Layered Constructions for Low-Delay Streaming Codes

Ahmed Badr, *Student Member, IEEE*, Pratik Patil, Ashish Khisti, *Member, IEEE*, Wai-Tian Tan, *Member, IEEE* and John Apostolopoulos, *Fellow, IEEE*

***Abstract*—We study error correction codes for multimedia streaming applications where a stream of source packets must be transmitted in real-time, with in-order decoding, and strict delay constraints. In our setup, the encoder observes a stream of source packets in a sequential fashion, and M channel packets must be transmitted between the arrival of successive source packets. Each channel packet can depend on all the source packets observed up to and including that time, but not on any future source packets. The decoder must reconstruct the source stream with a delay of T packets.**

We consider a class of packet erasure channels with burst and isolated erasures, where the erasure patterns are locally constrained. Our proposed model provides a tractable approximation to statistical models, such as the Gilbert-Elliott channel, for capacity analysis. When $M = 1$, i.e., when the source-packet arrival and channel-packet transmission rates are the equal, we establish upper and lower bounds on the capacity, that are within one unit of the decoding delay T. We also establish necessary and sufficient conditions on the column distance and column span of a code to be feasible, and in turn establish a fundamental tradeoff between these. Our proposed codes — Maximum Distance And Span (MiDAS) codes — achieve a near-optimal tradeoff between the column distance and column span, and involve a layered construction. When $M > 1$, we establish the capacity for the burst-erasure channel and an achievable rate in the general case. Extensive numerical simulations over Gilbert-Elliott and Fritchman channel models suggest that our codes also achieve significant gains in the residual loss probability over statistical channel models.

***Index Terms*—Delay Constrained Capacity, Application Layer Error Correction, Packet Erasure Channels, Real-Time Streaming Communication, Deterministic Channel Models**

I. Introduction

MULTIMEDIA applications such as interactive audio/video conferencing, mobile gaming, and cloud-computing require the transmission of a stream of source packets in real time, and under strict delay constraints. The transmitter must encode a source-stream sequentially, and the receiver must decode each source packet within a fixed playback deadline. In this paper, we investigate a systematic approach for constructing *streaming codes* for such applications. Classical error correction codes such as Maximum Distance Separable (MDS) and rateless codes are not ideal streaming codes. Their encoders operate on the source-stream in blocks, and introduce buffering delays. The decoders can only recover missing source packets simultaneously, without considering the different decoding deadlines.

Classical results in information theory provide little insights into real-time communication. Naturally the Shannon capacity is no longer the fundamental limit under delay constraints. A recent empirical study [1] notes that in the Skype conferencing application, the overhead used in the error correction codes far exceeds the Shannon limit without delay constraints. Furthermore, the performance degradation due to burst losses is far more detrimental than random losses. This is again fundamentally different from classical systems without delay constraints where one can use interleaving or codes with long block-lengths to average out the effect of local burst erasures. In practice, channels introduce both burst and isolated losses often captured by statistical models such as the Gilbert-Elliott (GE) channel. Thus, burst losses are unavoidable and it is of practical interest to study the optimal coding schemes over such channels.

In the present paper, we study a class of packet-erasure channels that introduce both burst and isolated losses. Since the direct analysis of the GE channel under delay constraints appears intractable, we introduce a simplified channel that provides a useful approximation. We propose a *sliding-window erasure channel model* — $\mathcal{C}(N, B, W)$ — where in any sliding window of size W, the channel can introduce either an erasure burst of maximum length B or up to N erasures in arbitrary locations. Thus, error correction codes over such channels must correct both burst erasures and isolated erasures. We show that our proposed model is not only amenable to a tractable capacity analysis under delay constraints, but that the resulting streaming codes also provide significant gains in simulations over the GE and related channel models.

We model our streaming setup as follows. The encoder observes a stream of source packets in a sequential fashion. Between the arrival of two consecutive source packets, the encoder transmits M channel packets. Each channel packet can depend on all the source packets observed up to and including that time, but not on any future source packets. The decoder is required to reconstruct each source packet with a delay of T packets. In practice, M is a parameter determined by the application. For example, in high definition video streaming, each source frame may arrive every 40 ms, whereas each channel packet may be transmitted every millisecond, resulting in $M = 40$. On the other hand in low-rate VoIP applications,

A. Badr, P. Patil and A. Khisti are with University of Toronto, Toronto, ON, Canada. W. Tan and J. Apostolopoulos were with Hewlett Packard Laboratories, USA when this work was done. They are now with Cisco Systems, USA. The corresponding author is Ashish Khisti (ashish.khisti@gmail.com).

This work was supported by an Ontario Early Researcher Award, the Canada Research Chair program and by Hewlett Packard through a HP-IRP Award.

Part of this work was presented at the INFOCOM, Turin, Italy, 2013, CWIT, Toronto, ON, Canada, 2013, ISIT, Istanbul, Turkey, 2013 and the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, 2013.

M can be as small as 1.

In the first part of the paper, we treat the case when $M = 1$, i.e., the source-packet arrival rate and the channel-packet transmission rate are equal. A special case of this setup, involving the burst-erasure channel, was previously studied by Martinian et. al. [2], [3]. The authors establish the capacity, as well as the optimal code, for the $\mathcal{C}(N = 1, B, W \geq T + 1)$ channel[1]. These codes were called Maximally Short Code (MS) and involved constructing a block code with certain properties, and then converting it into a convolutional code. In the present paper, we revisit this construction and propose a modification that has certain advantages. First it does not require construction of a block code, but directly constructs the streaming code using a strongly-MDS convolutional code [4], [5], and a repetition code as constituent codes. We believe that our approach provides a more transparent viewpoint on how classical codes can be modified to achieve sequential recovery of source packets over the burst erasure channel. Furthermore, our construction for the burst erasure channel can be naturally extended to the sliding window erasure channel when $N > 1$. This is achieved by simply concatenating an additional layer of parity-checks to the burst-erasure code. We call this approach a *layered code* design. For any $\mathcal{C}(N, B, W)$ channel, our proposed layered code achieves within one unit of the optimal decoding delay. We note that our construction provides the first family of streaming codes that can correct both burst and isolated erasures in the streaming setup. The importance of studying such robust codes was also discussed in [2], and some specific examples were obtained using a computer search, but these do not appear to immediately lead to a general constructions.

The streaming codes by construction are convolutional codes, and hence it is natural to study their underlying distance properties. We show that any feasible streaming code over the $\mathcal{C}(N, B, W)$ channel must simultaneously have a certain minimum column distance and column span. The column distance is associated with isolated erasures, whereas the column span is associated with the recovery from burst erasures. As a corollary to our capacity bound, we characterize a new tradeoff between the column distance and column span for any convolutional code, which could be of independent interest. Furthermore, our proposed codes attain a near optimal tradeoff. Hence, we call them Maximum Distance and Span (MiDAS) codes.

In the second part of the paper, we consider the general case when $M > 1$, i.e., the source-packet arrival and channel-packet transmission rates are unequal. We propose an optimal construction for the burst erasure channel, and an achievable rate for the general case. Both the construction as well as the decoding analysis are a non-trivial extension of the case when $M = 1$. Finally we present extensive simulation results over the GE and Fritchman channels that suggest substantial performance gains over baseline codes for a wide range of channel parameters. We also discuss how the parameters of the sliding-window erasure channel can be selected based on the underlying statistical models of interest.

In other related works, references [6]–[10] study a multicast extension of [2], [3] to the case of two users and a common source stream. The stronger receiver's channel introduces shorter bursts and in turn, the decoding delay is required to be smaller. The weaker receiver's channel introduces longer bursts and the decoding delay can be longer. Such codes can also be used in applications where the decoding delay can vary based on channel conditions. The construction of these codes involves embedding the parity-checks of two single-user MS codes in a careful manner to simultaneously satisfy the constraints of both the receivers. References [11]–[13] study an extension of MS codes to parallel channels with burst erasures where the constructions involve a modification of MS codes to exploit the diversity across the parallel channels. In [14], streaming codes that can correct multiple bursts are proposed using an interleaving-type approach. We note that these references do not consider channels with burst and isolated erasure as in the present work. References [15]–[17] study streaming codes motivated by connections between streaming and unicast network coding and study channels with either burst erasures or i.i.d. erasures. However, to the best of our knowledge, these papers do not consider channels with *both* burst and isolated erasures, or the layered approach for coding, which is the focus of the present work. In the broader literature, problems involving real-time coding and compression have been studied from many different perspectives. Some structural properties of optimal codes have been studied in e.g., [18]–[20], and a dynamic programming based formulation is proposed. However, to the best of our knowledge, these papers do not consider explicit codes as considered here. Tree codes for streaming over i.i.d. channels are studied in [21]–[23]. There is also a significant body of literature on adapting various coding techniques for streaming systems, see e.g., [24]–[28] and references therein.

II. System Model and Main Results

In this section, we introduce the streaming setup and summarize the main results of the paper.

A. Sliding-Window Erasure Channel Model

We consider a class of packet erasure channels where the erasure patterns are locally constrained. In any sliding window of length W, the channel can introduce one of the following patterns: (i) a single erasure burst of maximum length B, or (ii) a maximum of N erasures in arbitrary locations. Note that the condition $N \leq B$ follows since a burst erasure is a special type of erasure pattern. We will assume throughout the paper that $B+1 \leq W$, so that in any window of length W there is at-least one non-erased packet[2]. We use the notation $\mathcal{C}(N, B, W)$ for such a channel. Note that the special case when $N = 1$ reduces the above model to a burst-only channel model. In this case, the guard separation between successive bursts is at-least $W - 1$.

In practice, we can view $\mathcal{C}(N, B, W)$ as an approximation of statistical models such as the Gilbert-Elliott (GE) channel

[1]The setup in [2], [3], only considers a single burst-erasure channel. But their construction also applies to the sliding window erasure channel when $W \geq T + 1$.

[2]If this condition is violated, it follows that the capacity is zero.

Fig. 1: An example of the sliding-window erasure channel $\mathcal{C}(N = 2, B = 3, W = 5)$. In any sliding window of length $W = 5$, there is either a single erasure burst of length no greater than $B = 3$, or no more than $N = 2$ isolated erasures.

model. A GE channel is in one of two states. In the good state, it behaves as an i.i.d. erasure channel, while in the bad state, it behaves as a burst-erasure channel. Thus, the interval consisting of a burst loss corresponds to the bad state, whereas a window comprising of isolated erasures corresponds to the good state. Fig. 1 provides an example of $\mathcal{C}(N = 2, B = 3, W = 5)$. The intervals $[0, 2]$ and $[12, 14]$ of burst erasures correspond to the bad state of the GE channel, whereas the interval $[3, 11]$ corresponds to the good state of the GE channel. The advantage of studying the $\mathcal{C}(N, B, W)$ is that it is a tractable model for capacity analysis. Furthermore, streaming codes developed using this model with suitable parameters can provide significant gains over statistical models. This will be verified in our simulation results over the Gilbert-Elliott and Fritchman channel models.

We separately treat the cases when the source-packet inter-arrival rates and channel-packet transmission rates are equal and when they are not. For convenience we refer to these as *source-channel inter-arrival rates.*

B. Equal Source-Channel Inter-arrival Rates

At each time-slot $i \geq 0$, the encoder observes a source packet $\mathbf{s}[i]$, and transmits a channel packet $\mathbf{x}[i]$ as shown in Fig. 2. We will assume throughout that the source packet consists of k symbols, while the channel packet consists of n symbols over a common finite field $\mathbb{F}_q$. The rate of the code equals $R = \frac{k}{n}$. The channel input at time i can depend causally on all the source packets observed up to and including time i, but not on any future packets, i.e., $\mathbf{x}[i] = f_i(\mathbf{s}[0], \ldots, \mathbf{s}[i])$. The channel output at time i is denoted by the packet $\mathbf{y}[i]$. Note that either $\mathbf{y}[i] = \mathbf{x}[i]$, or $\mathbf{y}[i] = \star$, when the channel introduces an erasure. The decoder is required to reconstruct each source packet with a delay of T, i.e., for each $i \geq 0$ we must have a decoding function, $\mathbf{s}[i] = g_i(\mathbf{y}[0], \ldots, \mathbf{y}[i + T])$. Such a collection of encoding and decoding functions constitutes the *streaming code*.

Definition 1 (Streaming Capacity - Equal Source-Channel Inter-arrival Rates)**.** *A rate R is achievable with a delay of T over the $\mathcal{C}(N, B, W)$ channel, if there exists a streaming code of this rate over some field of size q such that every source packet $\mathbf{s}[i]$ can be decoded at the destination with a delay of T packets. The supremum of all achievable rates is the streaming capacity.*

We establish the following upper and lower bounds on the streaming capacity.

Theorem 1 (Upper Bound)**.** *For the sliding-window erasure channel, $\mathcal{C}(N, B, W)$, an achievable rate R with delay $T \geq B$ must satisfy the following upper bound:*

$$\left(\frac{R}{1-R}\right) B + N \leq T_{\text{eff}} + 1, \tag{1}$$

where $T_{\text{eff}} \triangleq \min(T, W - 1)$. □

Theorem 1 provides an upper bound on the rate for any code, linear or not, and any field size q. Alternatively, when the rate R and delay T are fixed, Eq. (1) illustrates a tradeoff between the achievable values of B and N. We cannot have a streaming code that can simultaneously correct long erasure bursts and many isolated erasures. The proof of Theorem 1 is provided in Section IV-A.

Note that $T_{\text{eff}} + 1 = \min(T + 1, W)$, where $T + 1$ denotes the *active* duration of each source packet, i.e., each source packet $\mathbf{s}[i]$ arrives at time $t = i$ and must be decoded by time $t = i + T$. When $W \geq T + 1$ the right hand side in the upper bound in (1) depends on the decoding delay T and not on the window W; otherwise it depends on W. In the latter case, the upper bound corresponds to the capacity of a certain periodic erasure channel with $T = \infty$. The intuition here is that the $\mathcal{C}(N, B, W)$ channel only controls the allowable erasure patterns in a local window of length W. If $W < T + 1$, one can construct an erasure pattern such that the channel packets received after time $i + W - 1$ cannot be useful for the decoding of source packet $\mathbf{s}[i]$. Thus the effective delay reduces to $W - 1$. Indeed our lower bound involves the construction of a streaming code the delay of $W - 1$, instead of T.

Theorem 2 (Lower Bound)**.** *For any channel $\mathcal{C}(N, B, W)$ and delay $T \geq B$, there exists a code of rate R that satisfies*

$$\left(\frac{R}{1-R}\right) B + N > T_{\text{eff}}, \tag{2}$$

where $T_{\text{eff}} \triangleq \min(T, W - 1)$. □

The proof of Theorem 2 is presented in Section IV-C. Upon examining the upper bound (1) and the lower bound (2) we note that the right hand side only differs by one. Thus, the proposed codes are optimal within one unit of T_{eff}.

The streaming codes proposed in Theorem 2 is a new family of codes and will be called *Maximum Distance And Span tradeoff (MiDAS) codes*. Their construction is based on a two-step, layered coding approach. In the first step, we construct an optimal streaming code for $\mathcal{C}(N' = 1, B, W)$ channel. Then we append an additional layer of parity-check packets that enables the recovery from the N isolated erasures. While the recovery from the burst erasure does not use the parity-checks generated in the second step, the recovery from isolated erasures uses parity-checks from both the steps. To our knowledge MiDAS codes constitute the first family of streaming codes that can recover from both burst and isolated erasures and demonstrate significant gains over the Gilbert-Elliott channel and Fritchman channel over a wide range of parameters.

Note that Theorem 2 does not explicitly state the field-size q. The underlying constructions are based on systematic Strongly-MDS convolutional codes [4], [5] and their vari-

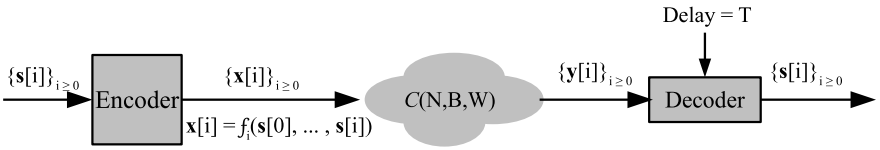


Fig. 2: The source stream $\mathbf{s}[t]$ for $t \geq 0$ is causally encoded to a channel stream $\mathbf{x}[t]$ which is transmitted over the sliding window erasure channel $\mathcal{C}(N, B, W)$. The decoder tolerates a maximum delay of T packets.

ants, which are known to exist for field-sizes that increase exponentially in T_{eff}. However, we also provide an alternate construction in Section IV-D, that satisfies (2), and whose field-size increases as $\mathcal{O}(T_{\text{eff}}^3)$.

Remark 1 (Column Distance and Column Span Property)**.** *By definition the streaming code is a (n, k) convolutional code, where the source stream $\mathbf{s}[i] \in \mathbb{F}_q^k$ is the input and $\mathbf{x}[i] \in \mathbb{F}_q^n$ is the output. Traditional constructions for convolutional codes maximize the underlying free distance [29]. However, in the present setup, the* column distance *and* column span *(see Def. 3 and 4 in Appendix A) determine the error correction properties. In Appendix A, we establish necessary and sufficient conditions on the column distance and column span of any feasible convolutional code for the $\mathcal{C}(N, B, W)$ channel.*

The following result, which is a consequence of Theorem 1 and Theorem 2, appears new and could be of independent interest.

Proposition 1 (A Fundamental Tradeoff between Column Distance and Column Span)**.** *For any (n, k) convolutional code with rate $R = \frac{k}{n}$, and an integer $T > 0$, the column distance d_T and the column span c_T must satisfy the upper following bound:*

$$\left(\frac{R}{1-R}\right) c_T + d_T \leq T + 1 + \frac{1}{1-R}. \tag{3}$$

Furthermore, there exists a convolutional code with column distance d_T and column span c_T, over a sufficiently large field-size that satisfy:

$$\left(\frac{R}{1-R}\right) c_T + d_T \geq T + \frac{1}{1-R}. \tag{4}$$

□

Proof: See Appendix A. ■

C. Unequal Source-Channel Inter-arrival Rates

We discuss a generalization of the setup in Section II-B where one source packet arrives every M channel uses. As before each source packet consists of k symbols and channel packets consists of n symbols, each over a finite field $\mathbb{F}_q$. For convenience, the collection of M channel packets is termed as a macro-packet. The index of each macro-packet is denoted using the letter i, i.e.,

$$\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \ldots \mid \mathbf{x}[i, M]] \in \mathbb{F}_q^{n \times M} \tag{5}$$

denotes the macro-packet i consisting of M channel packets. At the start of macro-packet i, the encoder observes the source packet $\mathbf{s}[i] \in \mathbb{F}_q^k$ and generates M packets $\mathbf{x}[i, j] \in \mathbb{F}_q^n$, for $j \in \{1, \ldots, M\}$ which can depend on all the observed source packets up to that time, i.e.,

$$\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \cdots, \mathbf{s}[i]). \tag{6}$$

These packets are transmitted in the M time-slots corresponding to the macro-packet i. Fig. 3 shows the system model. Note that for the case when $M = 1$ the setup reduces to that in Section II-B.

The jth channel output packet in the macro-packet i is denoted by $\mathbf{y}[i, j]$. When the channel input is not erased, we have, $\mathbf{y}[i, j] = \mathbf{x}[i, j]$, whereas when the channel input is erased, $\mathbf{y}[i, j] = \star$. The channel output macro-packets are expressed as $\mathbf{Y}[i, :] = [\mathbf{y}[i, 1] \mid \ldots \mid \mathbf{y}[i, M]]$. The decoder is required to decode each source packet with a maximum delay of T macro-packets, i.e.,

$$\mathbf{s}[i] = g_i(\mathbf{Y}[0, :], \mathbf{Y}[1, :], \cdots, \mathbf{Y}[i + T, :]). \tag{7}$$

We define the rate of this code by $R = \frac{k}{Mn}$. In the above definition, we are normalizing the size of each source packet with the size of each macro-packet. This is due to the fact that in our proposed setup, a total of Mn channel symbols are transmitted for each k source symbols.

Definition 2 (Streaming Capacity - Unequal Source-Channel Inter-arrival Rates)**.** *A rate R is achievable with a delay of T macro-packets over $\mathcal{C}(N, B, W)$ if there exists a streaming code of this rate over some field of size q such that every source packet $\mathbf{s}[i]$ can be decoded with a delay of T macro-packets. The supremum of all achievable rates is the streaming capacity.*

For the above setup, the capacity has been obtained when $N = 1$ and $W \geq M(T + 1)$.

Theorem 3. *For the channel $\mathcal{C}(N = 1, B, W)$, and any M and delay T, such that $W \geq M(T + 1)$ the capacity is expressed as follows. Let b and B' be defined via*

$$B = bM + B', \quad B' \in \{0, \ldots, M - 1\}, b \in \mathbb{N}^0. \tag{8}$$

The capacity C for $T > b$ is given by:

$$C = \begin{cases} \frac{T}{T+b}, & 0 \leq B' \leq \frac{b}{T+b} M, \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & \frac{b}{T+b} M < B' \leq M - 1. \end{cases} \tag{9}$$

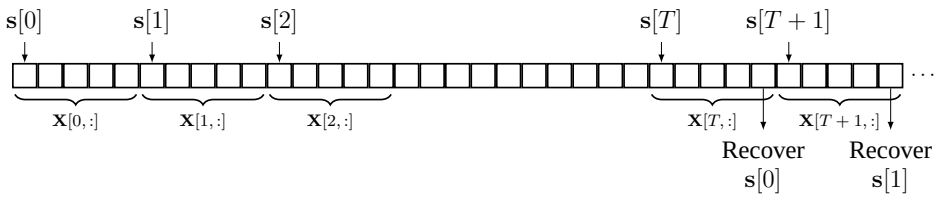


Fig. 3: Each source symbol $\mathbf{s}[i]$ arrives just before the transmission of $\mathbf{X}[i,:]$ and needs to be reconstructed at the destination after a delay of T macro-packets.

For the minimum delay case, $T = b$, we have:

$$C = \begin{cases} \frac{1}{2}, & 0 \leq B' \leq \frac{M}{2}, \\ \frac{M-B'}{M}, & \frac{M}{2} < B' \leq M-1, \end{cases} \tag{10}$$

Finally, $C = 0$ for $T < b$. □

The proof of Theorem 3 is divided into two main parts. The code construction is illustrated in Section V-B while the converse appears in Section V-E.

We make several remarks pertaining to Theorem 3. Recall that since the delay T is expressed in terms of the macro-packets (see Fig. 1), the case when $T = b$ corresponds to the minimum possible delay. In this special case, the capacity can be attained using a repetition code. When $T > b$ the optimal codes, as well as the decoding analysis, are a non-trivial extension of the case when $M = 1$. Secondly note that the capacity expression in (9) involves two cases. In the first case, which corresponds to $bM \leq B \leq bM + \frac{b}{T+b}M$, the capacity stays constant at $C = \frac{T}{T+b}$ even as B is increased in this range. To explain this, note that the threshold $\frac{b}{T+b}M = (1-C)M$, equals the number of parity-check packets in a given macro-packet. This observation can be used to construct a code such that up to B' erasures in the last macro-packet do not reduce the capacity. This will be further explained in Section V-A.

We remark that the capacity result in Theorem 3 can be used to obtain bounds on the symbol-level column span of a convolutional code. This is in contrast to our treatment in section II-B where the packet-level column span was considered. We do not discuss the associated results in this paper, but refer the reader to [30].

Finally note that the constructions in Theorem 3 only apply to the burst-erasure channel. Based on the layered approach in Theorem 2 we also propose a robust construction for the case when $N > 1$ in Section V-F. However, the optimal construction is left for future work.

This completes the discussion of the main results in the paper. The rest of the paper is organized as follows. In Section III, we review some previously proposed codes. We treat the case of when source channel inter-arrival rates are equal in Section IV and propose our MiDAS codes. The case of unequal rates is treated in Section V. Simulation results are presented in Section VI and Conclusions are presented in section VII.

III. Performance Analysis of Baseline Schemes

We review two constructions — m-MDS codes and Maximally Short codes — that have been proposed in earlier works. While these codes are optimal only in some special cases, they constitute important building blocks in our proposed constructions.

A. m-MDS Codes

In the traditional approach to erasure correction, the decoder must wait till sufficiently many parity-check symbols are collected so that all the source symbols can be recovered simultaneously by inverting a full-rank system of equations. In *random-linear codes*, see e.g., [24], [31], [32], the coefficients of the linear code are selected at random to guarantee near optimal recovery with high probability. However, instead of random codes, we consider a class of *deterministic code* constructions with optimal distance properties [4], [5] in this section.

Consider a $(\bar{n}, \bar{k}, \bar{m})$ convolutional code that maps an input source stream $\mathbf{s}[i] = (s_0[i], \ldots, s_{\bar{k}-1}[i])^\dagger \in \mathbb{F}_q^{\bar{k}}$ to an output $\mathbf{x}[i] = (x_0[i], \ldots, x_{\bar{n}-1}[i])^\dagger \in \mathbb{F}_q^{\bar{n}}$ using a memory $\bar{m}$ encoder[3]. In particular, let

$$\mathbf{x}[i] = \left(\sum_{t=0}^{\bar{m}} \mathbf{s}^\dagger[i-t] \cdot \mathbf{G}_t \right)^\dagger, \tag{11}$$

where $\mathbf{G}_0, \ldots, \mathbf{G}_{\bar{m}}$ are $\bar{k} \times \bar{n}$ matrices with elements in $\mathbb{F}_q$. Furthermore, the convolutional code is systematic[4] if we can express each generator matrix in the following form,

$$\mathbf{G}_0 = [\mathbf{I}_{\bar{k}\times\bar{k}} \ \mathbf{H}_0], \qquad \mathbf{G}_t = [\mathbf{0}_{\bar{k}\times\bar{k}} \ \mathbf{H}_t], \ t = 1, \ldots, \bar{m} \tag{12}$$

where $\mathbf{I}_{\bar{k}\times\bar{k}}$ denotes the $\bar{k} \times \bar{k}$ identity matrix, $\mathbf{0}_{\bar{k}\times\bar{k}}$ denotes the $\bar{k} \times \bar{k}$ zero matrix, and $\mathbf{H}_t \in \mathbb{F}_q^{\bar{k}\times(\bar{n}-\bar{k})}$ for $t = 0, 1, \ldots, \bar{m}$. For a systematic convolutional code, (11) reduces to

$$\mathbf{x}[i] = \begin{bmatrix} \mathbf{s}[i] \\ \mathbf{p}[i] \end{bmatrix}, \qquad \mathbf{p}[i] = \left(\sum_{t=0}^{\bar{m}} \mathbf{s}^\dagger[i-t] \cdot \mathbf{H}_t \right)^\dagger. \tag{13}$$

The m-MDS codes (see e.g. [5, Corollary 2.5]), correspond to a certain choice of $\mathbf{H}_t$ that result in the maximum column

[3]We use † to denote the vector/matrix transpose operation. Throughout this paper, we will treat $\mathbf{s}[i]$ and $\mathbf{x}[j]$ as column vectors and therefore $\mathbf{s}^\dagger[i]$ and $\mathbf{x}^\dagger[j]$ denote the associated row vectors. For convenience, we will not use the † notation when the dimensions are clear.

[4]Throughout the paper, we only consider systematic m-MDS codes and thus the word systematic is dropped for convenience

distance (see Appendix B). This in turn results in following error correction properties in the streaming setup.

Lemma 1. *Consider a systematic $(\bar{n}, \bar{k}, \bar{m})$ m-MDS code and suppose that the symbols in each packet $\mathbf{x}[i]$, i.e.,*

$$\mathbf{x}[i] = \big(s_0[i], \ldots, s_{\bar{k}-1}[i], p_0[i], \ldots, p_{\bar{n}-\bar{k}-1}[i]\big) \tag{14}$$

are transmitted sequentially in the time interval $[i \cdot \bar{n}, (i+1) \cdot \bar{n} - 1]$ over the channel[5]. The following properties hold for each $j = 0, 1, \ldots, \bar{m}$.

- *L1. If $\hat{N}$ transmitted symbols are erased in the interval $[0, (j+1)\bar{n} - 1]$ where $\hat{N} \leq (\bar{n} - \bar{k})(j+1)$, then $\mathbf{s}[0] = (s_0[0], \ldots, s_{\bar{k}-1}[0])$ can be recovered by time $(j+1)\bar{n} - 1$.*
- *L2. If the channel introduces an erasure-burst of length $\hat{B}$ symbols in the interval $[c, c + \hat{B} - 1]$, where $\hat{B} \leq (\bar{n} - \bar{k})(j+1)$ and $0 \leq c \leq \bar{k} - 1$, then all erased source packets are recovered by time $(j+1)\bar{n} - 1$.*
- *L3. If the channel introduces an erasure burst of length $\hat{B}$ symbols in the interval $[c, c+\hat{B}-1]$, where $0 \leq c \leq \bar{k}-1$, followed by a total of no more than $\hat{I}$ isolated erasures such that $\hat{B} + \hat{I} \leq (\bar{n} - \bar{k})(j+1)$, then all the erased packets in the burst are recovered by time $(j+1)\bar{n} - 1$.*

Proof: See Appendix B. ■

We now discuss how the properties in Lemma 1 can be applied to our system model. In the case in Section II-B, when the source and channel inter-arrival rates are equal, Lemma 1 immediately yields the following.

Corollary 1. *Consider a systematic $(\bar{n}, \bar{k}, \bar{m})$ m-MDS code of rate $R = \frac{\bar{k}}{\bar{n}}$ which transmits the entire channel packet $\mathbf{x}[i] = (x_0[i], \ldots, x_{\bar{n}-1}[i]) \in \mathbb{F}_q^{\bar{n}}$ in time-slot i. For each $j = 0, 1, \ldots, \bar{m}$, we have the following,*

- *P1. Suppose that in the window $[0, j]$, the channel introduces $N \leq (1-R)(j+1)$ erasures in arbitrary locations, then $\mathbf{s}[0]$ is recovered by time $t = j$.*
- *P2. Suppose an erasure burst happens in the interval $[0, B-1]$, where $B \leq (1-R)(j+1)$, then all the packets $\mathbf{s}[0], \ldots, \mathbf{s}[B-1]$ are simultaneously recovered by time $t = j$.*

□

Proof: To establish property P1 we invoke property L1 in Lemma 1. Note that in P1 we consider the transmission of channel packets whereas in Lemma 1 we consider the transmission of symbols. Note that N packet erasures leads to $\hat{N} = nN$ symbol erasures. Thus, $N \leq (1-R)(j+1)$ is equivalent to $\hat{N} \leq (n-k)(j+1)$ symbol erasures. Furthermore, the interval $[0, j]$ in P1 associated with the transmission of the first $j+1$ packets maps to the interval $[0, (j+1)n-1]$ in L1. Thus, the entire packet $\mathbf{s}[0]$ is guaranteed to be recovered.

Property P2 follows in an analogous fashion upon using property L2 in Lemma 1 with $c = 0$.

■

[5]Note that in this statement we are only transmitting symbols over $\mathbb{F}_q$ over the channel. Subsequently, we will adapt these properties for transmitting packets over $\mathbb{F}_q^{\bar{n}}$

From Corollary 1, it follows that any (N, B) pair that satisfies

$$N \leq (1-R)(T+1), \quad B \leq (1-R)(T+1) \tag{15}$$

is achieved using a (n, k, T) m-MDS code of rate $R = \frac{k}{n}$ with delay T and $W \geq T+1$. In particular, if the channel introduces up to $(1-R)(T+1)$ erasures in the window $[0, T]$, it follows from Property P1 in Corollary 1 that $\mathbf{s}[0]$ is recovered at $t = T$. Once $\mathbf{s}[0]$ has been recovered, its effect can be subtracted out from all parity-checks involving $\mathbf{s}[0]$. By the same property, $\mathbf{s}[1]$ is guaranteed to be recovered at time $t = T+1$. This argument can be successively repeated until all the erased packets are recovered. Furthermore, upon substituting $B = N$ in (1), we note that the m-MDS attain one extreme point on the tradeoff, namely when $N = B$. This is clearly the largest feasible value of N in (1).

In a similar fashion, it can be shown that for the case of unequal source-channel inter-arrival rates in Section II-C , when $W \geq M(T+1)$, any (N, B) is that satisfies

$$N \leq M(1-R)(T+1), \quad B \leq M(1-R)(T+1) \tag{16}$$

is achieved using a (Mn, k, T) m-MDS code with rate $R = \frac{k}{Mn}$.

B. Maximally Short (MS) Codes

While the m-MDS codes achieve the extreme point of the upper bound (1) corresponding to $N = B$, the Maximally Short (MS) codes [2], [3] achieve the other extreme point, corresponding to $N = 1$. In particular, the maximum value of B with $N = 1$ is given in the following result.

Lemma 2 (Martinian and Sundberg [2], Martinian and Trott [3])**.** *Consider the channel $\mathcal{C}(N = 1, B, W)$ with $W \geq T+1$ and $M = 1$. There exists an MS code of rate R satisfying*

$$R = \begin{cases} \frac{T}{T+B}, & B \leq T, \\ 0, & \text{else}. \end{cases} \tag{17}$$

Furthermore, R in (17) *is the maximum achievable rate for $\mathcal{C}(N = 1, B, W \geq T+1)$ channel.* □

The construction of MS codes presented in [2], [3] involves first constructing a specific low-delay block code and then converting it into a streaming code using a diagonal interleaving technique. Thus, the problem of constructing a streaming code is reduced to the problem of constructing a block code with certain properties. While such a simplification is appealing, unfortunately it does not appear to easily generalize when seeking extensions of MS codes. Note that the above MS codes can only achieve $N = 1$ and are highly sensitive to isolated losses over the channel. In [2], some examples of codes with higher N were reported using a numerical search but a general approach for constructing robust streaming codes remained elusive. In Section IV-B, we present an alternative perspective that easily extends to achieve a near optimal rate for any (N, B).

For the case of unequal source-channel inter-arrival rates, a straightforward adaptation of the MS codes is as follows.

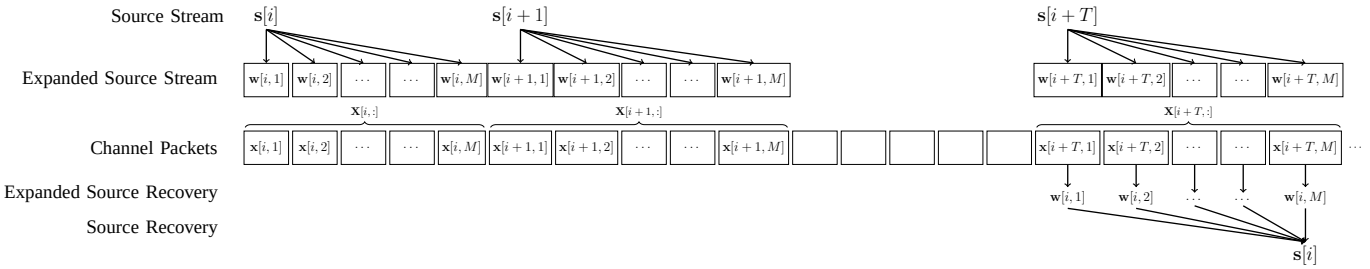


Fig. 4: Each source symbol $\mathbf{s}[i]$ is split into M sub-packets i.e., $\mathbf{s}[i] = (\mathbf{w}[i,1], \mathbf{w}[i,2], \ldots, \mathbf{w}[i,M])$. The expanded source stream is then encoded using a Maximally-Short code. The decoder recovers each $\mathbf{w}[i,j]$ once $\mathbf{y}[i+T,j]$ is received which ensures that $\mathbf{s}[i]$ is recovered by the end of the macro-packet $i+T$.

TABLE I: Achievable (N,B) for channel $\mathcal{C}(N,B,W \geq T+1)$ for equal source-channel inter-arrival rates. Similar Tradeoffs for the first three codes can be achieved for $W < T+1$ by replacing T with $W-1$.

Code	N	B
m-MDS Codes	$(1-R)(T+1)$	$(1-R)(T+1)$
Maximally Short Codes	1	$T \cdot \min\left(\frac{1}{R}-1, 1\right)$
MiDAS Codes	$\min\left(B, T-\frac{R}{1-R}B\right)$	$B \in [1,T]$
E-RLC Codes [33] $\Delta \in [R(T+1), T-1]$, $R \geq 1/2$	$\frac{1-R}{R}(T-\Delta)+1$	$\frac{1-R}{R}\Delta$

We split each packet $\mathbf{s}[i]$ into M sub-packets, one for each time-slot in the macro-packet and then apply a MS code in Lemma 2 to this expanded source stream with delay $T' = MT$ (cf. Fig. 4). In particular, we assume that $\mathbf{s}[i] \in (\mathbb{F}_q)^{kM}$ and proceed as follows.

- Split each $\mathbf{s}[i] = (\mathbf{w}[i,1], \ldots, \mathbf{w}[i,M])$ where $\mathbf{w}[i,j] \in \mathbb{F}_q^k$.
- Apply a MS code in Lemma 2 for the $\mathcal{C}(N=1,B,W)$ channel with delay $T' = MT$ (channel packets) and $W \geq M(T+1)$.
- Transmit the associated channel packet $\mathbf{x}[i,j] \in \mathbb{F}_q^n$ in slot j of the macro-packet i.

From (17) we have that

$$R = \frac{MT}{MT+B} = \frac{T}{T+b+\frac{B'}{M}} \tag{18}$$

is achievable when $B \leq MT$. Note that in the second equality in (18), we use (8). Note that the delay of $T' = M \cdot T$ channel packets in the expanded stream, implies that $\mathbf{w}[i,j]$ is recovered when $\mathbf{y}[i+T,j]$ is received for each $j \in \{1,2,\ldots,M\}$. Thus, the entire source packet $\mathbf{s}[i]$ is guaranteed to be recovered at the end of macro-packet $i+T$, thus satisfying the delay constraint. We note that the rate in (18) is only positive if $B \leq MT$ and attains the capacity in Theorem 3 in the special case when $B' = 0$. If $B > MT$ the above construction is not feasible and the rate attained is zero.

C. Numerical Comparisons

1) Equal Source-Channel Inter-arrival Rates: Table I summarizes the feasible values of N and B for different codes[6]. For a fixed rate R and delay T we indicate the values of N and B achieved by various codes in the case of equal source-channel inter-arrival rates. The first row corresponds to the m-MDS in Section III-A, while the second row corresponds to the MS codes in Section III-B. The third row corresponds to our proposed construction — MiDAS codes — in Theorem 2. In contrast to the m-MDS codes and MS codes, that only attain specific values of N and B, the family of MiDAS codes can attain a range of (N,B) for a given R and T. The last row corresponds to another family of codes – Embedded Random Linear Codes (E-RLC) – proposed in [33]. While such constructions are optimal for $R = 1/2$, they are far from optimal in general and will not be discussed in this paper.

We further numerically illustrate the achievable (N,B) pairs for various codes in Fig. 5. We fix the rate to $R = 0.6$. As stated before, the m-MDS and MS codes in Sections III-A and III-B respectively only achieve the extreme points on the tradeoff. The MiDAS codes achieve a tradeoff, very close to the upper bound for all rates. The E-RLC codes, illustrated with the red plot, are generally far from optimal except for $R = 0.5$ which is not the case in this figure.

2) Unequal Source-Channel Inter-arrival Rates: Fig. 6 illustrates the capacity and rates achieved with baseline schemes for the case of unequal source-channel inter-arrival rates. In this example, we consider $M = 20$ and a delay of $T = 5$ macro-packets and plot the rate vs. correctable burst length. The capacity is shown by the blue-curve marked with squares. Note that it is constant in the intervals $B \in [40,45], [60,67], [80,88], [100,110]$. The red curve marked with circles denotes the rate achieved by a suitable modification of the MS code (18). We note that the curves intersect whenever B is an integer multiple of M, indicating the optimality of the MS codes for these special values: $B \in \{40,60,80,100\}$. Furthermore, for burst lengths $B > MT = 100$, the MS codes are no longer feasible and the associated rate is zero. The dotted black line shows the performance of the m-MDS codes in (16). Since these codes do not perform sequential recovery, their achievable rate is significantly lower than the capacity.

[6]We note that the floor of the values given in Table I should be considered as the values might not be integers

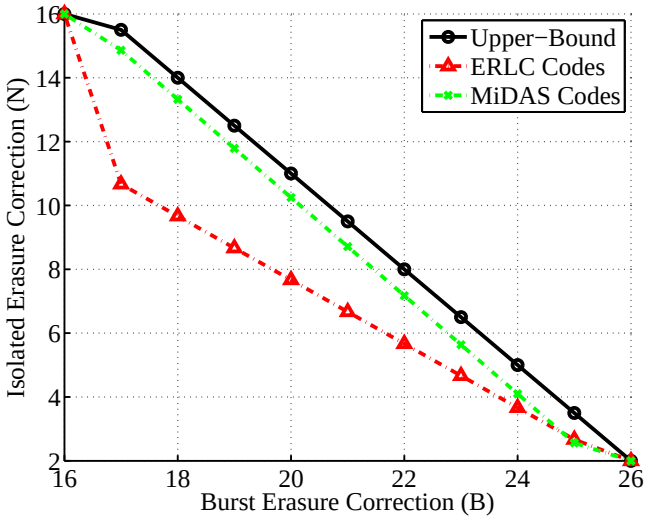


Fig. 5: Achievable tradeoff between N and B for equal source-channel inter-arrival rates. The rate is fixed to $R = 0.6$ and the delay is fixed to $T = 40$ and $W = T + 1$. The uppermost curve (solid black lines with 'o') is the upper bound in (1). The MiDAS codes are shown with broken green lines with '$\times$' and are very close to the upper bound. The E-RLC codes in [33] are shown with broken red lines with '$\triangle$'.

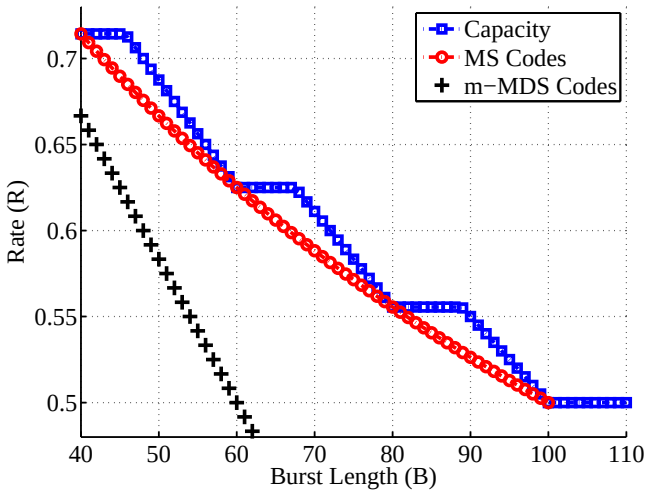


Fig. 6: Achievable rates for different code constructions for the case of unequal source-channel inter-arrival rates for the $\mathcal{C}(N = 1, B, W = M(T + 1))$ channel. We fix the delay to $T = 5$ macro-packets and let $M = 20$. The blue plot (marked with squares) corresponds to the capacity in Theorem 3. The red curve (marked with circles) corresponds to the rate achieved by the adapted MS code (18) whereas the black line corresponds to the rate of the m-MDS code (16).

IV. Equal Source-Channel Inter-arrival Rates

In this section, we consider the case when source and channel inter-arrival rates are equal, i.e., $M = 1$. We start by establishing the upper-bound in Theorem 1 in Section IV-A. In Section IV-B, we revisit the Maximally Short codes for the $\mathcal{C}(N = 1, B, W)$ channel and propose a modification that uses m-MDS and the repetition code as its constituent codes. We then extend these constructions to the $\mathcal{C}(N, B, W)$ channel in Section IV-C to construct MiDAS codes. In Section IV-D, we provide an alternative construction of MiDAS codes achieving the same tradeoff in Theorem 2 but with a smaller field-size. Finally, we compare the performance of the two constructions through an example in Section IV-E.

A. Upper-bound

To establish the upper bound in Theorem 1, we separately consider the cases where $W \geq T + 1$ and $W < T + 1$. When $W \geq T + 1$, consider a periodic erasure channel with a period of $\tau_P = T + B - N + 1$ and suppose that in every such period the first B packets are erased (see Fig. 7). While such a channel is not included in $\mathcal{C}(N, B, W)$, we nonetheless show that any code for $\mathcal{C}(N, B, W)$ and delay T is also feasible for the proposed periodic erasure channel.[7]

Consider the first period that spans the interval $[0, \tau_P - 1]$. We note the following

- The first $B - N + 1$ packets, $\{\mathbf{s}[i]\}_{0 \leq i \leq B-N}$, must be all recovered with delay T since the recovery window $[i, i + T]$ of each such packet only have a burst of length B or smaller. Thus, all these packets are recovered by time $t = \tau_P - 1$.
- The recovery window of each of the $N - 1$ packets, $\{\mathbf{s}[i]\}_{B-N+1 \leq i \leq B-1}$ is $[i, i + T]$ which sees two bursts. The first burst spans $[i, B - 1]$ and is of length $B - i$. The second burst spans $[T + B - N + 1, i + T]$ and is of length $i + N - B$. Thus, the total number of erased packets in each recovery period is exactly N. Thus, any feasible code over the $\mathcal{C}(N, B, W)$ channels guarantees that each such packet is also recovered at time $i + T$.
- The recovery window of each of the remaining packets in the first period, $\mathbf{s}[B], \ldots, \mathbf{s}[\tau_P - 1]$, again sees a single-erasure burst of length B at the end of the window. Hence, each of these packets is also guaranteed to be recovered with delay no more than T, in particular, by time $\tau_P - 1$.

We have thus shown that all the packets in the first period spanning $[0, \tau_P - 1]$ can be recovered with delay T. We can repeat the same argument for all the remaining periods and thus the claim follows. Thus, using the capacity of the periodic erasure channel, we have

$$R \leq 1 - \frac{B}{T + B - N + 1}. \tag{19}$$

For the case when $W < T + 1$, we consider a periodic erasure channel with a period of $\tau_P = W + B - N$ where in each period the first B packets are erased and the remaining $W - N$ packets are not erased. Such a channel by construction is a $\mathcal{C}(N, B, W)$ channel. In any window $\mathcal{W}_i = [i, i + W - 1]$

[7] A similar converse argument involving periodic erasure channel for the burst-erasure channel is also presented in [2], [7], [8]. For a rigorous information theoretic argument, we refer the reader to [9], [10], [13] for the case of burst erasure channel. A similar approach can be used in the present setup, but it will not be presented.

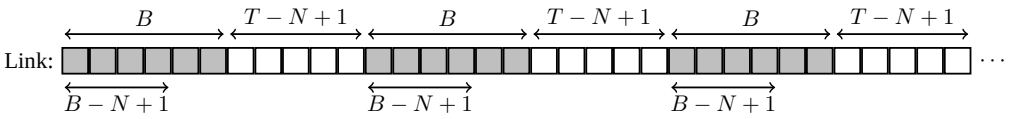


Fig. 7: The periodic erasure channel in the proof Theorem 1. The shaded symbols are erased while the remaining ones are received by the destination.

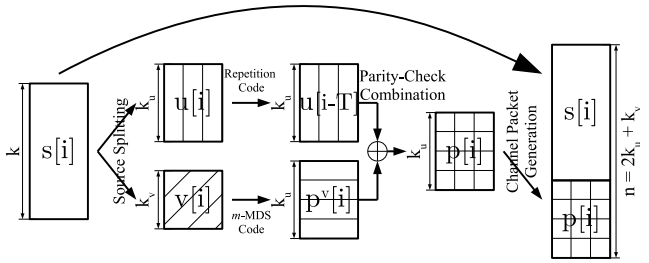


Fig. 8: A block diagram illustrating the encoding steps of a Generalized MS code. The source packet is first split into two packets and a different code is applied to each packet. The resulting parity-checks are then combined to form the overall parity-check packet. Finally, the parity-check packet and the source packet are concatenated to generate the channel packet.

of length W, there exists either a single burst of maximum length B, or up to N isolated erasures. Thus, every erased packet on such a channel must be recovered, i.e., we have that

$$R \leq 1 - \frac{B}{W+B-N}. \tag{20}$$

Rearranging (19) and (20) and using $T_{\text{eff}} = \min(W-1, T)$, we easily recover (1). This completes the proof of the upper bound.

B. Generalized MS Codes

In this section, we present a generalization of the MS codes introduced in Section III-B. The proposed construction applies to any $W \geq B+1$, and eliminates the intermediate step of constructing a block code in [2], [3]. This method can be then generalized to correct both burst and isolated erasures for the $\mathcal{C}(N,B,W)$ channel.

Proposition 2. *Let $T_{\text{eff}} \triangleq \min(W-1, T)$. For the $\mathcal{C}(N=1,B,W)$ channel, there exists a streaming code with delay T and rate*

$$R = \begin{cases} \frac{T_{\text{eff}}}{T_{\text{eff}}+B}, & T_{\text{eff}} \geq B \\ 0, & \textit{else}. \end{cases} \tag{21}$$

□

The encoding steps, illustrated in Fig. 8, are as follows:

- **Source Splitting:** Split each source packet $\mathbf{s}[i] \in \mathbb{F}_q^k$ into two sub-packets[8] $\mathbf{u}[i] \in \mathbb{F}_q^{k^u}$ and $\mathbf{v}[i] \in \mathbb{F}_q^{k^v}$ as follows,
$$\mathbf{s}[i] = (\underbrace{u_0[i], \ldots, u_{k^u-1}[i]}_{=\mathbf{u}[i]}, \underbrace{v_0[i], \ldots, v_{k^v-1}[i]}_{=\mathbf{v}[i]}), \tag{22}$$
where $k^u + k^v = k$, i.e., $\mathbf{u}[i]$ constitutes the first k^u symbols in $\mathbf{s}[i]$ whereas $\mathbf{v}[i]$ constitutes the remaining k^v symbols.
- **m-MDS Parity-Checks:** Apply a $(k^u+k^v, k^v, T_{\text{eff}})$ m-MDS code of rate $R^v = \frac{k^v}{k^u+k^v}$ on the sub-packets $\mathbf{v}[i]$ and generate parity-check packets
$$\mathbf{p}^v[i] = \left(\sum_{j=0}^{T_{\text{eff}}} \mathbf{v}^\dagger[i-j] \cdot \mathbf{H}_j^v \right)^\dagger, \quad \mathbf{p}^v[i] \in \mathbb{F}_q^{k^u}, \tag{23}$$
where the matrices $\mathbf{H}_j^v \in \mathbb{F}_q^{k^v \times k^u}$ are associated with a m-MDS code (12).
- **Repetition Code:** Superimpose the $\mathbf{u}[\cdot]$ sub-packets onto $\mathbf{p}^v[\cdot]$ and let
$$\mathbf{q}[i] = \mathbf{p}^v[i] + \mathbf{u}[i - T_{\text{eff}}]. \tag{24}$$
- **Channel Packet Generation:** Concatenate the generated parity-checks to the source packets so that the channel input at time i is given by $\mathbf{x}[i] = (\mathbf{u}[i], \mathbf{v}[i], \mathbf{q}[i]) \in \mathbb{F}_q^n$, where $n = 2k^u + k^v$.

In our construction discussed above, we select $k = T_{\text{eff}}$, $k^u = B$, $k^v = T_{\text{eff}} - B$ and $n = T_{\text{eff}} + B$. Clearly the rate of the proposed code $R = \frac{k}{n}$ matches the expression in (21).

For decoding, we suppose that the first erasure burst of length B spans the interval $[0, B-1]$. Since the code is time-invariant a completely analogous argument applies when the erasure burst spans the interval $[i, i+B-1]$ for any $i > 0$ and the source packets upto time $i-1$ have been already recovered. By the definition of the sliding window erasure channel, there can be no other erasures for T_{eff} packets following the erasure burst and in particular all the channel packets $\mathbf{s}[i]$ for $i \in [B, T_{\text{eff}}+B-1]$ are recovered. We claim that each $\mathbf{s}[0], \mathbf{s}[1], \ldots, \mathbf{s}[B-1]$ is recovered by time $t = T_{\text{eff}}, T_{\text{eff}}+1, \ldots, T_{\text{eff}}+B-1$ respectively.

The decoder proceeds in two steps as illustrated in Table II:

- *Simultaneously recover* $\mathbf{v}[0], \ldots, \mathbf{v}[B-1]$ by time $t = T_{\text{eff}} - 1$. In this step, the decoder proceeds as follows. For each $j \in \{B, \ldots, T_{\text{eff}}-1\}$, the decoder recovers the parity-check packets $\mathbf{p}^v[j]$, by subtracting the unerased $\mathbf{u}[j - T_{\text{eff}}]$ from the associated $\mathbf{q}[j] = \mathbf{p}^v[j] + \mathbf{u}[j - T_{\text{eff}}]$ packets. These recovered parity-checks can then be used to recover $\mathbf{v}[0], \ldots, \mathbf{v}[B-1]$.

[8]Throughout the paper, we will use packets to denote a vector of source, parity and channel symbols, respectively, whereas we use sub-packets for $\mathbf{u}[\cdot]$ and $\mathbf{v}[\cdot]$.

TABLE II: An illustration of the decoding steps in a Generalized MS code. Each column denotes a channel packet transmitted at the time index shown in the first row. In each interval, the parity check packets used for recovery are highlighted in red, whereas the parities that are computed and cancelled are crossed out.

	0	$\dots$	$B-1$	B	$\dots$	$T_{\text{eff}}-1$	T_{eff}	$\dots$	$T_{\text{eff}}+B-1$
k_u	$\mathbf{u}[0]$	$\dots$	$\mathbf{u}[B-1]$	$\mathbf{u}[B]$	$\dots$	$\mathbf{u}[T_{\text{eff}}-1]$	$\mathbf{u}[T_{\text{eff}}]$	$\dots$	$\mathbf{u}[T_{\text{eff}}+B-1]$
k_v	$\mathbf{v}[0]$	$\dots$	$\mathbf{v}[B-1]$	$\mathbf{v}[B]$	$\dots$	$\mathbf{v}[T_{\text{eff}}-1]$	$\mathbf{v}[T_{\text{eff}}]$	$\dots$	$\mathbf{v}[T_{\text{eff}}+B-1]$
k_u	$\mathbf{u}[-T_{\text{eff}}]$ $+\mathbf{p}[0]$	$\dots$	$\mathbf{u}[B-T_{\text{eff}}-1]$ $+\mathbf{p}[B-1]$	~~$\mathbf{u}[B-T_{\text{eff}}]$~~ $+\mathbf{p}[B]$	$\dots$	~~$\mathbf{u}[-1]$~~$+$ $\mathbf{p}[T_{\text{eff}}-1]$	$\mathbf{u}[0]+$ ~~$\mathbf{p}[T_{\text{eff}}]$~~	$\dots$	$\mathbf{u}[B-1]+$ ~~$\mathbf{p}[T_{\text{eff}}+B-1]$~~
							$\Downarrow$		$\Downarrow$
	Burst Erasure			Simultaneously Recover $\mathbf{v}[0],\dots,\mathbf{v}[B-1]$			$\mathbf{u}[0]$	$\dots$	$\mathbf{u}[B-1]$ Sequentially

Note that using property P2 in Corollary 1 and substituting $R=R^v$ and $j=T_{\text{eff}}-1$ we get,

$$(1-R^v)T_{\text{eff}}=B, \tag{25}$$

and hence the recovery of $\mathbf{v}[0],\dots,\mathbf{v}[B-1]$ by time $t=T_{\text{eff}}-1$ is guaranteed.

- *Sequentially recover* $\mathbf{u}[0],\dots,\mathbf{u}[B-1]$ at times $T_{\text{eff}},\dots,T_{\text{eff}}+B-1$, respectively. Consider the parity-checks $\mathbf{q}[j]=\mathbf{u}[j-T_{\text{eff}}]+\mathbf{p}^v[j]$ for $j\in\{T_{\text{eff}},\dots,T_{\text{eff}}+B-1\}$, which are available to the decoder. Upon the recovery of $\mathbf{v}[0],\dots,\mathbf{v}[B-1]$ in the previous step, the required $\mathbf{p}^v[j]$ can be computed, subtracted from $\mathbf{q}[j]$, and the underlying $\mathbf{u}[\cdot]$ sub-packets can be sequentially recovered by their deadlines.

Upon completion of the two steps stated above, the recovery of $\mathbf{s}[i]$ for $i\in\{0,\dots,B-1\}$ follows. Any subsequent burst, starting at time $t\geq T_{\text{eff}}+B$, can be corrected in a similar fashion. Since the rate of the code is clearly given by (21), the proof of Prop. 2 is complete.

Remark 2. *The generalized MS code construction makes the structure of the optimal streaming code for the burst erasure channel more transparent. Note that m-MDS is an inter-packet code that combines the sub-packets, $\mathbf{v}[\cdot]$, across different time instants. Such a code can only simultaneously recover all the erased $\mathbf{v}[\cdot]$ sub-packets and does not provide the sequential recovery. The repetition code applied to $\mathbf{u}[\cdot]$ sub-packets is a simple intra-packet code. It does not combine packets across different time and can be sequentially recovered. The proposed construction splits each source packet into two parts, applies the inter-packet code to one group, the intra-packet repetition code to the other group, and then superimposes the resulting parity-check packets. Thus, the optimal code involves balancing the contributions of the inter-packet and intra-packet codes through an appropriate sub-packetization.*

The Generalized MS code is no longer feasible when $N>1$. To see this consider two isolated erasures one at $t=0$ and the other at $t=T_{\text{eff}}$. In this case, both $\mathbf{u}[0]$ as well as its repeated copy are erased. We propose a modification to these codes that can deal with any value of N in Theorem 2.

C. MiDAS: Code Construction

Our proposed construction is based on a layered approach. We first construct a Generalized MS code for $\mathcal{C}(N=1,B,W)$ channel and then concatenate an additional layer of parity packets when $N>1$. We again assume that $\mathbf{s}[i]\in\mathbb{F}_q^k$ and split it into two sub-packets $\mathbf{u}[i]$ and $\mathbf{v}[i]$ as in (22) and generate the parity-checks $\mathbf{q}[i]$ as in (24). The resulting code up to this point can only correct burst erasures. We further apply a $(k^u+k^s,k^u,T_{\text{eff}})$ m-MDS code of rate $R^u=\frac{k^u}{k^u+k^s}$ to the $\mathbf{u}[i]$ sub-packets and generate additional parity-check packets,

$$\mathbf{p}^u[i]=\left(\sum_{j=0}^{T_{\text{eff}}}\mathbf{u}^\dagger[i-j]\cdot\mathbf{H}_j^u\right)^\dagger,\qquad \mathbf{p}^u[i]\in\mathbb{F}_q^{k^s}, \tag{26}$$

where $\mathbf{H}_j^u\in\mathbb{F}_q^{k^u\times k^s}$ are matrices associated with a m-MDS code (12). We simply concatenate the parity-checks $\mathbf{q}[i]$ and $\mathbf{p}^u[i]$ with the source packets, i.e., $\mathbf{x}[i]=(\mathbf{u}[i],\mathbf{v}[i],\mathbf{q}[i],\mathbf{p}^u[i])$. Fig. 9 illustrates the layered approach in our code construction. Note that $\mathbf{x}[i]\in\mathbb{F}_q^n$, where $n=2k^u+k^v+k^s$ and the associated rate is given by $R=\frac{k}{n}$.

In our construction, we select $k^u=B$, $k^v=T_{\text{eff}}-B$, $k=k^u+k^v=T_{\text{eff}}$ and,

$$k^s=\frac{N}{T_{\text{eff}}-N+1}k^u. \tag{27}$$

Remark 3. *We note that if the value of k^s in (27) is non-integer, extra source splitting by a certain factor of m is needed. In particular, we set $k^u=mB$, $k^v=m(T_{\text{eff}}-B)$, $k=k^u+k^v=mT_{\text{eff}}$ and $k^s=\frac{N}{T_{\text{eff}}-N+1}k^u=\frac{N}{T_{\text{eff}}-N+1}mB$. It can be clearly seen that choosing $m=T_{\text{eff}}-N+1$ is*

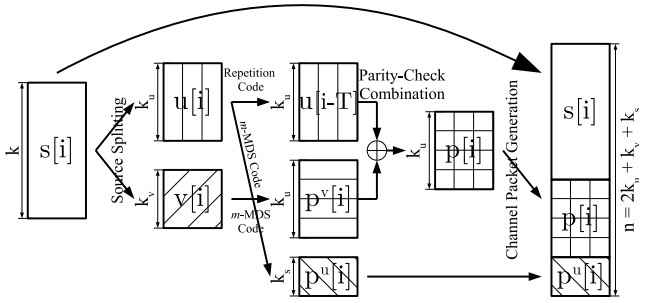


Fig. 9: A block diagram illustrating the encoding steps of a MiDAS code. The top part is equivalent to that of a Generalized MS code (cf. Fig. 8). The lower part shows the extra layer of parity-checks added which is generated by applying a m-MDS code to the $\mathbf{u}[\cdot]$ sub-packets.

sufficient for k^s to be an integer.

Decoder Analysis: In the analysis of the decoder, we consider the interval $[0, T_{\text{eff}}]$ and show that the decoder can recover $\mathbf{s}[0]$ by time $t = T_{\text{eff}}$ if there is either an erasure burst of length B or smaller, or up to N isolated erasures in this interval. Once we show the recovery of $\mathbf{s}[0]$ by time $t = T_{\text{eff}}$, we can cancel its effect from all future parity-check packets if necessary. The same argument can then be used to show that $\mathbf{s}[1]$ can be recovered by time $T_{\text{eff}} + 1$ if there are no more than N isolated erasures or a single burst erasure of maximum length B in the interval $[1, T_{\text{eff}} + 1]$. Recursively continuing this argument we are guaranteed the recovery of each $\mathbf{s}[i]$ by time $i + T_{\text{eff}}$.

If there is a burst of length B in the interval $[0, T_{\text{eff}}]$ our construction of $\mathbf{q}[\cdot]$ already guarantees the recovery of $\mathbf{s}[0]$ by time $t = T_{\text{eff}}$ (cf. Section IV-B). Thus, we only need to consider the case when there are N isolated erasures in the interval $[0, T_{\text{eff}}]$. We show that the decoder is guaranteed to recover $\mathbf{v}[0]$ at time $t = T_{\text{eff}} - 1$ using the parity-checks $\mathbf{q}[\cdot]$ and $\mathbf{u}[0]$ at time $t = T_{\text{eff}}$ using the parity-checks $\mathbf{p}^u[\cdot]$.

The recovery of $\mathbf{v}[0]$ by time $T_{\text{eff}} - 1$ follows in a fashion similar to the *simultaneous recovery* step above (25) in the previous section. However, we use P1 in corollary 1 instead. Recall from (24) that $\mathbf{q}[i] = \mathbf{p}^v[i] + \mathbf{u}[i - T_{\text{eff}}]$, where $\mathbf{p}^v[i]$ are the parity-checks of the m-MDS code (23). Since the interfering $\mathbf{u}[i-T_{\text{eff}}]$ sub-packets in the interval $i \in [0, T_{\text{eff}}-1]$ are not erased, they can be canceled out from $\mathbf{q}[i]$ and the corresponding parity-checks $\mathbf{p}^v[i]$ are recovered at the decoder. Since the code $(\mathbf{v}[i], \mathbf{p}^v[i])$ is a m-MDS code of rate $R^v = \frac{T_{\text{eff}}-B}{T_{\text{eff}}}$, applying property P1 in Corollary 1 the number of isolated erasures under which the recovery of $\mathbf{v}[0]$ is possible is given by $N^v = (1-R^v)T_{\text{eff}} = B$. Since $N \leq B$ holds, the recovery of $\mathbf{v}[0]$ by time $t = T_{\text{eff}} - 1$ is guaranteed by the code construction.

For recovering $\mathbf{u}[0]$ at time $t = T_{\text{eff}}$, we use the $\mathbf{p}^u[\cdot]$ parity-checks in the interval $[0, T_{\text{eff}}]$. Note that the associated code $(\mathbf{u}[i], \mathbf{p}^u[i])$ is a m-MDS code with rate $R^u = \frac{k^u}{k^u+k^s}$ and hence it follows from P1 in Corollary 1 that the number of isolated erasures under which the recovery of $\mathbf{u}[0]$ is possible is given by

$$(1 - R^u)(T_{\text{eff}} + 1) = \frac{k^s}{k^s + k^u}(T_{\text{eff}} + 1) = N, \tag{28}$$

where we substitute (27) in the last equality. This completes the proof that $\mathbf{s}[0] = (\mathbf{u}[0], \mathbf{v}[0])$ can be recovered at time $t = T_{\text{eff}}$ when there are N isolated erasures in the interval $[0, T_{\text{eff}}]$. ■

It remains to show that our proposed code parameter satisfy the lower bound in Theorem 2.

$$R = \frac{k^u + k^v}{2k^u + k^v + k^s} = \frac{T_{\text{eff}}}{T_{\text{eff}} + B + B\frac{N}{T_{\text{eff}}-N+1}} \tag{29}$$

$$> \frac{T_{\text{eff}}}{T_{\text{eff}} + B + B\frac{N}{T_{\text{eff}}-N}}$$

$$= \frac{T_{\text{eff}} - N}{T_{\text{eff}} - N + B} \tag{30}$$

where (29) follows by substituting in (27). Rearranging (30) we have that

$$\frac{R}{1-R}B + N > T_{\text{eff}}. \tag{31}$$

The proof of Theorem 2 is thus completed.

Example - MiDAS $(N, B, T) = (2, 3, 4)$ and $W \geq T + 1 = 5$: Table III illustrates a MiDAS construction for $(N, B) = (2, 3)$ and $T = 4$ and $T_{\text{eff}} = T$. The encoding steps are as follows:

- Split each source packet $\mathbf{s}[i]$ into $k = T = 4$ symbols. The first $k^u = B = 3$ symbols are $\mathbf{u}[i] = (u_0[i], u_1[i], u_2[i])$ while the last $k^v = T-B = 1$ symbol is $v_0[i]$.
- Apply a $(k^u + k^v, k^v, T) = (4, 1, 4)$ m-MDS code of rate $R^v = \frac{1}{4}$ to the $\mathbf{v}[\cdot]$ sub-packets generating the parity-check packets,

$$\mathbf{p}^v[i] = (p_0^v[i], p_1^v[i], p_2^v[i]) = \sum_{j=0}^{4} v_0[i-j]\mathbf{H}_j^v. \tag{32}$$

- Combine the $\mathbf{u}[\cdot]$ with $\mathbf{p}^v[\cdot]$ and generate $\mathbf{q}[i] = \mathbf{p}^v[i] + \mathbf{u}[i - T]$.
- Apply a $(k^u+k^s, k^u, T) = (5, 3, 4)$ m-MDS code of rate $R^u = \frac{3}{5}$ to the $\mathbf{u}[\cdot]$ sub-packets generating parity-check packets each with $k^s = \frac{N}{T-N+1}k^u = 2$ symbols,

$$\mathbf{p}^u[i] = (p_0^u[i], p_1^u[i])$$
$$= \sum_{j=0}^{4} \begin{bmatrix} u_0[i-j] & u_1[i-j] & u_2[i-j] \end{bmatrix} \mathbf{H}_j^u. \tag{33}$$

.

The channel packet at time i is given by,

$$\mathbf{x}[i] = (\mathbf{u}[i], \mathbf{v}[i], \mathbf{q}[i], \mathbf{p}^u[i]), \tag{34}$$

whose rate is $R = \frac{k^u+k^v}{2k^u+k^v+k^s} = \frac{T}{T+B+\frac{NB}{T-N+1}} = \frac{4}{9}$.

For decoding, first assume that an erasure burst spans the interval $[i, i+2]$. We first recover $p_0^v[i+3], p_1^v[i+3], p_2^v[i+3]$ by subtracting $u_0[i-1], u_1[i-1], u_2[i-1]$ from the parity-check symbols $q_0[i+3], q_1[i+3], q_2[i+3]$ respectively. In the interval $[i, i+T-1] = [i, i+3]$, the channel introduces a burst of length 3. Thus, the $(4, 1, 4)$ m-MDS code suffices for recovering the three erased packets $v_0[i]$, $v_0[i+1]$ and $v_0[i+2]$ by time $i+3$ since $(1 - R^v)T = 3$. Once all the erased $\mathbf{v}[t]$ are recovered, we can compute the parity-check packets $\mathbf{p}^v[t]$ for $t \in \{i+4, i+5, i+6\}$ and subtract them from the corresponding $\mathbf{q}[t]$ to recover $\mathbf{u}[i], \mathbf{u}[i+1], \mathbf{u}[i+2]$ at time $i+4, i+5, i+6$ respectively, i.e., within a delay of $T = 4$.

In the case of isolated erasures, we consider a channel introducing $N = 2$ isolated erasures in the interval $[i, i+4]$ of length $T + 1 = 5$. We first recover the unerased parity-check packets $\mathbf{p}^v[\cdot]$ in the interval $[i, i+3]$ by subtracting the corresponding $\mathbf{u}[\cdot]$ sub-packets. The $(4, 1, 4)$ suffices for recovering $v_0[i]$ by time $i+T-1 = i+3$ since $(1-R^v)T = 3 > 2 = N$. Also, $\mathbf{u}[0]$ can be recovered by time $i+4$ using the $(5, 3, 4)$ m-MDS code in the interval $[i, i+4]$ since $(1 - R^u)(T + 1) = 2 = N$.

TABLE III: MiDAS code construction for $(N, B) = (2, 3)$, a delay of $T = 4$ and rate $R = 4/9$.

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$
$k^u = 3$	$u_0[i]$ $u_1[i]$ $u_2[i]$	$u_0[i+1]$ $u_1[i+1]$ $u_2[i+1]$	$u_0[i+2]$ $u_1[i+2]$ $u_2[i+2]$	$u_0[i+3]$ $u_1[i+3]$ $u_2[i+3]$	$u_0[i+4]$ $u_1[i+4]$ $u_2[i+4]$
$k^v = 1$	$v_0[i]$	$v_0[i+1]$	$v_0[i+2]$	$v_0[i+3]$	$v_0[i+4]$
$k^u = 3$	$u_0[i-4] + p_0^v[i]$ $u_1[i-4] + p_1^v[i]$ $u_2[i-4] + p_2^v[i]$	$u_0[i-3] + p_0^v[i+1]$ $u_1[i-3] + p_1^v[i+1]$ $u_2[i-1] + p_2^v[i+1]$	$u_0[i-2] + p_0^v[i+2]$ $u_1[i-2] + p_1^v[i+2]$ $u_2[i-2] + p_2^v[i+2]$	$u_0[i-1] + p_0^v[i+3]$ $u_1[i-1] + p_1^v[i+3]$ $u_2[i-1] + p_2^v[i+3]$	$u_0[i] + p_0^v[i+4]$ $u_1[i] + p_1^v[i+4]$ $u_2[i] + p_2^v[i+4]$
$k^s = 2$	$p_0^u[i]$ $p_1^u[i]$	$p_0^u[i+1]$ $p_1^u[i+1]$	$p_0^u[i+2]$ $p_1^u[i+2]$	$p_0^u[i+3]$ $p_1^u[i+3]$	$p_0^u[i+4]$ $p_1^u[i+4]$

D. MiDAS Codes with Improved Field-Size

Our constructions in Section IV-C are based on m-MDS codes [4], [5]. Such codes are guaranteed to exist only when the underlying field-sizes are very large. In particular, the field-size must increase exponentially in T_{eff} except in some special cases [5]. In this section, we suggest an alternative construction that uses block-MDS codes instead of m-MDS codes. This construction requires a field-size that only increases as $\mathcal{O}(T_{\text{eff}}^3)$. While this alternate construction also attains the tradeoff in Theorem 2, it does come at a price. It incurs some performance loss in simulations and is less robust to non-ideal erasure patterns as discussed in Section IV-E.

Proposition 3. *For the channel $\mathcal{C}(N, B, W)$ and delay T, there exists a streaming code of rate R that satisfies* (2) *in Theorem 2 with a field-size that increases as $\mathcal{O}(T_{\text{eff}}^3)$.* □

We start by giving two examples and then discuss the general code construction. The key step is to replace the m-MDS code in (23) and (26) by two block MDS codes applied diagonally to the $\mathbf{v}[\cdot]$ and $\mathbf{u}[\cdot]$ sub-packets.

1) Example - MiDAS $(N, B, T) = (2, 3, 4)$ and $W \geq T + 1 = 5$: Table IV illustrates a MiDAS construction using MDS as constituent codes. The rate of this code is $R = \frac{T}{T+B+\frac{NB}{T-N+1}} = \frac{4}{9}$ from (29). Note that this code has the same parameters as in Table III in Section IV-C. The encoding steps, stated below, are also similar except that the m-MDS codes are replaced with block MDS codes.

- Split each source packet $\mathbf{s}[i]$ into $k = T = 4$ symbols. The first $k^u = B = 3$ symbols are $\mathbf{u}[i] = (u_0[i], u_1[i], u_2[i])$, while the last $k^v = T - B = 1$ symbol is $v_0[i]$.
- Apply a $(T, T-B) = (4, 1)$ MDS code[9] to the $\mathbf{v}[\cdot]$ sub-packets generating parity-check packets $\mathbf{p}^v[\cdot]$ each with $B = 3$ symbols, i.e., $\mathbf{p}^v[i] = (p_0^v[i], p_1^v[i], p_2^v[i])$. Hence, at time i, the generated codeword is,

$$\mathbf{c}^v[i] = (v_0[i], p_0^v[i+1], p_1^v[i+2], p_2^v[i+3]) \tag{35}$$

  and is shown using the shaded boxes in Table IV.
- Combine $\mathbf{u}[\cdot]$ with $\mathbf{p}^v[\cdot]$ packets and generate $\mathbf{q}[t] = \mathbf{p}^v[t] + \mathbf{u}[t-T]$.
- Apply a $(T+1, T-N+1) = (5, 3)$ MDS code diagonally to the $\mathbf{u}[\cdot]$ sub-packets generating $N = 2$ parity-check symbols $\mathbf{p}^u[i] = (p_0^u[i], p_1^u[i])$. The codeword starting at time i is given by,

$$\mathbf{c}^u[i] = (u_0[i], u_1[i+1], u_2[i+2], p_0^u[i+3], p_1^u[i+4]) \tag{36}$$

  and is marked by the unshaded boxes in Table IV for convenience.

The channel packet at time i is given by,

$$\mathbf{x}[i] = (\mathbf{u}[i], \mathbf{v}[i], \mathbf{q}[i], \mathbf{p}^u[i]), \tag{37}$$

whose rate is $R = \frac{3+1}{3+1+3+2} = \frac{4}{9}$ which is consistent with (29).

For decoding, first assume that an erasure burst spans the interval $[i, i+2]$. We first recover $p_0^v[i+3], p_1^v[i+3], p_2^v[i+3]$ at time $t = i+3$ from the parity-check packets $q_0[i+3], q_1[i+3], q_2[i+3]$. We can use the underlying MDS codes to recover $v_0[i], v_1[i+1], v_2[i+2]$ at time $t = i+3$ by considering $\mathbf{c}^v[i], \mathbf{c}^v[i+1], \mathbf{c}^v[i+2]$ respectively (see (35)). Once all the erased $\mathbf{v}[t]$ are recovered, we recover $\mathbf{u}[i]$ at time $t = i+4$, $\mathbf{u}[i+1]$ at time $t = i+5$ and $\mathbf{u}[i+2]$ at time $t = i+6$.

In the case of isolated erasures, we assume a channel introducing $N = 2$ isolated erasures in the interval $[0, 4]$ of length $T + 1 = 5$. Note that the codeword $\mathbf{c}^v[i]$ in (35) terminates at time $t = i+3$. Thus, there are no more than $N = 2$ erasures on it and thus the recovery of $v_0[i]$ is guaranteed at time $t = i+3$. Likewise the codewords $\mathbf{c}^u[i-2], \mathbf{c}^u[i-1], \mathbf{c}^u[i]$ in (36) combining $u_2[i], u_1[i], u_0[i]$, respectively, terminate at time $t = i+4$ and there are no more than $N = 2$ erasures on any of them. Thus, the recovery of $u_j[i]$ for $j = 0, 1, 2$ is guaranteed at time $t = i+4$.

However, splitting each source packet into $k = T$ symbols is not enough in general. In particular, applying a $(T, T-B)$ MDS code to the $\mathbf{v}[\cdot]$ sub-packets requires that the $\mathbf{v}[\cdot]$ sub-packets are split into a multiple of $T - B$ symbols. Similarly, applying a $(T+1, T-N+1)$ MDS code to the $\mathbf{u}[\cdot]$ sub-packets requires splitting them into a multiple of $T - N + 1$ symbols. On the other hand, achieving the tradeoff in (2) requires that the ratio between the size of $\mathbf{u}[\cdot]$ to $\mathbf{v}[\cdot]$ to be $\frac{B}{T-B}$. Thus, splitting the $\mathbf{u}[\cdot]$ sub-packets to $B(T-N+1)$ symbols and splitting the $\mathbf{v}[\cdot]$ sub-packets into $(T-N+1)(T-B)$ symbols fulfills all the previous constraints. The following example illustrates this case.

2) Example - MiDAS $(N, B, T) = (2, 3, 5)$ and $W \geq T + 1 = 6$: Table V illustrates a MiDAS construction using MDS as constituent codes. The rate of this code is $R = \frac{T}{T+B+\frac{NB}{T-N+1}} = \frac{10}{19}$. The encoding steps are as follows.

[9]This can be a simple repetition code, i.e., $p_0^v[i+1] = p_1^v[i+2] = p_2^v[i+3] = v_0[i]$.

TABLE IV: MiDAS code construction for $(N, B) = (2, 3)$, a delay of $T = 4$ and rate $R = 4/9$ with a block MDS constituent code.

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$
$k^u = 3$	$u_0[i]$	$u_0[i+1]$	$u_0[i+2]$	$u_0[i+3]$	$u_0[i+4]$
	$u_1[i]$	$u_1[i+1]$	$u_1[i+2]$	$u_1[i+3]$	$u_1[i+4]$
	$u_2[i]$	$u_2[i+1]$	$u_2[i+2]$	$u_2[i+3]$	$u_2[i+4]$
$k^v = 1$	$v_0[i]$	$v_0[i+1]$	$v_0[i+2]$	$v_0[i+3]$	$v_0[i+4]$
$k^u = 3$	$u_0[i-4] + p_0^v[i]$	$u_0[i-3] + p_0^v[i+1]$	$u_0[i-2] + p_0^v[i+2]$	$u_0[i-1] + p_0^v[i+3]$	$u_0[i] + p_0^v[i+4]$
	$u_1[i-4] + p_1^v[i]$	$u_1[i-3] + p_1^v[i+1]$	$u_1[i-2] + p_1^v[i+2]$	$u_1[i-1] + p_1^v[i+3]$	$u_1[i] + p_1^v[i+4]$
	$u_2[i-4] + p_2^v[i]$	$u_2[i-1] + p_2^v[i+1]$	$u_2[i-2] + p_2^v[i+2]$	$u_2[i-1] + p_2^v[i+3]$	$u_2[i] + p_2^v[i+4]$
$k^s = 2$	$p_0^u[i]$	$p_0^u[i+1]$	$p_0^u[i+2]$	$p_0^u[i+3]$	$p_0^u[i+4]$
	$p_1^u[i]$	$p_1^u[i+1]$	$p_1^u[i+2]$	$p_1^u[i+3]$	$p_1^u[i+4]$

TABLE V: MiDAS code construction for $(N, B) = (2, 3)$, a delay of $T = 5$ and rate $R = 10/19$ with a block MDS constituent code. We note that each of the parity-check sub-symbols $p_j^v[t]$ is combined with $u_j[t-5]$ for $j = \{0, 1, \ldots, 11\}$ but the latter are omitted in the above table for simplicity.

	$[i]$	$[i+1]$	$[i+2]$	$[i+3]$	$[i+4]$	$[i+5]$
$k^u = 12$	$u_0[i]$	$u_0[i+1]$	$u_0[i+2]$	$u_0[i+3]$	$u_0[i+4]$	$u_0[i+5]$
	$u_1[i]$	$u_1[i+1]$	$u_1[i+2]$	$u_1[i+3]$	$u_1[i+4]$	$u_1[i+5]$
	$u_2[i]$	$u_2[i+1]$	$u_2[i+2]$	$u_2[i+3]$	$u_2[i+4]$	$u_2[i+5]$
	$u_3[i]$	$u_3[i+1]$	$u_3[i+2]$	$u_3[i+3]$	$u_3[i+4]$	$u_3[i+5]$
	$u_4[i]$	$u_4[i+1]$	$u_4[i+2]$	$u_4[i+3]$	$u_4[i+4]$	$u_4[i+5]$
	$u_5[i]$	$u_5[i+1]$	$u_5[i+2]$	$u_5[i+3]$	$u_5[i+4]$	$u_5[i+5]$
	$u_6[i]$	$u_6[i+1]$	$u_6[i+2]$	$u_6[i+3]$	$u_6[i+4]$	$u_6[i+5]$
	$u_7[i]$	$u_7[i+1]$	$u_7[i+2]$	$u_7[i+3]$	$u_7[i+4]$	$u_7[i+5]$
	$u_8[i]$	$u_8[i+1]$	$u_8[i+2]$	$u_8[i+3]$	$u_8[i+4]$	$u_8[i+5]$
	$u_9[i]$	$u_9[i+1]$	$u_9[i+2]$	$u_9[i+3]$	$u_9[i+4]$	$u_9[i+5]$
	$u_{10}[i]$	$u_{10}[i+1]$	$u_{10}[i+2]$	$u_{10}[i+3]$	$u_{10}[i+4]$	$u_{10}[i+5]$
	$u_{11}[i]$	$u_{11}[i+1]$	$u_{11}[i+2]$	$u_{11}[i+3]$	$u_{11}[i+4]$	$u_{11}[i+5]$
$k^v = 8$	$v_0[i]$	$v_0[i+1]$	$v_0[i+2]$	$v_0[i+3]$	$v_0[i+4]$	$v_0[i+5]$
	$v_1[i]$	$v_1[i+1]$	$v_1[i+2]$	$v_1[i+3]$	$v_1[i+4]$	$v_1[i+5]$
	$v_2[i]$	$v_2[i+1]$	$v_2[i+2]$	$v_2[i+3]$	$v_2[i+4]$	$v_2[i+5]$
	$v_3[i]$	$v_3[i+1]$	$v_3[i+2]$	$v_3[i+3]$	$v_3[i+4]$	$v_3[i+5]$
	$v_4[i]$	$v_4[i+1]$	$v_4[i+2]$	$v_4[i+3]$	$v_4[i+4]$	$v_4[i+5]$
	$v_5[i]$	$v_5[i+1]$	$v_5[i+2]$	$v_5[i+3]$	$v_5[i+4]$	$v_5[i+5]$
	$v_6[i]$	$v_6[i+1]$	$v_6[i+2]$	$v_6[i+3]$	$v_6[i+4]$	$v_6[i+5]$
	$v_7[i]$	$v_7[i+1]$	$v_7[i+2]$	$v_7[i+3]$	$v_7[i+4]$	$v_7[i+5]$
$k^u = 12$	$p_0^v[i]$	$p_0^v[i+1]$	$p_0^v[i+2]$	$p_0^v[i+3]$	$p_0^v[i+4]$	$p_0^v[i+5]$
	$p_1^v[i]$	$p_1^v[i+1]$	$p_1^v[i+2]$	$p_1^v[i+3]$	$p_1^v[i+4]$	$p_1^v[i+5]$
	$p_2^v[i]$	$p_2^v[i+1]$	$p_2^v[i+2]$	$p_2^v[i+3]$	$p_2^v[i+4]$	$p_2^v[i+5]$
	$p_3^v[i]$	$p_3^v[i+1]$	$p_3^v[i+2]$	$p_3^v[i+3]$	$p_3^v[i+4]$	$p_3^v[i+5]$
	$p_4^v[i]$	$p_4^v[i+1]$	$p_4^v[i+2]$	$p_4^v[i+3]$	$p_4^v[i+4]$	$p_4^v[i+5]$
	$p_5^v[i]$	$p_5^v[i+1]$	$p_5^v[i+2]$	$p_5^v[i+3]$	$p_5^v[i+4]$	$p_5^v[i+5]$
	$p_6^v[i]$	$p_6^v[i+1]$	$p_6^v[i+2]$	$p_6^v[i+3]$	$p_6^v[i+4]$	$p_6^v[i+5]$
	$p_7^v[i]$	$p_7^v[i+1]$	$p_7^v[i+2]$	$p_7^v[i+3]$	$p_7^v[i+4]$	$p_7^v[i+5]$
	$p_8^v[i]$	$p_8^v[i+1]$	$p_8^v[i+2]$	$p_8^v[i+3]$	$p_8^v[i+4]$	$p_8^v[i+5]$
	$p_9^v[i]$	$p_9^v[i+1]$	$p_9^v[i+2]$	$p_9^v[i+3]$	$p_9^v[i+4]$	$p_9^v[i+5]$
	$p_{10}^v[i]$	$p_{10}^v[i+1]$	$p_{10}^v[i+2]$	$p_{10}^v[i+3]$	$p_{10}^v[i+4]$	$p_{10}^v[i+5]$
	$p_{11}^v[i]$	$p_{11}^v[i+1]$	$p_{11}^v[i+2]$	$p_{11}^v[i+3]$	$p_{11}^v[i+4]$	$p_{11}^v[i+5]$
$k^s = 6$	$p_0^u[i]$	$p_0^u[i+1]$	$p_0^u[i+2]$	$p_0^u[i+3]$	$p_0^u[i+4]$	$p_0^u[i+5]$
	$p_1^u[i]$	$p_1^u[i+1]$	$p_1^u[i+2]$	$p_1^u[i+3]$	$p_1^u[i+4]$	$p_1^u[i+5]$
	$p_2^u[i]$	$p_2^u[i+1]$	$p_2^u[i+2]$	$p_2^u[i+3]$	$p_2^u[i+4]$	$p_2^u[i+5]$
	$p_3^u[i]$	$p_3^u[i+1]$	$p_3^u[i+2]$	$p_3^u[i+3]$	$p_3^u[i+4]$	$p_3^u[i+5]$
	$p_4^u[i]$	$p_4^u[i+1]$	$p_4^u[i+2]$	$p_4^u[i+3]$	$p_4^u[i+4]$	$p_4^u[i+5]$
	$p_5^u[i]$	$p_5^u[i+1]$	$p_5^u[i+2]$	$p_5^u[i+3]$	$p_5^u[i+4]$	$p_5^u[i+5]$

- Split each source packet $\mathbf{s}[i]$ into $k=(T-N+1)T=20$ symbols. The first $k^u=(T-N+1)B=12$ of which are $(u_0[i],\ldots,u_{11}[i])$ while the last $k^v=(T-N+1)(T-B)=8$ are $(v_0[i],\ldots,v_7[i])$.
- Apply a $(T,T-B)=(5,2)$ MDS code diagonally to the symbols in the $\mathbf{v}[\cdot]$ sub-packets with an interleaving factor of $T-N+1=4$. Hence, at time i, four codewords are generated as follows,

$$\begin{aligned}
\mathbf{c}_0^v[i] &= (v_0[i], v_4[i+1], p_0^v[i+2], p_4^v[i+3], p_8^v[i+4])\\
\mathbf{c}_1^v[i] &= (v_1[i], v_5[i+1], p_1^v[i+2], p_5^v[i+3], p_9^v[i+4])\\
\mathbf{c}_2^v[i] &= (v_2[i], v_6[i+1], p_2^v[i+2], p_6^v[i+3], p_{10}^v[i+4])\\
\mathbf{c}_3^v[i] &= (v_3[i], v_7[i+1], p_3^v[i+2], p_7^v[i+3], p_{11}^v[i+4])
\end{aligned} \tag{38}$$

  The codeword $\mathbf{c}_0^v[i]$ is shown using the shaded boxes in Table V. According to (38), $(T-N+1)B=12$ parity-check symbols are generated, namely $(p_0^v[i],\ldots,p_{11}^v[i])$.
- Combine the $\mathbf{u}[\cdot]$ sub-packets with $\mathbf{p}^v[\cdot]$ packets and generate $\mathbf{q}[t]=\mathbf{p}^v[t]+\mathbf{u}[t-T]$. For simplicity we do not show these in Table V.
- Apply a $(T+1,T-N+1)=(6,4)$ MDS code to the u packets with an interleaving factor of $B=3$ generating $BN=6$ parity-check symbols $(p_0^u[i],\ldots,p_5^u[i])$. The resulting codewords are as follows,

$$\begin{aligned}
\mathbf{c}_0^u[i] &= (u_0[i], u_3[i+1], u_6[i+2], u_9[i+3], p_0^u[i+4],\\
&\qquad p_3^u[i+5])\\
\mathbf{c}_1^u[i] &= (u_1[i], u_4[i+1], u_7[i+2], u_{10}[i+3], p_1^u[i+4],\\
&\qquad p_4^u[i+5])\\
\mathbf{c}_2^u[i] &= (u_2[i], u_5[i+1], u_8[i+2], u_{11}[i+3], p_2^u[i+4],\\
&\qquad p_5^u[i+5])
\end{aligned} \tag{39}$$

  The codeword $\mathbf{c}_0^u[i]$ is marked by the unshaded boxes in Table V for convenience.

The channel packet at time i is given by,

$$\mathbf{x}[i]=(\mathbf{u}[i],\mathbf{v}[i],\mathbf{q}[i],\mathbf{p}^u[i])\,, \tag{40}$$

whose rate is $R=\frac{12+8}{12+8+12+6}=\frac{10}{19}$.

For decoding, first assume that an erasure burst spans the interval $[i,i+2]$. The decoding steps are as follows,

- Recover $\mathbf{p}^v[t]=(p_0^v[t],\ldots,p_{11}^v[t])$ for $t=\{i+3,i+4\}$ by subtracting $\mathbf{u}[t-5]$ from $\mathbf{q}[t]$.
- Recover $\mathbf{v}[i]$, $\mathbf{v}[i+1]$ and $\mathbf{v}[i+2]$ using the underlying $(5,2)$ MDS codes as follows. For $j\in\{0,\ldots,3\}$,
  - $\mathbf{c}_j^v[i-1]=(v_j[i-1], v_{j+4}[i], p_j^v[i+1], p_{j+4}^v[i+2], p_{j+8}^v[i+3])$ has 3 erasures at i, $i+1$ and $i+2$. Hence, the $v_{j+4}[i]$ symbols are recovered by time $i+3$.
  - $\mathbf{c}_j^v[i]=(v_j[i], v_{j+4}[i+1], p_j^v[i+2], p_{j+4}^v[i+3], p_{j+8}^v[i+4])$ has 3 erasures at i, $i+1$ and $i+2$. Hence, the $v_j[i]$ and $v_{j+4}[i+1]$ symbols are recovered by time $i+4$.
  - $\mathbf{c}_j^v[i+1]=(v_j[i+1], v_{j+4}[i+2], p_j^v[i+3], p_{j+4}^v[i+4], p_{j+8}^v[i+5])$ has 3 erasures at $i+1$, $i+2$ and $i+5$[10]. Hence, the $v_j[i+1]$ and $v_{j+4}[i+2]$ symbols are recovered by time $i+4$.
  - $\mathbf{c}_j^v[i+2]=(v_j[i+2], v_{j+4}[i+3], p_j^v[i+4], p_{j+4}^v[i+5], p_{j+8}^v[i+6])$ has 3 erasures at $i+2$, $i+5$ and $i+6$. Hence, the $v_j[i+2]$ symbols are recovered by time $i+4$.

  In other words, all the erased $\mathbf{v}[\cdot]$ sub-packets are recovered by time $i+4$.
- Compute the parity-check packets $\mathbf{p}^v[t]$ for $t\in\{i+5,i+6,i+7\}$ as they only combine $\mathbf{v}[\cdot]$ sub-packets that are either unerased or recovered in the previous step. These parity-check packets can be subtracted from the corresponding $\mathbf{p}[t]$ packets to recover $\mathbf{u}[i-T]$ sub-packets within a delay of $T=5$. In other words, we recover $\mathbf{u}[i]$ at time $t=i+5$, $\mathbf{u}[i+1]$ at time $t=i+6$ and $\mathbf{u}[i+2]$ at time $t=i+7$.

In the case of isolated erasures, we assume a channel introducing $N=2$ isolated erasures in a the interval $[0,5]$ of length $T+1=6$. Note that the codewords $\mathbf{c}_j^v[i]$ in (38) terminate at time $t=i+4$. Thus, there are no more than $N=2$ erasures on either of them and thus the recovery of $v_j[i]$ is guaranteed at time $i+4$. Likewise the codewords $\mathbf{c}_j^u[i]$ in (39) terminate at time $t=i+5$ and there are no more than $N=2$ erasures on any of them. Thus, the recovery of $u_j[i]$ is guaranteed at time $t=i+5$.

3) Code Construction: The general construction achieving Prop. 3 is as follows.

- **Source Splitting:** We assume that each source packet $\mathbf{s}[i]\in\mathbb{F}_q^k$ and partition the k symbols into two sub-packets $\mathbf{u}_{\text{vec}}[i]\in\mathbb{F}_q^{k^u}$ and $\mathbf{v}_{\text{vec}}[i]\in\mathbb{F}_q^{k^v}$ as follows,

$$\begin{aligned}
\mathbf{s}[i] &= (s_0[i],\ldots,s_{k-1}[i])\\
&= (\underbrace{u_0[i],\ldots,u_{k^u-1}[i]}_{\mathbf{u}_{\text{vec}}[i]}, \underbrace{v_0[i],\ldots,v_{k^v-1}[i]}_{\mathbf{v}_{\text{vec}}[i]})
\end{aligned} \tag{41}$$

  where we select

$$\begin{aligned}
k^u &= (T_{\text{eff}}-N+1)B,\\
k^v &= (T_{\text{eff}}-N+1)(T_{\text{eff}}-B).
\end{aligned} \tag{42}$$

- **MDS Parity-Checks for $\mathbf{v}[\cdot]$ sub-packets:** Construct $T_{\text{eff}}-N+1$ systematic MDS codes of parameters $(T_{\text{eff}},T_{\text{eff}}-B)$ starting at time i whose associated codewords are,

$$\mathbf{c}_j^v[i]=\begin{bmatrix}
v_j[i]\\
v_{j+(T_{\text{eff}}-N+1)}[i+1]\\
v_{j+2(T_{\text{eff}}-N+1)}[i+2]\\
\vdots\\
v_{j+(T_{\text{eff}}-N+1)(T_{\text{eff}}-B-1)}[i+T_{\text{eff}}-B-1]\\
p_j^v[i+T_{\text{eff}}-B]\\
p_{j+(T_{\text{eff}}-N+1)}^v[i+T_{\text{eff}}-B+1]\\
\vdots\\
p_{j+(T_{\text{eff}}-N+1)(B-1)}^v[i+T_{\text{eff}}-1]
\end{bmatrix}, \tag{43}$$

[10]We note that the parity-check symbols $p_{j+8}^v[i+5]$ for $j\in\{0,\ldots,3\}$ are counted as erasures since they are combined with $u_{j+8}[i]$ which are erased.

for $j \in \{0, 1, \ldots, T_{\text{eff}} - N\}$. Notice that each codeword $\mathbf{c}_j^v[i]$ spans the interval $[i, i + T_{\text{eff}} - 1]$ and the adjacent symbols have an interleaving factor of $T_{\text{eff}} - N + 1$. The resulting parity-check packets at time i are expressed as: $\mathbf{p}^v[i] = (p_0^v[i], \ldots, p_{(T_{\text{eff}}-N+1)B-1}^v[i])$

- **Repetition of $\mathbf{u}[\cdot]$ sub-packets:** Combine the $\mathbf{u}[\cdot]$ sub-packets with the parity-check packets $\mathbf{p}^v[\cdot]$ after applying a shift of T_{eff}, i.e., $\mathbf{q}[i] = \mathbf{p}^v[i] + \mathbf{u}[i - T_{\text{eff}}]$.
- **MDS Parity-Checks for $\mathbf{u}[\cdot]$ sub-packets:** Construct B systematic MDS codes of parameters $(T_{\text{eff}}+1, T_{\text{eff}}-N+1)$ at time i whose associated codewords are,

$$\mathbf{c}_j^u[i] = \begin{bmatrix} u_j[i] \\ u_{j+B}[i+1] \\ u_{j+2B}[i+2] \\ \vdots \\ u_{j+B(T_{\text{eff}}-N)}[i+T_{\text{eff}}-N] \\ p_j^u[i+T_{\text{eff}}-N+1] \\ p_{j+B}^u[i+T_{\text{eff}}-N+2] \\ \vdots \\ p_{j+B(N-1)}^u[i+T_{\text{eff}}] \end{bmatrix}, \tag{44}$$

for $j \in \{0, 1, \ldots, B-1\}$. Notice that each codeword $\mathbf{c}_j^u[i]$ spans the interval $[i, i + T_{\text{eff}}]$ and consists of symbols with an interleaving factor of B. The resulting parity-check packets at time i are denoted by $\mathbf{p}^u[i] = (p_0^u[i], \ldots, p_{BN-1}^u[i])$.

- **Concatenation of Parity-Checks:** Concatenate the parity-check packets $\mathbf{p}^u[\cdot]$ and $\mathbf{q}[\cdot]$, i.e., the channel input at time i is given by,

$$\mathbf{x}[i] = (\mathbf{u}[i], \mathbf{v}[i], \mathbf{q}[i], \mathbf{p}^u[i]). \tag{45}$$

Note that the rate of the code equals

$$\begin{aligned} R &= \frac{(T_{\text{eff}} - N + 1)T_{\text{eff}}}{(T_{\text{eff}} - N + 1)T_{\text{eff}} + B(T_{\text{eff}} + 1)} \\ &= \frac{T_{\text{eff}}}{T_{\text{eff}} + \frac{B(T_{\text{eff}}+1)}{T_{\text{eff}}-N+1}} \end{aligned} \tag{46}$$

which is identical to the expression in (29).

The decoding steps are similar to that discussed in the previous examples and is provided in Appendix C.

4) Field-Size Computation: To compute the required field-size, note that splitting each source packet into $(T_{\text{eff}} - N + 1)T_{\text{eff}}$ symbols requires that each source packet consist of $q_1 = (T_{\text{eff}} - N + 1)T_{\text{eff}}$ symbols. We therefore need to determine the field-size of each symbol. Using the well-known fact that an (n, k) MDS code exists for any field-size greater than n, we note that the field-size needed for both $(T_{\text{eff}}, T_{\text{eff}} - B)$ and $(T_{\text{eff}} + 1, T_{\text{eff}} - N + 1)$ MDS codes to simultaneously exist is $q_2 = \Theta(T_{\text{eff}})$. Thus, a field-size of $q = q_1 \cdot q_2$ which is of the order $\mathcal{O}(T_{\text{eff}}^3)$ is sufficient.

E. Non-Ideal Erasure Patterns

Even though the construction in Section IV-D attains the same optimal tradeoff over the deterministic erasure channel model with a smaller field-size, their performance is more sensitive compared to the construction in Section IV-C when non-ideal erasure patterns are considered. To illustrate this we focus on the case when $N = 2$, $B = 3$, $T = 5$ and $W \geq 6$ in our discussion. The MiDAS construction with block MDS constituent code for these parameters is illustrated in Table V. The MiDAS codes using m-MDS codes has a similar structure except that the parity-checks $p_j^v[\cdot]$ and $p_j^u[\cdot]$ are generated using the m-MDS code.

We consider an erasure pattern that introduces a burst of length 2 in the interval $[i, i + 1]$ and an additional isolated erasure at time $i+3$. Clearly such a pattern violates a $\mathcal{C}(N = 2, B = 3, W = 6)$. Nonetheless, we argue that the MiDAS codes are able to completely recover from this erasure pattern but the alternative construction using block MDS codes in Table V cannot.

In particular, note that the the parity packets $\mathbf{p}[i + 2]$ and $\mathbf{p}[i + 4]$ contribute a total of 24 symbols which suffice to recover $\mathbf{v}[i], \mathbf{v}[i + 1]$ and $\mathbf{v}[i + 3]$, each of which involves 8 symbols. Thus, by time $i + 4$ all the symbols in the erased $\mathbf{v}[\cdot]$ sub-packets are recovered and we can proceed to recover $\mathbf{u}[i], \mathbf{u}[i + 1]$ and $\mathbf{u}[i + 3]$ at time $i + 5, i + 6$ and $i + 8$, respectively, i.e., a delay of $T = 5$ packets.

In the MiDAS construction with MDS constituent codes, illustrated in Table V, we either use $\mathbf{c}^u[\cdot]$ or $\mathbf{c}^v[\cdot]$ codewords to recover $\mathbf{u}[i]$.

- Using $\mathbf{c}^u[\cdot]$ codewords: Here, a $(T + 1, T - N + 1) = (6, 4)$ block MDS code is applied to each of the $\mathbf{u}[\cdot]$ sub-packets. Each of the codewords $\mathbf{c}_j^u[i]$ for $j \in \{0, 1, 2\}$ in (39) has 3 erasures at i, $i + 1$ and $i + 3$ and hence the recovery of $\mathbf{u}[0]$ is impossible.
- Using $\mathbf{c}^v[\cdot]$ codewords: Also, the $\mathbf{v}[\cdot]$ sub-packets are protected using a $(T, T - B) = (5, 2)$ MDS codes. Let us consider the codewords $\mathbf{c}_j^v[i+3] = (v_j[i+3], v_{j+4}[i+4], p_j[i+5], p_{j+4}[i+6], p_{j+8}[i+7])$ for $j \in \{0, 1, 2, 3\}$ in (38). Each of these codewords has an erasure at time $i+3$ and the parity-check packets $p_j[i+5]$ and $p_{j+4}[i+6]$ are combined with $u_j[i]$ and $u_{j+4}[i+1]$ which are erased by the channel. Thus, a total of 3 erasures at times $i+3$, $i + 5$ and $i + 6$, which implies that $v_j[i + 3]$ can be recovered at time $i + 7$. Now, the decoder can compute $p_j[i+5]$ and $p_{j+4}[i+6]$ and subtract them from $q_j[i+5]$ and $q_{j+4}[i+6]$ to recover $u_j[i]$ and $u_j[i+1]$ with a delay of 7 and 6, respectively, i.e., exceeds the delay of $T = 5$.

Thus, unlike the case of MiDAS codes based on m-MDS , it is not possible to recover $\mathbf{u}[i]$ with a delay of $T = 5$ when a constituent block code is used. We will also see performance loss from using MDS block codes instead of m-MDS codes in our simulation results.

V. Unequal Source-Channel Inter-arrival Rates

In this section, we study the case when the source and channel inter-arrival rates are unequal, i.e., $M > 1$. We start by revisting the capacity expression in Theorem 3 in Section V-A. In Section V-B, we provide the code construction achieving such capacity. The decoding analysis is discussed in Section V-C. We illustrate both the encoding and decoding steps through a numerical example in Section V-D. We then provide

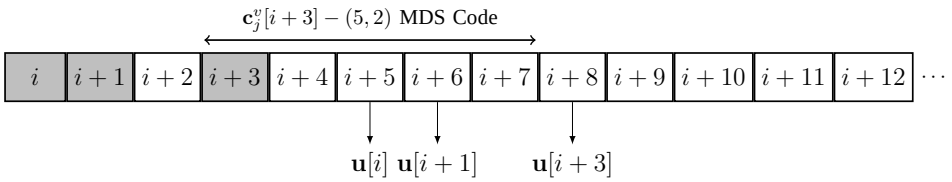


Fig. 10: A non-ideal erasure pattern in Section IV-E.

the converse proof of Theorem 3 in Section V-E. Finally, we present constructions that are robust against isolated erasures in Section V-F.

A. Capacity Expression

We note that $C = 0$, if $T < b$.[11] This follows since an erasure burst of length B can span all underlying channel packets in macro-packets $[i, i+T]$ thus making the recovery of $\mathbf{s}[i]$ by macro-packet $i+T$ impossible. This trivial case will therefore not be discussed further in the paper. When $T = b$, the capacity in Theorem 3 is given by:

$$C = \begin{cases} \frac{1}{2}, & 0 \le B' \le \frac{M}{2},\ T = b, \\ \frac{M-B'}{M}, & \frac{M}{2} < B' \le M-1,\ T = b. \end{cases} \tag{47}$$

In this special case of minimum delay, during the recovery of $\mathbf{s}[i]$ we can only use the unerased packets in $\mathbf{Y}[i,:]$ and $\mathbf{Y}[i+b,:]$ as all the intermediate macro-packets are completely erased. It turns out that a simple repetition code that uses $\min\left(M-B', \frac{M}{2}\right)$ information packets and an identical number of parity-check packets in each macro-packet achieves the capacity when $T = b$.

When $T > b$ the capacity in Theorem 3 reduces to the following.

$$C = \begin{cases} \frac{T}{T+b}, & 0 \le B' \le \frac{b}{T+b}M, \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & \frac{b}{T+b}M < B' \le M-1. \end{cases} \tag{48}$$

We propose the associated code construction below.

B. Code Construction

As illustrated in Fig. 11 we split each source packet into two packets as was the case in the generalized MS construction in Section IV-B. However, our construction involves an additional step of *reshaping* as illustrated in Fig. 12 to re-arrange the symbols in each macro-packet. We separately consider three cases below.

1) Encoding: $T \ge b$ and $B' \le \frac{b}{T+b}M$: We let

$$n = T + b, \quad k = MT, \tag{49}$$

throughout this case. Note that the rate $R = \frac{k}{Mn}$ reduces to the first case in both (47) and (48).

- **Source Splitting:** We assume that each source packet $\mathbf{s}[i] \in \mathbb{F}_q^k$ and partition the k symbols into two sub-packets $\mathbf{u}_{\rm vec}[i] \in \mathbb{F}_q^{k^u}$ and $\mathbf{v}_{\rm vec}[i] \in \mathbb{F}_q^{k^v}$ as follows,

$$\begin{aligned} \mathbf{s}[i] &= (s_0[i], \ldots, s_{k-1}[i]) \\ &= (\underbrace{u_0[i], \ldots, u_{k^u-1}[i]}_{\mathbf{u}_{\rm vec}[i]}, \underbrace{v_0[i], \ldots, v_{k^v-1}[i]}_{\mathbf{v}_{\rm vec}[i]}) \end{aligned} \tag{50}$$

  where we select

$$k^u = Mb, \quad k^v = M(T-b). \tag{51}$$

- **m-MDS Parity-Checks:** Apply a $(k^v + k^u, k^v, T)$ m-MDS code of rate $\frac{k^v}{k^v+k^u}$ to the sub-stream of $\mathbf{v}_{\rm vec}[\cdot]$ sub-packets generating k^u parity-check packets, $(p_0[i], \ldots, p_{k^u-1}[i]) = \mathbf{p}_{\rm vec}[i] \in \mathbb{F}_q^{k^u}$ for each macro-packet. In particular, we have that

$$\mathbf{p}_{\rm vec}[i] = \left(\sum_{j=0}^{T} \mathbf{v}_{\rm vec}^{\dagger}[i-j] \cdot \mathbf{H}_j \right)^{\dagger} \tag{52}$$

  where $\mathbf{H}_j \in \mathbb{F}_q^{k^v \times k^u}$ are the matrices associated with the m-MDS code (12).
- **Parity-Check Generation:** Combine the $\mathbf{u}_{\rm vec}[\cdot]$ sub-packets with the $\mathbf{p}_{\rm vec}[\cdot]$ parity-checks after applying a

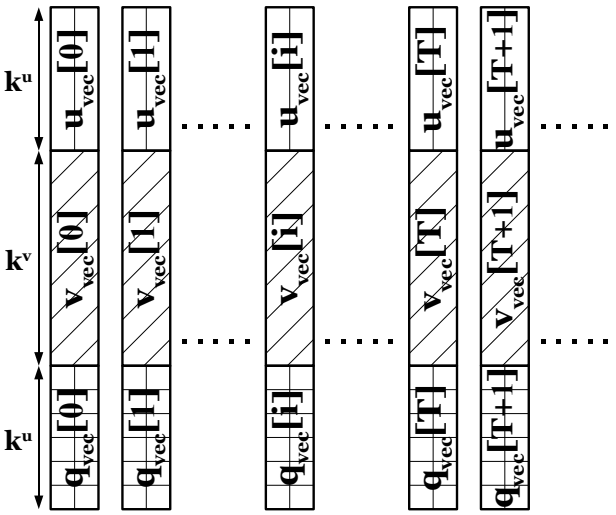


Fig. 11: Construction of Parity-Check Packets. As in the MS code, each source packet $\mathbf{s}[t]$ is divided into two sub-packets, $\mathbf{u}_{\rm vec}[t]$ and $\mathbf{v}_{\rm vec}[t]$. A m-MDS code is applied to the $\mathbf{v}_{\rm vec}[\cdot]$ sub-packets and a repetition code is applied to the $\mathbf{u}_{\rm vec}[\cdot]$ sub-packets. The resulting parities are then combined to generate the parity-check packets $\mathbf{q}_{\rm vec}[t] = \mathbf{p}_{\rm vec}[t] + \mathbf{u}_{\rm vec}[t-T]$.

[11] Recall from (8) that we express $B = bM + B'$ where $B' \in [0, M-1]$.

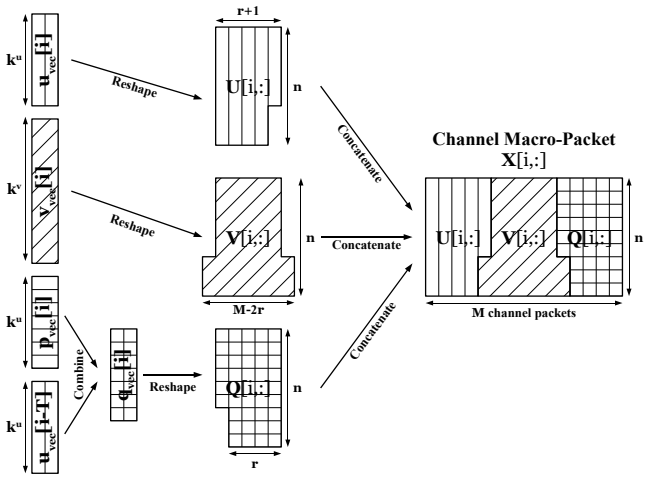


Fig. 12: Reshaping of Channel Packets. The three groups, $\mathbf{u}_{\text{vec}}[i]$, $\mathbf{v}_{\text{vec}}[i]$ and $\mathbf{q}_{\text{vec}}[i]$ are reshaped into $\mathbf{U}[i,:]$, $\mathbf{V}[i,:]$ and $\mathbf{Q}[i,:]$ which are denoted by vertically, diagonally and grid hatched boxes, respectively. These reshaped packets are then concatenated to form the channel macro-packet $\mathbf{X}[i,:]$.

shift of T to the former, i.e.,

$$\mathbf{q}_{\text{vec}}[i] = \mathbf{p}_{\text{vec}}[i] + \mathbf{u}_{\text{vec}}[i-T], \tag{53}$$

where $\mathbf{q}_{\text{vec}}[i] \in \mathbb{F}_q^{k^u}$.

- **Re-shaping:** In order to construct the macro-packet, we reshape $\mathbf{u}_{\text{vec}}[i]$, $\mathbf{v}_{\text{vec}}[i]$ and $\mathbf{q}_{\text{vec}}[i]$ into groups each of n symbols generating the matrices, $\mathbf{U}[i,:]$, $\mathbf{V}[i,:]$ and $\mathbf{Q}[i,:]$, respectively, as shown in (54).
  In (54), we define $r \in \mathbb{N}^0$ and $r' \in \{0,1,\ldots,n-1\}$ via
  $$k^u = r \cdot n + r'. \tag{57}$$
  Note that $\mathbf{u}[i,l] \in \mathbb{F}_q^n$ for each $l \in \{1,\ldots,r\}$ and $\mathbf{u}[i,r+1] \in \mathbb{F}_q^{r'}$. The splitting of $\mathbf{q}_{\text{vec}}[i]$ into $\mathbf{q}[i,j]$ in (54) follows in an analogous manner. We can express
  $$\mathbf{q}[i,j] = \mathbf{u}[i-T,j] + \mathbf{p}[i,j], \quad j = 1,2,\ldots,r+1 \tag{58}$$
  where $\mathbf{p}[i,j]$ is a sub-sequence of $\mathbf{p}_{\text{vec}}[i]$ defined in a similar manner. In the splitting of $\mathbf{v}_{\text{vec}}[i]$ into $\mathbf{v}[i,j]$, we note that $\mathbf{v}[i,1], \mathbf{v}[i,M-2r] \in \mathbb{F}_q^{n-r'}$ whereas $\mathbf{v}[i,j] \in \mathbb{F}_q^n$ for $2 \le j \le M-2r-1$. It can be easily verified that $M-2r > 0$ for our selected code parameters. When $M-2r = 1$ the structure of $\mathbf{V}[i,:]$ is as follows,
  $$\mathbf{V}[i,:] = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}[i,1] \\ \mathbf{0} \end{bmatrix}, \tag{59}$$
  where $\mathbf{v}[i,1] \in \mathbb{F}_q^{n-2r'}$.
- **Macro-Packet Generation:** Concatenate $\mathbf{U}[i,:]$, $\mathbf{V}[i,:]$ and $\mathbf{Q}[i,:]$ to construct the channel macro-packet $\mathbf{X}[i,:]$ as in (56). Note that the channel macro-packet at time i is denoted by $\mathbf{X}[i,:] \in \mathbb{F}_q^{n \times M}$ and the j-th channel packet in $\mathbf{X}[i,:]$ by $\mathbf{x}[i,j] \in \mathbb{F}_q^n$ for $j \in \{1,\ldots,M\}$.

Note that in the minimum delay case, i.e., $T = b$ we have that $k^v = M(T-b) = 0$. , This construction degenerates into a repetition code, and the corresponding rate of such repetition code is $R = \frac{k^u}{2k^u} = \frac{1}{2}$, which meets the capacity expression in the first case in (47). The construction achieving the second case with $T = b$ and $B' > \frac{M}{2}$ also involves a repetition code and is discussed later in this section.

This completes the description of the encoding function for the first case in (9) and (10). Fig. 13 illustrates the overall encoder structure.

2) Encoding: $T > b$ and $B' > \frac{b}{T+b}M$: We begin by choosing the following values of n and k,

$$n = T + b + 1, \quad k = M(T+b+1) - B \tag{60}$$

and note that the rate $R = \frac{k}{Mn}$ reduces to the second case in (48).

- Split each source $\mathbf{s}[i] \in \mathbb{F}_q^k$ into k symbols and divide them into two sub-packets $\mathbf{u}_{\text{vec}}[i] \in \mathbb{F}_q^{k^u}$ and $\mathbf{v}_{\text{vec}}[i] \in \mathbb{F}_q^{k^v}$ as in (50). This time we select
  $$k^u = B = Mb + B', \quad k^v = M(T+b+1) - 2B \tag{61}$$
- Apply a $(k^v + k^u, k^v, T)$ m-MDS code of rate $\frac{k^v}{k^v+k^u}$ to the sub-stream of $\mathbf{v}_{\text{vec}}[\cdot]$ sub-packets generating k^u parity-check packets, $(p_0[i],\ldots,p_{k^u-1}[i]) = \mathbf{p}_{\text{vec}}[i] \in \mathbb{F}_q^{k^u}$ for each macro-packet as in (52).
- Combine the $\mathbf{u}_{\text{vec}}[\cdot]$ sub-packets with the $\mathbf{p}_{\text{vec}}[\cdot]$ parity-checks after applying a shift of T to the former, i.e., $\mathbf{q}_{\text{vec}}[i] = \mathbf{p}_{\text{vec}}[i] + \mathbf{u}_{\text{vec}}[i-T]$.
- Reshape the $\mathbf{u}_{\text{vec}}[i]$, $\mathbf{v}_{\text{vec}}[i]$ and $\mathbf{q}_{\text{vec}}[i]$ vectors into matrices $\mathbf{U}[i,:]$, $\mathbf{V}[i,:]$ and $\mathbf{Q}[i,:]$ as in (54). In particular, we let r and r' be such that $k^u = r \cdot n + r'$ as in (57). As in (55) we split $\mathbf{u}_{\text{vec}}[i]$ into $\{\mathbf{u}[i,j]\}_{1 \le j \le (r+1)}$ where $\mathbf{u}[i,j] \in \mathbb{F}_q^n$ for $1 \le j \le r$ and $\mathbf{u}[i,r+1] \in \mathbb{F}_q^{r'}$ holds. In a similar manner, we split $\mathbf{q}_{\text{vec}}[i]$ into vectors $\{\mathbf{q}[i,j]\}_{1 \le j \le (r+1)}$ where $\mathbf{q}[i,j] \in \mathbb{F}_q^n$ for $1 \le j \le r$ and $\mathbf{q}[i,r+1] \in \mathbb{F}_q^{r'}$ holds. Finally we split $\mathbf{v}_{\text{vec}}[i]$ into $\{\mathbf{v}[i,j]\}_{1 \le j \le (M-2r)}$ where $\mathbf{v}[i,1], \mathbf{v}[i,M-2r] \in \mathbb{F}_q^{n-r'}$ and $\mathbf{v}[i,j] \in \mathbb{F}_q^n$ for $2 \le j \le (M-2r-1)$.
- Generate the Macro-Packet $\mathbf{X}[i,:]$ by concatenating $\mathbf{U}[i,:]$, $\mathbf{V}[i,:]$ and $\mathbf{Q}[i,:]$ as in (56).

3) Encoding: $T = b$ and $B' > \frac{M}{2}$: A simple repetition scheme is used. We split each source packet into $M - B'$ packets, i.e., $\mathbf{s}[i] = (s_0[i],\ldots,s_{M-B-1'}[i])$ and assign the channel packets as follows,

$$\mathbf{x}[i,j] = \begin{cases} s_{j-1}[i] & j \in [1, M-B'] \\ 0 & j \in [M-B'+1, B'] \\ s_{j-B'-1}[i-T] & j \in [B'+1, M]. \end{cases} \tag{62}$$

The rate of such code is clearly $R = \frac{M-B'}{M}$ as stated in the second case in (47). In this case, by inspection we can check that the code described above is decodable within the decoding delay $T = b$. Thus, we will only focus on the previous two cases in our decoding analysis.

$$\mathbf{U}[i,:] = \left[\begin{array}{c|c|c|c} \mathbf{u}[i,1] & \cdots & \mathbf{u}[i,r] & \begin{array}{c}\mathbf{u}[i,r+1]\\ \hline \mathbf{0}\end{array} \end{array}\right] \in \mathbb{F}_q^{n\times r+1}$$
$$\mathbf{V}[i,:] = \left[\begin{array}{c|c|c|c} \begin{array}{c}\mathbf{0}\\ \hline \mathbf{v}[i,1]\end{array} & \mathbf{v}[i,2] & \cdots & \mathbf{v}[i,M-2r-1] \end{array}\right] \left[\begin{array}{c} \begin{array}{c}\mathbf{0}\\ \hline \mathbf{v}[i,M-2r]\end{array} \end{array}\right] \in \mathbb{F}_q^{n\times M-2r} \tag{54}$$
$$\mathbf{Q}[i,:] = \left[\begin{array}{c|c|c|c} \begin{array}{c}\mathbf{q}[i,r+1]\\ \hline \mathbf{0}\end{array} & \mathbf{q}[i,r] & \cdots & \mathbf{q}[i,1] \end{array}\right] \in \mathbb{F}_q^{n\times r+1},$$

where

$$\mathbf{u}_{\text{vec}}[i] = \begin{bmatrix} \mathbf{u}[i,1]\\ \mathbf{u}[i,2]\\ \vdots\\ \mathbf{u}[i,r]\\ \mathbf{u}[i,r+1] \end{bmatrix}, \qquad \mathbf{v}_{\text{vec}}[i] = \begin{bmatrix} \mathbf{v}[i,1]\\ \mathbf{v}[i,2]\\ \vdots\\ \mathbf{v}[i,M-2r-1]\\ \mathbf{v}[i,M-2r] \end{bmatrix}, \qquad \mathbf{q}_{\text{vec}}[i] = \begin{bmatrix} \mathbf{q}[i,1]\\ \mathbf{q}[i,2]\\ \vdots\\ \mathbf{q}[i,r]\\ \mathbf{q}[i,r+1] \end{bmatrix} \tag{55}$$

$$\mathbf{X}[i,:] = [\mathbf{x}[i,1]|\ldots|\mathbf{x}[i,M]] =$$
$$\begin{cases} \left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c} \mathbf{u}[i,1] & \cdots & \mathbf{u}[i,r] & \begin{array}{c}\mathbf{u}[i,r+1]\\ \mathbf{v}[i,1]\end{array} & \mathbf{v}[i,2] & \cdots & \begin{array}{c}\mathbf{q}[i,r+1]\\ \mathbf{v}[i,M-2r]\end{array} & \mathbf{q}[i,r] & \cdots & \mathbf{q}[i,1] \end{array}\right], & M-2r>1 \\ \left[\begin{array}{c|c|c|c|c|c|c} \mathbf{u}[i,1] & \cdots & \mathbf{u}[i,r] & \begin{array}{c}\mathbf{u}[i,r+1]\\ \mathbf{v}[i,1]\\ \mathbf{q}[i,r+1]\end{array} & \mathbf{q}[i,r] & \cdots & \mathbf{q}[i,1] \end{array}\right] & M-2r=1 \end{cases} \tag{56}$$

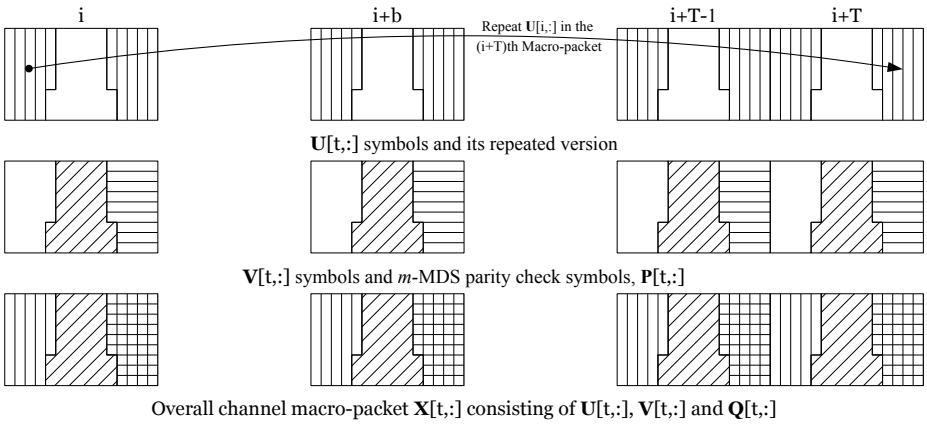


Fig. 13: Encoding of source packets into macro-packets. Each source packet is split into two groups. A repetition code is applied to the $\mathbf{U}[t,:]$ group with a delay of T macro-packets and is denoted by vertically hatched boxes as shown in the first figure. A m-MDS code is applied to the $\mathbf{V}[t,:]$ group which is denoted by diagonally hatched boxes to generate the parity-checks $\mathbf{P}[i,:]$ denoted by the horizontally hatched boxes as shown in the second figure. The combination of the resulting parity-checks of the two groups is indicated in the last figure with grid hatched boxes.

C. Decoding Analysis

Consider a channel that introduces a burst of length $B = bM + B'$ starting from $\mathbf{x}[i,j]$ for $j \in \{1,\ldots,M\}$. We first show how to recover $\mathbf{s}[i]$ by the macro-packet $i+T$. Note that since our code is time invariant, it suffices to consider only the recovery of $\mathbf{s}[i]$. Once $\mathbf{s}[i]$ is recovered, we can compute $\mathbf{X}[i,:]$ and repeat the same procedure with the smaller burst that starts at $\mathbf{x}[i+1,1]$ to recover $\mathbf{s}[i+1]$ and so on.

The decoding steps are as follows,

1) Step 1: In each macro-packet $\mathbf{X}[t,:]$, for $t \in [i+b, i+T-1]$, recover all the unerased symbols of $\mathbf{p}_{\text{vec}}[t]$ by subtracting out $\mathbf{u}_{\text{vec}}[t-T]$ from the corresponding $\mathbf{q}_{\text{vec}}[t]$ as the former are not erased. Since, $\mathbf{u}[i,1],\ldots,\mathbf{u}[i,j-1]$ are not erased, we can subtract these packets from the corresponding $\mathbf{q}_{\text{vec}}[i+T]$ to recover the respective $\mathbf{p}_{\text{vec}}[i+T]$ packets.
2) Step 2: Recover all erased $\mathbf{v}_{\text{vec}}[\cdot]$ sub-packets by the macro-packet $i+T$ using the underlying (k^u+k^v, k^v, T) m-MDS code. This step will be justified later in the sequel.
3) Step 3: Compute $\mathbf{p}_{\text{vec}}[i+T]$ as it combines $\mathbf{v}_{\text{vec}}[\cdot]$ sub-packets that are either not erased or recovered in the previous step.

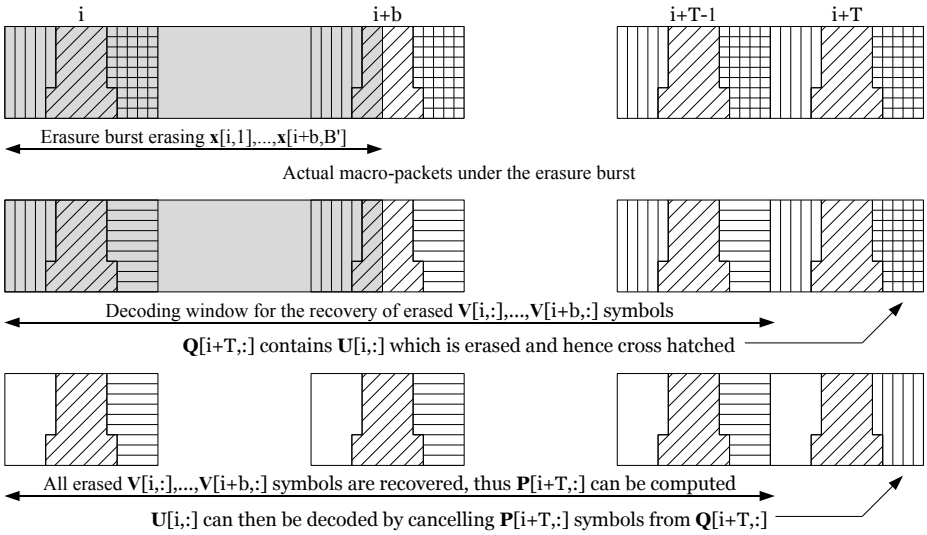


Fig. 14: Decoding for the burst pattern starting from $\mathbf{x}[i,1]$. The grey boxes denote an erasure burst of length B. The horizontally hatched parity-checks in the second figure are used to recover the erased $\mathbf{V}[i,:],\ldots,\mathbf{V}[i+b,:]$ packets. The third figure shows the recovery of $\mathbf{u}[i]$ using the parity-checks in macro-packet $i+T$.

4) Step 4: Subtract $\mathbf{p}_{\text{vec}}[i+T]$ from $\mathbf{q}_{\text{vec}}[i+T]$ to recover $\mathbf{u}_{\text{vec}}[i]$ within a delay of T macro-packets. At this point both $\mathbf{u}_{\text{vec}}[i]$ and $\mathbf{v}_{\text{vec}}[i]$ have been recovered (and hence $\mathbf{s}[i]$) with a delay of T macro-packets as required.

It only remains to show the sufficiency of the m-MDS code in Step 2. To do that we use the following lemma.

Lemma 3. *Consider any erasure burst of length B starting at $\mathbf{x}[i,j]$ for some $j \in \{1,\ldots,M-r\}$. After Step 1 of cancelling $\mathbf{u}_{\text{vec}}[t]$ sub-packets, the total number of unrecovered symbols in the sequence $\{(\mathbf{v}_{\text{vec}}[t],\mathbf{p}_{\text{vec}}[t])\}_{i\le t\le i+T}$ is at most $k^u(T+1)$.*

Proof: See Appendix D. ∎

We next claim that the decoder can recover all the erased $\mathbf{v}_{\text{vec}}[t]$ sub-packets by the end of macro-packet $i+T$. To prove this, we recall that $(\mathbf{v}_{\text{vec}}[t],\mathbf{p}_{\text{vec}}[t])$ is a m-MDS code with parameters (k^v+k^u,k^v,T). We consider the following cases:

- If the burst starts at $j\in\{1,...,r+1\}$ then all the symbols in $\{(\mathbf{v}_{\text{vec}}[t],\mathbf{p}_{\text{vec}}[t])\}_{i\le t\le i+b-1}$ are erased whereas a portion of the symbols in $\{(\mathbf{v}_{\text{vec}}[i+b],\mathbf{p}_{\text{vec}}[i+b])\}$ are erased until the termination of the erasure burst. Furthermore, $\{\mathbf{p}_{\text{vec}}[i+T,l]\}_{j\le l\le r+1}$ are also considered to be erased since they are interfered by the erased $\mathbf{u}_{\text{vec}}[i,l]$ sub-packets from macro-packet i. Note that all the erased symbols involving $\mathbf{v}_{\text{vec}}[t]$ will occur in a single erasure burst. Thus, applying property L3 in Lemma 1 with $j=T$ and $c=0$ and using $\hat{B}+\hat{I}\le k^u(T+1)=(n-k)(j+1)$, which follows from Lemma 3, we are guaranteed that all the erased $\mathbf{v}_{\text{vec}}[t]$ are recovered at the end of macro-packet $i+T$.
- If the burst starts at $j\in\{r+2,\ldots,M-r\}$ then none of the sub-packets $\mathbf{u}_{\text{vec}}[i]$ are erased and can be subtracted out from $\mathbf{q}_{\text{vec}}[i+T]$ to recover $\mathbf{p}_{\text{vec}}[i+T]$. All the erased symbols thus occur in a burst. Thus, using property L2 in Lemma 1, and using $\hat{B}\le(n-k)(T+1)$ which follows from Lemma 3, we are guaranteed that all the erased $\mathbf{v}_{\text{vec}}[t]$ are recovered at the end of macro-packet $i+T$.
- If $j\in\{M-r+1,\ldots,M-1\}$ then none of the symbols in either $\mathbf{u}_{\text{vec}}[i]$ or $\mathbf{v}_{\text{vec}}[i]$ are erased. Thus, we can proceed to block $i+1$ and apply the first step.

Finally as mentioned in Step 4 above, once all the erased sub-packets $\mathbf{v}_{\text{vec}}[t]$ have been recovered by macro-packet $i+T$, their effect can be canceled and $\mathbf{u}_{\text{vec}}[t]$, for $t\in\{i,i+1,\ldots,i+b\}$ can be sequentially recovered from macro-packet $t+T$ by computing and subtracting $\mathbf{p}_{\text{vec}}[t+T]$ from $\mathbf{q}_{\text{vec}}[t+T]$. Thus, each $\mathbf{s}[t]=(\mathbf{u}_{\text{vec}}[t],\mathbf{v}_{\text{vec}}[t])$ can be recovered by the end of macro-packet $t+T$. This completes the decoding analysis.

Remark 4. *We discuss intuition on the fact that the capacity function does not decrease with B' in the first case in* (48) *defined by $B'\le\frac{b}{T+b}M$. Recall that for this case the parameters that are selected are $k^u=Mb$ and $n=T+b$. Consider an erasure burst that starts at $\mathbf{x}[i,1]$ and terminates at $\mathbf{x}[i+b,B']$. We claim that for such an erasure burst, as long as $B'\le\frac{Mb}{T+b}$, only the $\mathbf{u}[\cdot]$ sub-packets are erased in macro-packet $\mathbf{X}[i+b,:]$. In particular, the number of symbols that are erased in marco-packet $\mathbf{X}[i+b,:]$ is equal to $nB'=(T+b)B'\le Mb=k^u$. Since the $\mathbf{u}[\cdot]$ sub-packets appears before any other packets in each macro-packet only these packets are erased. Thus, during the recovery process, the number of parity-checks available for recovering $\mathbf{v}[\cdot]$ sub-packets does not decrease as B' is increased from 0 to $\frac{Mb}{T+b}$. Thus, the same code parameters can be used. The above argument assumes that the burst starts at the beginning of a macro-packet. In Appendix D, in the proof of Lemma 3, we show that this is indeed the worst case pattern. If the burst starts anywhere else, the number of available parity-checks could only increase. This explains why, remarkably, the capacity is not a strictly decreasing function of B.*

As a final remark, note that the above construction achieves

the capacity in Theorem 3 for $W \geq M(T+1)$. For the case when $W < M(T+1)$, the same construction can be used with replacing the delay T with the effective delay $T_{\text{eff}} = \lfloor \frac{W}{M} \rfloor - 1$.

D. Example

In this section, we show a code construction for parameters $M = 2$, $B = 3$, $T = 3$. Note that $b = 1$ and $B' = 1 > \frac{b}{T+b} M$. Thus, the capacity is given by $C = \frac{M(T+b+1)-B}{M(T+b+1)} = \frac{7}{10}$, which can be achieved using the code illustrated in Table VI.

Encoding:

1) Split each source packet into $M(T+b+1) - B = 7$ packets, i.e., $\mathbf{s}[i] = (s_0[i], \cdots, s_6[i])$.
2) Divide these into two sub-packets, $\mathbf{u}_{\text{vec}}[i]$ and $\mathbf{v}_{\text{vec}}[i]$ with $k^u = B = 3$ and $k^v = M(T+b+1) - 2B = 4$ symbols, respectively, as in (50). We let $\mathbf{u}_{\text{vec}}[i] = (u_0[i], \cdots, u_2[i]) = (s_0[i], \cdots, s_2[i])$ and $\mathbf{v}_{\text{vec}}[i] = (v_0[i], \cdots, v_3[i]) = (s_3[i], \ldots, s_6[i])$.
3) We place $B = 3$ parity packets $\mathbf{q}_{\text{vec}}[i] = (q_0[i], q_1[i], q_2[i])$ into the last channel packet of each macro-packet. These parities consist of two components, $\mathbf{q}_{\text{vec}}[i]$=$\mathbf{p}_{\text{vec}}[i]$+$\mathbf{u}_{\text{vec}}[i-3]$. The parity packets $\mathbf{p}[i]$ are generated using a m-MDS code.

Decoding: Since $M = 2$, there are two burst patterns that need to be checked.

1) Burst that erases $\mathbf{x}[0,1]$, $\mathbf{x}[0,2]$ and $\mathbf{x}[1,1]$.
   Recovery of v packets: We first subtract $\mathbf{u}_{\text{vec}}[t-T]$ from $\mathbf{q}_{\text{vec}}[t]$ for $t = \{1,2\}$ to recover the corresponding $\mathbf{p}_{\text{vec}}[t]$. These are a total of 6 symbols and thus can be used to recover $v_0[0], \cdots, v_3[0]$ as well as $v_0[1], v_1[1]$. In other words, all erased v symbols are recovered by the end of the macro-packet $\mathbf{X}[2,:]$.
   Recovery of u packets: With all the erased v packets now recovered, we can compute the $\mathbf{p}_{\text{vec}}[t]$ packets for $t = \{3,4\}$ and subtract them from $\mathbf{q}_{\text{vec}}[t]$ to recover $\mathbf{u}_{\text{vec}}[0]$ and $\mathbf{u}_{\text{vec}}[1]$ at their respective deadlines.
2) Burst that erases $\mathbf{x}[0,2], \mathbf{x}[1,1], \mathbf{x}[1,2]$.
   Recovery of v packets: Since $\mathbf{u}_{\text{vec}}[0]$ is not erased, we can subtract it from $\mathbf{q}_{\text{vec}}[3]$ to recover $\mathbf{p}_{\text{vec}}[3]$. This together with $\mathbf{p}_{\text{vec}}[2]$ is a total of 6 symbols. Thus, they can be used to recover the erased v packets $(v_2[0], v_3[0])$ and $(v_0[1], \cdots, v_3[1])$.
   Recovery of u packets: Similar to the previous burst pattern, we compute the value of the parity-check packets $\mathbf{p}_{\text{vec}}[4]$ and subtract it from $\mathbf{q}_{\text{vec}}[4]$ to recover $\mathbf{u}[1]$ by its deadline.

E. Converse

In order to establish the converse, we first consider the case when $T > b$. We show that any feasible rate satisfies

$$R \leq R^+ = \min\left(\frac{M(T+b+1)-B}{M(T+b+1)}, \frac{T}{T+b}\right). \tag{63}$$

Consider a periodic erasure channel as shown in Figure 15. Each period consists of $\tau_P = T+b+1$ macro-packets. In each such period, the first B channel packets are erased and the subsequent $M(b+T+1) - B$ are not. Consider the first period with the burst starting at $\mathbf{x}[0,1]$. By definition we require that $\mathbf{s}[0]$ be recovered by the end of macro-packet T, $\mathbf{s}[1]$ by macro-packet $T+1$ and likewise the last erased source packet $\mathbf{s}[b]$ by macro-packet $T+b$. Thus, all the lost source packets are recovered by macro-packet $t = T+b$. Once these erased packets are recovered, we can treat these erasures as having never happened and simply repeat the argument for the next period and so on. Therefore, our proposed streaming code must be a feasible code for the periodic erasure channel. Since the capacity of the erasure channel is simply the fraction of the non-erased channel packets, it follows that

$$R^+ = \frac{M(T+b+1) - (bM + B')}{M(T+b+1)}. \tag{64}$$

is an upper bound on the rate of any feasible streaming code.

To establish the other inequality in (63) we consider a periodic erasure channel consisting of $\tau_P = T+b$ macro-packets and assume that in each period the first $\hat{B} = Mb \leq B$ channel packets are erased. Thus, in the proposed channel, the first b macro-packets are completely erased in each period and the remaining T macro-packets are not erased. In particular, in the first period, $\mathbf{s}[0], \ldots, \mathbf{s}[b-1]$ must be recovered at the end of macro-packets $T, \ldots, T+b-1$ respectively. At this point all the erased source packets have been recovered and we can proceed to the recovery of the second burst starting at macro-packet $T+b$. Thus, the streaming code must also be feasible on this erasure channel whose capacity is clearly $\frac{T}{T+b}$, and thus the upper bound follows.

When $T = b$ we show that

$$C \leq \min\left(\frac{M-B'}{M}, \frac{1}{2}\right). \tag{65}$$

When $B' \leq M/2$, the second condition $C \leq \frac{1}{2}$ dominates. This bound immediately follows from (63) by substituting $T = b$ in the second expression in (63). Thus, we only need to show that when $B' > \frac{M}{2}$ and $T = b$ the upper bound $C \leq \frac{M-B'}{M}$ is valid.

We start by considering a channel that erases the first $B = bM + B'$ channel packets $\mathbf{x}[i,1], \ldots, \mathbf{x}[i+b, B']$. Since the delay constraint for $\mathbf{s}[i]$ is $i+T = i+b$, the following equation should be satisfied,

$$\begin{aligned} &\mathsf{H}(\mathbf{s}[i] | \mathbf{x}[i+b, B'+1], \ldots, \mathbf{x}[i+b, M]) = 0 \\ &\Rightarrow \mathsf{H}(\mathbf{s}) \leq (M-B')\mathsf{H}(\mathbf{x}), \end{aligned} \tag{66}$$

which implies that $R = \frac{H(\mathbf{s})}{MH(\mathbf{x})} \leq \frac{M-B'}{M}$ as required. This completes the proof of the upper bound.

F. Robust Extensions

In Section V-B, we provided capacity achieving codes for $\mathcal{C}(1, B, W \geq M(T+1))$. In order to extend the codes for channels with $N > 1$, we apply the approach used in the MiDAS construction in Section IV-C. In particular, we construct an optimal burst erasure code and then append additional parity-checks for the $\mathbf{u}[\cdot]$ sub-packets to deal with isolated losses. In particular, we extend the macro-packet construction in (56) as follows,

TABLE VI: Code construction for $(M = 2, B = 3, T = 3)$ achieving a rate of $R = \frac{7}{10}$.

$\mathbf{X}[0,:]$		$\mathbf{X}[1,:]$		$\mathbf{X}[2,:]$		$\mathbf{X}[3,:]$	
$\mathbf{x}[0,1]$	$\mathbf{x}[0,2]$	$\mathbf{x}[1,1]$	$\mathbf{x}[1,2]$	$\mathbf{x}[2,1]$	$\mathbf{x}[2,2]$	$\mathbf{x}[3,1]$	$\mathbf{x}[3,2]$
$u_0[0]$	$v_2[0]$	$u_0[1]$	$v_2[1]$	$u_0[2]$	$v_2[2]$	$u_0[3]$	$v_2[3]$
$u_1[0]$	$v_3[0]$	$u_1[1]$	$v_3[1]$	$u_1[2]$	$v_3[2]$	$u_1[3]$	$v_3[3]$
$u_2[0]$	$u_0[-3]+p_0[0]$	$u_2[1]$	$u_0[-2]+p_0[1]$	$u_2[2]$	$u_0[-1]+p_0[2]$	$u_2[3]$	$u_0[0]+p_0[3]$
$v_0[0]$	$u_1[-3]+p_1[0]$	$v_0[1]$	$u_1[-2]+p_1[1]$	$v_0[2]$	$u_1[-1]+p_1[2]$	$v_0[3]$	$u_1[0]+p_1[3]$
$v_1[0]$	$u_2[-3]+p_2[0]$	$v_1[1]$	$u_2[-2]+p_2[1]$	$v_1[2]$	$u_2[-1]+p_2[2]$	$v_1[3]$	$u_2[0]+p_2[3]$

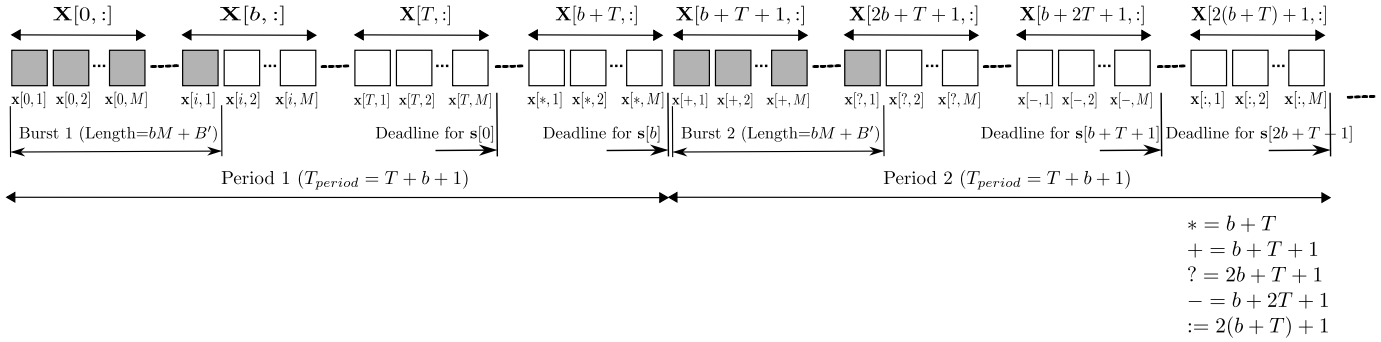


Fig. 15: Periodic Erasure Channel used in the Converse Proof of Theorem 3. We assume that the burst starts in macro-block 0 at the first packet and terminates in macro-block b. The period of the channel spans $T + b + 1$ macro-blocks as shown.

$$\mathbf{X}[i,:] = [\mathbf{x}[i,1]|\dots|\mathbf{x}[i,M]] = \left[\begin{array}{c|c|c|c|c} \begin{matrix}\mathbf{u}[i,1]\\ \mathbf{p}^u[i,1]\end{matrix} & \dots & \begin{matrix}\mathbf{u}[i,r]\\ \mathbf{p}^u[i,r]\end{matrix} & \begin{matrix}\mathbf{u}[i,r+1]\\ \mathbf{v}[i,1]\\ \mathbf{p}^u[i,r+1]\end{matrix} & \dots \\ \multicolumn{5}{c}{\begin{matrix}\mathbf{q}[i,r+1]\\ \mathbf{v}[i,M-2r]\\ \mathbf{p}^u[i,M-r]\end{matrix} \quad \begin{matrix}\mathbf{q}[i,r]\\ \mathbf{p}^u[i,M-r+1]\end{matrix} \quad \dots \quad \begin{matrix}\mathbf{q}[i,1]\\ \mathbf{p}^u[i,M]\end{matrix}} \end{array}\right] \tag{67}$$

where $\mathbf{u}[i,j]$, $\mathbf{v}[i,j]$ sub-packets and $\mathbf{q}[i,j]$ packets are obtained from the optimal code for the $\mathcal{C}(N = 1, B, W)$ channel. We apply another $(k^u + Mk^s, k^u, T)$ m-MDS code to the $\mathbf{u}_{\text{vec}}[\cdot]$ sub-packets generating Mk^s parity-check symbols $(p_1^u[i], \dots, p_{Mk^s}^u[i]) = \mathbf{p}_{\text{vec}}^u[i] \in \mathbb{F}_q^{Mk^s}$. We then concatenate the generated parities after splitting them into M equal groups to each channel packet, $\mathbf{p}_{\text{vec}}^u[i] = (\mathbf{p}^u[i,1], \dots, \mathbf{p}^u[i,M])$ as shown in (67). The corresponding rate of such code is clearly $R = \frac{k^u+k^v}{M(n+k^s)}$, where k^u, k^v and n are based on the optimal code for the burst-only channel.

Proposition 4. *Consider the layered code design for recovering from isolated erasures. To recover from any $N \leq \left\lfloor \frac{T}{T+b} Mb \right\rfloor$ isolated erasures when $W \geq M(T+1)$ and $T > b$, it suffices to select*

$$k^s = \left\lceil \frac{Nn}{M(T+1) - N} \right\rceil. \tag{68}$$

where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote the ceil and floor functions respectively.

Proof: We recall that there are two m-MDS codes underlying our construction in (67). A $(k^u + k^v, k^v, T)$ m-MDS code is applied to $\mathbf{v}_{\text{vec}}[\cdot]$ sub-packets to generate parity-checks $\mathbf{p}_{\text{vec}}[\cdot]$ and $\mathbf{q}_{\text{vec}}[t] = \mathbf{p}_{\text{vec}}[t] + \mathbf{u}_{\text{vec}}[t-T]$ are transmitted. Furthermore, a $(k^u + Mk^s, k^u, T)$ m-MDS code is applied to the $\mathbf{u}_{\text{vec}}[\cdot]$ sub-packets to generate parity-checks $\mathbf{p}_{\text{vec}}^u[\cdot]$.

Let us consider the window of length T consisting of the macro-packets $\mathbf{X}[i,:], \dots, \mathbf{X}[i+T-1,:]$ and assume that there are N erasures in arbitrary positions. Note that in $\mathbf{q}_{\text{vec}}[t] = \mathbf{p}_{\text{vec}}[t] + \mathbf{u}_{\text{vec}}[t-T]$ for $t \in [i, i+T-1]$, the $\mathbf{u}_{\text{vec}}[\cdot]$ are from time $i-1$ or before, and can be canceled to recover $\mathbf{p}_{\text{vec}}[t]$. The $(k^u + k^v, k^v, T)$ m-MDS code can recover $\mathbf{v}_{\text{vec}}[i]$ if no more than $k^u T$ symbols are erased among $(\mathbf{v}_{\text{vec}}[i], \mathbf{q}_{\text{vec}}[i], \dots, \mathbf{v}_{\text{vec}}[i+T-1], \mathbf{q}_{\text{vec}}[i+T-1])$. Since these symbols are reshaped into columns each having no more than n symbols, the number of erasures that are guaranteed to be corrected is given by,

$$N^v = \left\lfloor \frac{k^u T}{n} \right\rfloor$$
$$\geq \min\left(\left\lfloor \frac{(bM+B')T}{T+b+1} \right\rfloor \Big|_{B' \geq \frac{b}{T+b}M}, \left\lfloor \frac{MbT}{T+b} \right\rfloor \Big|_{B' < \frac{b}{T+b}M} \right) \tag{69}$$
$$= \left\lfloor \frac{MbT}{T+b} \right\rfloor \geq N, \tag{70}$$

where we use

$$(k^u, n) = \begin{cases} (B, T+b+1), & B' \geq \frac{b}{T+b}M \\ (Mb, T+b), & B' < \frac{b}{T+b}M \end{cases} \tag{71}$$

to get (69) and substitute for $B' \geq \frac{b}{T+b}M$ in the first term in (69) to get (70).

Next we consider the number of erased packets that can be corrected by the $(k^u + Mk^s, k^u, T)$ m-MDS code. Using

Lemma 1, one can see that this code can recover from $Mk^s(T+1)$ erasures in the window of interest. Since each channel input can have up to $n+k^s$ symbols belonging to this code, the total number of erasures that can be corrected is given by,

$$N^u = \left\lfloor \frac{Mk^s(T+1)}{n+k^s} \right\rfloor \tag{72}$$

which upon re-arranging gives (68). ■

Remark 5. *Unlike the case of MiDAS codes, we do not claim the optimality of the proposed robust codes. Nevertheless in the simulation results we observe that in some cases these codes outperform baseline schemes.*

VI. Simulation Results

In this section, we study the validity of our proposed code constructions over statistical channel models. We consider two classes of channels that introduce both burst and isolated erasures. A Gilbert-Elliott channel is a two-state Markov model. In the "good state", each channel packet is lost with a probability of ε whereas in the "bad state" each channel packet is lost with a probability of 1. We note that the average loss rate of the Gilbert-Elliott channel is given by

$$\Pr(\mathcal{E}) = \frac{\beta}{\beta+\alpha}\varepsilon + \frac{\alpha}{\alpha+\beta}. \tag{73}$$

where α and β denote the transition probability from the good state to the bad state and vice versa. As long as the channel stays in the bad state the channel behaves as a burst-erasure channel. The length of each burst is a Geometric random variable with mean of $\frac{1}{\beta}$. When the channel is in the good state it behaves as an i.i.d. erasure channel with an erasure probability of ε. The gap between two successive bursts is also a geometric random variable with a mean of $\frac{1}{\alpha}$. Finally note that $\varepsilon = 0$ results in a Gilbert Channel [34], which only results in burst losses.

Fig. 17 shows a Fritchman channel model [35] with a total of $\mathcal{N}+1$ states. One of the states is the good state and the remaining $\mathcal{N}$ states are bad states. We again let the transition probability from the good state to the first bad state E_1 to be α whereas the transition probability from each of the bad states equals β. Let ε be the probability of a packet loss in good state. We lose packets in any bad state with probability 1. The burst length distribution in a Fritchman model is a hyper-geometric random variable instead of a geometric random variable. Fritchman and related higher order Markov models are commonly used to model fade-durations in mobile links.

In a conferencing application with 2 Mbps video and packet of size 512 bytes, the inter-packet time is about 2 millisecond. A moderate decoding delay of 100 ms would correspond to $T = 50$ packets. With this in mind, our results in this Section will adopt different values of T from 12 to 80. Furthermore, we note that a benign playback disruption of once every 30 minutes corresponds to a packet loss rate of 10^{-6}, while an unacceptable disruption every two seconds correspond to packet loss rate of 10^{-3}. In our simulations, we would vary channel parameters over a wide range to provide comparison of different schemes over significant portions of packet loss rates between 10^{-3} to 10^{-6}.

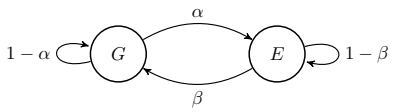


Fig. 16: Gilbert-Elliott Channel Model

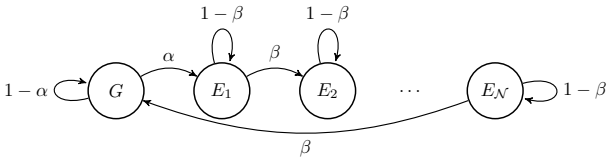
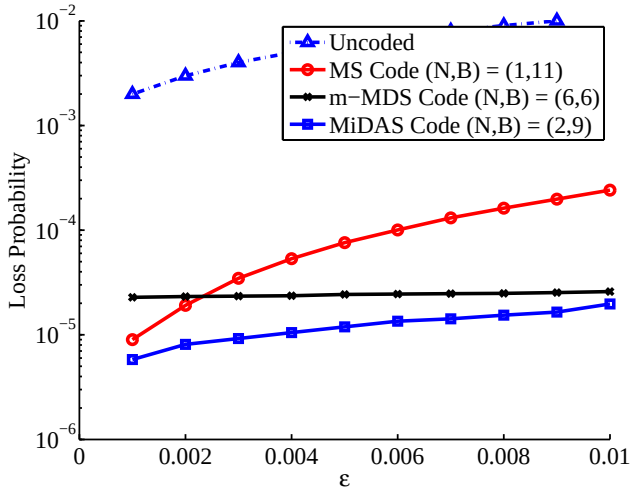


Fig. 17: Fritchman Channel Model

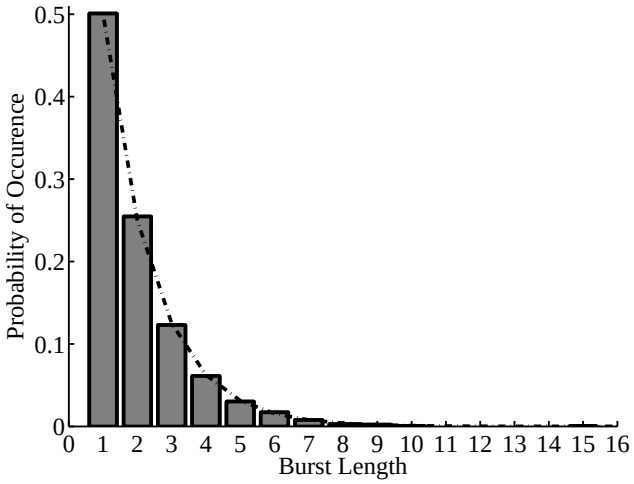
A. Equal Source-Channel Inter-arrival Rates

In Fig. 18(a) and Fig. 19(a), we study the performance of various streaming codes over the Gilbert-Elliott channel. The channel parameters and code parameters are shown in Table VII and VIII respectively. Fig. 18(b) and 19(b) indicate the histogram of the burst lengths observed for the two channels. The channel parameters for the $T = 12$ case are the same as those used in [2, Section 4-B, Fig. 5]. We remark that for this choice of α, the contribution from failures due to small guard periods between bursts is not dominant. When the inter-burst gaps are smaller we believe that an extension of MiDAS codes that control the number of losses in such events may be necessary and is left for a future investigation.

All codes in Fig. 18(a) are selected to have a rate of $R = 12/23 \approx 0.52$ and the delay is $T = 12$. For reference the uncoded loss-rate is also shown by the upper-most dotted blue line marked with triangles. The black horizontal line is the loss rate of the m-MDS code. It achieves $B = N = 6$. Thus, its performance is limited by its burst-correction capability and thus is consistent with the probability of observing bursts longer than 6 which is given by $\approx 2 \times 10^{-5}$. The red-curve which deteriorates rapidly as we increase ε is the Maximally Short code (MS). It achieves $B = 11$ and $N = 1$. Thus, in general it cannot recover from even two losses occurring in a window of length $T+1$. The remaining curve marked with squares shows the MiDAS code which achieve $B = 9$ and $N = 2$. The loss probability also deteriorates with ε but at a much lower rate. Thus, a slight decrease in B, while improving N from 1 to 2 exhibits noticeable gains over both MS and m-MDS codes. At the left most point, i.e., when $\varepsilon = 10^{-3}$, the loss probability is dominated by burst losses, while as ε is increased, the effect of isolated losses becomes more significant. In Fig. 19(a), the rate of all codes is set to $R = 50/83 \approx 0.6$. The delay is set to $T = 50$. The m-MDS code (black horizontal plot) achieves $B = N = 20$ whereas the MS code (red plot) achieves $N = 1$ and $B = 33$. Both codes suffer from the same phenomenon discussed in the previous case. We also consider the MiDAS code (blue plot) with $N = 4$ and $B = 30$. We observe that its performance deteriorates as ε is increased and eventually crosses the m-MDS codes.

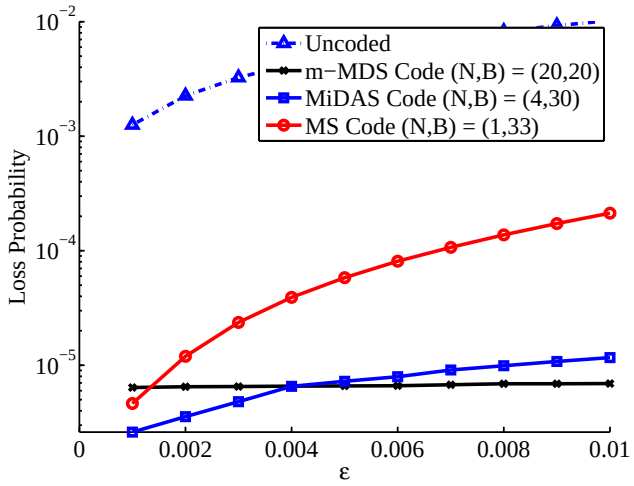


(a) Simulation results. All codes are evaluated using a decoding delay of $T = 12$ packets and a rate of $R = 12/23 \approx 0.52$.

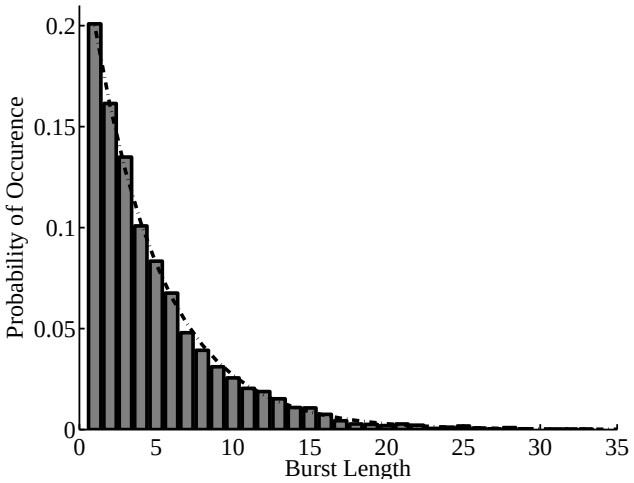


(b) The actual burst histogram (in bars) follows a geometric distribution (dotted line) with a success probability of $\beta = 0.5$.

Fig. 18: Simulation Experiments for Gilbert-Elliott Channel Model with $(\alpha, \beta) = (5 \times 10^{-4}, 0.5)$.



(a) Simulation results. All codes are evaluated using a decoding delay of $T = 50$ packets and a rate of $R = 50/83 \approx 0.6$.



(b) The actual burst histogram (in bars) follows a geometric distribution (dotted line) with a success probability of $\beta = 0.2$.

Fig. 19: Simulation Experiments for Gilbert-Elliott Channel Model with $(\alpha, \beta) = (5 \times 10^{-5}, 0.2)$.

We believe that despite the relatively large value of N, this performance deterioration is due to burst and isolated erasures being observed in the *same* decoding window. Such patterns, which occur during the transition period between good and bad states, are not covered in our sliding window erasure channel. We refer the reader to our follow-up work [30], [36], where the layered construction is exploited further to handle these patterns.

In Fig. 20 and Fig. 21, we evaluate streaming codes over the Fritchman channel in Fig. 17. The channel parameters and code parameters are shown in Table IX and X respectively. We let the transition probability from the good state to the first bad state E_1 to be α whereas the transition probability from each of the bad states equals β. Let ε be the probability of a packet loss in good state. We lose packets in any bad state with probability 1. Fig. 20(b) and 21(b) indicate the histogram of the burst lengths observed for the two channels.

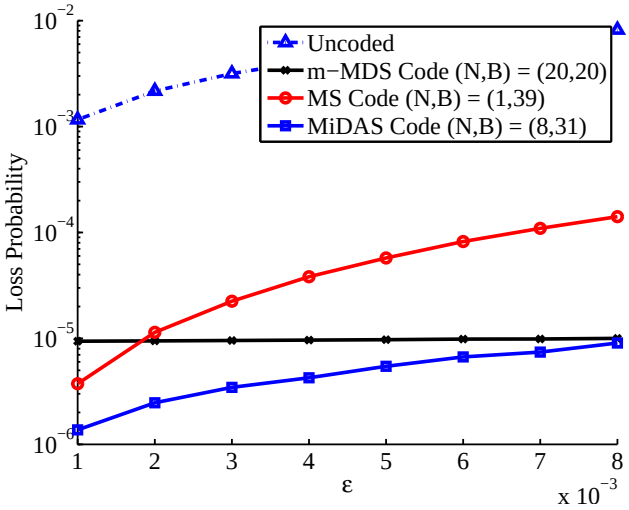
In Fig. 20 and Fig. 21, the uncoded loss rate is shown by the upper-most plot while the black horizontal line is the performance of m-MDS code. Note that the performance of this code is essentially independent of ε in the interval of interest. As in the case of GE channels, the m-MDS codes recover all the losses in the good state and fail against burst lengths longer than its burst erasure correction capability. Thus, their loss rate is consistent with the probability of observing bursts longer than 20 and 16 which can be calculated to be $\approx 10^{-5}$ and $\approx 3 \times 10^{-5}$, respectively. The performance of the MS codes is shown by the red-plot in both figures. We note that it is better than the m-MDS codes for $\varepsilon = 10^{-3}$, but deteriorates quickly as we increase ε. The performance gains from MiDAS codes are significantly more noticeable for the Fritchman channel because the hyper-geometric burst-

TABLE VII: Gilbert-Elliott Channel Parameters

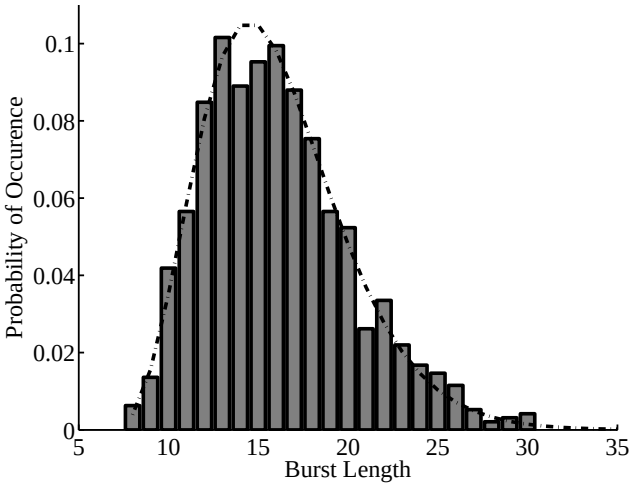
	Fig. 18(a)	Fig. 19(a)
Delay T	12	50
(α, β)	$(5 \times 10^{-4}, 0.5)$	$(5 \times 10^{-5}, 0.2)$
Channel Length	10^7	10^8
Rate R	$12/23$	$50/83$

TABLE VIII: Achievable N and B for different streaming codes

	Fig. 18(a)		Fig. 19(a)	
Code	N	B	N	B
MiDAS Code	2	9	4	30
m-MDS	6	6	20	20
MS Codes	1	11	1	33



(a) Simulation over a $\mathcal{N}+1 = 9$-States Fritchman Channel with $(\alpha, \beta) = (10^{-5}, 0.5)$. All codes are evaluated using a decoding delay of $T = 40$ packets and a rate of $R = 40/79 \approx 0.5$.



(b) The actual burst histogram (in bars) follows a negative binomial distribution (dotted line) with $\mathcal{N} = 8$ failures and a success probability of $\beta = 0.5$.

Fig. 20: Simulation Experiments for Fritchman Channel Model with $(\mathcal{N}, \alpha, \beta) = (8, 10^{-5}, 0.5)$.

TABLE IX: Fritchman Channel Parameters

	Fig. 20	Fig. 21
Channel States	9	12
Delay T	40	40
(α, β)	$(10^{-5}, 0.5)$	$(2 \times 10^{-5}, 0.75)$
Channel Length	10^8	10^8
Rate R	$40/79 \approx 0.5$	$40/67 \approx 0.6$

TABLE X: Achievable N and B for different streaming codes

	Fig. 20		Fig. 21	
Code	N	B	N	B
MiDAS Codes	8	31	4	24
m-MDS	20	20	16	16
MS Codes	1	39	1	27

TABLE XI: Unequal Source Channel Inter-arrival Rates

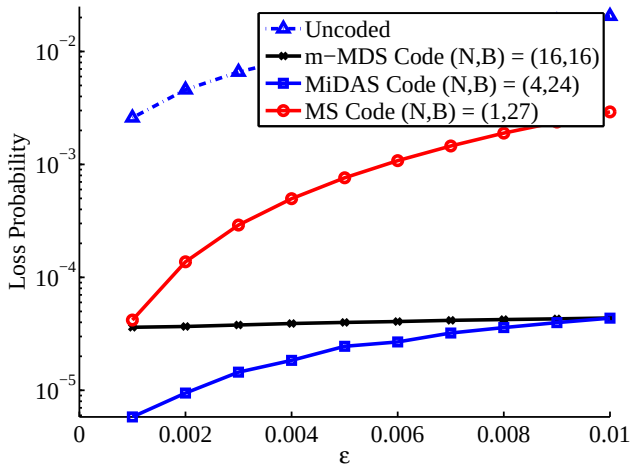
	Fig. 23	Fig. 24
Channel States	2	20
M	20	40
T	4	2
(α, β)	$(10^{-5}, [0.05, 0.15])$	$(10^{-5}, 0.5)$
Channel Length	10^9	10^9
Rate R	$9/14 \approx 0.64$	$40/63 \approx 0.63$

TABLE XII: Achievable N and B for different streaming codes

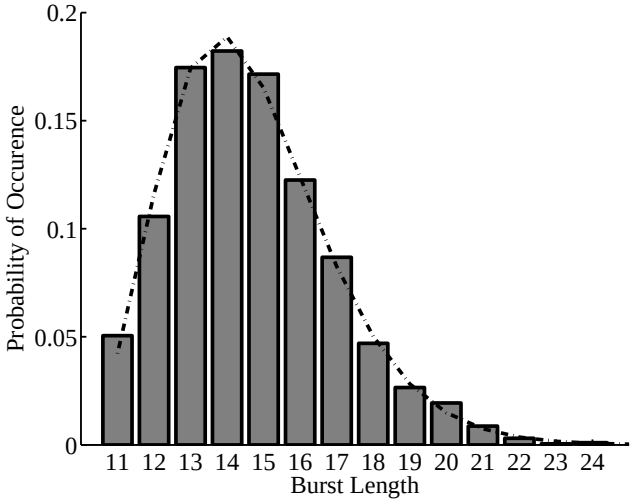
	Fig. 23		Fig. 24	
Code	N	B	N	B
Reshaped Code	1	50	1	58
Robust Reshaped Code	N/A	N/A	5	53
MiDAS Code	N/A	N/A	5	42
m-MDS Code	35	35	43	43
MS Codes	1	44	1	45

length distribution favors longer bursts over shorter ones. As in the case of GE Channels, we expect further performance gains to be possible by considering more sophisticated erasure patterns, such as burst plus isolated losses, but leave such an investigation for a future work.

In Fig. 22, we compare the performance of MiDAS and MS codes obtained by replacing the m-MDS constituent code with a diagonally interleaved block MDS code (cf. Section IV-D). We consider the same GE channel in Fig. 18(a) and delay $T = 12$. The codes involving m-MDS codes are plotted using a solid line whereas the codes involving block MDS codes are shown by the dotted lines of the same color. We note that in all cases there is a noticeable increase in the loss rate when a block MDS code is used despite the fact that these codes achieve the same (N, B) values over deterministic channels. This loss in performance is due to their sensitivity to non-ideal erasure patterns as discussed in Section IV-E.



(a) Simulation over a $\mathcal{N}+1 = 12$-States Fritchman Channel with $(\alpha, \beta) = (2\times10^{-5}, 0.75)$. All codes are evaluated using a decoding delay of $T = 40$ packets and a rate of $R = 40/67 \approx 0.6$.



(b) The actual burst histogram (in bars) follows a negative binomial distribution (dotted line) with $\mathcal{N} = 11$ failures and a success probability of $\beta = 0.75$.

Fig. 21: Simulation Experiments for Fritchman Channel Model with $(\mathcal{N}, \alpha, \beta) = (11, 2 \times 10^{-5}, 0.75)$.

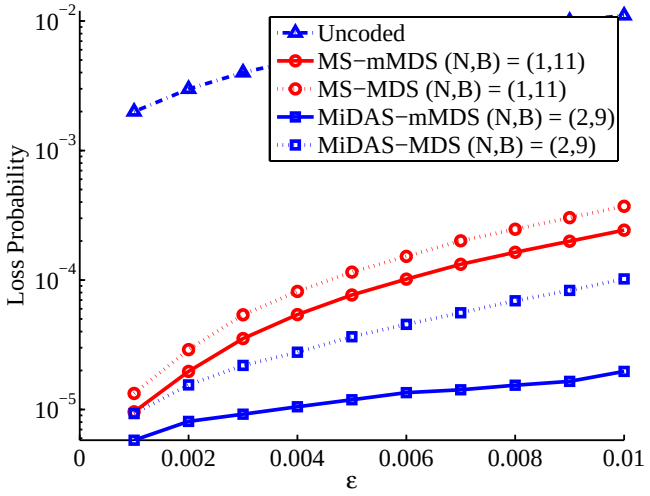


Fig. 22: Simulation over a Gilbert-Elliott Channel with $(\alpha, \beta) = (5 \times 10^{-4}, 0.5)$. All codes are evaluated using a decoding delay of $T = 12$ packets and a rate of $R = 12/23 \approx 0.52$.

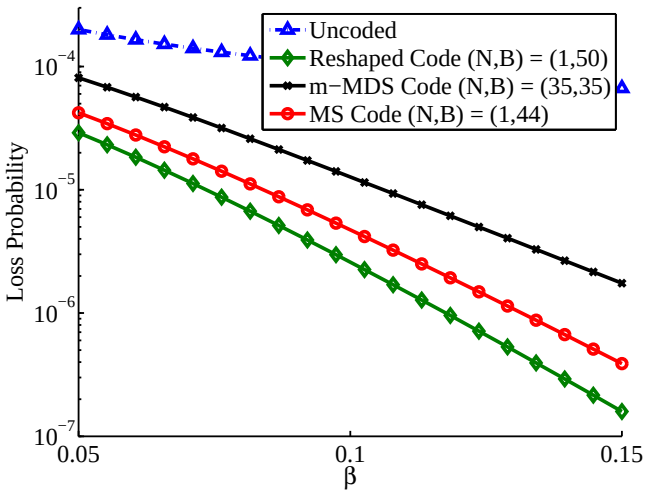
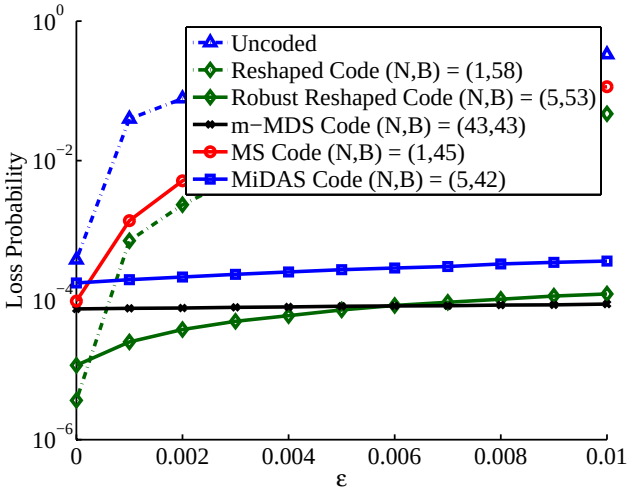


Fig. 23: Simulation over a Gilbert Channel with $\alpha = 10^{-5}$ and β varied on the x-axis. All codes are of rate $R = \frac{9}{14}$ and evaluated using a decoding delay of $T = 4$ macro-packets. Each macro-packet consists of $M = 20$ channel packets.

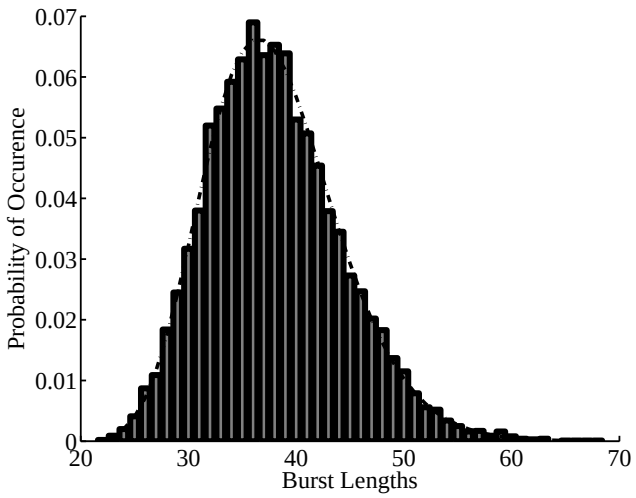
B. Unequal Source-Channel Inter-arrival Rates

In our simulations in Fig. 23, we consider a Gilbert channel model which is the same as a Gilbert-Elliott channel with $\varepsilon = 0$, i.e., the loss probability is 0 in the good state. We fix $\alpha = 10^{-5}$ and vary β on the x-axis in the interval $[0.05, 0.15]$ which in turn changes the burst length distribution. We further select $M = 20$, i.e., 20 channel packets are generated for every source packet received at the encoder. We fix the rate $R = 9/14$ and the delay $T = 4$ macro-packets. Under these conditions, the m-MDS code can correct burst erasures of length up to $B = 35$, whereas a Maximally Short code achieves $B = 44$. In contrast, for the optimal code we have $B = 50$. This gain in terms of correctable burst-length is reflected in Fig. 23 as one can see that codes designed for unequal source-channel inter-arrival rates, which are referred to as *reshaped codes*, achieve a lower loss probability. We note that the code parameters in Fig. 23 correspond to the second case in (9).

In Fig. 24, we consider a Fritchman channel with $(\alpha, \beta) = (10^{-5}, 0.5)$ and $\mathcal{N} + 1 = 20$ states. The corresponding burst distribution is illustrated in Fig. 24(b). In Fig. 24(a), we show the performance of different streaming codes in the case of unequal source-channel inter-arrival rates on such channel. The rate for all codes is fixed to $R = 0.64$ and the delay constraint is $T = 2$ macro-packets where each macro-packet has $M = 40$ packets. As the probability of erasure in the good state ε increases, the performance of m-MDS code (black curve) does



(a) Simulation results. All codes are evaluated using a decoding delay of $T = 2$ macro-packets and a rate of $R \approx 0.63$. Each macro-packet consists of $M = 40$ channel packets.



(b) The actual burst histogram (in bars) follows a negative binomial distribution (dotted line) with $\mathcal{N} = 20$ failures and a success probability of $\beta = 0.5$.

Fig. 24: Simulation Experiments for Fritchman Channel Model with $\mathcal{N} + 1 = 20$ states and $(\alpha, \beta) = (10^{-5}, 0.5)$.

not change. The loss rate of this code is $\approx 10^{-4}$ which is dominated by the fraction of erasures introduced by bursts longer than 43. On the other hand, both Maximally Short and reshaped codes achieve $N = 1$ and thus deteriorate as quickly as ε^2. For the left most point corresponding to $\varepsilon = 0$, the probability of loss of the Maximally Short code is $\approx 10^{-4}$ which reflects the number of erasures introduced by bursts longer than 45. Similarly, the loss probability of the reshaped code is $\approx 3 \times 10^{-6}$ which matches the fraction of losses introduced due to bursts longer than 58. The performance of the robust versions of these codes, namely MiDAS and robust reshaped codes does not deteriorate as fast. However, the robust reshaped code outperforms the MiDAS code as the former achieves $B = 53$ versus $B = 42$ achieved by the latter, when $N = 5$, $R = 0.63$ and $T = 2$.

VII. Conclusion

In this paper, we introduce a systematic approach for constructing low-delay error correction codes for real-time streaming communication over packet erasure channels that introduce both burst and isolated erasures. We introduce a class of sliding window erasure channels where the erasure sequences are constrained locally. Such models lead to a tractable analysis of the capacity and the resulting codes are observed to provide substantial gains in simulations over the Gilbert-Elliott and Fritchman channel models.

When the source-packet arrival rate and the channel-packet transmission rates are equal, we propose a near-optimal code construction, MiDAS Codes, using a layered coding approach that uses m-MDS codes and repetition codes as constituent codes. We also propose another class of codes that use block-MDS codes as constituent codes and require a considerably smaller field size. We establish the necessary and sufficient conditions on the column distance and column span of any feasible streaming code for the sliding-window erasure channel, and establish a fundamental tradeoff between these metrics which could be of independent interest.

For the case when the source-packet arrival rate and the channel-packet transmission rates are unequal, we characterize the capacity for the burst erasure channel. Our proposed codes are a non-trivial extension of the MiDAS codes and require a careful re-arrangement of the source symbols into the channel packets. We also present an achievable rate for the sliding window erasure channel with both burst and isolated erasures. Extensive numerical simulations indicate that our proposed constructions outperform traditional codes over statistical models.

We believe that the results in the paper are a promising first step towards construction of explicit error correction codes for real-time streaming applications over practical channel models. A number of further topics can be pursued, both from a theoretical viewpoint as well as practical viewpoint. On the theoretical side, the tradeoff between column distance and column span discovered in our analysis of the sliding-window erasure channel appears intriguing. It will be interesting to revisit it, perhaps using systems theoretic tools for convolutional codes [29], [37]. Furthermore, provable bounds on the achievable error probability over the Gilbert-Elliott and Fritchman channels, when using codes for the sliding-window erasure channel can be developed. In other directions, optimal streaming codes for the case when the source and channel rates are unequal and the channel introduces both burst and isolated erasures remain to be found. Furthermore, as noted in our simulations, improvements can be attained by considering streaming codes that correct both burst and isolated losses within the same decoding the window of interest. Finally our constructions are tuned to specific channel parameters. In practice, it is very desirable to extend such constructions that adapt to varying channel parameters with little or no feedback.

APPENDIX A
COLUMN DISTANCE AND COLUMN SPAN OF CONVOLUTIONAL CODES

In this section, we show that the error correction capability of a streaming code can be expressed in terms of its column distance and column span. In our discussion, we view the input packets $\mathbf{s}[i]$ as a length $\bar{k}$ vector over $\mathbb{F}_q$ and $\mathbf{x}[i]$ as a length $\bar{n}$ vector over $\mathbb{F}_q$. We restrict our attention to time-invariant linear $(\bar{n}, \bar{k}, \bar{m})$ convolutional codes specified by

$$\mathbf{x}[i] = \left(\sum_{j=0}^{\bar{m}} \mathbf{s}^{\dagger}[i-j]\mathbf{G}_j \right)^{\dagger},$$

where $\mathbf{G}_0, \ldots, \mathbf{G}_{\bar{m}}$ are generator matrices over $\mathbb{F}_q^{\bar{k}\times\bar{n}}$.

The first $T+1$ output packets can be expressed as,

$$[\mathbf{x}[0], \mathbf{x}[1], \ldots, \mathbf{x}[T]] = [\mathbf{s}[0], \mathbf{s}[1], \ldots, \mathbf{s}[T]] \cdot \mathbf{G}_T^s. \tag{74}$$

where

$$\mathbf{G}_T^s = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_T \\ 0 & \mathbf{G}_0 & & \mathbf{G}_{T-1} \\ \vdots & & \ddots & \vdots \\ 0 & & \ldots & \mathbf{G}_0 \end{bmatrix} \tag{75}$$

is the truncated generator matrix to the first $T+1$ columns. Note that $\mathbf{G}_j = 0$ if $j > m$.

Definition 3 (Column Distance)**.** *The column distance of $\mathbf{G}_T^s$ in* (75) *is defined as*

$$d_T = \min_{\substack{\mathbf{s}\equiv[\mathbf{s}[0],\mathbf{s}[1],\ldots,\mathbf{s}[T]] \\ \mathbf{s}[0]\neq 0}} \mathrm{wt}([\mathbf{x}[0], \ldots, \mathbf{x}[T]]) \tag{76}$$

where $\mathrm{wt}([\mathbf{x}[0], \ldots, \mathbf{x}[T]])$ counts the number of non-zero elements in the $T+1$ length vector.

Intuitively, the column distance of the convolutional code finds the codeword sequence of minimum Hamming weight in the interval $[0, T]$ that diverges from the all zero state at time $t = 0$. We refer the reader to [29, Chapter 3] for some properties of d_T.

Fact 1. *A convolutional code with a column distance of d_T can recover every information packet with a delay of T provided the channel introduces no more than $N = d_T - 1$ erasures in any sliding window of length $T+1$. Conversely there exists at-least one erasure pattern with d_T erasures in a window of length $T+1$ where the decoder fails to recover all source packets.*

Proof: Consider the interval $[0, T]$ and consider two input sequences $(\mathbf{s}[0], \ldots, \mathbf{s}[T])$ and $(\mathbf{s}'[0], \ldots, \mathbf{s}'[T])$ with $\mathbf{s}[0] \neq \mathbf{s}'[0]$. Let the corresponding output be $(\mathbf{x}[0], \ldots, \mathbf{x}[T])$ and $(\mathbf{x}'[0], \ldots, \mathbf{x}'[T])$. Note that the output sequences differ in at-least d_T indices since otherwise the output sequence $(\mathbf{x}[0] - \mathbf{x}'[0], \ldots, \mathbf{x}[T] - \mathbf{x}'[T])$ which corresponds to $(\mathbf{s}[0] - \mathbf{s}'[0], \ldots, \mathbf{s}[T] - \mathbf{s}'[T])$ has a Hamming weight less than d_T while the input $\mathbf{s}[0] - \mathbf{s}'[0] \neq 0$, which is a contradiction. Thus, if $(\mathbf{s}[0], \ldots, \mathbf{s}[T])$ is the input source sequence, for any sequence of $d_T - 1$ or fewer erasures, there will be at-least one packet where $(\mathbf{x}'[0], \ldots, \mathbf{x}'[T])$ differs from the received sequence. Thus, $\mathbf{s}[0]$ is recovered uniquely at time T. Once $\mathbf{s}[0]$ is recovered we can cancel its contribution from all the future packets and repeat the same argument for the interval $[1, T+1]$ to recover $\mathbf{s}[1]$ and proceed.

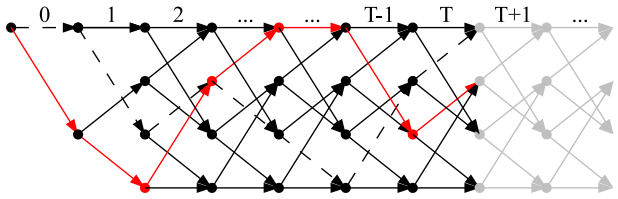


Fig. 25: Trellis diagram showing a streaming code. Since the delay constraint is T packets, the corresponding generator matrix is truncated at time T (cf. (74)) and hence nodes after time T are irrelevant. Also, according to the column distance and column span definitions in Def. 3 and Def. 4, only paths diverging from all zero path at the starting point are considered to calculate these metrics. The red solid path is one example of these paths while the dashed line is not considered.

Conversely there exists at-least one output sequence whose Hamming weight equals d_T and the input packet $\mathbf{s}[0] \neq 0$. By erasing all the non-zero d_T positions for this output sequence, we cannot distinguish it from the all-zero sequence. ■

To the best of our knowledge the column span of a convolutional code was first introduced in [2] in the context of low-delay codes for burst erasure channels.

Definition 4 (Column Span)**.** *The column span of $\mathbf{G}_T^s$ in* (75) *is defined as*

$$c_T = \min_{\substack{\mathbf{s}\equiv[\mathbf{s}[0],\mathbf{s}[1],\ldots,\mathbf{s}[T]] \\ \mathbf{s}[0]\neq 0}} \mathrm{span}([\mathbf{x}[0], \ldots, \mathbf{x}[T]]) \tag{77}$$

where $\mathrm{span}([\mathbf{x}[0], \ldots, \mathbf{x}[T]])$ equals the support of the underlying vector, i.e., $\mathrm{span}([\mathbf{x}[0], \ldots, \mathbf{x}[T]]) = j - i + 1$, where j is the last index where $\mathbf{x}$ is non-zero and i is the first such index.

Fact 2. *Consider a channel that introduces no more than a single erasure burst of maximum length B in any sliding window of length $T+1$. A necessary and sufficient condition for a convolutional code to recover every erased packet with a delay of T is that $c_T > B$.*

The justification is virtually identical to the proof of Fact 1 and is omitted.

It follows from Facts 1 and 2 that a necessary and sufficient condition for any convolutional code to recover each source packet with a delay of T over a channel $\mathcal{C}(N, B, W = T+1)$ is that both $d_T > N$ and $c_T > B$. Thus, specializing Theorem 1 and 2 to $W = T+1$ we are now able to prove Prop. 1, which is stated below for convenience.

Proposition (A Fundamental Tradeoff between Column Distance and Column Span)**.** *For any $(\bar{n}, \bar{k}, \bar{m})$ convolutional code and an integer $T > 0$ we have that the column distance*

d_T and column span c_T must satisfy

$$\frac{R}{1-R}c_T + d_T \leq T + 1 + \frac{1}{1-R} \tag{78}$$

where $R = \frac{\bar{k}}{\bar{n}}$ denotes the rate of the code. Furthermore, for any $T > 0$ there exists a $(\bar{n}, \bar{k}, \bar{m})$ convolutional code with column distance d_T and column span c_T, over a sufficiently large field-size such that,

$$\frac{R}{1-R}c_T + d_T \geq T + \frac{1}{1-R} \tag{79}$$

□

Proof: To establish (78), consider any convolutional code with a column distance d_T and column span c_T. From the sufficiency parts of Facts 1 and 2 such a code is feasible over the channel $\mathcal{C}(N = d_T - 1, B = c_T - 1, W = T + 1)$ with delay T. Thus, it must satisfy the upper bound (1). Substituting $N = d_T - 1$ and $B = c_T - 1$ immediately gives (78).

To establish (79), consider the code that satisfies the lower bound in (2) in Theorem 2. From the necessity parts of Facts 1 and 2 such a code must satisfy $c_T \geq B + 1$ and $d_T \geq N + 1$. Substituting in (2) immediately leads to (79).

■

As a final remark we note Facts 1 and 2 also immediately apply to any channel with $W \geq T + 1$. In particular, any erasure pattern for the $\mathcal{C}(N, B, W)$ channel with $W \geq T+1$ is also feasible for $\mathcal{C}(N, B, W = T+1)$ and thus the sufficiency follows. Furthermore, note that whenever $W \geq T + 1$, any erasure pattern in the interval $[0, T]$ used in the proof of the necessity part can also be used for the channel $\mathcal{C}(N, B, W)$.

Appendix B
Proof of Lemma 1

In order to establish L1, we use the following property regarding systematic m-MDS codes [5, Corollary 2.5]. Consider the window of the first $j + 1$ packets of a $(\bar{n}, \bar{k}, \bar{m})$ convolutional code and let the truncated codeword associated with the input sequence $(\mathbf{s}[0], \ldots, \mathbf{s}[j])$ be $(\mathbf{x}[0], \ldots, \mathbf{x}[j])$, where each $\mathbf{x}[i]$ is expressed as in (14). Then the j-th (symbol level) column distance[12] is defined as

$$d_j^c = \min_{\substack{\mathbf{s} \equiv (\mathbf{s}[0], \ldots, \mathbf{s}[j]) \\ \mathbf{s}[0] \neq 0}} \mathrm{wt}^c(\mathbf{x}[0], \ldots, \mathbf{x}[j]), \tag{80}$$

where recall that each channel packet $\mathbf{x}[i]$ has $\bar{n}$ symbols, i.e., $\mathbf{x}[i] = (x_0[i], \ldots, x_{\bar{n}-1}[i])$ and $\mathrm{wt}^c(\mathbf{v})$ counts the number of non-zero **symbols** in the codeword $\mathbf{v}$.

It is well-known that for any $(\bar{n}, \bar{k}, \bar{m})$ convolutional code $d_j^c \leq (\bar{n} - \bar{k})(j + 1) + 1$ for all $j \geq 0$. A special class of convolutional codes – systematic m-MDS codes – satisfy this bound with equality for $j = \{0, \ldots, \bar{m}\}$ [5, Corollary 2.5].

The proof of property L1 follows by using an argument similar to that in the proof of Fact 1 in Appendix A. We will omit it as the argument is completely analogous.

To establish L2, we use the notation $\mathcal{W}_i(l)$ to denote a window of length $l \cdot \bar{n}$ starting at time $i \cdot \bar{n}$, i.e., $\mathcal{W}_i(l) \triangleq [i \cdot \bar{n}, (i + l)\bar{n} - 1]$ (see Fig. 26). We show that the entire erasure burst can be recovered through the following steps.

- In the window $\mathcal{W}_0(j+1) = [0, (j+1)\bar{n}-1]$, the channel introduces $\hat{B} \leq (\bar{n}-\bar{k})(j+1)$ erasures. Hence, we use L1 to recover $\mathbf{s}[0] = (s_0[0], \ldots, s_{\bar{k}-1}[0])$ at time $(j+1)n-1$ among which only the last $\bar{k} - c$ symbols are erased. At this point we can also compute the $\bar{n} - \bar{k}$ symbols of $\mathbf{p}[0] = (p_0[0], \ldots, p_{\bar{n}-\bar{k}-1}[0])$. Thus, all the symbols until time $t = \bar{n} - 1$ have now been recovered by the decoder.
- The next window $\mathcal{W}_1(j) = [\bar{n}, (j + 1)\bar{n} - 1]$ has $\hat{B} - (\bar{n} - c) < j(\bar{n} - \bar{k})$ erasures since $c < \bar{k}$. Hence, L1 can be used to recover $\mathbf{s}[1]$ at time $(j+1)\bar{n}-1$ and $\mathbf{p}[1]$ can be computed consequently.
- Similarly, $\mathcal{W}_2(j-1) = [2\bar{n}, (j+1)\bar{n}-1]$ has $\hat{B}-(\bar{n}-c)-\bar{n} < (\bar{n} - \bar{k})(j - 1)$ erasures which implies the recovery of $\mathbf{s}[2]$ at time $(j + 1)\bar{n} - 1$.
- Repeating the previous step for $\mathcal{W}_i(j-i+1) = [i \cdot \bar{n}, (j+1)\bar{n}-1]$ and $i \cdot \bar{n} \leq c + \hat{B} - 1$, one can recover all erased packets in the erasure burst at time $(j + 1)\bar{n} - 1$.

The proof of L2 is thus complete. The claim in L3 is a generalization of L2, as it permits the erasure pattern to have both burst and isolated erasures, but only guarantees the recovery of the burst erasure. To establish L3 we can proceed in a similar fashion as above and stop when the recovery of the erasure burst is complete.

Appendix C
Decoding Analysis of MiDAS code with MDS constituent codes

In the decoding analysis, it is sufficient to show that each source packet $\mathbf{s}[i]$ can be recovered at time $t = i + T_{\text{eff}}$ if there is either an erasure burst of length B or up to N isolated erasures in the interval $[i, i + T_{\text{eff}}]$.

A. Burst Erasure

First consider the case when a burst erasure spans $[i, i + B - 1]$. Following this burst, we are guaranteed that for the $\mathcal{C}(N, B, W)$ channel, there are no erasures in the interval $[i + B, i+T_{\text{eff}}+B-1]$. We argue that the decoder can first recover $\mathbf{v}[i], \ldots, \mathbf{v}[i+B-1]$ simultaneously by time $t = i+T_{\text{eff}}-1$ and then recover $\mathbf{u}[i]$ at time $t = i+T_{\text{eff}}$ by computing $\mathbf{p}^v[i+T_{\text{eff}}]$ and then $\mathbf{u}[i] = \mathbf{q}[i+T_{\text{eff}}]-\mathbf{p}^v[i+T_{\text{eff}}]$. To show the recovery of $\mathbf{v}[i], \ldots, \mathbf{v}[i + B - 1]$, note that there are no erasures in the interval spanning $[i + B, i + T_{\text{eff}} - 1]$ and the interfering $\mathbf{u}[\cdot]$ sub-packets in $\mathbf{q}[t] = \mathbf{u}[t - T_{\text{eff}}] + \mathbf{p}^v[t]$ can be subtracted out to recover $\mathbf{p}^v[t]$. The diagonal codewords $\left\{\mathbf{c}_j^v[r]\right\}$ spanning $\mathbf{v}[i], \ldots, \mathbf{v}[i + B - 1]$ start at $r \in \{i - (T_{\text{eff}} - B) + 1, \ldots, i + B - 1\}$. Each such codeword belongs to a $(T_{\text{eff}}, T_{\text{eff}} - B)$ MDS code. Hence, if no more than B erasures take place in each codeword, the erased packets can be recovered. However, we still need to take the delay into account. We first note that the $\mathbf{v}[\cdot]$ sub-packets in the interval $[i, i + B - 1]$ are erased. Also, the $\mathbf{q}[\cdot]$ packets in the interval $[i+T_{\text{eff}}, i+T_{\text{eff}}+B-1]$ combine $\mathbf{u}[\cdot]$ which are erased and thus the corresponding $\mathbf{p}[\cdot]$ packets must also be treated as erased. We split the diagonal codewords of interest into two groups,

[12]This differs from (76) in that we measure the Hamming weight of symbols rather than the packets $\mathbf{x}[j]$.

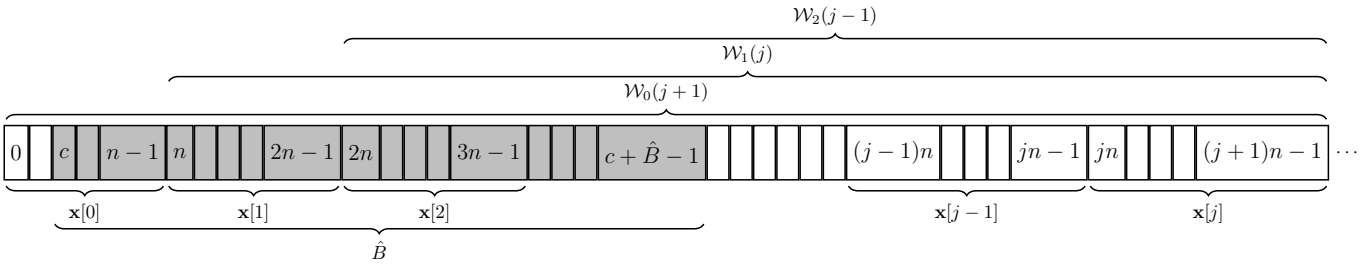


Fig. 26: An erasure channel with $\hat{B}$ erasures in a burst starting at time c used in proving L2 in Lemma 1. Grey and white squares resemble erased and unerased sub-symbols respectively.

- $r \in \{i-(T_{\text{eff}}-B)+1,\ldots,i\}$: These codewords end before time $i+T_{\text{eff}}$ where there are only B erased columns in the interval $[i, i+B-1]$.
- $r \in \{i+1,\ldots,i+B-1\}$: Each of these codewords has $T_{\text{eff}}-B$ symbols in the interval $[i+B, i+T_{\text{eff}}-1]$ which are not erased. Since the length of each codeword is T_{eff}, then the number of erasures are at most $T_{\text{eff}}-(T_{\text{eff}}-B) = B$.

We note that all the considered diagonal codewords $\left\{\mathbf{c}_j^v[r]\right\}$ for $r \in \{i-(T_{\text{eff}}-B)+1,\ldots,i+B-1\}$ end before time $i+T_{\text{eff}}+B$. Also, the $\mathbf{p}[\cdot]$ parities in the interval $[i+T_{\text{eff}}, i+T_{\text{eff}}+B-1]$ cannot be used as discussed earlier. Thus, it follows that the corresponding $\mathbf{v}[\cdot]$ sub-packets are recovered by $i+T_{\text{eff}}-1$.

B. Isolated Erasures

Next we show that when there are N erasures in arbitrary locations in the interval $[i, i+T_{\text{eff}}]$, then $\mathbf{u}[i]$ is guaranteed to be recovered by time $t = i+T_{\text{eff}}$, and $\mathbf{v}[i]$ is guaranteed to be recovered by time $t = i+T_{\text{eff}}-1$. For the recovery of $\mathbf{u}[i]$ we note that the codewords $\left\{\mathbf{c}_j^u[r]\right\}$ that include $\mathbf{u}[i]$ start at $r \in \{i-(T_{\text{eff}}-N),\ldots,i\}$. Since each $\mathbf{c}_j^u[r]$ is a $(T_{\text{eff}}+1, T_{\text{eff}}-N+1)$ MDS code, and there are no more than N erasures on each such sequence, it follows that all the erased packets are guaranteed to be recovered by time $i+T_{\text{eff}}$. The recovery of $\mathbf{u}[i]$ by time $t = i+T_{\text{eff}}$ now follows. For recovering $\mathbf{v}[i]$, we consider the non-erased parity-check packets $\mathbf{p}^v[t]$ for $t \in [i, i+T_{\text{eff}}-1]$, which can be obtained by cancelling the interfering $\mathbf{u}[t-T_{\text{eff}}]$ sub-packets from $\mathbf{q}[t]$ as discussed in the case of burst erasure above. Notice that the diagonal codewords $\left\{\mathbf{c}_j^v[r]\right\}$ spanning $\mathbf{v}[i]$ start at $r \in \{i-(T_{\text{eff}}-B)+1,\ldots,i\}$ and terminate by time $i+T_{\text{eff}}-1$. It follows that each such sequence has no more than N erasures and hence all the erased $\mathbf{v}[i]$ sub-packets are recovered by time $t = i+T_{\text{eff}}-1$.

APPENDIX D
PROOF OF LEMMA 3

We need to show that the total erased symbols $(\mathbf{v}_{\text{vec}}[\cdot], \mathbf{p}_{\text{vec}}[\cdot])$ between macro-packets i to macro-packet $i+T$, i.e., in the following sequence,

$$\begin{aligned}\left\{\mathbf{v}_{\text{vec}}[t], \mathbf{p}_{\text{vec}}[t]\right\}_{i\le t\le i+T} = \\ \Big(v_0[i],\ldots,v_{k^v-1}[i], p_0[i],\ldots,p_{k^u-1}[i], \\ v_0[i+1],\ldots,v_{k^v-1}[i+1], p_0[i+1],\ldots,p_{k^u-1}[i+1] \\ ,\ldots, \\ v_0[i+T],\ldots,v_{k^v-1}[i+T], p_0[i+T],\ldots,p_{k^u-1}[i+T]\Big),\end{aligned} \tag{81}$$

after the cancellation of $\mathbf{u}[\cdot]$ sub-packets does not exceed $k^u(T+1)$. We start by considering that the burst begins at $j=1$ and subsequently consider other cases in Table XIII. For the case when $j=1$ we consider two cases.

- $B' > \frac{b}{T+b}M$: We first show that the total number symbols erased in $\mathbf{v}_{\text{vec}}[\cdot]$ and $\mathbf{p}_{\text{vec}}[\cdot]$ due to the burst in the macro-packets $i, i+1,\ldots,i+b$ equals $k^uT = B\cdot T$. Furthermore, in macro-packet $i+T$, the parity-checks $\mathbf{p}_{\text{vec}}[i+T]$ combine with $\mathbf{u}_{\text{vec}}[i]$ which are also erased. Hence, these symbols contribute to additional k^u erasures leading to a total of $k^u(T+1)$ erased symbols.
  Note that the erasure burst spans the entire macro-packets $\mathbf{X}[i,:],\ldots,\mathbf{X}[i+b-1,:]$ as well as $\mathbf{x}[i+b,1],\ldots,\mathbf{x}[i+b,B']$. The total number of symbols in $\mathbf{v}_{\text{vec}}[t]$ and $\mathbf{p}_{\text{vec}}[t]$ in each macro-packet is $k^v+k^u = M(T+b+1)-B$. In the b-th macro-packet, we only have the first B' columns erased. Out of these, the first k^u symbols are from the $\mathbf{u}_{\text{vec}}[\cdot]$ sub-packets whereas the remaining $B'n-k^u = B'(T+b+1)-B$ symbols come from $\mathbf{v}_{\text{vec}}[i+b]$ and $\mathbf{p}_{\text{vec}}[i+b]$. It can be easily verified that $B'(T+b+1)-B \ge 0$. Hence, the total number of erased symbols of $\mathbf{v}_{\text{vec}}[t]$ and $\mathbf{p}_{\text{vec}}[t]$ is

$$\begin{aligned}&b(k^u+k^v)+B'n-k^u \\ &\quad= b(M(T+b+1)-B)+B'(T+b+1)-k^u \\ &\quad= B(T+b+1)-bB-k^u \\ &\quad= B(T+1)-k^u = Tk^u\end{aligned} \tag{82}$$

  where we use the fact that $k^u = B$ in our code construction and $B = bM+B'$.
- $B' \le \frac{b}{T+b}M$: Again the macro-packets $\mathbf{X}[i,:],\ldots,\mathbf{X}[i+b-1,:]$ are completely erased and each contributes to $b(k^v+k^u) = b(M(T+b)-Mb)$ erasures. In the $\mathbf{X}[i+b,:]$, only the symbols in $\mathbf{u}_{\text{vec}}[i+b]$ are erased as it can be

TABLE XIII: Different erasure patterns considered in the analysis of the decoder. The index j at the left of each row, indicates the starting location of each burst in macro-block i. The shaded blocks shows the symbols that are erased.

$j=1$	$\mathbf{u}[i,1].$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$
$j=2$	$\mathbf{u}[i,1]$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$
$j=r+1$	$\mathbf{u}[i,1]$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$
$j=r+2$	$\mathbf{u}[i,1]$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$
$j=r+3$	$\mathbf{u}[i,1]$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$
$j=M-r$	$\mathbf{u}[i,1]$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$
$j=M-r+1$	$\mathbf{u}[i,1]$	$\mathbf{u}[i,2]$	$\cdots$	$\mathbf{u}[i,r]$	$\mathbf{u}[i,r+1]$ $\mathbf{v}[i,1]$	$\mathbf{v}[i,2]$	$\mathbf{v}[i,3]$	$\cdots$	$\mathbf{v}[i,M-2r-1]$	$\mathbf{q}[i,r+1]$ $\mathbf{v}[i,M-2r]$	$\mathbf{q}[i,r]$	$\cdots$	$\mathbf{q}[i,1]$

easily verified that $B'n \leq k^u$. Finally, as in the previous case all the symbols in $\mathbf{v}_{\text{vec}}[i+T]$ in macro-packet $i+T$ that combine with $\mathbf{u}_{\text{vec}}[i]$ must be considered erased. Thus, the total number of erased symbols is $b(M(T+b)-Mb)+k^u = bMT+k^u = k^u(T+1)$.

To establish the claim for $j=2,3,\ldots,M-r$ it suffices to show the following lemma

Lemma 4. *Let N_j denote the total number of erased symbols in $\{\mathbf{v}_{\text{vec}}[t],\mathbf{p}_{\text{vec}}[t]\}$ after the cancellation of non-erased $\mathbf{u}_{\text{vec}}[\cdot]$ sub-packets when the erasure burst begins at $\mathbf{x}[i,j]$. Then we have that $N_j \leq N_{j-1}$ for each $j=2,3,\ldots,M-r$.*

Lemma 4 establishes that the worst case erasure sequence is the one that begins at $j=1$. Since we have already established that the total number of erasures in $\{\mathbf{v}_{\text{vec}}[t],\mathbf{p}_{\text{vec}}[t]\}$ in this case does not exceed $k^u(T+1)$, this will complete our claim.

To establish Lemma 4, we note that going from the burst pattern that starts at $\mathbf{x}[i,j]$ to the pattern that start at $\mathbf{x}[i,j+1]$ results in one extra erased channel packet at the end. Also, it results in revealing the first channel packet which is $\mathbf{x}[i,j]$. We assume (as a worst case) that the extra erased channel packet at the end contributes to n additional erased symbols of either $\mathbf{v}_{\text{vec}}[\cdot]$ or $\mathbf{p}_{\text{vec}}[\cdot]$. We consider the effect of revealing the channel packet $\mathbf{x}[i,j]$ and show that it always compensates exactly n new unerased packets of either $\mathbf{v}_{\text{vec}}[\cdot]$ or $\mathbf{p}_{\text{vec}}[\cdot]$. Thus, we do not increase the total number of erased packets in such a transition.

Recall that $\mathbf{x}[i,j]$ can be one of the following (cf. Table XIII),

$$\mathbf{x}[i,j] = \begin{cases} \mathbf{u}[i,j] & j=\{1,\ldots,r\} \\ \begin{bmatrix} \mathbf{u}[i,r+1] \\ \mathbf{v}[i,1] \end{bmatrix} & j=r+1 \\ \mathbf{v}[i,j-r] & j=\{r+2,\ldots,M-r-1\} \\ \begin{bmatrix} \mathbf{q}[i,r+1] \\ \mathbf{v}[i,j-r] \end{bmatrix} & j=M-r \\ \mathbf{q}[i,M-j+1] & j=\{M-r+1,\ldots,M\}. \end{cases} \tag{83}$$

- $j=\{1,\cdots,r\}$: In the case under consideration, the revealed $\mathbf{x}[i,j]$ is always $\mathbf{u}[i,j]$. It can be subtracted from $\mathbf{q}[i+T,j]$ to recover $\mathbf{p}[i+T,j] \in \mathbf{p}_{\text{vec}}[\cdot]$ having n symbols. Thus, it compensates for the n extra erased symbols.
- $j=r+1$: The r' symbols of $\mathbf{u}[i,r+1]$ helps in recovering the r' symbols of $\mathbf{p}[i+T,r+1] \in \mathbf{p}_{\text{vec}}[\cdot]$. This together with the revealed $n-r'$ symbols of $\mathbf{v}[i,1] \in \mathbf{v}_{\text{vec}}[\cdot]$ compensates for the n extra erasures.
- $j=\{r+2,\ldots,M-r-1\}$: In this case, the revealed channel packet is $\mathbf{x}[i,j]=\mathbf{v}[i,j-r] \in \mathbf{v}_{\text{vec}}[\cdot]$ and has n symbols which are now available at the decoder.
- $j=M-r$: As shown in Table XIII, the decoder can subtract $\mathbf{u}[i-T,r+1]$ from $\mathbf{q}[i,r+1]$ to recover the r' symbols $\mathbf{p}[i,r+1] \in \mathbf{p}_{\text{vec}}[\cdot]$. This together with the $n-r'$ symbols of $\mathbf{v}[i,j-r] \in \mathbf{v}_{\text{vec}}[\cdot]$ add up to n symbols and the claim follows.

This establishes Lemma 4 and in turn the proof of Lemma 3 is complete.

REFERENCES

[1] T. Huang, P. Huang, K. Chen, and P. Wang. Could Skype be more satisfying? a QoE-centric study of the FEC mechanism in an internet-scale VoIP system. *IEEE Network*, 24(2):42–48, Feb 2010.

[2] E. Martinian and C. W. Sundberg. Burst erasure correction codes with low decoding delay. *IEEE Transactions on Information Theory*, 50(10):2494–2502, 2004.

[3] E. Martinian and M. Trott. Delay-optimal burst erasure code construction. In *Proc. International Symposium on Information Theory (ISIT)*, Nice, France, July 2007.

[4] E. M. Gabidulin. Convolutional codes over large alphabets. In *Proc. International Workshop on Algebraic Combinatorial and Coding Theory*, pages 80–84, Varna, Bulgaria, 1988.

[5] H. Gluesing-Luerssen, J. Rosenthal, and R. Smarandache. Strongly-MDS convolutional codes. *IEEE Transactions on Information Theory*, 52(2):584–598, 2006.

[6] A. Khisti and J. Singh. On multicasting with streaming burst-erasure codes. In *Proc. International Symposium on Information Theory (ISIT)*, 2009.

[7] A. Badr, A. Khisti, and E. Martinian. Diversity Embedded Streaming erasure Codes (DE-SCo): Constructions and optimality. *IEEE Journal on Selected Areas in Communications (JSAC)*, 29(5):1042–1054, May 2011.

[8] A. Badr, A. Khisti, and E. Martinian. Diversity Embedded Streaming erasure Codes (DE-SCo): Constructions and optimality. In *Proc. Global Communications Conference (GLOBECOM)*, pages 1–5, 2010.

[9] A. Badr, D. Lui, and A. Khisti. Streaming-codes for multicast over burst erasure channels. *IEEE Transactions on Information Theory*, 2014, To Appear.

[10] A. Badr, D. Lui, and A. Khisti. Multicast Streaming Codes (Mu-SCo) for burst erasure channels. In *Proc. Allerton Conference on Communication, Control, and Computing*, 2010.

[11] E. Martinian. *Dynamic Information and Constraints in Source and Channel Coding*. PhD thesis, Massachusetts Institute of Technology (MIT), 2004.

[12] D. Lui, A. Badr, and A. Khisti. Streaming codes for a double-link burst erasure channel. In *Proc. Canadian Workshop on Information Theory (CWIT)*, 2011.

[13] D. Lui. Coding theorems for delay sensitive communication over burst-erasure channels. Master's thesis, University of Toronto, Toronto, ON, August 2011.

[14] Z. Li, A. Khisti, and B. Girod. Correcting erasure bursts with minimum decoding delay. In *Proc. Asilomar Conference on Signals, Systems & Computers*, 2011.

[15] O. Tekin, T. Ho, H. Yao, and S. Jaggi. On erasure correction coding for streaming. In *Information Theory and Applications Workshop (ITA)*, pages 221–226, 2012.

[16] D. Leong and T. Ho. Erasure coding for real-time streaming. In *Proc. International Symposium on Information Theory (ISIT)*, 2012.

[17] D. Leong, A. Qureshi, and T. Ho. On coding for real-time streaming under packet erasures. In *Proc. International Symposium on Information Theory (ISIT)*, 2013.

[18] H. S. Witsenhausen. On the structure of real-time source coders. *Bell Syst. Tech. J.*, 58(6):1437–1451, Jul-Aug 1979.

[19] D. Teneketzis. On the structure of optimal real-time encoders and decoders in noisy communication. *IEEE Transactions on Information Theory*, 52(9):4017–4035, sep 2006.

[20] T. Javidi and A. Goldsmith. Dynamic joint source-channel coding with feedback. In *Proc. International Symposium on Information Theory (ISIT)*, 2013.

[21] L. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

[22] A. Sahai. *Anytime Information Theory*. PhD thesis, Massachusetts Institute of Technology (MIT), 2001.

[23] R. Sukhavasi and B. Hassibi. Linear error correcting codes with anytime reliability. In *Proc. International Symposium on Information Theory (ISIT)*, pages 1748–1752, 2011.

[24] G. Joshi, Y. Kochman, and G. W. Wornell. On playback delay in streaming communication. In *ISIT*, pages 2856–2860, 2012.

[25] Y. Li and E. Soljanin. Rateless codes for single-server streaming to diverse users. *CoRR*, abs/0912.5055, 2009.

[26] H. Yao, Y. Kochman, and G. W. Wornell. A multi-burst transmission strategy for streaming over blockage channels with long feedback delay. *IEEE Journal on Selected Areas in Communications (JSAC)*, 29(10):2033–2043, 2011.

[27] S. Mehrotra, J. Li, and Y. Huang. Optimizing fec transmission strategy for minimizing delay in lossless sequential streaming. *IEEE Transactions on Multimedia*, 13(5):1066–1076, 2011.

[28] S. Kokalj-Filipovic, P. Spasojevic, and E. Soljanin. Doped Fountain coding for minimum delay data collection in circular networks. *CoRR*, abs/1001.3765, 2010.

[29] R. Johannesson and K. Zigangirov. *Fundamentals of Convolutional Coding*. Wiley-IEEE Press, 1999.

[30] A. Badr. *Error-Correcting Codes for Low-Delay Streaming Communications*. PhD thesis, University of Toronto, 2014.

[31] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.

[32] Y. Li, P. Vingelmann, M. Pedersen, and E. Soljanin. Round-robin streaming with generations. *CoRR*, abs/1206.3014, 2012.

[33] A. Badr, A. Khisti, W. Tan, and J. Apostolopoulos. Streaming codes for channels with burst and isolated erasures. In *Proc. International Conference on Computer Communications (INFOCOM)*, 2013.

[34] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Systems Technical Journal*, 39:1253–1265, September 1960.

[35] B. D. Fritchman. A binary channel characterization using partitioned markov chains. *IEEE Transactions on Information Theory*, 13:221–227, 1967.

[36] A. Badr, A. Khisti, W. Tan, and J. Apostolopoulos. Streaming Codes with Partial Recovery over Channels with Burst and Isolated Erasures. *submitted to IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 2014.

[37] Jr. G. D. Forney. Convolutional codes I: Algebraic structure. *IEEE Transactions on Information Theory*, 16:720 – 738, November 1970.