Explicit Rate-Optimal Streaming Codes with Smaller Field Size

Myna Vajha*, Vinayak Ramkumar*, M. Nikhil Krishnan†, P. Vijay Kumar*

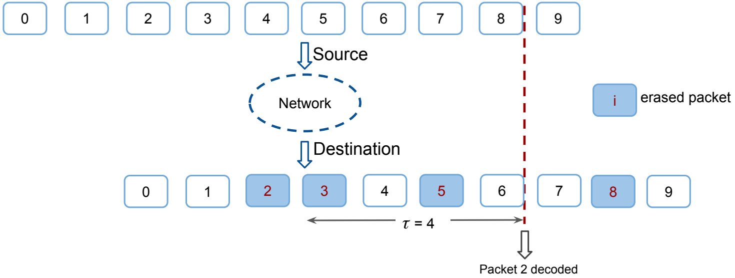
* Department of Electrical Communication Engineering, IISc Bangalore
† Department of Electrical and Computer Engineering, University of Toronto

IEEE International Symposium on Information Theory 2021
Melbourne, Victoria, Australia

12-20 July 2021

The Setting: Delay-Constrained Decoding of Packets

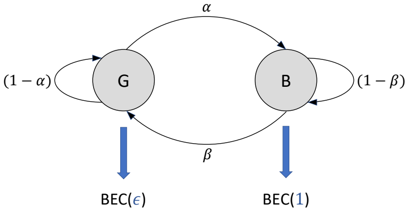
- A stream of packets sent over a network
- Some packets are dropped or lost (packet erasure)
  - congestion, deep fade in wireless link, arriving after deadline



- Latency gaurantees can be provided by allowing atmost τ future packets to be used in packet erasure recovery.
- This has applications in interactive voice & video, AR, VR, 5G URLLC etc

Modeling Packet Erasures

- Gilbert Elliot (GE) Channel is a commonly-accepted model
  - Not tractable, difficult to design codes

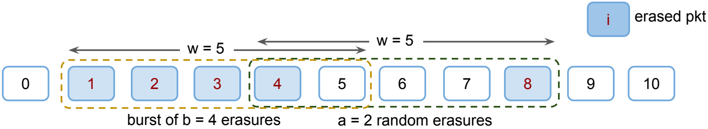


- Delay Constrained Sliding Window (DCSW) channel has been used a proxy to a GE channel [1]

[1] Badr et al., "Layered Constructions for Low-Delay Streaming Codes", IEEE Trans. Info. Theory, 2017.

DCSW Channel

- Admissible erasure patterns (AEP) of $\{a, b, w, \tau\}$ DCSW channel
  - within every sliding window of size w:
    either $\leq a$ random erasures or a burst of $\leq b$ erasures
  - decoding delay constraint τ



$$(a = 2, b = 4, w = 5, \tau = 4)$$

- DCSW Channel is deterministic. A recent analytical work on how well DCSW channel approximates the erasures in GE channel can be found at [2].

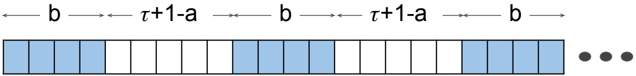
[2] M. Vajha, V. Ramkumar, M. Jhamtani, and P. V. Kumar, "On Sliding Window Approximation of Gilbert-Elliott Channel for Delay Constrained Setting," CoRR, vol. abs/2005.06921, 2020.

Streaming Code

- Streaming code is a packet-level code that can correct from all AEP of DCSW channel within the decoding delay constraint τ.

- Non-trivial only if $a \leq b \leq \tau$.

- Turns out WOLOG we can set $w = \tau + 1$. This reduces parameter set from $\{a, b, w, \tau\}$ to $\{a, b, \tau\}$.



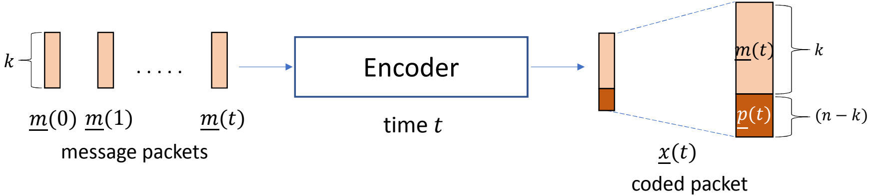
- The optimal rate of an (a, b, τ) streaming code is given by [3], [4].

$$R_{opt} = \frac{\tau + 1 - a}{\tau + 1 - a + b} \ .$$

[3] Badr et al., "Layered Constructions for Low-Delay Streaming Codes", IEEE Trans. Info. Theory, 2017.

[4] M. N. Krishnan, P. V. Kumar "Rate-optimal streaming codes for channels with burst and isolated erasures", ISIT 2018.

Redundancy through Packet Expansion Framework



- Can use scalar codes to come up with streaming codes.

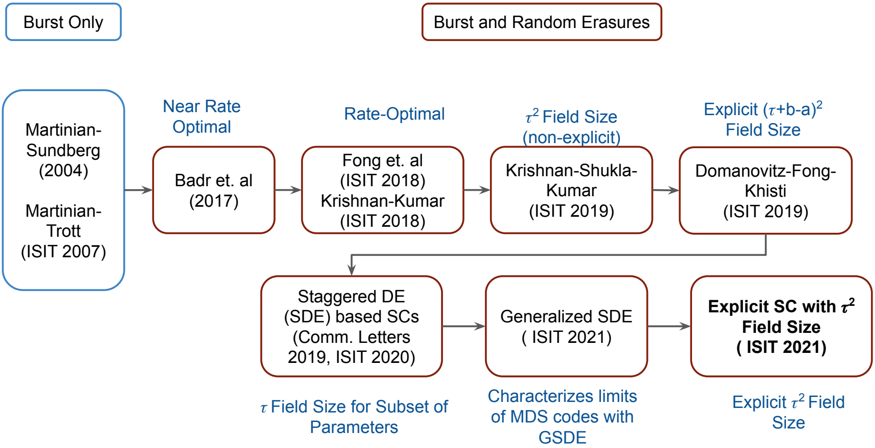
Diagonal Embedding (DE)

- Codewords of $[n, k]$ scalar block code are diagonally placed in the packet stream.

$\underline{x}(0)$	$\underline{x}(1)$	$\underline{x}(2)$	$\underline{x}(3)$	$\underline{x}(4)$	$\underline{x}(5)$	$\underline{x}(6)$	$\underline{x}(7)$	$\underline{x}(8)$	$\underline{x}(9)$	$\underline{x}(10)$	$\underline{x}(11)$	$\underline{x}(12)$	$\underline{x}(13)$
m_1	m_1	m_1											
	m_2	m_2	m_2										
		m_3	m_3	m_3									
			m_4	m_4	m_4								
				m_5	m_5	m_5							
					m_6	m_6	m_6						
						p_1	p_1	p_1					
							p_2	p_2	p_2				
								p_3	p_3	p_3			
									p_4	p_4	p_4		
										p_5	p_5	p_5	
											p_6	p_6	p_6

DE of $[12, 6]$ scalar code

Prior Work on Streaming Codes



DE based Streaming Code Construction

- This approach results in a rate-optimal $\{a, b, \tau\}$ streaming code only if:
  - $n - k \geq b$
  - $n \geq \tau + 1 - a + b$

$\underline{x}(0)$	$\underline{x}(1)$	$\underline{x}(2)$	$\underline{x}(3)$	$\underline{x}(4)$	$\underline{x}(5)$	$\underline{x}(6)$	$\underline{x}(7)$	$\underline{x}(8)$	$\underline{x}(9)$	$\underline{x}(10)$	$\underline{x}(11)$	$\underline{x}(12)$	$\underline{x}(13)$
m_1	m_1	m_1											
	m_2	m_2	m_2										
		m_3	m_3	m_3									
			m_4	m_4	m_4								
				m_5	m_5	m_5							
					m_6	m_6	m_6						
						p_1	p_1	p_1					
							p_2	p_2	p_2				
								p_3	p_3	p_3			
									p_4	p_4	p_4		
										p_5	p_5	p_5	
											p_6	p_6	p_6

DE of $[12, 6]$ scalar code where $a = 3, b = 6, \tau = 8$

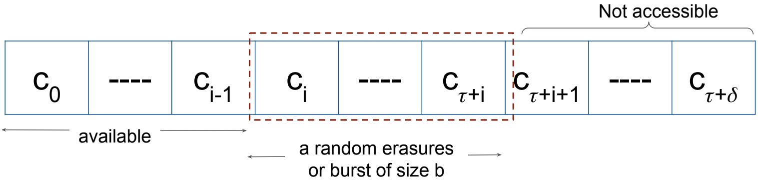
- To recover symbol m_1, symbols p_4, p_5 and p_6 are not available.

Scalar Code Properties

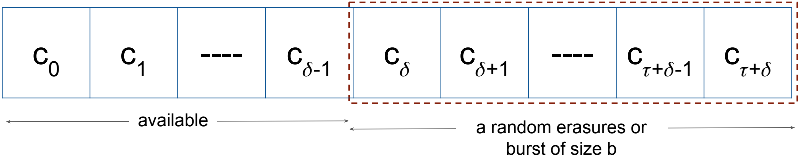
For a given $\{a, b, \tau\}$

$$n = \tau + 1 + \delta, k = n - b \quad \text{where} \quad \delta = b - a$$

- For $i \in [0 : \delta - 1]$, to recover c_i:



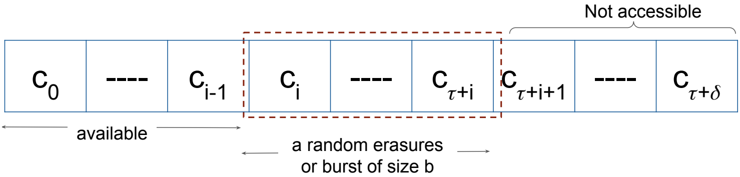
- Let $E \subset [\delta : \tau + \delta]$ be either the set of a random erasures or a set of consecutive b erasures. To recover $\{c_j \mid j \in E\}$:



Parity Check Properties

$$H = \left[\begin{array}{cccc} h_0 & h_1 & \cdots & h_{\tau+\delta} \end{array}\right] \text{ is an } (b \times \tau + \delta + 1) \text{ matrix .}$$

- For $i \in [0 : \delta - 1]$, to recover c_i:



- Let $H^{(i)} = \left[\begin{array}{cccc} h_0^{(i)} & h_1^{(i)} & \cdots & h_{\tau+i}^{(i)} \end{array}\right]$ be pc-matrix of scalar code punctured at indices $[\tau + i + 1 : \tau + \delta]$.

  R1: To recover from a erasures $E \subset [i : \tau + i]$ such that $i \in E$.

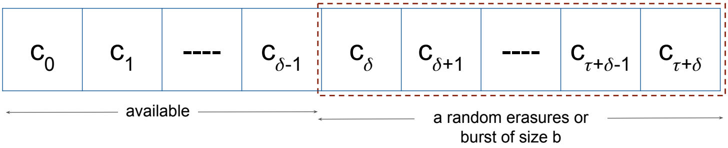
$$h_i^{(i)} \notin \text{Span}\left(h_j^{(i)} \mid j \in E \setminus \{i\}\right)$$

  B1: Given b length burst starting at i:

$$h_i^{(i)} \notin \text{Span}\left(h_j^{(i)} \mid j \in [i + 1 : i + b - 1]\right)$$

Parity Check Properties

- For $E \subseteq [\delta : \tau + \delta]$, to recover $\{c_j \mid j \in E\}$:



R2: To recover from $|E| = a$ random erasures,

$(h_j \mid j \in E)$ have to be linearly independant

B2: Given b length burst starting at i, $E = [i : i + b - 1]$:

$(h_j \mid j \in E)$ have to be linearly independant

Our Explicit Construction

- For all valid $\{a, b, \tau\}$, we present parity check matrix of $[n = \tau + 1 + \delta, k = n - b]$ scalar code where $\delta = b - a$ satisfying the properties.

Parity Check Matrix Design

First $\delta = b - a$ rows

$$H([0:\delta-1],[0:\tau+\delta]) = \left[\begin{array}{cccc|c|c|ccccc} \alpha & & & & & & \alpha & & & & 0 \\ & \alpha & & & \underbrace{0}_{(\delta\times a)} & \underbrace{\mathrm{P}^{a}_{\delta,\tau-b}}_{(\delta\times(\tau-b))} & & 1 & & & 0 \\ & & \ddots & & & & & & \ddots & & \vdots \\ & & & \alpha & & & & & & 1 & 0 \end{array}\right],$$

Last a rows

$$H([\delta:b-1],[0:\tau+\delta]) = \left[\begin{array}{cccc|c|c|c} 1 & & & & & & 1 \\ & 1 & & & \underbrace{\mathrm{C}}_{(a\times(\tau+1-a))} & \underbrace{0}_{(a\times(\delta-1))} & 0 \\ & & \ddots & & & & \vdots \\ & & & 1 & & & 0 \end{array}\right].$$

- The last a rows are the same as in construction by Krishnan et al [5] whereas the first δ rows support is changed along with explicit assignment of coefficients.
- C is a Cauchy matrix with elements in $\mathbb{F}_q$ such that $q \geq \tau$ and $\alpha \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$.

[5] M. Nikhil Krishnan, Deeptanshu Shukla and P. Vijay Kumar, "Rate-Optimal Streaming Codes for Channels With Burst and Random Erasures", IEEE Trans. Info. Theory, 2020

Parity Check Matrix Design

- The definition of $(u \times v)$ matrix $P^a_{u,v}$ is recursive.

$$
\mathrm{P}^a_{u,v} \quad = \quad \begin{cases} \begin{bmatrix} I_u & \underbrace{0}_{(u\times a)} & \mathrm{P}^a_{u,v-u-a} \end{bmatrix} & u+a<v \\ \begin{bmatrix} I_u & \underbrace{0}_{(u\times(v-u))} \end{bmatrix} & u \le v \le u+a \\ \begin{bmatrix} I_v \\ \mathrm{P}^a_{u-v,v} \end{bmatrix} & v<u \end{cases}
$$

For $(a=3, b=6, \tau=8)$, $\delta = 3$, $\tau - b = 2$

$$
H = \left[\begin{array}{ccc|ccc|cc|cccc} \alpha & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & \multicolumn{6}{c|}{} & 0 & 0 & 1 \\ 0 & 1 & 0 & \multicolumn{6}{c|}{\underbrace{\mathrm{C}}_{(3\times 6)}} & 0 & 0 & 0 \\ 0 & 0 & 1 & \multicolumn{6}{c|}{} & 0 & 0 & 0 \end{array}\right] \text{ as } P^3_{3,2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.
$$

Example ($a = 3, b = 6, \tau = 8$) R1 and B1

$$H^{(0)} = \left[\begin{array}{ccccccccc} \overset{0}{\alpha} & \overset{1}{0} & \overset{2}{0} & \overset{3}{0} & \overset{4}{0} & \overset{5}{0} & \overset{6}{1} & \overset{7}{0} & \overset{8}{\alpha} \\ \hline 0 & 1 & 0 & \multicolumn{6}{c}{\underbrace{C([1:2],[0:5])}_{(2\times 6)}} \\ 0 & 0 & 1 & & & & & & \end{array}\right]$$

- B1 Can use 0-th row to recover c_0
- R1 No two columns in $[1:8]$ can linearly combine to give 0's in last two rows.

$$H^{(2)} = \left[\begin{array}{ccc|cccccc|cc} \overset{0}{\alpha} & \overset{1}{0} & \overset{2}{0} & \overset{3}{0} & \overset{4}{0} & \overset{5}{0} & \overset{6}{1} & \overset{7}{0} & \overset{8}{\alpha} & \overset{9}{} & \overset{10}{} \\ & \alpha & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \\ & & \alpha & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & \multicolumn{6}{c|}{\underbrace{C([1:2],[0:5])}_{(2\times 6)}} & 0 & 0 \\ 0 & 0 & 1 & & & & & & & 0 & 0 \end{array}\right]$$

- B1 Can recover c_2 using 2-nd row combined with 0-th row
- R1 No two columns in $[3:10]$ can linearly combine to give 2-nd column.

Example ($a = 3, b = 6, \tau = 8$) B2

$E = [i : i+5]$ be the burst erasures starting at index i. $b \times b$ submatrix of H_E is as shown below:

$$\begin{array}{cccccc} 3 & 4 & 5 & 6 & 7 & 8 \end{array}$$
$$\left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \multicolumn{6}{c}{\mathrm{C}([0:2, 0:5])} \end{array}\right]$$

$$|H_E| = \alpha|(C[0:2],[0:2])|$$

$$\begin{array}{cccccc} 4 & 5 & 6 & 7 & 8 & 9 \end{array}$$
$$\left[\begin{array}{ccccc|c} 0 & 0 & 1 & 0 & \alpha & \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline \multicolumn{5}{c|}{\mathrm{C}([0:2],[1:5])} & 0 \end{array}\right]$$

$$|H_E| = \alpha|(C[0:2],\{1,2,4\})|$$

$$\begin{array}{cccccc} 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$
$$\left[\begin{array}{cccc|cc} 0 & 1 & 0 & \alpha & & \\ 0 & 0 & 1 & 0 & 1 & \\ 0 & 1 & 0 & 0 & 0 & 1 \\ \hline \multicolumn{4}{c|}{\mathrm{C}([0:2],[2:5])} & \multicolumn{2}{c}{0} \end{array}\right]$$

$$|H_E| = |C[0:2],\{2,4,5\}| + \alpha|C[0:2],[2:4]|$$

$$\begin{array}{cccccc} 6 & 7 & 8 & 9 & 10 & 11 \end{array}$$
$$\left[\begin{array}{ccc|cc|c} 1 & 0 & \alpha & & & \\ 0 & 1 & 0 & 1 & & \\ 1 & 0 & 0 & 0 & 1 & \\ \hline \multicolumn{3}{c|}{\mathrm{C}([0:2],[3:5])} & \multicolumn{2}{c|}{0} & \begin{array}{c}1\\0\\0\end{array} \end{array}\right]$$

$$|H_E| = |C[1:2],\{4,5\}| + \alpha|C[1:2],\{3,4\}|$$

Example $(a = 3, b = 6, \tau = 8)$ R2

$$E \subseteq [3:11] \text{ such that } |E| = 3.$$

$$\begin{array}{cccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array}$$

$$H = \left[\begin{array}{cccccccccccc}
\alpha & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 \\
0 & \alpha & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & \alpha & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
\hline
1 & 0 & 0 & \multicolumn{6}{|c|}{} & 0 & 0 & 1 \\
0 & 1 & 0 & \multicolumn{6}{|c|}{\underbrace{C}_{(3\times 6)}} & 0 & 0 & 0 \\
0 & 0 & 1 & \multicolumn{6}{|c|}{} & 0 & 0 & 0
\end{array}\right]$$

- $E \subset [3:8]$ than by cauchy property can recover $|E| = 3$ erasures
- Otherwise i.e, $|E \cap [3:8]| < 3$ then by the cauchy property, can recover $E \cap [3:8]$ using last 2 rows. Remaining erasures $E \setminus [3:8]$ recovery follows from B2 property.

Comparison to a Related Work

- Hollmann-Tolhuizen [6] have come up with burst only correcting streaming codes i.e., $(a = 1, b, \tau)$.
- The parity check of construction by Hollmann-Tolhuizen is given by

$$H = \begin{bmatrix} I_b & P_{b,\tau-b} & I_b \end{bmatrix}, \quad P_{u,v} = \begin{cases} \begin{bmatrix} I_u & P_{u,v-u} \end{bmatrix} & v > u \\ I_u & v = u \\ \begin{bmatrix} I_v \\ P_{u-v,v} \end{bmatrix} & v < u \end{cases}$$

- Notice that the $P^{a}_{\delta,\tau-b}$ follows similar structure but introduces a zero columns in the column-wise expansion.

[6] H. D. Hollmann and L. M. Tolhuizen, "Optimal Codes for Correcting a Single (wrap-around) Burst of Erasures," Trans. in Information Theory, 2008

Summary

- We provide explicit rate-optimal streaming code construction for any $\{a, b, \tau\}$ with field size q^2 where $q \geq \tau$.
- This is an improvement over the best known explicit construction by Domanovitz et al. [7] where $q \geq \tau + b - a$.
- Our construction has the same field size requirement as the best known non-explicit construction by Krishnan et al. [8].

[7] E. Domanovitz, S. L. Fong, and A. Khisti, "An Explicit Rate-Optimal Streaming Code for Channels with Burst and Arbitrary Erasures", ITW 2019

[8] M. Nikhil Krishnan, Deeptanshu Shukla and P. Vijay Kumar, "Rate-Optimal Streaming Codes for Channels With Burst and Random Erasures", Trans. in Information Theory, 2020

Thanks!