combinations of symbols become redundant.

B. Decoder

We set the code parameters as follows:

$$\kappa = T_{\text{eff}} + B(1 - \frac{p}{r}), \kappa_u = B, \kappa_v = T_{\text{eff}} - B\frac{p}{r}, \kappa_p = B\frac{p}{r}. \tag{11}$$

It is assumed that (11) yields integer valued results, but this may not always be the case in practice. When a parameter is not an integer, we multiply all code parameters by a normalizing factor. As ν is simply a function of the parameters, the normalizing factor affects it as well, leaving the code rate unchanged.

The urgent source sub-packet is protected by an MRD code $\mathcal{C}[B(r+p), Br]$ and the non-urgent source sub-packet is protected by an MSR code $\mathcal{C}[T_{\text{eff}} r, T_{\text{eff}} r - Bp, T_{\text{eff}}]$. We show below that the decoder with the parameters in (11) can completely recover all packets from a burst of length B when there is a sufficiently large gap $G \geq B$. The procedure consists of two steps. The non-urgent source symbols erased in the burst are simultaneously recovered first. Then, the urgent source symbols are iteratively recovered with delay T_{eff}.

Suppose that there is a burst beginning at time t, i.e., $\text{rank}\,(\mathbf{A}_i) = r - p$ for $i \in [t, t+B-1]$. All prior source packets are known to the decoder and communication is perfect afterwards for $[t+B, t+B+T_{\text{eff}}-1]$. Recall that $T_{\text{eff}} \leq G$. We show that all source packets $\mathbf{s}_{[t,t+B+T_{\text{eff}}-1]}$ are recoverable within their respective decoding deadlines. The steps to recover $\mathbf{s}_t$ by time $t + T_{\text{eff}}$ are detailed below. The remaining source packets affected by the burst are decoded using the same steps.

Step 1: The decoder recovers the non-urgent $\mathbf{s}_{[t,t+T_{\text{eff}}]}^v$ by time $t + T_{\text{eff}}$. This step is divided into two actions. To begin, $\mathbf{s}_{[t,t+T_{\text{eff}}-1]}^v$ are recovered simultaneously at time $t + T_{\text{eff}} - 1$. At time t, the decoder then recovers $\mathbf{s}_{t+T_{\text{eff}}}^v$.

All source packets before time t are known to the decoder, meaning $\mathbf{p}_{[t-T_{\text{eff}},t-1]}^u$ can be computed and negated from the associated overlapped symbols in the window $[t, t+T_{\text{eff}}-1]$. The remaining non-urgent and overlapped symbols are all linear combinations of only $\mathbf{x}_{[t,t+T_{\text{eff}}-1]}^v$. The MSR code protecting the non-urgent source symbols is now decodable. From Statement 2 of Lemma 1, all non-urgent source symbols can be recovered at time $t + T_{\text{eff}} - 1$ if (5) is satisfied. By setting $R = \frac{\kappa_v}{\kappa_v + \kappa_p}$

and $T = T_{\text{eff}} - 1$, we find

$$Bp \leq \frac{\kappa_p}{\kappa_v + \kappa_p} T_{\text{eff}} r. \tag{12}$$

Substituting the code parameters in (11) to the above reveals that the condition is met with equality. Then $\mathbf{s}^v_{[t,t+T_{\text{eff}}-1]}$ are all recoverable by time $t + T_{\text{eff}} - 1$.

The non-urgent source sub-packet at time $t + T_{\text{eff}}$ is recovered immediately by considering a window of length 1. After negating the effects of the prior source packets, the receiver effectively observes

$$\mathbf{s}^v_{t+T_{\text{eff}}} \mathbf{G}_0 \operatorname{diag}(\mathbf{A}_{t+T_{\text{eff}}}; \kappa_p + \kappa_v) + (\mathbf{p}^u_t, \mathbf{0}_{\kappa_v r}) \operatorname{diag}(\mathbf{A}_{t+T_{\text{eff}}}; \kappa_p + \kappa_v).$$

The second term is assumed to have caused a rank deficiency in the first κ_p columns of the channel matrix. Consequently, we discard the affected columns and associated received symbols. Due to $\mathbf{G}_0$ possessing the properties of an MRD generator matrix, Theorem 2 can be used to prove that $\mathbf{s}^v_{t+T_{\text{eff}}}$ is recoverable.

Step 2: The decoder recovers the urgent $\mathbf{s}^u_t$ at time $t + T_{\text{eff}}$. Because the non-urgent $\mathbf{s}^v_{[t,t+T_{\text{eff}}]}$ are all known to the decoder, $\mathbf{x}^v_{t+T_{\text{eff}}}$ is computable and can be negated from the overlapped symbols at time $t + T_{\text{eff}}$. Thus, $\mathbf{p}^u_t$ is available to the decoder at time $t + T_{\text{eff}}$. The MRD code protecting the urgent source sub-packet is now decodable. Using (9) shows that the receiver observes

$$\mathbf{s}^u_t \mathbf{G} \begin{pmatrix} \operatorname{diag}(\mathbf{A}^*_t; \kappa_u) & \\ & \operatorname{diag}(\mathbf{A}_{t+T_{\text{eff}}}; \kappa_p) \end{pmatrix},$$

the product of the codeword $(\mathbf{s}^u_t, \mathbf{p}^u_t)$ with a concatenation of the channel matrices at time t and $t + T_{\text{eff}}$. The rank of the channel matrix is $\kappa_u(r - p) + \kappa_p r$, being the summation of the ranks of the matrices $\mathbf{A}^*_t$ and $\mathbf{A}_{t+T_{\text{eff}}}$. By Theorem 2, the source can be completely recovered if

$$\kappa_u(r - p) + \kappa_p r \geq \kappa_u r. \tag{13}$$

Substituting the values of (11) to the above reveals the condition is met with equality. Thus, $\mathbf{s}^u_t$ is recovered at time $t + T_{\text{eff}}$, completing recovery of the entire first source packet.

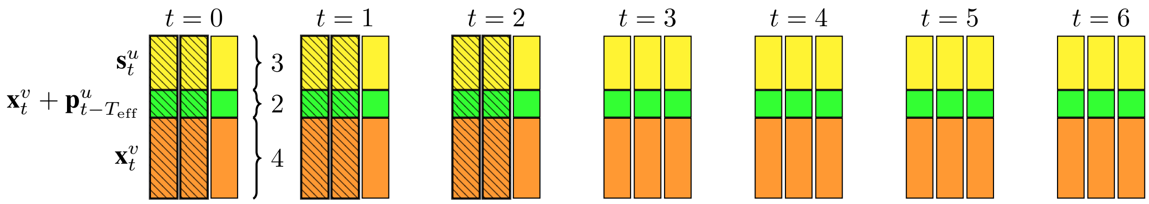
The next source packet $\mathbf{s}_{t+1}$ must be recovered by time $t + T_{\text{eff}} + 1$. All prior non-urgent source symbols are known to the decoder, so $\mathbf{s}^v_{t+T_{\text{eff}}+1}$ can be computed using Step 1. Once $\mathbf{x}^v_{t+T_{\text{eff}}+1}$ is computed, Step 2 allows $\mathbf{s}^u_{t+1}$ to be recovered at time $t + T_{\text{eff}} + 1$. We repeat this technique for the subsequent packets in the burst by first recovering the non-urgent symbols at each time instance. Each urgent source is thus sequentially recovered iteratively with delay T_{eff}.

C. Decoding Example

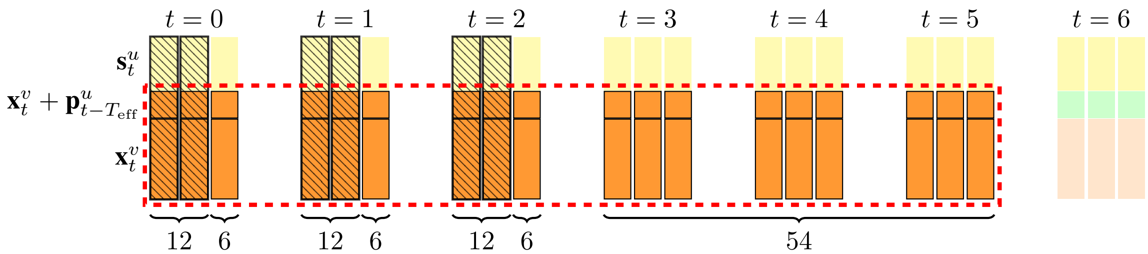
In Fig. 5, we provide an example of the encoding and decoding steps. Consider a channel $\mathcal{CH}(3, 2, 3, 6)$, which experiences a burst affecting the first two paths for $0 \leq t \leq 2$. For simplicity, we assume that no linear transformations occur, i.e., $\mathbf{A} = \mathbf{I}_{3\times 3}$ is an Identity matrix and $\mathbf{A}^*$ contains only the third column of $\mathbf{I}_{3\times 3}$. The receiver simply receives all transmitted channel packets when there are no losses and only one of the packets when there are. The rank deficiency in the channel matrix is then equivalent to causing erasures. The network in the window $[0, 6]$ is given in Fig. 5a, where the erased links are shaded.

We use a ROBIN code with memory $T_{\text{eff}} = 6$. Using (11), the source $\mathbf{s}_t$ is split into urgent $\mathbf{s}^u_t \in \mathbb{F}^9_{q^M}$ and non-urgent $\mathbf{s}^v_t \in \mathbb{F}^{12}_{q^M}$. The non-urgent source is protected with an MSR code $\mathcal{C}[18, 12, 6]$, which generates $\mathbf{x}^v_t$, whereas the urgent source is protected with an MRD code $\mathcal{C}[15, 9]$, which generates $(\mathbf{s}^u_t, \mathbf{p}^u_{t-T_{\text{eff}}})$. In each channel packet, the encoder transmits $\kappa_u = 3$ urgent source symbols, $\kappa_p = 2$ overlapped symbols, and $\kappa_v = 4$ non-urgent symbols.

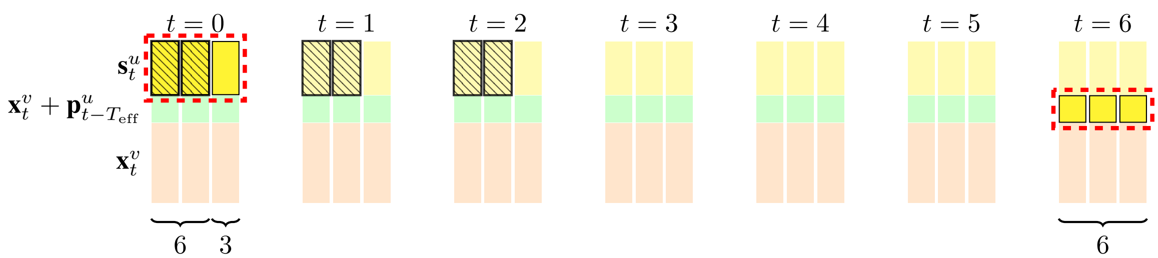
The decoding steps to recover $\mathbf{s}_0$ at time $t = 6$ are given as follows. We first show how the steps in the previous sub-section are followed. We then remark how, due to our restricted channel matrix for this example, the results are equivalent to counting un-erased symbols. In Fig. 5b, the decoder first negates $\mathbf{p}^u_{[-6,-1]}$ from the other overlapped symbols. This gives the decoder complete access to $\mathbf{x}^v_{[0,5]}$. With $B = 3, T_{\text{eff}} = 6$, note that (12) is satisfied and $\mathbf{s}^v_{[0,5]}$ recoverable by time $t = 5$. From the alternate perspective, there are 12 erased non-urgent symbols per time slot for $0 \leq t \leq 2$, leading to a total of 36 erased



(a) A slot consists of $r = 3$ channel packets—depicted as columns—each of which includes $\kappa_u = 3$ urgent source, $\kappa_p = 2$ overlapped, and $\kappa_v = 4$ non-urgent symbols. Failing links in the window $[0, 2]$ force several columns (shown as shaded) of the respective macro-packets redundant.



(b) $\mathbf{p}^u_{[-6,-1]}$ is negated from the overlapped symbols. The overlapped and non-urgent symbols form an MSR code with 12 rank losses per shot in the first three shots. There are a total of 36 rank losses and 72 correctly received symbols, allowing the recovery of $\mathbf{s}^v_{[0,5]}$.



(c) After recovering and negating $\mathbf{s}_6^v$, the overlapped section at $t = 6$ and urgent section at $t = 0$ form an MRD code with 6 rank losses and 9 correctly received symbols, which are sufficient to recover all symbols of the urgent source.

Fig. 5: Example of a ROBIN code with memory $T_{\text{eff}} = 6$ recovering from a burst $B = 3$ in a network with $r = 3$. Each column represents a received channel packet along a given path, which is a linear combination of the transmitted packets. The hatched columns are the linearly dependent combinations.

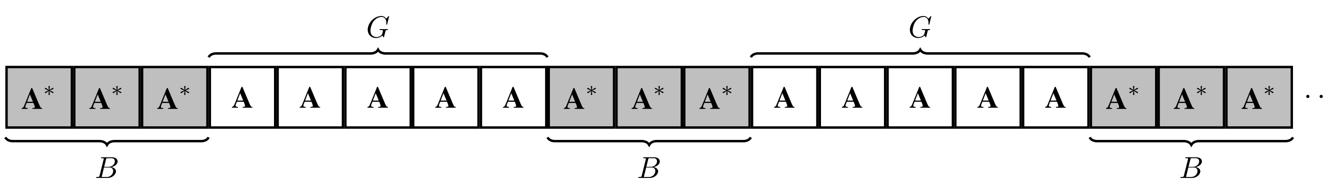


Fig. 6: A periodic burst rank loss network, where the channel matrix $\text{rank}(\mathbf{A}^*) = r - p$ during bursts and $\text{rank}(\mathbf{A}) = r$ during perfect communication.

non-urgent symbols. Furthermore note that there are a total of 72 correctly transmitted non-urgent symbols in the window $[0, 5]$. This is sufficient to recover all 72 non-urgent symbols of $\mathbf{s}^v_{[0,5]}$. Then at time $t = 6$, the decoder recovers $\mathbf{s}_6^v$ by ignoring the overlapped received symbols and using only the 12 non-urgent received symbols of $\mathbf{x}_6^v$.

In Fig. 5c, the decoder recovers $\mathbf{s}_0^u$ at time $t = 6$. Having recovered $\mathbf{s}^v_{[0,6]}$, the decoder reconstructs $\mathbf{x}_6^v$ and negates it from the overlapped symbols to recover $\mathbf{p}_0^u$ that was transmitted at time $t = 6$. This gives us a Gabidulin codeword $(\mathbf{s}_0^u, \mathbf{p}_0^u)$, where a part of $\mathbf{s}_0^u$ is affected by a rank-deficient matrix. Setting all of the code parameter values reveals that (13) is satisfied. From the erased symbols perspective, there are 3 correctly received symbols of $\mathbf{s}_0^u$ and 6 recovered symbols of $\mathbf{p}_0^u$, which are sufficient to recover all 9 symbols that comprise $\mathbf{s}_0^u$ within the deadline.

V. Upper Bound on the Burst Loss Network

In this section, we prove the converse to Theorem 1. The proof for an upper bound on achievable rates for $\mathcal{CH}(r, p, B, G)$ follows similar to prior proofs establishing the capacity of single-link burst erasure channels [1], [5].

A. *Upper Bound for $T \geq B$*

We first address the case of $T \geq B$. A network with r paths, out of which p paths are unavailable in a burst, is described in Fig. 6. This network has a period of $B + G$, where the ranks of the channel matrices $\mathbf{A}_t$ in the i-th period is given by

$$\operatorname{rank}(\mathbf{A}_t) = \begin{cases} r - p & t \in [i(B+G), i(B+G) + B - 1] \\ r & t \in [i(B+G) + B, (i+1)(B+G) - 1] \end{cases}$$

for $0 \leq i \leq L$, where L is the total number of periods over which communication occurs. When the channel matrix is full-rank, we denote it $\mathbf{A}_t = \mathbf{A}$, whereas the rank-deficient channel matrix is $\mathbf{A}_t = \mathbf{A}^*$. Every period consists of the network experiencing a burst of length B followed by a gap of full-rank channel matrices for G shots, meaning we can describe it as $\mathcal{CH}(r, p, B, G)$. We use a counting argument for the number of linearly independent received combinations of symbols over the first period $[0, B + G - 1]$ to derive an upper bound on the rate:

$$\begin{aligned} (B+G)nR &\leq B(n - \nu p) + Gn \\ R &\leq \frac{Gr + B(r-p)}{(B+G)r}. \end{aligned} \tag{14}$$

When $T \geq G$, we substitute $G = T_{\text{eff}}$ to (14), returning the streaming capacity. Note that this result can be obtained also by directly solving for the Shannon capacity of the periodic network.

This counting argument does not lead to the capacity when $T < G$. By considering the decoding delay constraint for a streaming code, the upper bound in (14) can be tightened for this case. Consider a new periodic network with period $B + T$. The channel matrices in the i-th period have the following ranks:

$$\operatorname{rank}(\mathbf{A}_t) = \begin{cases} r - p & t \in [i(B+T), i(B+T) + B - 1] \\ r & t \in [i(B+T) + B, (i+1)(B+T) - 1] \end{cases}. \tag{15}$$

For this network, every burst of length B is followed by a gap of only T shots. The channel matrices remain $\mathbf{A}^*$ and $\mathbf{A}$ respectively as described above. This channel cannot be described by $\mathcal{CH}(r, p, B, G)$ as the gaps are insufficiently short. Nonetheless, recall that any code that is feasible over $\mathcal{CH}(r, p, B, G)$ must recover every source packet with delay T. We argue the following claim.

Claim 1. Any code feasible over $\mathcal{CH}(r, p, B, G)$ is also feasible over the periodic network defined in (15).

A simplified argument is presented here, with a rigorous information theoretic proof in Appendix B. The key insight is that when the decoder is concerned with recovering $\mathbf{s}_t$, the channel is only relevant in the window $[0, t + T]$. In analysis, the periodic network can be replaced by a hypothetical effective network $\operatorname{diag}(\mathbf{A}_0, \dots, \mathbf{A}_{t+T}, \mathbf{A}, \dots)$, comprised of the original channel for $[0, t + T]$ with all subsequent shots assumed to be transmitted perfectly. We prove feasibility in two steps for each period of the network, beginning with the first period $[0, B + T - 1]$.

Step 1: First consider the packets affected by the burst, i.e., $t \in [0, B - 1]$. The maximum tolerable delay is T, but we relax the constraints to require that every packet affected by the burst is completely recovered by time $B + T - 1$, before the second burst begins. From the perspective of these packets, the network is equivalent to a hypothetical effective network where the channel is rank-deficient in the window $[0, B - 1]$ and full-rank for all subsequent shots. A feasible streaming code guarantees that every packet in the burst is recoverable within the decoding constraint for this network.

Step 2: Next, we recover the source packets for $t \in [B, B + T - 1]$ remaining in the period. Similar to the previous packets, we relax the requirement to every packet in this window being completely recovered by time $2B + 2T - 1$ at the end of the second period. At time $B + T - 1$, the packets affected by the first burst have all been recovered and their effect can be negated

	Fig. 7	Fig. 8
α	5×10^{-4}	5×10^{-4}
β	$(0.1, \ldots, 0.8)$	0.3
r	6	16
p	3	$(2, \ldots, 16)$
Channel Length	10^8	10^8
Rate R	$\frac{10}{13}$	$(0.94, \ldots, 0.52)$
Delay T	7	17
	B	B
ROBIN Code	6	16
MSR Code	3	8
m-MDS Code	1	$(1, \ldots, 8)$
MS Code	3	$(1, \ldots, 16)$

TABLE I: Parameters used in simulations.

for the current packets of interest. We permit the network to start a new burst for B shots. The network is then equivalent to a hypothetical effective network where the channel follows (15) for $[0, 2B + T - 1]$, before becoming full-rank again for all subsequent shots. A feasible streaming code guarantees that every packet in the window $[B, B + T - 1]$ is recoverable within the decoding constraint for this network.

The packets affected by the second burst at $[B + T, 2B + T - 1]$ are then considered. The effect of all packets in the first period is negated by time $2B + 2T - 1$, so we reuse the argument for the packets affected by the first burst, i.e., all source packets in the window $[B + T, 2B + T - 1]$ are completely recovered by time $2B + 2T - 1$. The first burst has been recovered, so the effective network contains only a single burst of length B and a feasible code guarantees recoverability. The remaining packets in the second period are then recovered in the same way as in the above by time $3B + 3T - 1$.

A feasible decoder effectively completes decoding from one period of the network before pursuing the next. Suppose that this periodic channel continues for L periods. Furthermore, we permit an $L + 1$-th grace period that ensures enough transmissions for the packets in the L-th period to be recovered. Note that we are not concerned with recovering the packets in this period. Reusing the counting argument reveals

$$L(B + T)nR \leq (L + 1)B(n - \nu p) + (L + 1)Tn.$$

By substituting $T = T_{\text{eff}}$ and letting L grow asymptotically, this equation is re-arranged to recover (2). Thus, all achievable rates are bounded by the capacity for both cases. This completes the converse for $T \geq B$.

B. Upper Bound for $T < B$

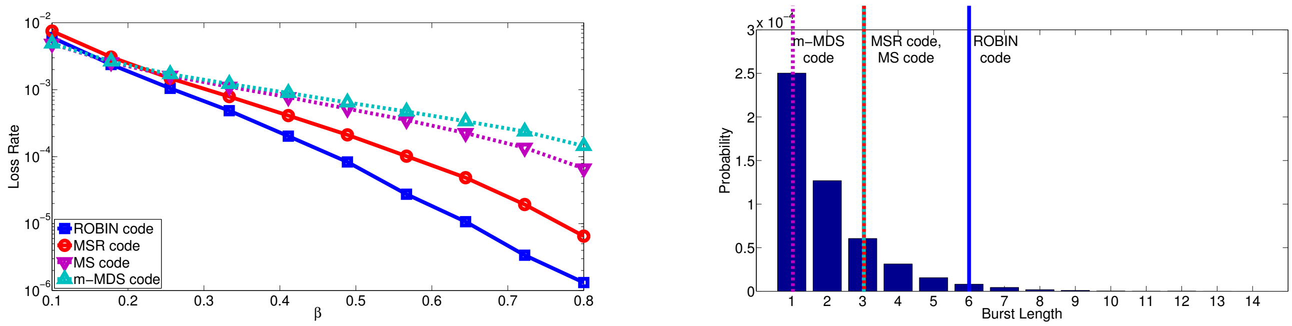
When $T < B$, the first packet in a burst of length B must be recovered before the burst ends. In fact, assuming that all prior source packets are known, any packet $\mathbf{s}_t$ is expected to be recoverable by the end of an window $[t, t + T]$ experiencing only bursts. Consequently, the inter-burst gap is irrelevant and any feasible code over $\mathcal{CH}(r, p, B, G)$ is also feasible over a perpetual rank-deficient network, i.e., $\text{rank}\,(\mathbf{A}_t) = r - p$ for all $t \geq 0$. Decoding is performed similar to the prior scenario where $T < G$ but by considering each source packet sequentially. Once $\mathbf{s}_t$ is recovered at time $t + T$, we move to the next time instance and negate the effect of the recovered packet. Then $\mathbf{s}_{t+1}$ is recovered in the same manner. Using the previous counting argument, we can bound the rate

$$nR \leq n - \nu p$$

An information theoretic argument similar to the one given in Appendix B can be made to show the same results.

VI. Simulation Results

We use simulations to evaluate the performance of ROBIN codes over statistical networks. Single-link burst erasure streamins have been modelled in previous works using a Gilbert channel [4], [5]. We appropriate this model to characterize a network



(a) Loss probability measured over Gilbert Channels with varying β.

(b) Probability of burst lengths for $(\alpha, \beta) = (5 \times 10^{-4}, 0.5)$. B for each code is marked.

Fig. 7: Simulations over a Gilbert Channel modelling a network with $\mathrm{rank}\,(\mathbf{A}_t) = 6$ in the good-state and 3 in the bad-state. The channel and code parameters are given in Table I.

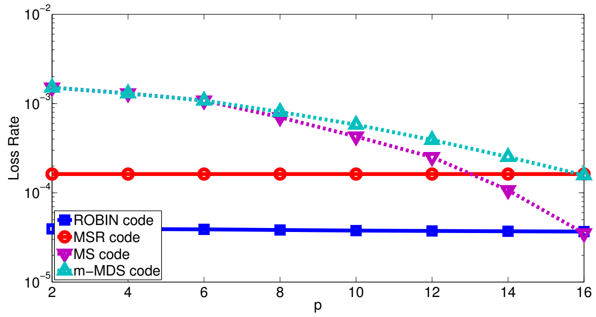
with bursts of rank-deficient matrices. The Gilbert channel is a Markov model with two states: a good-state where the channel matrix $\mathbf{A}_t = \mathbf{A}$ is full-rank, i.e., $\rho_t = r$, and a bad-state where the channel matrix $\mathbf{A}_t = \mathbf{A}^*$ is rank-deficient, i.e., $\rho_t = r - p$. The bad-state represents a burst loss event, i.e., p paths deactivate forcing the receiver to observe p linearly dependent combinations of the channel packets for each shot while the channel remains in the bad-state. The transition probability from good-state to bad-state is given by the parameter α whereas the transition probability from bad-state to good-state is given by β. The length of a burst is a geometric random variable with mean $\frac{1}{\beta}$, whereas the length of the gaps between bursts is geometric with mean $\frac{1}{\alpha}$. The adversarial Burst Rank Loss Network approximates this Gilbert channel, with B and G chosen in proportion to $\frac{1}{\beta}$ and $\frac{1}{\alpha}$ respectively.

Decoders are not explicitly implemented in the experiments. The analysis consists of calculating whether a given packet is recoverable by counting the number of available linear combinations. To simplify the computations, we assume that the MSR and m-MDS codes have infinite memory, including when used as constituents. Consequently, source packets can be recovered after the deadline, but are not considered successfully decoded. We measure the packet loss rate, given by the frequency of source packets that are not completely recoverable within their respective deadlines. The burst length parameter β is varied along the x-axis in the first experiment, whereas in the second, we vary the rank-deficiency p. We plot loss probability on the y-axis.

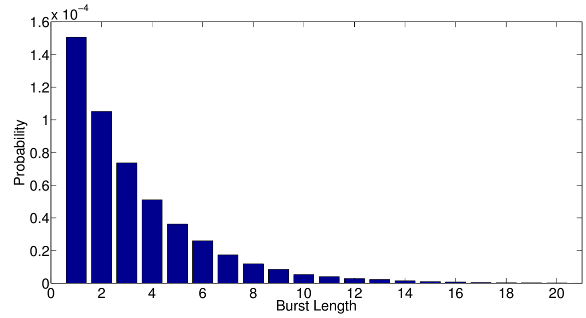
A. Variable Burst Length Parameter

For this experiment, the bad-state transition probability β is varied from 0.1 to 0.8. The good-state transition probability α is equal to 5×10^{-4}. Both $r = 6$ and $p = 3$ are fixed. Fig. 7b shows the burst length distribution for $\beta = 0.5$. The code rate is set at $R = \frac{20}{26} \approx 0.77$ and the decoder permits a maximum tolerable delay $T = 7$. These channel and code parameters are summarized in Table I. In Fig. 7a, the packet loss rate of the ROBIN code is measured and compared to the loss rate of the MSR code and their single-link counterparts: the MS and m-MDS codes.

- **m-MDS code:** The m-MDS code loss rate is the dashed cyan line with '$\triangle$'. Having been designed for single-link channels, the m-MDS code discards the macro-packet if the channel matrix is rank-deficient. The maximum burst length for perfect recovery with delay $T = 6$ is $B = 1$, meaning if the channel is in the bad-state for two consecutive time instances, the first packet is not recoverable within the deadline. However, this code is capable of recovering part of a burst within the delay constraint. For example, if a burst of length 2 occurs at time t, both source packets are recovered simultaneously at time $t + 7$. Consequently, the second source packet meets its deadline even though the first packet fails. This ability to recover a fraction of packets affected by a burst is referred to as 'partial recovery'.
- **MS code:** The dashed purple line with '$\triangledown$' represents the MS code loss rate. Having been designed for single-link streaming, this code also considers a rank-deficient channel as an erasure of the entire macro-packet. The MS code nonetheless affords a larger $B = 3$ than the m-MDS code, as marked in Fig. 7b. As a result, it achieves a slightly lower loss rate in comparison to the m-MDS code.



(a) Loss probability measured over Gilbert Channels with varying p.



(b) Probability distribution of burst lengths for $(\alpha, \beta) = (5 \times 10^{-4}, 0.3)$.

Fig. 8: Simulations over a Gilbert Channel modelling a network with $\mathrm{rank}\,(\mathbf{A}_t) = 16$ in the good-state and $16 - p$ in the bad-state. The channel and code parameters are given in Table I.

Note that both the MS and the m-MDS codes perform slightly better than the rank metric codes for $\beta = 0.1$. This region is characterized by long bursts which are not recoverable by any of the codes. The m-MDS code (and by extension, the MS code) are systematic constructions, meaning that packets after a burst are immediately recovered simply by reading the source symbols. In contrast, the MSR and ROBIN codes are non-systematic. As a result, even the packets after a burst are not recoverable until all prior packets are recovered. Using a systematic code gives a technical improvement over the non-systematic codes, but the effect is not visible except in the extreme case of long bursts.

- **MSR code:** The loss rate of the MSR code is given by the solid red line marked '◯'. This code does not discard the entire macro-packet for a rank-deficient channel. Furthermore, this code is capable of partial recovery for bursts exceeding the maximum tolerable length. These improvements permit the MSR code to have a significantly lower loss probability in comparison to the single-link codes. If a burst of length 4 occurs at time t, the last three source packets in the burst are recoverable within their respective deadlines using an MSR code. In contrast, all four source packets are lost when using a MS code. The MSR code shows that rank metric codes designed for streaming over a network provide significant gains in comparison to single-link codes. However, the maximum tolerable burst length $B = 3$ is significantly smaller than that of the ROBIN code, as marked in Fig. 7b.
- **ROBIN code:** The solid blue line marked '□' shows the loss rate of the ROBIN code. Similar to the MSR code, this code is capable of decoding using the linearly independent packets in rank-deficient channel matrices. In addition, the ROBIN code guarantees the largest tolerable burst length $B = 6$. As a result, the ROBIN code easily outperforms all of the other codes for nearly all values of β.

  The only region where ROBIN codes do not significantly outperform the others is when $\beta < 0.2$. This region is indicative of long bursts, with the mean burst length exceeding 10. Here, all codes fail to recover any packets affected by the bursts and therefore all achieve similarly high loss rates. The m-MDS and MS codes slightly outperform the ROBIN codes for $\beta = 0.1$ as previously discussed. This is due to ROBIN codes using a non-systematic construction, which makes it unable to recover packets after a long burst with low delay.

B. *Variable Rank-deficiency*

For this experiment, we fix the Gilbert channel parameters $\alpha = 5 \times 10^{-4}$ and $\beta = 0.3$. The decoder delay $T = 17$ and network rate $r = 16$ are also fixed. We vary the bad-state rank-deficiency p to compare the performance to the achievable rates shown in Fig. 3. For every tested value of p, we change the rate R of the codes, while keeping the ratio $\frac{1-R}{p}$ constant. This ensures that for every pair (p, R), the maximum tolerable burst lengths B of the ROBIN Code and the MSR Code remain fixed. Naturally, the single-link codes are not dependent on p and therefore their B values decrease as R increases for fixed $\frac{1-R}{p}$. The channel and code parameters are summarized in Table I. The packet loss rates of the codes are displayed in Fig. 8b. Comparing with Fig. 3, the simulations match the predicted results and further, justifying the adversarial approach.

- **m-MDS code:** The m-MDS code loss rate is the dashed cyan line with '△'. Predictably, this code reveals the worst performance throughout the experiment. Matching the achievable rates in Fig. 3, the m-MDS code and MSR code have the highest disparity for small p and achieve the same performance when p approaches r. By forcing constant $\frac{1-R}{p}$, the code rate decreases as p increases, permitting larger bursts to be recoverable.

- **MS code:** The dashed purple line with '$\triangledown$' represents the MS code loss rate. The performance matches the m-MDS code when p is small, but converges to the ROBIN code as p increases. Because the network for $p = r$ effectively behaves as an erasure channel, the MS code and m-MDS code match single-link streaming simulations in [5]. MS codes use a similar layering technique to ROBIN codes. At $p = r$, their code parameters converge to the same values, and thus, they are effectively equivalent.
- **MSR code:** The loss rate of the MSR code is given by the solid red line marked '$\bigcirc$'. This code has a constant tolerable burst length B for every pair (p, R). As a result, the performance remains constant throughout the experiment. The MSR and m-MDS code performances naturally become the same when $p = r$, similar to the comparison in Fig. 3 of achievable rates. When $p = r$, burst recoverability condition in (5) for MSR codes is equivalent to that for m-MDS codes [20].
- **ROBIN code:** The solid blue line marked '$\square$' shows the loss rate of the ROBIN code. By keeping the ratio $\frac{1-R}{p}$ constant, we fix B of the ROBIN code. Thus, the loss rate does not change as p increases. When $p = r$, the channel behaves as an erasure channel and the MS and ROBIN code effectively become the same. Once again, this result validates the predictions from Fig. 3. The key difference between MS and ROBIN codes is the use of a non-systematic encoder to protect the non-urgent symbols by the ROBIN code. However, this does not yield a significant difference in the loss rates for the above simulations.

VII. Conclusion

In this paper, we address the problem of network streaming where links fail in a burst event. Random linear network codes are assumed rather than naive routing. As a consequence channel packets mix amongst themselves before arriving at the destination. We propose the Burst Rank Loss Network, extending the classic burst erasure channel to consider rank-deficient channel matrices. Layered coding techniques are used to construct a new family of codes: ROBIN codes. These codes use MSR and MRD codes as constituents and guarantee perfect low-delay recovery of every packet transmitted during a burst of link failures. Furthermore, ROBIN codes achieve the streaming capacity of the Burst Rank Loss Network for the practical case of $T \geq B$. Simulations over statistical networks are performed to compare these to baseline rank metric codes such as the MSR and MRD codes as well as single-link erasure correcting streaming codes such as the m-MDS and MS codes.

We remark on the size of the required field $\mathbb{F}_{q^M}$ to construct ROBIN codes. Both the constituent MSR and MRD codes naturally require sufficiently large fields in order to be constructed. Given the ROBIN code parameters in (11), we write $M \geq \max(M_1, M_2)$, where $M_1 \geq B(r+p)$ is the necessary field size to guarantee the construction of an appropriate Gabidulin code [14] and $M_2 \geq q^{T_{\text{eff}} r (T_{\text{eff}}+2)-1}$ is the necessary field size to guarantee an appropriate MSR code [8]. However, simulation results can show that both structured and random linear code constructions exist over smaller fields for the substituents. Given these alternatives, we conjecture that fields smaller than $\max(M_1, M_2)$ can be found to harbour ROBIN codes.

There exist several problems to be addressed in future works. A current concern is that in contrast to single-link burst correcting MS codes which use systematic constructions, ROBIN codes use a non-systematic MSR constituent. It is not immediately obvious whether a systematic MSR code exists to satisfy the delay constraints, but assuming their existence, a hypothetical systematic ROBIN code is more easily decodable and offers protection in the event of long bursts. A second problem to consider is the region of $G < B$. Currently, general achievable codes for the single-link problem under this scenario are not known and may provide insights to the network scenario. Finally, the layered approach to solve the Burst Rank Loss Network can be implemented for more sophisticated networks. Further sliding window erasure channels such as the robust burst erasure channel and multi-cast burst erasure channels naturally possess network counterparts. As the layered technique using MSR constituent codes has proven successful for burst correction, it would be interesting to observe whether the results of streaming over more sophisticated erasure channels channels can be fully generalized to the network scenario.

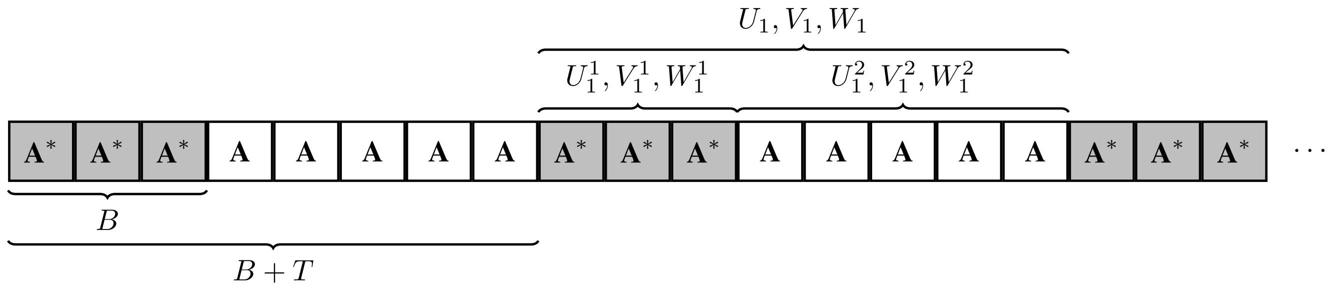


Fig. 9: The periodic rank loss channel, with the sets U_i, V_i, W_i identified for the i-th period.

Appendix A
Proof of Statement 1 in Lemma 1

Recall the assumption in Section II on the code length $n = \nu r$. The proof below is nearly identical to the one provided in [8], except for the key difference being that only $\nu = 1$ was considered in the prior work.

Proof: We assume only that $\operatorname{rank}(\mathbf{A}_{[t,t+j]}) \geq R(j+1)r$ for the first packet in a sequence to be recoverable. Consider two source sequences $\mathbf{s}_{[t,t+T]} = (\mathbf{s}_t, \ldots, \mathbf{s}_{t+T})$ and $\hat{\mathbf{s}}_{[t,t+T]} = (\hat{\mathbf{s}}_t, \ldots, \hat{\mathbf{s}}_{t+T})$, where $\mathbf{s}_t \neq \hat{\mathbf{s}}_t$. Suppose they respectively generate macro-packet sequences $\mathbf{X}_{[t,t+T]}$ and $\hat{\mathbf{X}}_{[t,t+T]}$, where $\mathbf{X}_{[t,t+T]}\mathbf{A}^*_{[t,t+T]} = \hat{\mathbf{X}}_{[t,t+T]}\mathbf{A}^*_{[t,t+T]}$. In vector notation, $\mathbf{X}_{[t,t+T]}\mathbf{A}^*_{[t,t+T]}$ expands to

$$(\mathbf{x}_t, \ldots, \mathbf{x}_{t+T}) \begin{pmatrix} \operatorname{diag}(\mathbf{A}^*_t; \nu) & & \\ & \ddots & \\ & & \operatorname{diag}(\mathbf{A}^*_{t+T}; \nu) \end{pmatrix}.$$

Note that $\operatorname{rank}(\operatorname{diag}(\mathbf{A}_t; \nu)) = \nu \operatorname{rank}(\mathbf{A}_t)$. The difference $\mathbf{x}_{[t,t+T]} - \hat{\mathbf{x}}_{[t,t+T]}$ is a hypothetical codeword sequence whose sum rank is at least $(n-k)(T+1)+1$. However, $(\mathbf{x}_i - \hat{\mathbf{x}}_i)\operatorname{diag}(\mathbf{A}^*_i; \nu) = 0$ implies $\operatorname{rank}(\phi_n(\mathbf{x}_i) - \phi_n(\hat{\mathbf{x}}_i)) \leq n - \nu \operatorname{rank}(\mathbf{A}^*_i)$ for $i \in [t, t+T]$. By summing each of the inequalities, we arrive at the following contradiction on the sum rank of this codeword sequence:

$$\begin{aligned} \sum_{i=t}^{t+T} \operatorname{rank}(\phi_n(\mathbf{x}_i) - \phi_n(\hat{\mathbf{x}}_i)) &\leq n(T+1) - \nu \operatorname{rank} \mathbf{A}^*_{[t,t+T]} \\ &\leq n(T+1) - \nu R(T+1)r \\ &\leq (n-k)(T+1). \end{aligned}$$

■

Appendix B
Converse Proof of Theorem 1 for $T < G$

This proof follows similarly to the one in [4]. Consider the periodic network in Fig. 9 and suppose the network is in use for L periods. An $L+1$-th period provides sufficient information to recover the packets in the L-th period, and the packets in the grace period do not need to be recovered. We use the following notation to facilitate the proof:

$$\left.\begin{aligned} U_i^1 &= (\mathbf{s}_{(T+B)i}, \ldots, \mathbf{s}_{(T+B)i+B-1}) \\ U_i^2 &= (\mathbf{s}_{(T+B)i+B}, \ldots, \mathbf{s}_{(T+B)(i+1)-1}) \end{aligned}\right\} U_i = (U_i^1, U_i^2)$$
$$\left.\begin{aligned} V_i^1 &= (\mathbf{X}_{(T+B)i}, \ldots, \mathbf{X}_{(T+B)i+B-1}) \\ V_i^2 &= (\mathbf{X}_{(T+B)i+B}, \ldots, \mathbf{X}_{(T+B)(i+1)-1}) \end{aligned}\right\} V_i = (V_i^1, V_i^2)$$
$$\left.\begin{aligned} W_i^1 &= (\mathbf{Y}_{(T+B)i}, \ldots, \mathbf{Y}_{(T+B)i+B-1}) \\ W_i^2 &= (\mathbf{Y}_{(T+B)i+B}, \ldots, \mathbf{Y}_{(T+B)(i+1)-1}) \end{aligned}\right\} W_i = (W_i^1, W_i^2).$$

Here, $i \in \{0, \ldots, L-1\}$ denotes the period of interest. U_i, V_i, and W_i denote the source packets, transmitted macro-packets, and received macro-packets respectively in one period of transmission. The superscript 1 refers to the packets in the period that are affected by the burst, whereas the superscript 2 refers to the packets after the burst. We use the following inequalities:

$$H(W_i^1) \le H(V_i^1)\frac{r-p}{r} \tag{16a}$$
$$H(W_i^2) = H(V_i^2), \tag{16b}$$

in order to bound the entropy of W_i. Both relationships arise from the channel transfer matrix. The packets in W_i^2 and V_i^2 are related by a full rank channel matrix $\mathbf{A}$ and thus, the linear transformation does not change the entropy. However, the packets in W_i^1 and V_i^1 are related by a rank-deficient $\mathbf{A}^*$. As the received packets formed from linearly dependent combinations of the channel packets can simply be re-written using the remaining linearly independent channel packets, they decrease the entropy accordingly.

Before proving the converse, we assert two remaining relevant properties of a streaming code:

$$H(U_i^1 \mid V_{[0,i-1]}, W_i) = 0 \tag{17a}$$
$$H(U_i^2 \mid V_{[0,i-1]}, V_i^1, W_i^2, W_{i+1}) = 0. \tag{17b}$$

Both properties are due to the fact that for a feasible code, each source packet must be recovered with delay T. Now, let $\mathcal{S}$ denote the alphabet of source symbols and $\mathcal{X}$ the alphabet of channel symbols. Then, we can show

$$\begin{aligned}
L(B+T)H(\mathcal{S}) &= H(U_{[0,L-1]}) \\
&\le H(U_{[0,L-1]}, W_{[0,L]}) \\
&= H(W_{[0,L]}) + \sum_{i=0}^{L-1} H(U_i \mid U_{[0,i-1]}, W_{[0,L]}) \\
&= H(W_{[0,L]}) + \sum_{i=0}^{L-1} \Big(H(U_i^1 \mid U_{[0,i-1]}, W_{[0,L]}) + H(U_i^2 \mid U_i^1, U_{[0,i-1]}, W_{[0,L]}) \Big) \\
&= H(W_{[0,L]}) + \sum_{i=0}^{L-1} \Big(H(U_i^1 \mid V_{[0,i-1]}, W_{[0,L]}) + H(U_i^2 \mid V_i^1, V_{[0,i-1]}, W_{[0,L]}) \Big) && (18) \\
&= H(W_{[0,L]}) && (19) \\
&\le (L+1)(H(W_i^1) + H(W_i^2)) \\
&\le (L+1)(B\frac{r-p}{r} + T)\log|\mathcal{X}|. && (20)
\end{aligned}$$

In the above, (18) is due to the channel macro-packets being causally generated from source packets. Using (17) gives (19). We arrive at (20) due to (16). The inequality can be re-arranged to bound the rate.

$$\begin{aligned}
R = \frac{k}{n} \le \frac{H(\mathcal{S})}{\log|\mathcal{X}|} &= \left(\frac{L+1}{L}\right)\frac{Tr + B(r-p)}{(T+B)r} \\
&\xrightarrow{L\to\infty} \frac{Tr + B(r-p)}{(T+B)r}
\end{aligned}$$

Replacing $T_{\text{eff}} = T$ returns the capacity in (2).

References

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

[2] V. Tomas, J. Rosenthal, and R. Smarandache, "Decoding of convolutional codes over the erasure channel," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 90–108, 2012.

[3] D. Leong, A. Qureshi, and T. Ho, "On coding for real-time streaming under packet erasures," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2013, pp. 1012–1016.

[4] A. Badr, D. Lui, and A. Khisti, “Streaming codes for multicast over burst erasure channels,” *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4181–4208, 2015.

[5] A. Badr, P. Patil, A. Khisti, W. Tan, and J. Apostolopoulos, “Layered constructions for low-delay streaming codes,” *IEEE Trans. Inf. Theory*, 2017.

[6] M. Ellis, D. P. Pezaros, and C. Perkins, “Performance analysis of AL-FEC for RTP-based streaming video traffic to residential users,” in *IEEE Packet Video Workshop*, 2012.

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

[8] R. Mahmood, A. Badr, and A. Khisti, “Convolutional codes with maximum column sum rank for network streaming,” *IEEE Trans. Information Theory*, vol. 62, no. 6, pp. 3039–3052, 2016. [Online]. Available: http://dx.doi.org/10.1109/TIT.2016.2555949

[9] D. Lui, A. Badr, and A. Khisti, “Streaming codes for a double-link burst erasure channel,” in *Information Theory (CWIT), 2011 12th Canadian Workshop on*. IEEE, 2011, pp. 147–150.

[10] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[11] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Allerton Conf. on Comm., Control, Comp.*, 2003.

[12] T. Ho, M. Medard, R. Kotter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 413–430, 2006.

[13] S. Jalali, M. Effros, and T. Ho, “On the impact of a single edge on the network coding capacity,” in *Inf. Theory and App. (ITA)*, 2011, pp. 1–5.

[14] E. M. Gabidulin, “Theory of codes with maximum rank distance,” *Probl. Inf. Transm.*, vol. 21, no. 1, pp. 1–12, 1985.

[15] R. Roth, “Maximum-rank array codes and their application to crisscross error correction,” *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 328–336, 1991.

[16] D. Silva, F. R. Kschischang, and R. Kotter, “A rank-metric approach to error control in random network coding,” *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[17] C. Feng, R. W. Nóbrega, F. R. Kschischang, and D. Silva, “Communication over finite-chain-ring matrix channels,” *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5899–5917, 2014.

[18] G. Forney, “Burst-correcting codes for the classic bursty channel,” *IEEE Trans. Comm. Technol.*, vol. 19, no. 5, pp. 772–781, 1971.

[19] R. W. Nobrega and B. F. Uchoa-Filho, “Multishot codes for network coding using rank-metric codes,” in *IEEE Wireless Netw. Coding Conf. (WiNC)*, 2010, pp. 1–6.

[20] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache, “Strongly-MDS convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 584–598, 2006.

[21] R. Kotter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.

[22] R. Kotter and F. R. Kschischang, “Coding for errors and erasures in random network coding,” *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.

[23] A. Khisti., D. Silva, and F. R. Kschischang, “Secure-broadcast codes over linear-deterministic channels,” in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2010, pp. 555–559.

[24] Z. Li, A. Khisti, and B. Girod, “Correcting erasure bursts with minimum decoding delay,” in *ASILOMAR 2011*, 2011, pp. 33–39.

[25] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. IEEE Press, 1999.