Layered Constructions for Low-Delay Streaming Codes

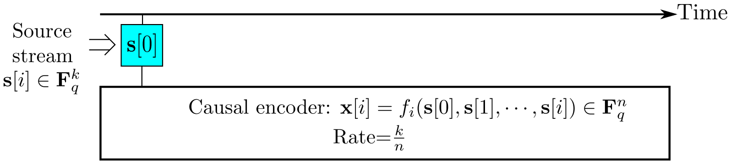
Ashish Khisti
University of Toronto

Joint Work:
Ahmed Badr (Toronto),
Wai-Tian Tan (Cisco),
John Apostolopoulos (Cisco).

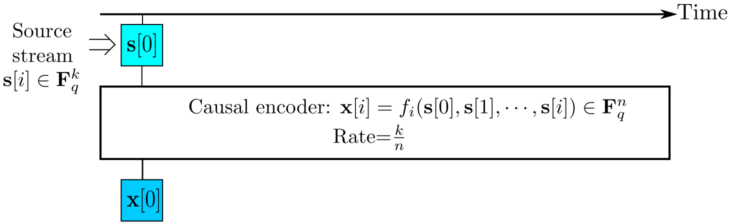
ITA, 2014

Real-Time Communication System

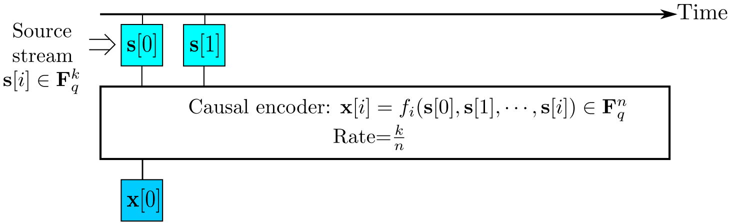
Real-Time Communication System



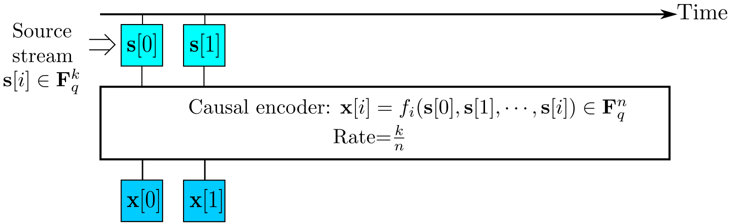
Real-Time Communication System



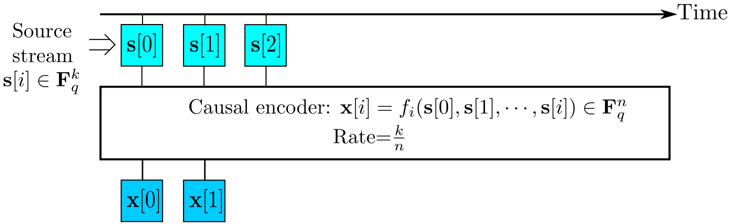
Real-Time Communication System



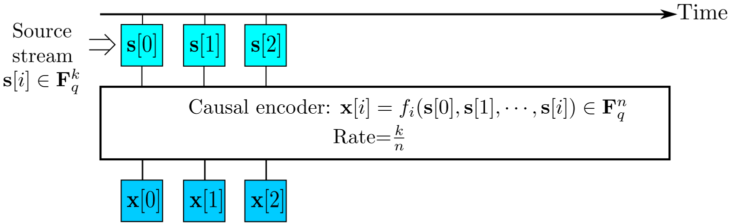
Real-Time Communication System



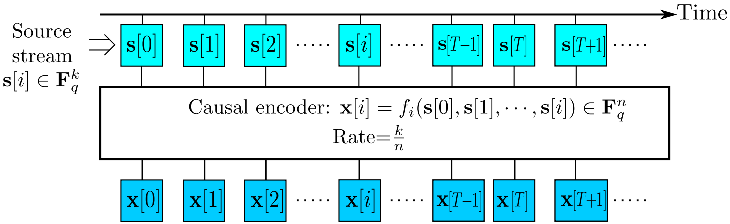
Real-Time Communication System



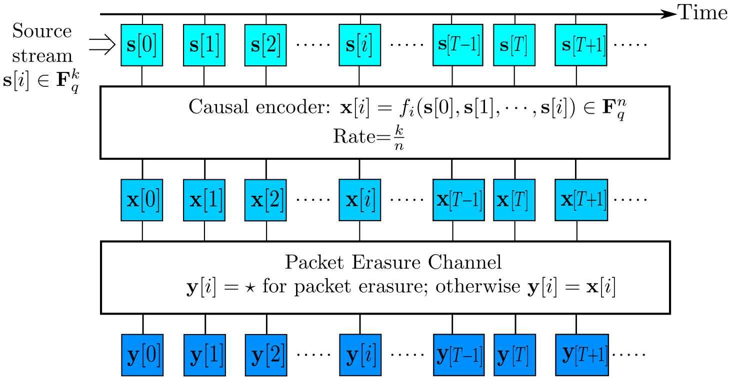
Real-Time Communication System



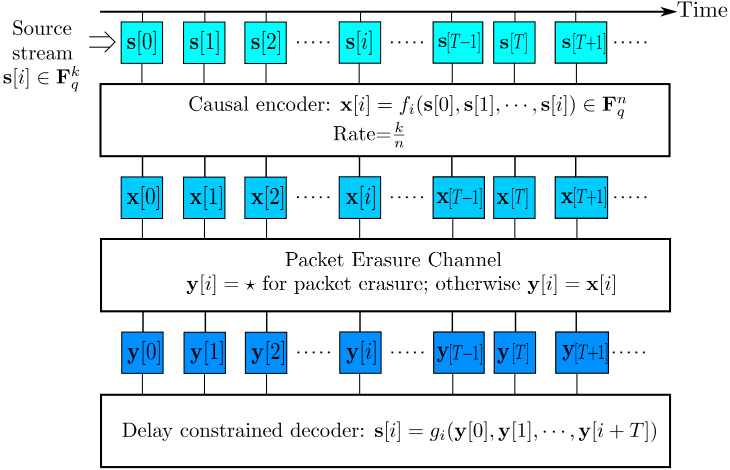
Real-Time Communication System



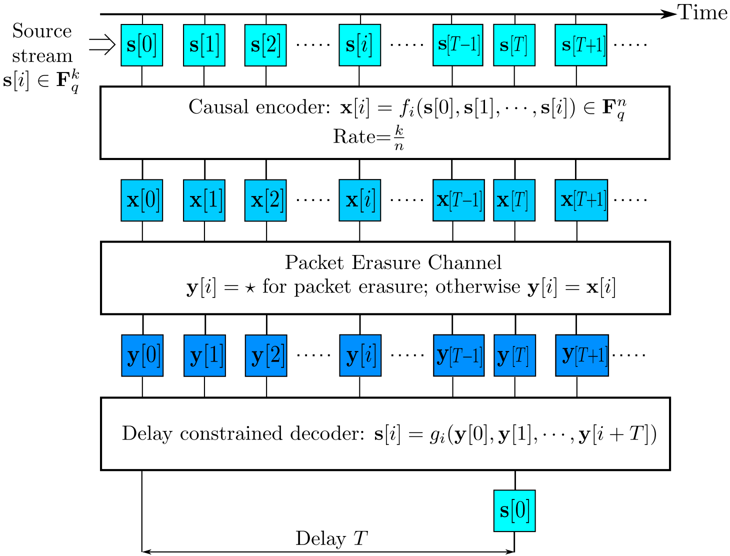
Real-Time Communication System



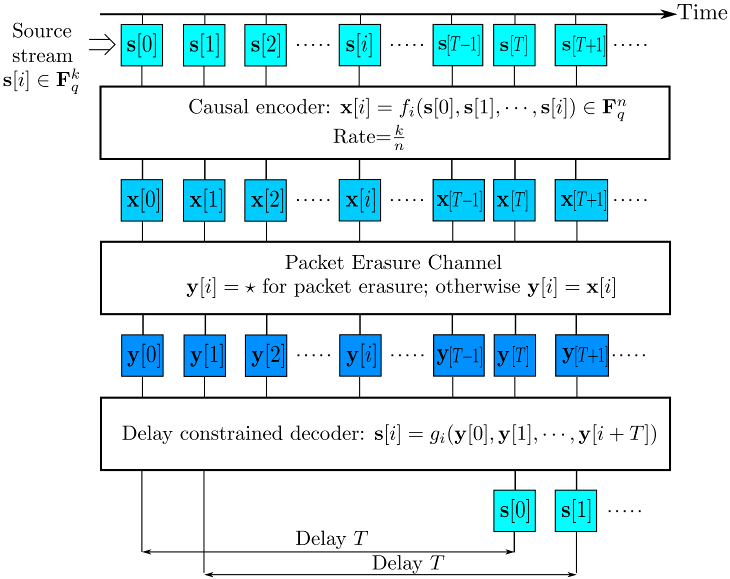
Real-Time Communication System



Real-Time Communication System



Real-Time Communication System

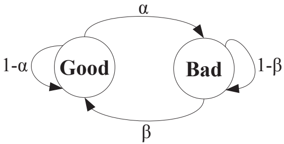


Problem Setup

- **Source Model** : i.i.d. sequence $\mathsf{s}[t] \sim p_{\mathsf{s}}(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- **Streaming Encoder**: $\mathsf{x}[t] = f_t\left(\mathsf{s}[1], \ldots, \mathsf{s}[t]\right)$, $\mathsf{x}[t] \in (\mathbb{F}_q)^n$
- Erasure Channel (To be specified)
- **Delay-Constrained Decoder**: $\hat{\mathsf{s}}[t] = g_t(\mathsf{y}[1], \ldots, \mathsf{y}[t+T])$
- Rate $R = \frac{k}{n}$

Problem Setup

- **Source Model** : i.i.d. sequence $s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- **Streaming Encoder**: $x[t] = f_t(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel (To be specified)
- **Delay-Constrained Decoder**: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$
- Rate $R = \frac{k}{n}$



Gilbert-Elliott Model

- How much performance gains can we obtain?
- What are the fundamental metrics for low-delay error correction codes?

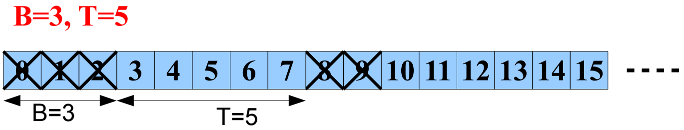
Prior Work - Real Time Streaming

- Structural Results on Real Time Coding: Witsenhausen (1979), Teneketzis (2006), Mahajan (2009), Kaspi and Merhav (2012), Asnani and Weissman (2013) ...
- Delay-Universal Codes: Schulman (IT 1996), Sahai (2001), Martinian and Wornell (Allerton 2004), Sukhavasi and Hassibi (2011)
- Low-Delay Codes: Martinian and Sundberg (2004), Martinian (2004)

Streaming Codes - Burst Erasure Channel

Martinian and Sundberg (2004)

- **Source Model** : i.i.d. sequence $s[t] \sim p_s(\cdot) = \mathrm{Unif}\{(\mathbb{F}_q)^k\}$
- **Streaming Encoder**: $x[t] = f_t\,(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel (Burst Erasure Channel)
- **Delay-Constrained Decoder**: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$



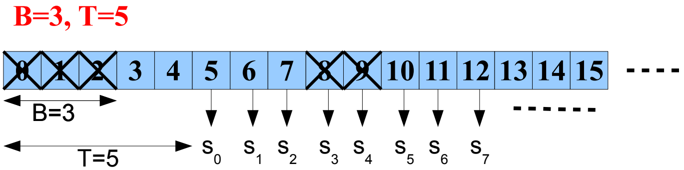
Capacity $C(B,T)$

The maximum rate R such that there exists a rate $R = \frac{k}{n}$ streaming code over a sufficiently large field q such that $\Pr(s[t] \neq \hat{s}[t]) = 0$, for all $t \geq 0$.

Streaming Codes - Burst Erasure Channel

Martinian and Sundberg (2004)

- **Source Model** : i.i.d. sequence $s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- **Streaming Encoder**: $x[t] = f_t\,(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel (Burst Erasure Channel)
- **Delay-Constrained Decoder**: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$



Capacity $C(B,T)$

The maximum rate R such that there exists a rate $R = \frac{k}{n}$ streaming code over a sufficiently large field q such that $\Pr(s[t] \neq \hat{s}[t]) = 0$, for all $t \geq 0$.

Streaming Codes - Burst Erasure Channel

Martinian and Sundberg (2004)

Capacity Result

Streaming Capacity:

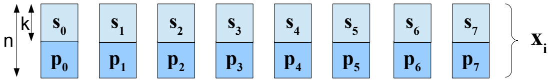
$$C(B,T) = \begin{cases} \frac{T}{T+B}, & T \geq B \\ 0, & T < B. \end{cases}$$

Code Construction

MS Codes (Maximally-Short Codes) Construction:

- Step 1: Construct a low-delay block code
- Step 2: Use diagonal interleaving to construct a streaming code.

Baseline Codes

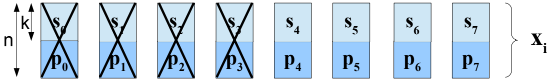


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

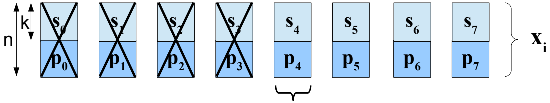


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

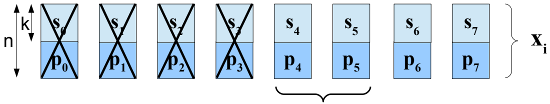


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

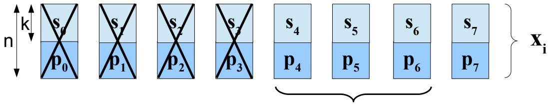


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

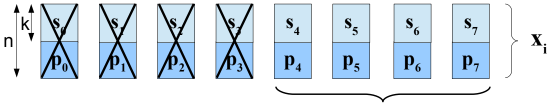


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

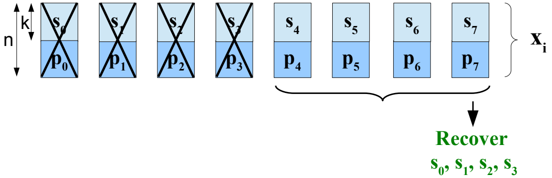


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

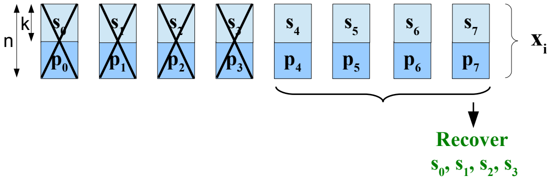


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes

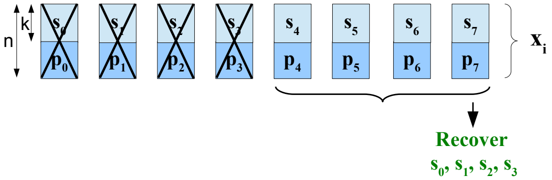


$$\mathbf{p}_i = \mathbf{s}_i \cdot \mathbf{H}_0 + \mathbf{s}_{i-1} \cdot \mathbf{H}_1 + \ldots + \mathbf{s}_{i-M} \cdot \mathbf{H}_M, \qquad \mathbf{H}_i \in \mathbb{F}_q^{k \times n-k}$$

Erasure Codes:

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

Baseline Codes



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

Erasure Codes:

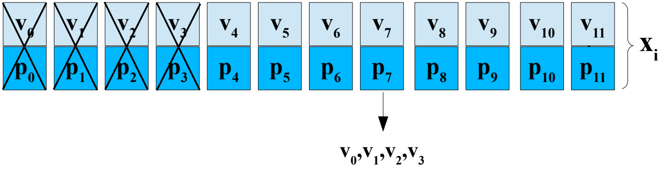
- Random Linear Codes
- Strongly-MDS Codes (Gabidulin'88, Gluesing-Luerssen'06)

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

Streaming Code - Example

$B = 4$, $T = 8$

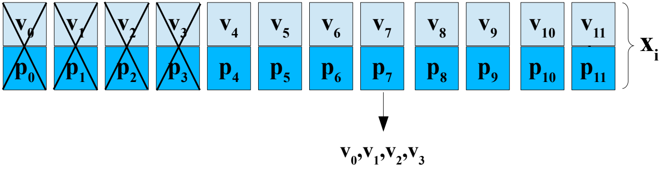
Rate $1/2$ Baseline Erasure Codes, $T = 7$



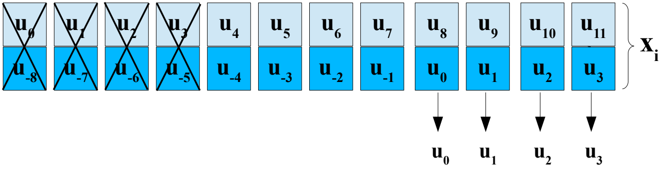
Streaming Code - Example

$B = 4$, $T = 8$

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

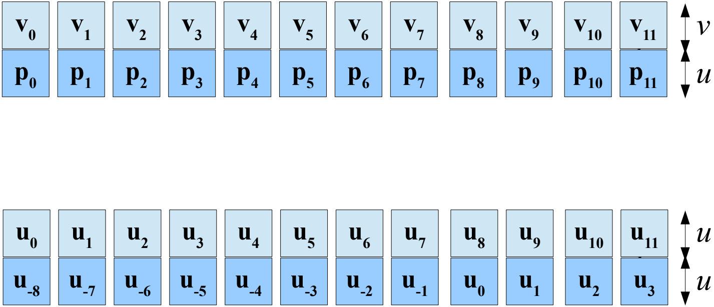


Rate $1/2$ Repetition Code, $T = 8$



Streaming Code - Example

$B = 4$, $T = 8$



Streaming Code - Example

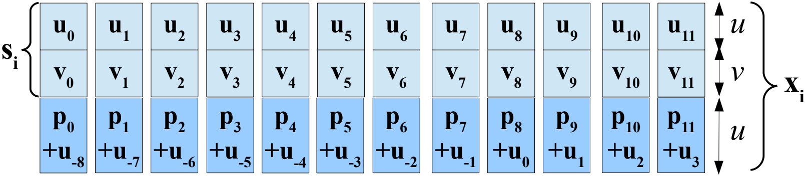
$B = 4$, $T = 8$

$\mathbf{u}_0$	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	$\mathbf{u}_4$	$\mathbf{u}_5$	$\mathbf{u}_6$	$\mathbf{u}_7$	$\mathbf{u}_8$	$\mathbf{u}_9$	$\mathbf{u}_{10}$	$\mathbf{u}_{11}$	u
$\mathbf{v}_0$	$\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$	$\mathbf{v}_9$	$\mathbf{v}_{10}$	$\mathbf{v}_{11}$	v
$\mathbf{p}_0$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_6$	$\mathbf{p}_7$	$\mathbf{p}_8$	$\mathbf{p}_9$	$\mathbf{p}_{10}$	$\mathbf{p}_{11}$	u
$\mathbf{u}_{-8}$	$\mathbf{u}_{-7}$	$\mathbf{u}_{-6}$	$\mathbf{u}_{-5}$	$\mathbf{u}_{-4}$	$\mathbf{u}_{-3}$	$\mathbf{u}_{-2}$	$\mathbf{u}_{-1}$	$\mathbf{u}_0$	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	u

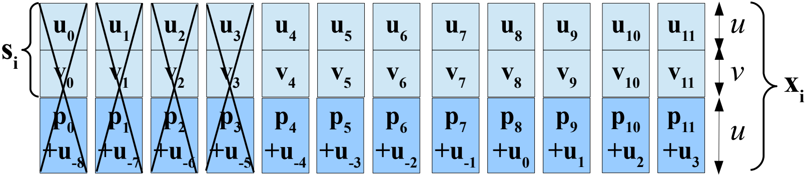
$$R = \frac{u+v}{3u+v} = \frac{1}{2}$$

Streaming Code - Example

$B = 4$, $T = 8$



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



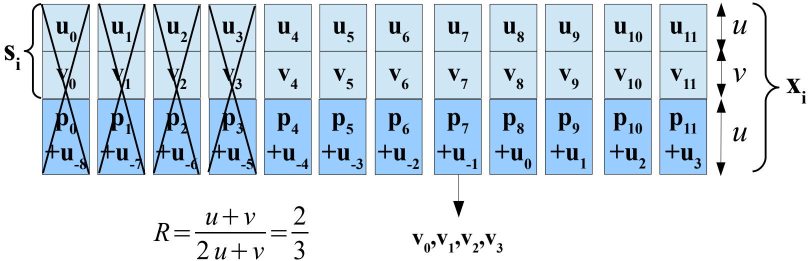
$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

Encoding:

1. Split each source symbol into 2 groups $\mathbf{s}_i = (\mathbf{u}_i, \mathbf{v}_i)$
2. Apply Erasure code to the $\mathbf{v}_i$ stream generating $\mathbf{p}_i$ parities
3. Repeat the $\mathbf{u}_i$ symbols with a shift of T
4. Combine the repeated $\mathbf{u}_i$'s with the $\mathbf{p}_i$'s

Streaming Code - Example

$B = 4$, $T = 8$



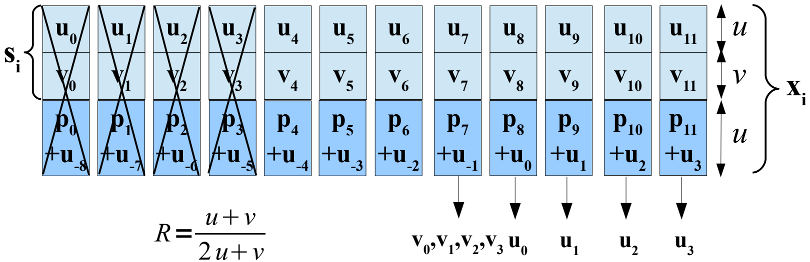
$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

Encoding:

1. Split each source symbol into 2 groups $\mathbf{s}_i = (\mathbf{u}_i, \mathbf{v}_i)$
2. Apply Erasure code to the $\mathbf{v}_i$ stream generating $\mathbf{p}_i$ parities
3. Repeat the $\mathbf{u}_i$ symbols with a shift of T
4. Combine the repeated $\mathbf{u}_i$'s with the $\mathbf{p}_i$'s

Streaming Code - Example

$B = 4$, $T = 8$

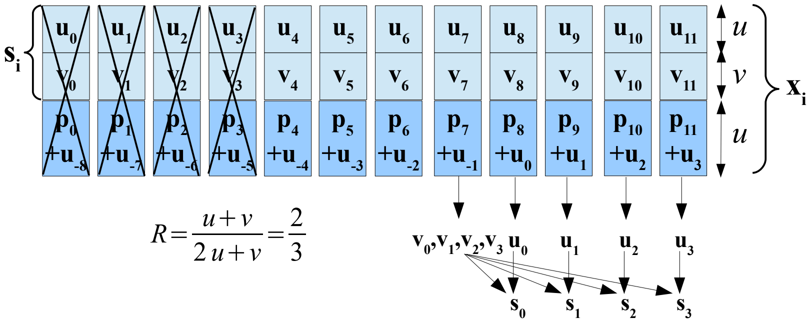


Encoding:

1. Split each source symbol into 2 groups $\mathbf{s}_i = (\mathbf{u}_i, \mathbf{v}_i)$
2. Apply Erasure code to the $\mathbf{v}_i$ stream generating $\mathbf{p}_i$ parities
3. Repeat the $\mathbf{u}_i$ symbols with a shift of T
4. Combine the repeated $\mathbf{u}_i$'s with the $\mathbf{p}_i$'s

Streaming Code - Example

$B = 4$, $T = 8$



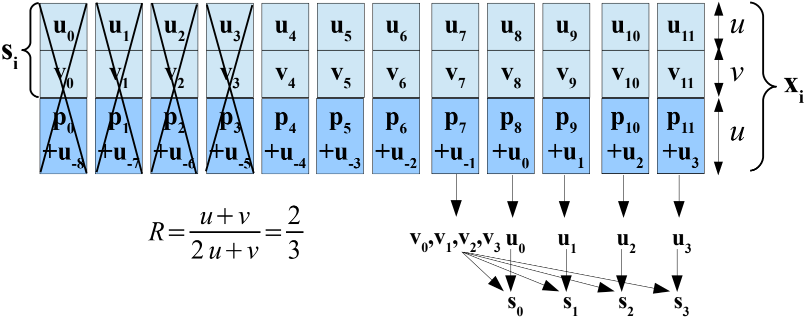
$$R = \frac{u+v}{2u+v} = \frac{2}{3}$$

Encoding:

1. Split each source symbol into 2 groups $\mathbf{s}_i = (\mathbf{u}_i, \mathbf{v}_i)$
2. Apply Erasure code to the $\mathbf{v}_i$ stream generating $\mathbf{p}_i$ parities
3. Repeat the $\mathbf{u}_i$ symbols with a shift of T
4. Combine the repeated $\mathbf{u}_i$'s with the $\mathbf{p}_i$'s

Streaming Code - Example

$B = 4$, $T = 8$



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

Encoding:

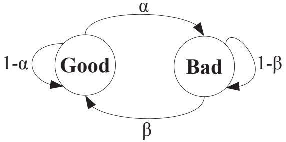
1. Split each source symbol into 2 groups $\mathbf{s}_i = (\mathbf{u}_i, \mathbf{v}_i)$
2. Apply Erasure code to the $\mathbf{v}_i$ stream generating $\mathbf{p}_i$ parities
3. Repeat the $\mathbf{u}_i$ symbols with a shift of T
4. Combine the repeated $\mathbf{u}_i$'s with the $\mathbf{p}_i$'s

Simulation

Gilbert-Eliott Channel $(\alpha, \beta) = (5 \times 10^{-4}, 0.5)$, $T = 12$ and $R = 12/23$

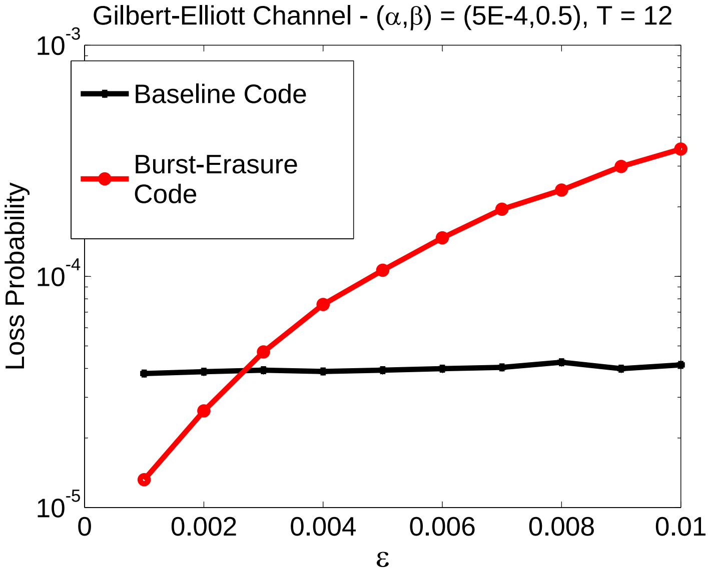
Gilbert Elliott Channel

- Good State: $\Pr(\text{loss}) = \varepsilon$
- Bad State: $\Pr(\text{loss}) = 1$



Simulation

Gilbert-Eliott Channel $(\alpha, \beta) = (5 \times 10^{-4}, 0.5)$, $T = 12$ and $R = 12/23$



Problem Setup

- Source Model : i.i.d. sequence $\mathsf{s}[t] \sim p_{\mathsf{s}}(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- Streaming Encoder: $\mathsf{x}[t] = f_t\left(\mathsf{s}[1], \ldots, \mathsf{s}[t]\right)$, $\mathsf{x}[t] \in (\mathbb{F}_q)^n$
- Erasure Channel: (Sliding Window Model)
- Delay-Constrained Decoder: $\hat{\mathsf{s}}[t] = g_t(\mathsf{y}[1], \ldots, \mathsf{y}[t+T])$
- Rate $R = \frac{k}{n}$

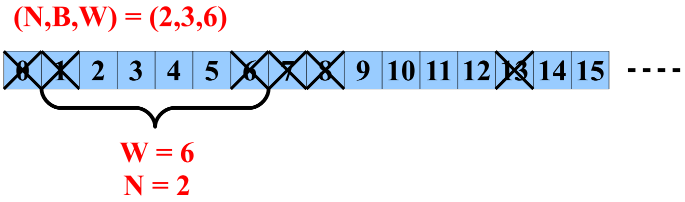
Sliding Window Erasure Channel

In any sliding window of length W, the channel can introduce only one of the following:

- An erasure burst of maximum length B
- Upto N erasures in arbitrary positions

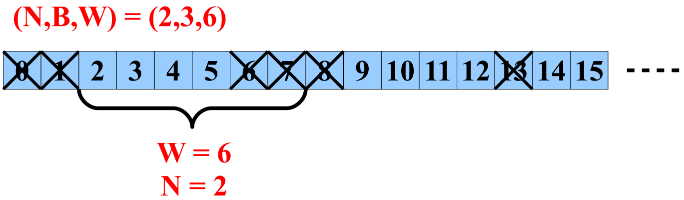
Problem Setup

- Source Model : i.i.d. sequence $s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- Streaming Encoder: $x[t] = f_t\,(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel: (Sliding Window Model)
- Delay-Constrained Decoder: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$
- Rate $R = \frac{k}{n}$



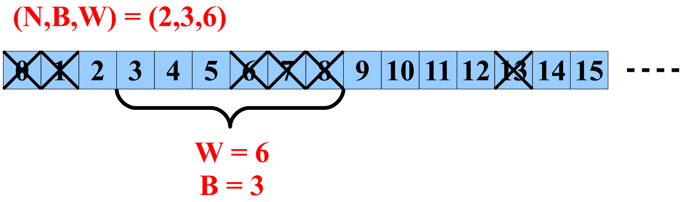
Problem Setup

- Source Model : i.i.d. sequence $s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- Streaming Encoder: $x[t] = f_t(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel: (Sliding Window Model)
- Delay-Constrained Decoder: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$
- Rate $R = \frac{k}{n}$



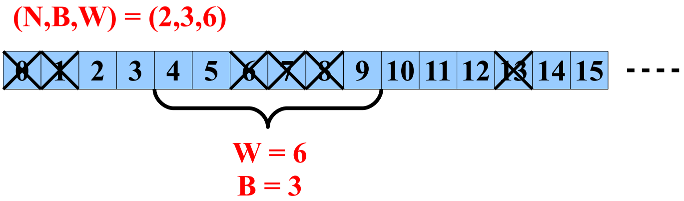
Problem Setup

- Source Model : i.i.d. sequence $s[t] \sim p_s(\cdot) = \mathrm{Unif}\{(\mathbb{F}_q)^k\}$
- Streaming Encoder: $x[t] = f_t\,(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel: (Sliding Window Model)
- Delay-Constrained Decoder: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$
- Rate $R = \frac{k}{n}$



Problem Setup

- Source Model : i.i.d. sequence $s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- Streaming Encoder: $x[t] = f_t\,(s[1], \ldots, s[t])$, $x[t] \in (\mathbb{F}_q)^n$
- Erasure Channel: (Sliding Window Model)
- Delay-Constrained Decoder: $\hat{s}[t] = g_t(y[1], \ldots, y[t+T])$
- Rate $R = \frac{k}{n}$



Capacity: $C(N, B, W, T)$

Main Result

Theorem

Consider the $\mathcal{C}(N, B, W)$ channel, with $W \geq B+1$, and let the delay be T.

Upper-Bound *For any rate R code, we have:*

$$\left(\frac{R}{1-R}\right) B + N \leq \min(W, T+1)$$

Lower-Bound: *There exists a rate R code that satisfies:*

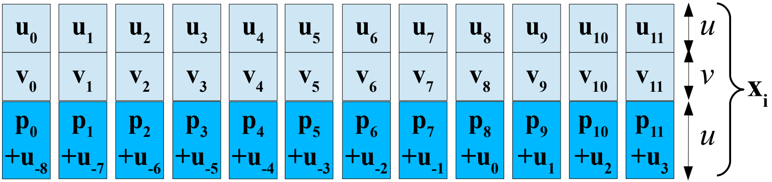
$$\left(\frac{R}{1-R}\right) B + N \geq \min(W, T+1) - 1.$$

The gap between the upper and lower bound is 1 unit of delay.

Streaming Codes - Isolated Erasures

$C(N \geq 2$, B, W)

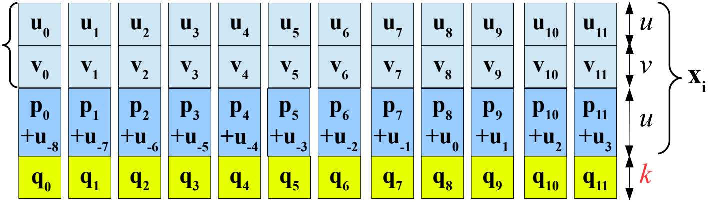
$$T = 8$$



- Erasures at time $t = 0$ and $t = 8$
- $\mathbf{u}_0$ cannot be recovered due to a repetition code

Proposed Approach: Layering

$C(N \geq 2$, B, W)



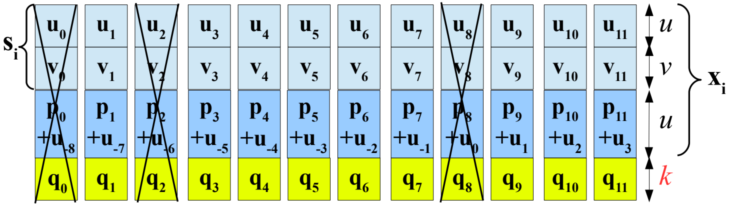
Layered Code Design

- Burst-Erasure Streaming Code $\mathcal{C}_1 : (\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T})$
- Erasure Code: $\mathbf{q}_i = \sum_{t=1}^{M} \mathbf{u}_{i-t} \cdot \mathbf{H}_t^u$, $\mathbf{q}_i \in \mathbb{F}_q^k$
- $\mathcal{C}_2$: $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T}, \mathbf{q}_i)$

$$R = \frac{T}{T+B+k}, \qquad k = \frac{N}{T-N+1}B$$

Proposed Approach: Layering

$C(N \geq 2$, B, W)



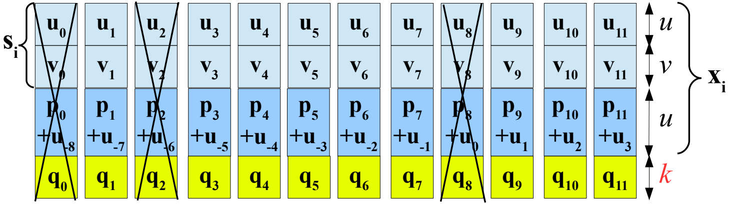
Layered Code Design

- Burst-Erasure Streaming Code $\mathcal{C}_1 : (\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T})$
- Erasure Code: $\mathbf{q}_i = \sum_{t=1}^{M} \mathbf{u}_{i-t} \cdot \mathbf{H}_t^u$, $\mathbf{q}_i \in \mathbb{F}_q^k$
- $\mathcal{C}_2$: $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T}, \mathbf{q}_i)$

$$R = \frac{T}{T+B+k}, \qquad k = \frac{N}{T-N+1}B$$

Proposed Approach: Layering

$C(N \geq 2$, B, W)



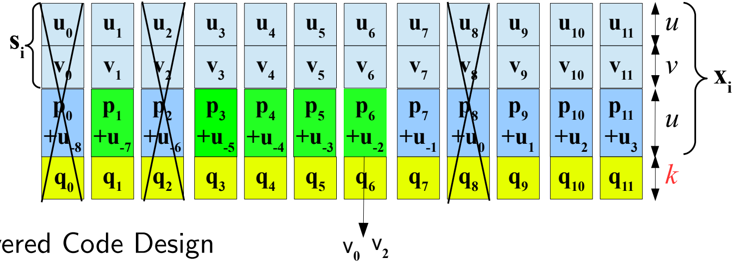
Layered Code Design

- Burst-Erasure Streaming Code $\mathcal{C}_1 : (\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T})$
- Erasure Code: $\mathbf{q}_i = \sum_{t=1}^{M} \mathbf{u}_{i-t} \cdot \mathbf{H}_t^u$, $\mathbf{q}_i \in \mathbb{F}_q^k$
- $\mathcal{C}_2$: $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T}, \mathbf{q}_i)$

$$R = \frac{T}{T+B+k}, \qquad k = \frac{N}{T-N+1}B$$

Proposed Approach: Layering

$C(N \geq 2$, B, W)



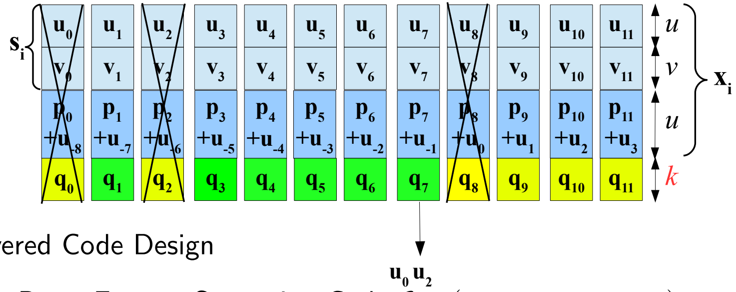
Layered Code Design

- Burst-Erasure Streaming Code $\mathcal{C}_1 : (\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T})$
- Erasure Code: $\mathbf{q}_i = \sum_{t=1}^{M} \mathbf{u}_{i-t} \cdot \mathbf{H}_t^u$, $\mathbf{q}_i \in \mathbb{F}_q^k$
- $\mathcal{C}_2$: $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T}, \mathbf{q}_i)$

$$R = \frac{T}{T+B+k}, \qquad k = \frac{N}{T-N+1} B$$

Proposed Approach: Layering

$C(N \geq 2$, B, W)



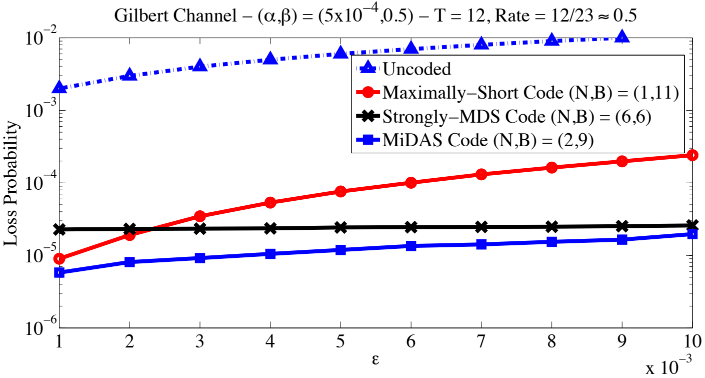
Layered Code Design

- Burst-Erasure Streaming Code $\mathcal{C}_1$: $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T})$
- Erasure Code: $\mathbf{q}_i = \sum_{t=1}^{M} \mathbf{u}_{i-t} \cdot \mathbf{H}_t^u$, $\mathbf{q}_i \in \mathbb{F}_q^k$
- $\mathcal{C}_2$: $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{p}_i + \mathbf{u}_{i-T}, \mathbf{q}_i)$

$$R = \frac{T}{T+B+k}, \qquad k = \frac{N}{T-N+1} B$$

Simulation Results

Gilbert-Eliott Channel $(\alpha, \beta) = (5 \times 10^{-4}, 0.5)$, $T = 12$ and $R = 12/23$

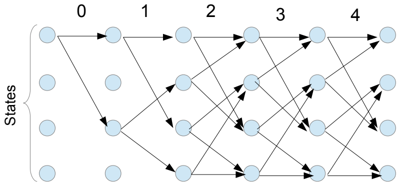


Code	N	B	Code	N	B
Strongly MDS	6	6	MiDAS	2	9
Burst-Erasure	1	11			

Distance and Span Properties

Distance and Span Properties

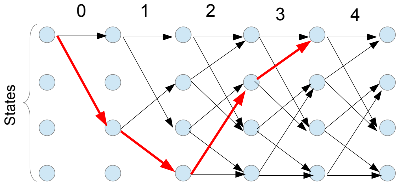
Consider (n, k, m) Convolutional code: $\mathbf{x}_i = \sum_{j=0}^{m} \mathbf{s}_{i-j} \mathbf{G}_j$



Trellis Diagram

Distance and Span Properties

Consider (n, k, m) Convolutional code: $\mathbf{x}_i = \sum_{j=0}^{m} \mathbf{s}_{i-j}\mathbf{G}_j$



Trellis Diagram – Free Distance

Distance and Span Properties

Consider (n,k,m) Convolutional code: $\mathbf{x}_i = \sum_{j=0}^{m} \mathbf{s}_{i-j}\mathbf{G}_j$

Column Distance: d_T

$$d_T = \min_{\substack{[\mathbf{s}_0,\ldots,\mathbf{s}_T] \\ \mathbf{s}_0 \neq 0}} \mathrm{wt}\left(\begin{bmatrix} \mathbf{s}_0 & \ldots & \mathbf{s}_T \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_T \\ 0 & \mathbf{G}_0 & \ldots & \mathbf{G}_{T-1} \\ \vdots & & \ddots & \vdots \\ 0 & & \ldots & \mathbf{G}_0 \end{bmatrix} \right)$$

Distance and Span Properties

Consider (n, k, m) Convolutional code: $\mathbf{x}_i = \sum_{j=0}^{m} \mathbf{s}_{i-j}\mathbf{G}_j$

Column Distance: d_T

$$d_T = \min_{\substack{[\mathbf{s}_0,\ldots,\mathbf{s}_T] \\ \mathbf{s}_0 \neq 0}} \mathrm{wt}\left(\begin{bmatrix} \mathbf{s}_0 & \ldots & \mathbf{s}_T \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_T \\ 0 & \mathbf{G}_0 & \ldots & \mathbf{G}_{T-1} \\ \vdots & & \ddots & \vdots \\ 0 & & \ldots & \mathbf{G}_0 \end{bmatrix} \right)$$

Distance and Span Properties

Consider (n, k, m) Convolutional code: $\mathbf{x}_i = \sum_{j=0}^{m} \mathbf{s}_{i-j} \mathbf{G}_j$

Column Distance: d_T

$$d_T = \min_{\substack{[\mathbf{s}_0, \ldots, \mathbf{s}_T] \\ \mathbf{s}_0 \neq 0}} \mathrm{wt} \left(\begin{bmatrix} \mathbf{s}_0 & \ldots & \mathbf{s}_T \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_T \\ 0 & \mathbf{G}_0 & \ldots & \mathbf{G}_{T-1} \\ \vdots & & \ddots & \vdots \\ 0 & & \ldots & \mathbf{G}_0 \end{bmatrix} \right)$$

Column Span: c_T

$$c_T = \min_{\substack{[\mathbf{s}_0, \ldots, \mathbf{s}_T] \\ \mathbf{s}_0 \neq 0}} \mathrm{span} \left(\begin{bmatrix} \mathbf{s}_0 & \ldots & \mathbf{s}_T \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \ldots & \mathbf{G}_T \\ 0 & \mathbf{G}_0 & \ldots & \mathbf{G}_{T-1} \\ \vdots & & \ddots & \vdots \\ 0 & & \ldots & \mathbf{G}_0 \end{bmatrix} \right)$$

Column-Distance & Column Span Tradeoff

Theorem

Consider a $\mathcal{C}(N, B, W)$ channel with delay T and $W \geq T + 1$. A streaming code is feasible over this channel if and only if it satisfies: $d_T \geq N + 1$ and $c_T \geq B + 1$

Column-Distance & Column Span Tradeoff

Theorem

Consider a $\mathcal{C}(N, B, W)$ *channel with delay* T *and* $W \geq T + 1$*. A streaming code is feasible over this channel if and only if it satisfies:* $d_T \geq N + 1$ *and* $c_T \geq B + 1$

Theorem

For any rate R *convolutional code and any* $T \geq 0$ *the Column-Distance* d_T *and Column-Span* c_T *satisfy the following:*

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

There exists a rate R *code (MiDAS Code) over a sufficiently large field that satisfies:*

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

Extensions

- **Burst plus Isolated Erasures**: Layered Coding Approach (Badr-K-Tan-Apostolopoulos 2013, Badr 2014)
- **Mismatched Source-Channel Rates**: (Patil-Badr-K-Tan 2013), (Leong-Ho 2012), (Leong-Qureshi-Ho 2013)
- **Multicast Streaming Codes**: Optimal Codes for certain parameters; Column-Span profile (Badr-Lui-K 2014)
- **Parallel Channels**: Lui-Badr-Khisti 2011
- **Multi-Source Models**: Lui 2011

Conclusions

Streaming Codes for Real-Time Streaming over Channels with Burst and Isolated Erasures

- Sliding Window Erasure Channel Model
- MiDAS Codes: Near Optimal Distance/Span Tradeoff
- Layering Approach
- Distance and Span Metrics

Future Work

- Improved constructions for short-inter burst gaps
- Systems Theoretic Approach (e.g. Dual Codes for MiDAS Codes)
- Analysis of probabilistic channels