

Delay-Optimal Streaming Codes under Source-Channel Rate Mismatch

Pratik Patil
University of Toronto

Joint work with Ahmed Badr (U of T), Ashish Khisti (U of T), Wai-Tian Tan (HP Labs)

Asilomar 2013

Motivation - Delay Sensitive Communication

- Delay is a central issue in many applications¹

Application	Bit-Rate	MSDU (B)	Delay (ms)	Delay (pkts)	PLR
Video Conf.	2 Mbps	1500	100 ms	16	10^{-4}
Interactive Gaming	1Mbps	512	50 ms	12	10^{-4}
SDTV	4Mbps	1500	200 ms	66	10^{-6}

¹IEEE Usage Model Proposal (doc.: IEEE 802.11-03/802r23)

Motivation - Delay Sensitive Communication

- Delay is a central issue in many applications¹

Application	Bit-Rate	MSDU (B)	Delay (ms)	Delay (pkts)	PLR
Video Conf.	2 Mbps	1500	100 ms	16	10^{-4}
Interactive Gaming	1Mbps	512	50 ms	12	10^{-4}
SDTV	4Mbps	1500	200 ms	66	10^{-6}

- Various delays: network delay - processing delay, propagation delay, queuing delay and **coding delay**.

¹IEEE Usage Model Proposal (doc.: IEEE 802.11-03/802r23)

Motivation - Delay Sensitive Communication

- Delay is a central issue in many applications¹

Application	Bit-Rate	MSDU (B)	Delay (ms)	Delay (pkts)	PLR
Video Conf.	2 Mbps	1500	100 ms	16	10^{-4}
Interactive Gaming	1Mbps	512	50 ms	12	10^{-4}
SDTV	4Mbps	1500	200 ms	66	10^{-6}

- Various delays: network delay - processing delay, propagation delay, queuing delay and coding delay.
- Communication medium: wireless channel.

¹IEEE Usage Model Proposal (doc.: IEEE 802.11-03/802r23)

Motivation - Delay Sensitive Communication

- Delay is a central issue in many applications¹

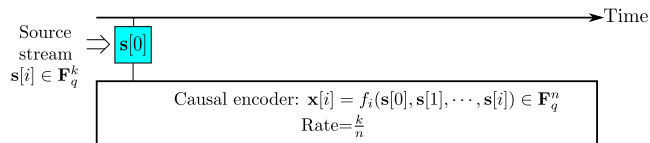
Application	Bit-Rate	MSDU (B)	Delay (ms)	Delay (pkts)	PLR
Video Conf.	2 Mbps	1500	100 ms	16	10^{-4}
Interactive Gaming	1Mbps	512	50 ms	12	10^{-4}
SDTV	4Mbps	1500	200 ms	66	10^{-6}

- Various delays: network delay - processing delay, propagation delay, queuing delay and **coding delay**.
- Communication medium: wireless channel.
- Why **forward error correction** (FEC) and not automatic repeat request (ARQ)?
 - Multicast streaming: feedback not feasible (e.g. digital video broadcasting, IPTV)
 - Large round-trip delay: delayed feedback (possibly lossy too)

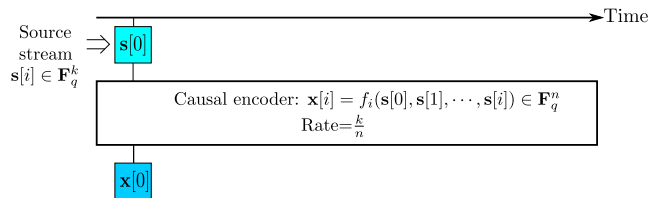
¹IEEE Usage Model Proposal (doc.: IEEE 802.11-03/802r23)

Real-time Streaming Setup

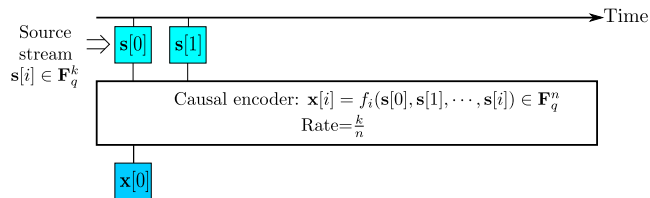
Real-time Streaming Setup



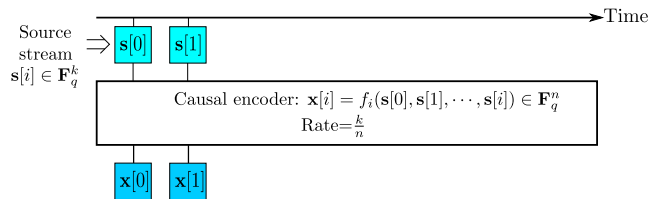
Real-time Streaming Setup



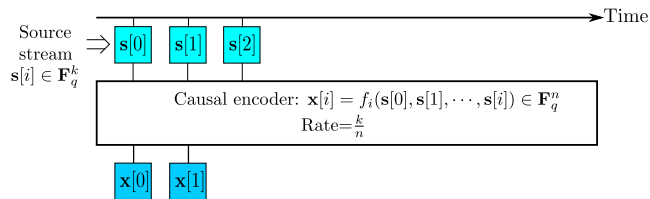
Real-time Streaming Setup



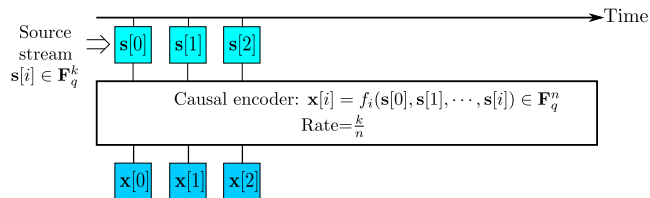
Real-time Streaming Setup



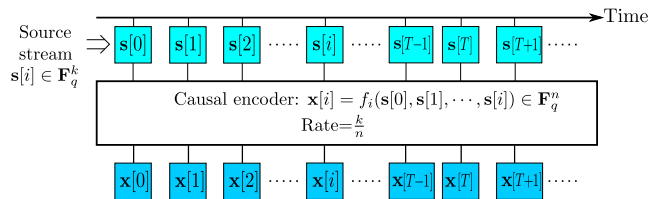
Real-time Streaming Setup



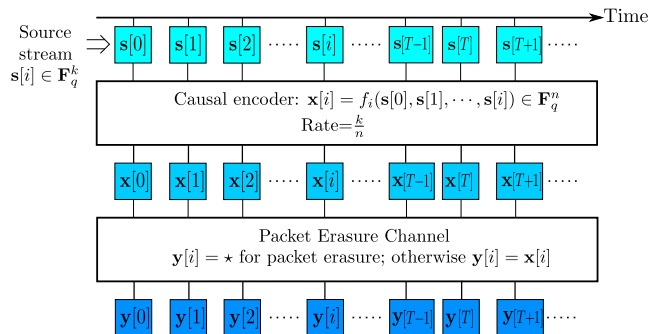
Real-time Streaming Setup



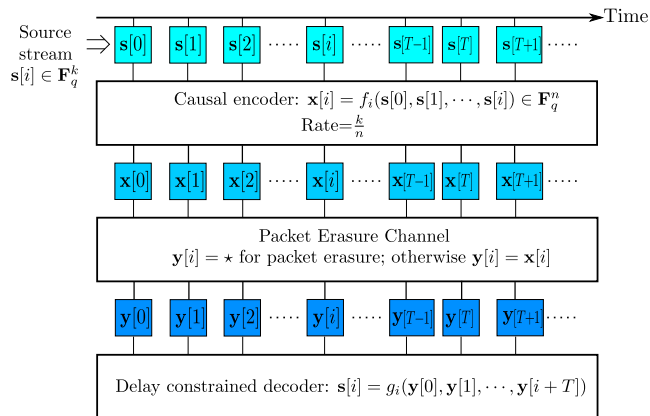
Real-time Streaming Setup



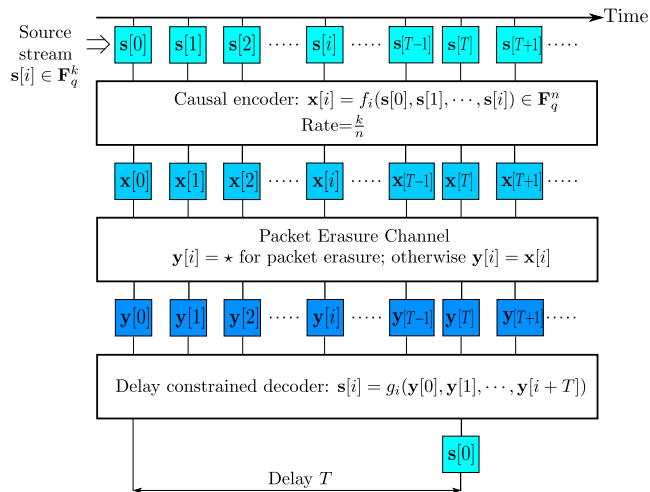
Real-time Streaming Setup



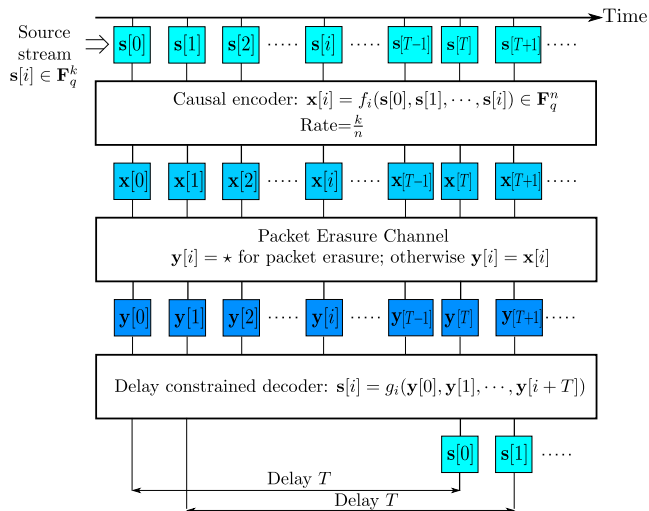
Real-time Streaming Setup



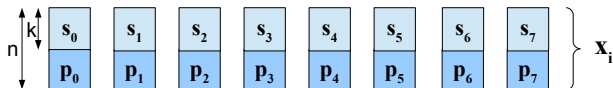
Real-time Streaming Setup



Real-time Streaming Setup



Error Correction for Burst Erasure Channel: Baseline Techniques



$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasures Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

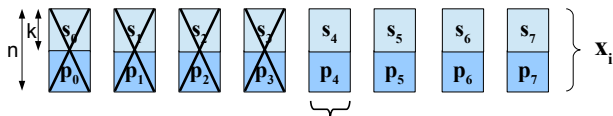


$$\mathbf{p}_j = \mathbf{s}_j \cdot \mathbf{H}_0 + \mathbf{s}_{j-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{j-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

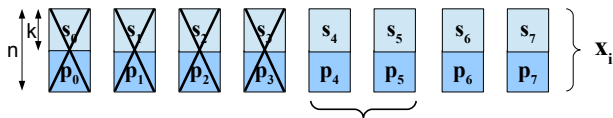


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

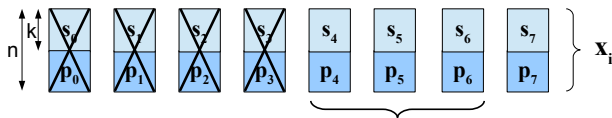


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

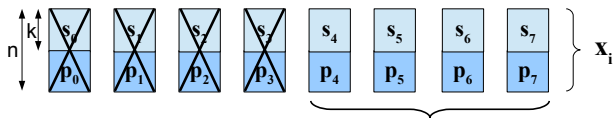


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

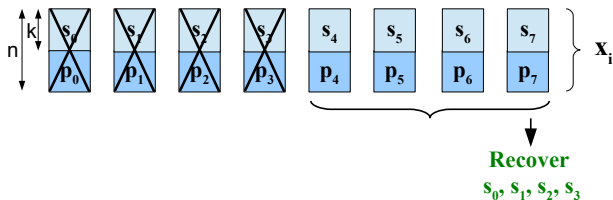


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

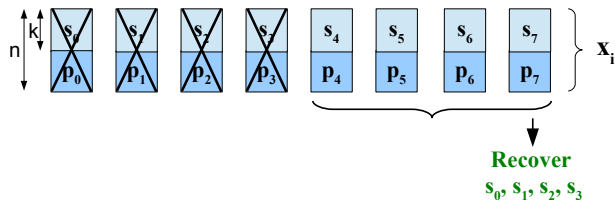


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques

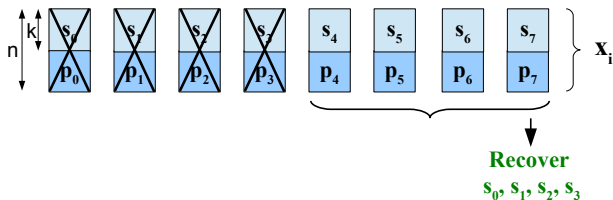


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

Error Correction for Burst Erasure Channel: Baseline Techniques



$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \dots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \quad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

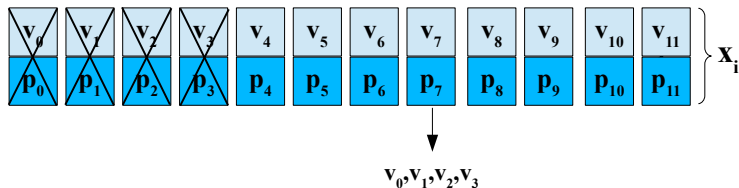
- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

$$\begin{bmatrix} \mathbf{p}_4 \\ \mathbf{p}_5 \\ \mathbf{p}_6 \\ \mathbf{p}_7 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_3 & \mathbf{H}_2 & \mathbf{H}_1 \\ \mathbf{H}_5 & \mathbf{H}_4 & \mathbf{H}_3 & \mathbf{H}_2 \\ 0 & \mathbf{H}_5 & \mathbf{H}_4 & \mathbf{H}_3 \\ 0 & 0 & \mathbf{H}_5 & \mathbf{H}_4 \end{bmatrix}}_{\text{full rank}} \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix}$$

Streaming Codes - Burst Erasure Channel

$$B = 4, T = 8$$

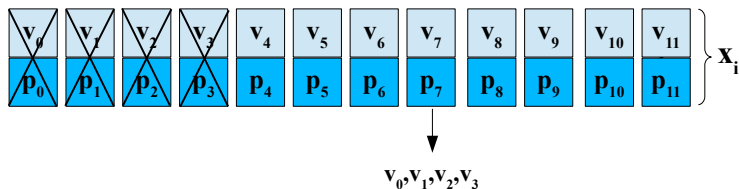
Rate 1/2 Baseline Erasure Codes, $T = 7$



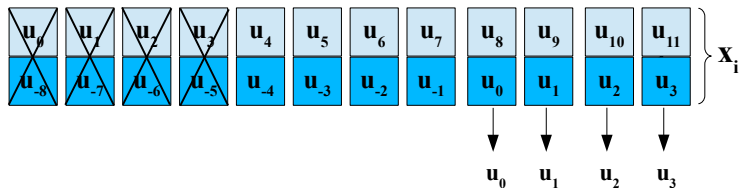
Streaming Codes - Burst Erasure Channel

$B = 4, T = 8$

Rate 1/2 Baseline Erasure Codes, $T = 7$

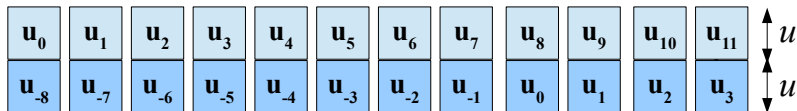
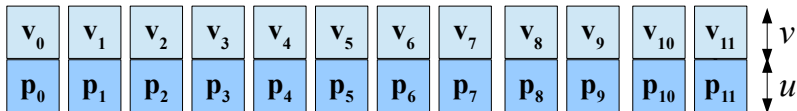


Rate 1/2 Repetition Code, $T = 8$



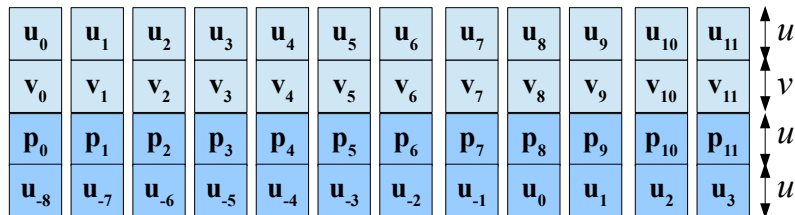
Burst-Erasure Streaming Codes

$B = 4, T = 8$



Burst-Erasure Streaming Codes

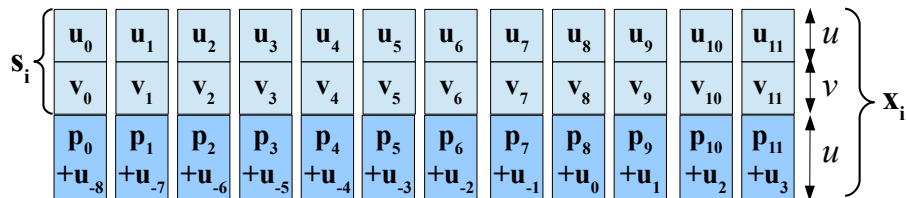
$B = 4, T = 8$



$$R = \frac{u+v}{3u+v} = \frac{1}{2}$$

Burst-Erasure Streaming Codes

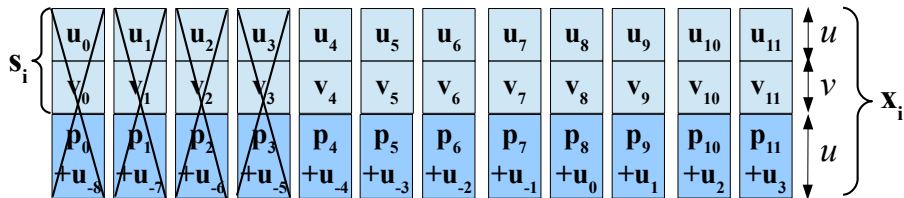
$B = 4, T = 8$



$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

Burst-Erasure Streaming Codes

$B = 4, T = 8$



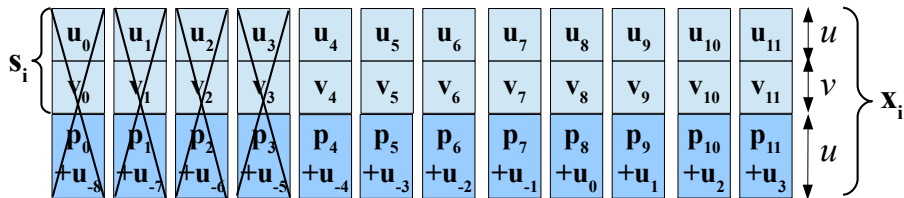
$$R = \frac{u + v}{2u + v}$$

Encoding:

- 1 Split each source symbol into 2 groups $s_i = (u_i, v_i)$
- 2 Apply Erasure code to the v_i stream generating p_i parities
- 3 Repeat the u_i symbols with a shift of T
- 4 Combine the repeated u_i 's with the p_i 's

Burst-Erasure Streaming Codes

$B = 4, T = 8$



$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

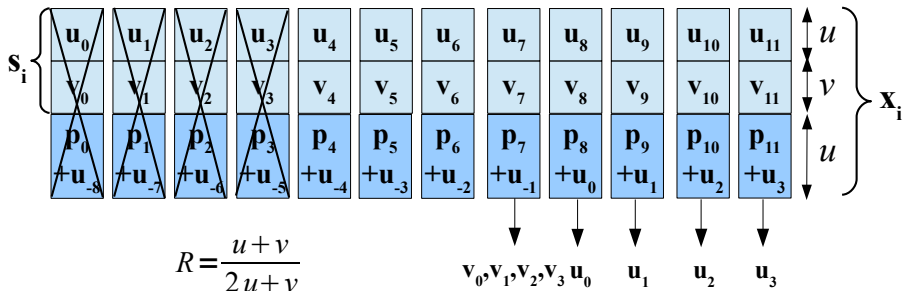
v_0, v_1, v_2, v_3

Encoding:

- 1 Split each source symbol into 2 groups $s_i = (u_i, v_i)$
- 2 Apply Erasure code to the v_i stream generating p_i parities
- 3 Repeat the u_i symbols with a shift of T
- 4 Combine the repeated u_i 's with the p_i 's

Burst-Erasure Streaming Codes

$B = 4, T = 8$

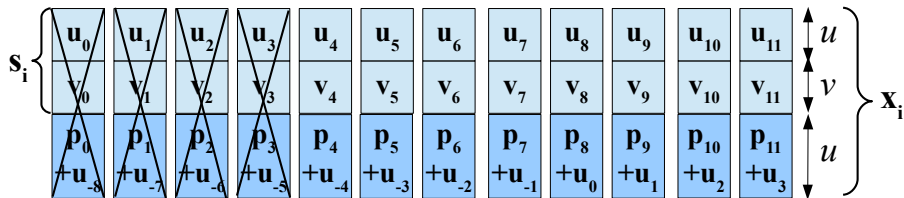


Encoding:

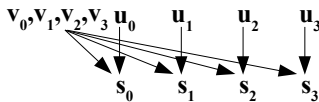
- 1 Split each source symbol into 2 groups $s_i = (u_i, v_i)$
- 2 Apply Erasure code to the v_i stream generating p_i parities
- 3 Repeat the u_i symbols with a shift of T
- 4 Combine the repeated u_i 's with the p_i 's

Burst-Erasure Streaming Codes

$B = 4, T = 8$



$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

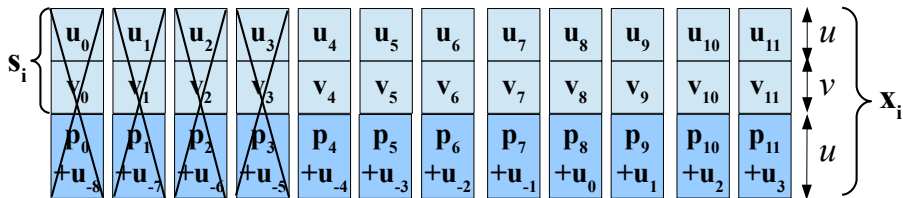


Encoding:

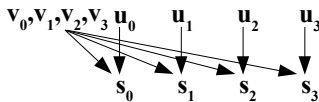
- 1 Split each source symbol into 2 groups $s_i = (u_i, v_i)$
- 2 Apply Erasure code to the v_i stream generating p_i parities
- 3 Repeat the u_i symbols with a shift of T
- 4 Combine the repeated u_i 's with the p_i 's

Burst-Erasure Streaming Codes

$$B = 4, T = 8$$



$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

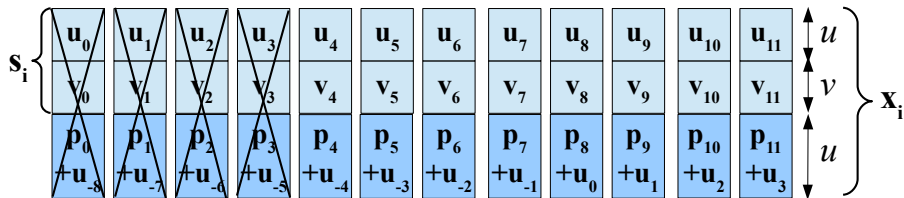


Encoding:

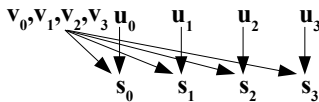
- 1 Split each source symbol into 2 groups $s_i = (u_i, v_i)$
 - 2 Apply Erasure code to the v_i stream generating p_i parities
 - 3 Repeat the u_i symbols with a shift of T
 - 4 Combine the repeated u_i 's with the p_i 's
- Choosing $u = B$ and $v = T - B$, $R = \frac{T}{T+B}$ (Optimal) [Badr, Khisti-Infocom '13]

Burst-Erasure Streaming Codes

$$B = 4, T = 8$$



$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$



Encoding:

- 1 Split each source symbol into 2 groups $s_i = (u_i, v_i)$
 - 2 Apply Erasure code to the v_i stream generating p_i parities
 - 3 Repeat the u_i symbols with a shift of T
 - 4 Combine the repeated u_i 's with the p_i 's
- Choosing $u = B$ and $v = T - B$, $R = \frac{T}{T+B}$ (Optimal) [Badr, Khisti-Infocom '13]
 - Capacity first analyzed by Martinian and Sundberg (IT-2004) (Maximally Short Codes)

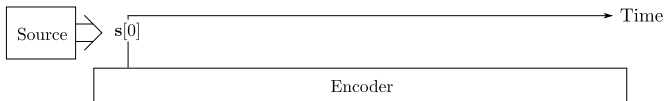
Streaming Setup under Source-Channel Rate Mismatch

Streaming Setup under Source-Channel Rate Mismatch



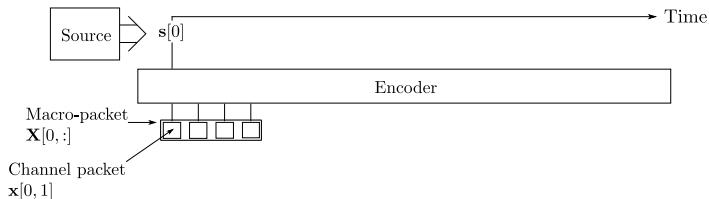
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim \text{uniform over } \mathbf{F}_q^k$

Streaming Setup under Source-Channel Rate Mismatch



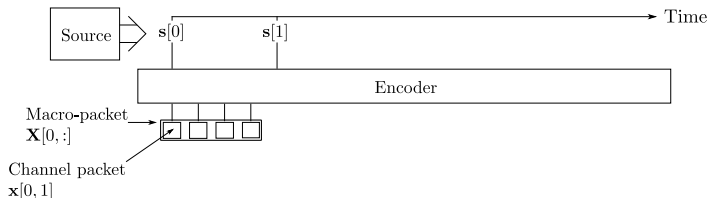
- **Source model:** i.i.d. process with $s[i] \sim \text{uniform over } \mathbf{F}_q^k$

Streaming Setup under Source-Channel Rate Mismatch



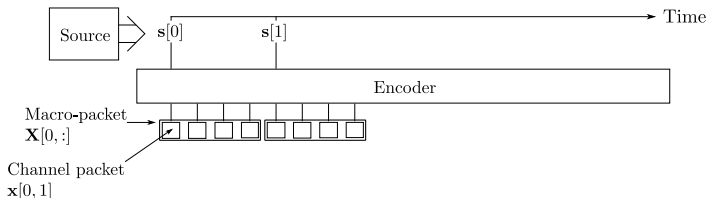
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \in \mathbf{F}_q^n$
- **Macro-packet:** $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
- **Rate:** $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$

Streaming Setup under Source-Channel Rate Mismatch



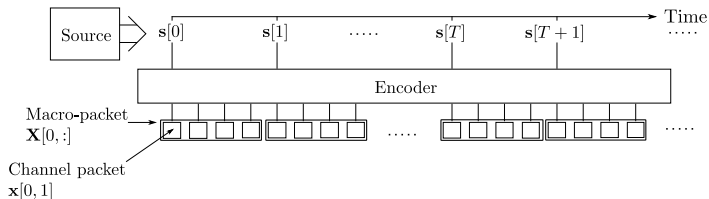
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim$ uniform over \mathbf{F}_q^k
 - **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \in \mathbf{F}_q^n$
- Macro-packet:** $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
- Rate:** $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$

Streaming Setup under Source-Channel Rate Mismatch



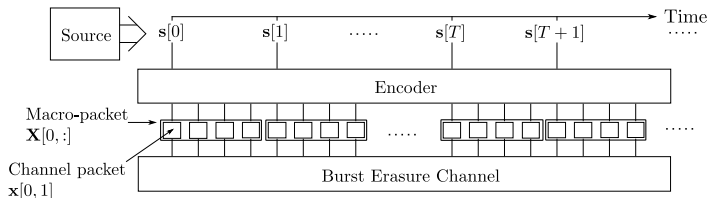
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \in \mathbf{F}_q^n$
Macro-packet: $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
Rate: $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$

Streaming Setup under Source-Channel Rate Mismatch



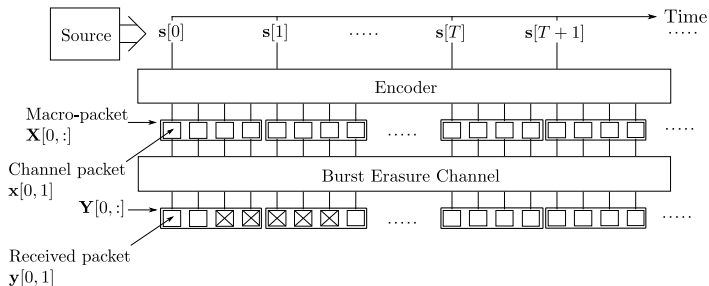
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim$ uniform over \mathbf{F}_q^k
 - **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \in \mathbf{F}_q^n$
- Macro-packet:** $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
- Rate:** $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$

Streaming Setup under Source-Channel Rate Mismatch



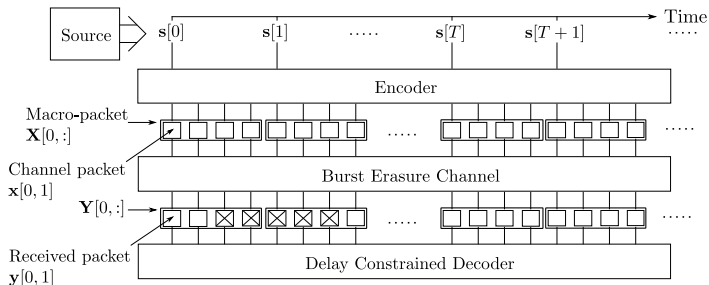
- **Source model:** i.i.d. process with $s[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(s[0], s[1], \dots, s[i]) \in \mathbf{F}_q^n$
Macro-packet: $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
Rate: $R = \frac{H(s)}{n \times M} = \frac{k}{n \times M}$
- **Packet erasure channel:** erasure burst of maximum B channel packets

Streaming Setup under Source-Channel Rate Mismatch



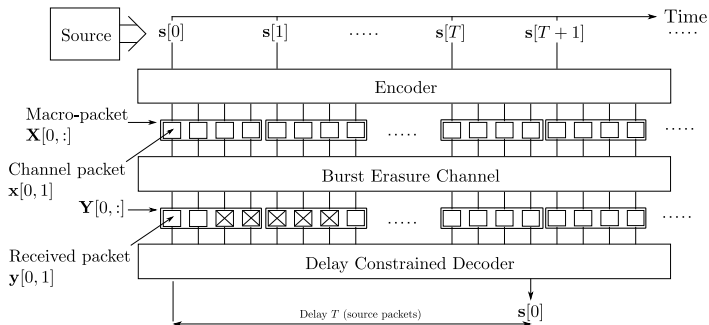
- **Source model:** i.i.d. process with $s[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(s[0], s[1], \dots, s[i]) \in \mathbf{F}_q^n$
Macro-packet: $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
Rate: $R = \frac{H(s)}{n \times M} = \frac{k}{n \times M}$
- **Packet erasure channel:** erasure burst of maximum B channel packets

Streaming Setup under Source-Channel Rate Mismatch



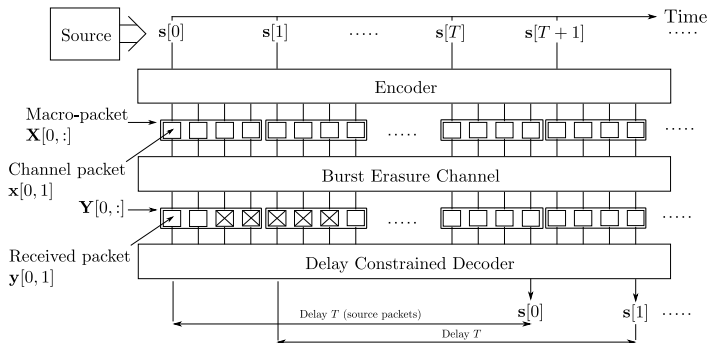
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \in \mathbf{F}_q^n$
Macro-packet: $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
Rate: $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$
- **Packet erasure channel:** erasure burst of maximum B channel packets
- **Delay-constrained decoder:** $\mathbf{s}[i]$ needs to be recovered by macro-packet $i + T$

Streaming Setup under Source-Channel Rate Mismatch



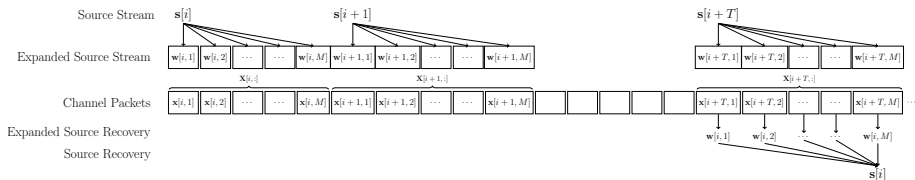
- **Source model:** i.i.d. process with $\mathbf{s}[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(\mathbf{s}[0], \mathbf{s}[1], \dots, \mathbf{s}[i]) \in \mathbf{F}_q^n$
Macro-packet: $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
Rate: $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$
- **Packet erasure channel:** erasure burst of maximum B channel packets
- **Delay-constrained decoder:** $\mathbf{s}[i]$ needs to be recovered by macro-packet $i + T$

Streaming Setup under Source-Channel Rate Mismatch



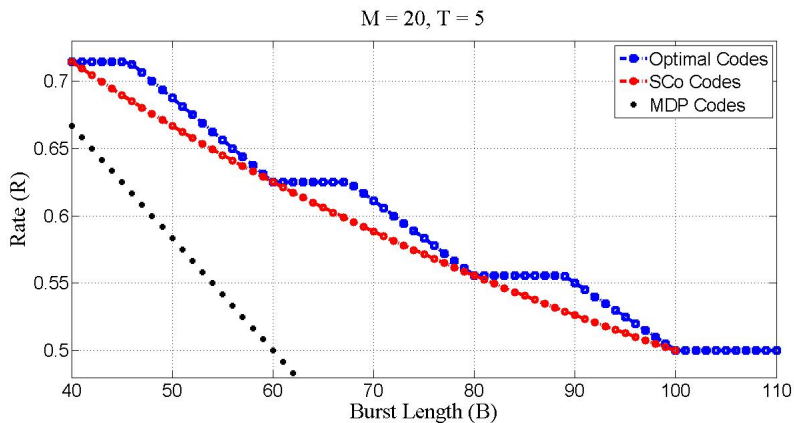
- **Source model:** i.i.d. process with $s[i] \sim$ uniform over \mathbf{F}_q^k
- **Streaming encoder:** $\mathbf{x}[i, j] = f_{i,j}(s[0], s[1], \dots, s[i]) \in \mathbf{F}_q^n$
Macro-packet: $\mathbf{X}[i, :] = [\mathbf{x}[i, 1] \mid \dots \mid \mathbf{x}[i, M]]$
Rate: $R = \frac{H(\mathbf{s})}{n \times M} = \frac{k}{n \times M}$
- **Packet erasure channel:** erasure burst of maximum B channel packets
- **Delay-constrained decoder:** $s[i]$ needs to be recovered by macro-packet $i + T$

Adaptation of Code Construction for $M=1$ to any M



- Split each source-packet $s[i]$ into M sub-source-packets $w[i, 1], \dots, w[i, M]$
- Apply code for $M = 1$ to this expanded source stream
- Modified delay (w.r.t.) each sub-source-packet is MT , burst length is still B
- Achievable rate: $R = \frac{MT}{MT+B}$
- Only optimal for $B = cM$, for any $c \in \mathbb{N}$

Numerical Comparison



Optimal Code Construction

Key Idea:

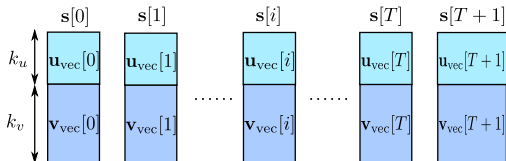
- ① Apply code for $M = 1$ on complete source-packet to generate the entire macro-packet.
- ② Map/reshape the generated macro-packet into M individual channel-packets.

Optimal Code Construction (cont.)

1 Source splitting

- split $\mathbf{s}[i]$ into two groups

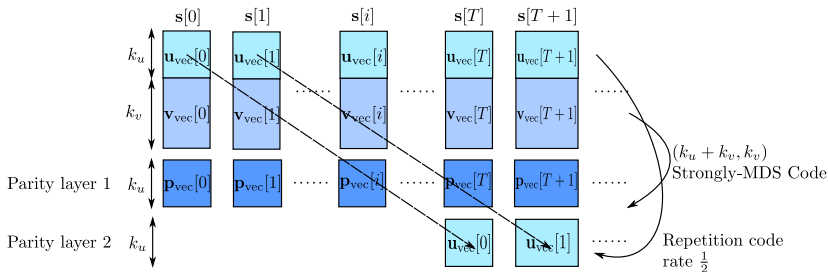
$$\begin{aligned}\mathbf{s}[i] &= (s_1[i], \dots, s_k[i]) \\ &= (\underbrace{u_1[i], \dots, u_{k_u}[i]}_{\mathbf{u}_{\text{vec}}[i]}, \underbrace{v_1[i], \dots, v_{k_v}[i]}_{\mathbf{v}_{\text{vec}}[i]})\end{aligned}$$



Optimal Code Construction (cont.)

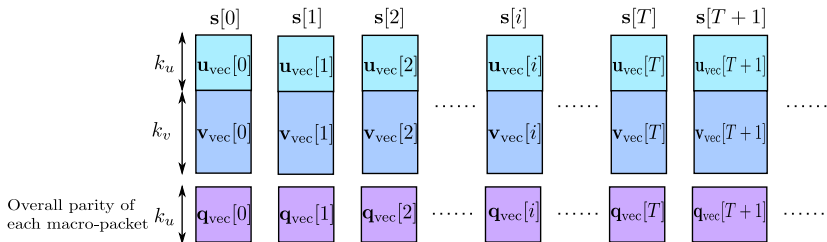
2 Parity generation

- layer 1: $(k_v + k_u, k_v, T)$ Strongly-MDS code applied to $\mathbf{v}_{\text{vec}}[\cdot]$ generating $\mathbf{p}_{\text{vec}}[i]$
- layer 2: repetition code on \mathbf{u}_{vec} with a shift of T



Optimal Code Construction (cont.)

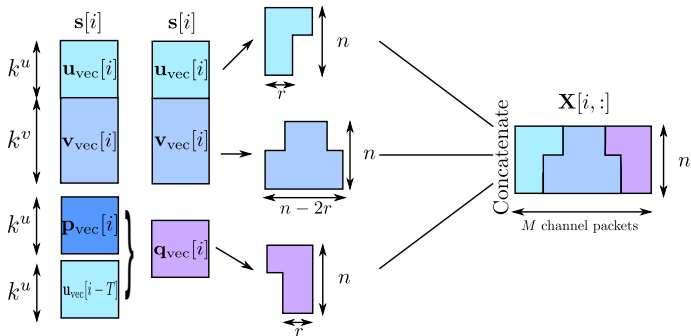
- Overall combined parity: $\mathbf{q}_{\text{vec}}[i] = \mathbf{p}_{\text{vec}}[i] + \mathbf{u}_{\text{vec}}[i - T]$



Optimal Code Construction (cont.)

3 Mapping of macro-packet to individual channel-packets

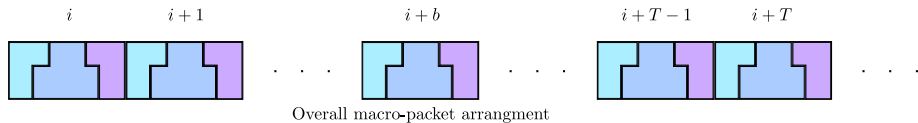
- Map the generated macro-packet into M individual channel-packets



$$\text{Rate of the code} = \frac{k_u + k_v}{2k_u + k_v}$$

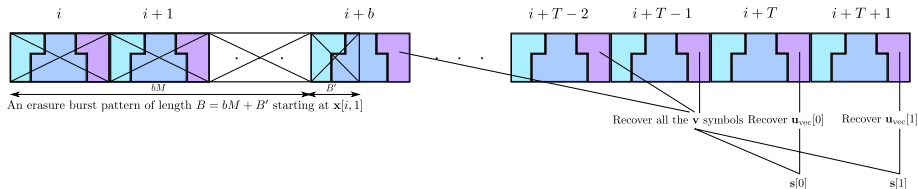
Optimal Code Construction (cont.)

Overall macro-packet structure:



Decoding Analysis

Key Fact: The worst case burst pattern starts at the beginning of the macro-packet.



Let $B = bM + B'$. Two cases depending on whether $B' \leq (1 - R)M$:

Burst only erases symbols from $\mathbf{u}_{\text{vec}}[i + b]$

- Recovery of \mathbf{v} symbols:
 $(k_u + k_v)b = k_u T$

- Optimal spitting ratio:
 $\frac{k_u}{k_v} = \frac{b}{T-b}$

- $R = \frac{T}{T+b}$

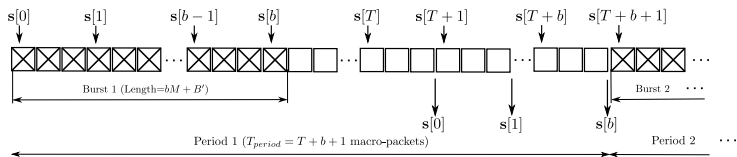
Burst erases symbols from $\mathbf{v}_{\text{vec}}[i + b]$

- Recovery of \mathbf{v} symbols:
 $(k_u + k_v)b + (B'n - k_u) = k_u T$

- Optimal spitting ratio:
 $\frac{k_u}{k_v} = \frac{B}{M(T+b+1) - 2B}$

- $R = \frac{M(T+b+1) - B}{M(T+b+1)}$

- $T > b$: Technique of periodic erasure channel



- Enough parities to recover from full burst $bM + B'$ $\implies 1 - R \geq \frac{bM + B'}{M(T + b + 1)}$
- Enough parities to recover from burst length $bM \implies 1 - R \geq \frac{bM}{M(T + b)}$
- $T = b$: Periodic channel argument not tight, different argument

Capacity Result

Theorem

For the streaming setup considered, with any M , T and B , the streaming capacity C is given by the following expression:

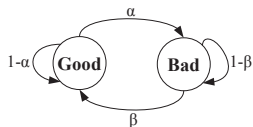
$$C = \begin{cases} \frac{T}{T+b}, & B' \leq \frac{b}{T+b}M, \quad T \geq b, \\ \frac{M(T+b+1)-B}{M(T+b+1)}, & B' > \frac{b}{T+b}M, \quad T > b, \\ \frac{M-B'}{M}, & B' > \frac{M}{2}, \quad T = b, \\ 0, & T < b. \end{cases}$$

where the constants b and B' are defined via

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



Simulation Result



Gilbert Channel

- Bad state: $\Pr(\text{loss}) = 1$
- Good state: no losses

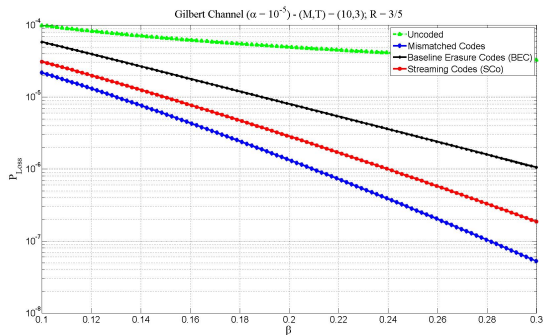


Figure: $(M, T, R) = (10, 3, 3/5)$

Robust Extension

Extending the channel model to account for i.i.d. isolated erasures:

Any sliding window of length W contains

- A **burst** of maximum length B , or,
- No more than N erasures in **arbitrary** positions.

Robust Extension

Extending the channel model to account for i.i.d. isolated erasures:

Any sliding window of length W contains

- A **burst** of maximum length B , or,
- No more than N erasures in **arbitrary** positions.

$$(N, B, W) = (2, 3, 6)$$



$$W = 6$$

$$N = 2$$

Robust Extension

Extending the channel model to account for i.i.d. isolated erasures:

Any sliding window of length W contains

- A **burst** of maximum length B , or,
- No more than N erasures in **arbitrary** positions.

$$(N, B, W) = (2, 3, 6)$$



$$W = 6$$

$$N = 2$$

Robust Extension

Extending the channel model to account for i.i.d. isolated erasures:

Any sliding window of length W contains

- A **burst** of maximum length B , or,
- No more than N erasures in **arbitrary** positions.

$$(N, B, W) = (2, 3, 6)$$



$$W = 6$$

$$B = 3$$

Robust Extension

Extending the channel model to account for i.i.d. isolated erasures:

Any sliding window of length W contains

- A **burst** of maximum length B , or,
- No more than N erasures in **arbitrary** positions.

$$(N, B, W) = (2, 3, 6)$$

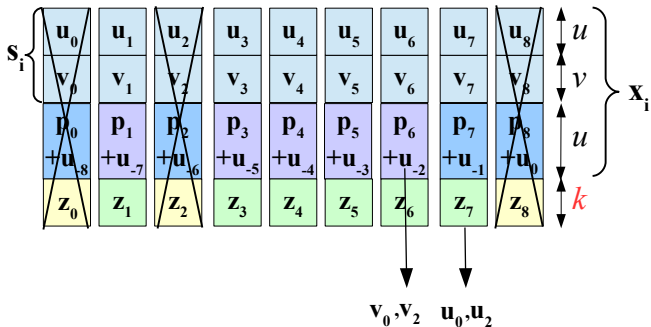


$$W = 6$$

$$B = 3$$

Robust Extension for $M = 1$

- Append an additional layer parity checks containing Strongly-MDS code on \mathbf{u}



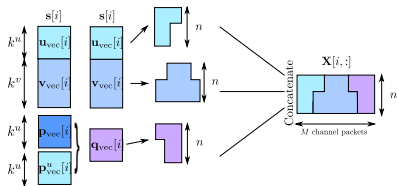
- $R = \frac{u+v}{2u+v+k}$.
- Very close to being optimal for $k = \frac{N}{T-N+1} B$

Robust Extension for any M

Many possibilities for the placement of Strongly-MDS parities for u !

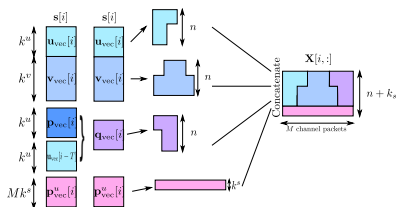
Approach I:

- Repetition code replaced by Strongly-MDS code for \mathbf{u}
- Rate of the code unchanged.
- $N = r$ is achievable.



Approach II

- Append additional layer of Strongly-MDS code for \mathbf{u}
- $R = \frac{k_u + k_v}{2k_u + k_v + k_s}$.
- k_s chosen according to given N .



Conclusions

- ① Extension of the previously studied streaming setup ($M = 1$) to any general M (source-channel rate mismatch).
- ② Capacity characterization for a certain class of burst-erasure channels.
- ③ Layered code construction approach.
- ④ Robust code extension for protection against i.i.d. isolated erasures.

Thanks for listening!

Any questions/comments/thoughts?