(Structured) Coding for Real-Time Streaming Communication

Ashish Khisti

Department of Electrical and Computer Engineering
University of Toronto

Joint work with:
Ahmed Badr (Toronto), Farrokh Etezadi (Toronto), Wai-Tian Tan (Cisco), John Apostolopoulos (Cisco), Mitchell Trott (Luminate Wireless)

October 9, 2014
University of Michigan, Ann Arbor
Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Bit-Rate (Mbps)</th>
<th>MSDU (B)</th>
<th>Delay (ms)</th>
<th>PLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Conf.</td>
<td>2 Mbps</td>
<td>1500</td>
<td>100 ms</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Interactive Gaming</td>
<td>1 Mbps</td>
<td>512</td>
<td>50 ms</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Video Streaming</td>
<td>4 Mbps</td>
<td>1500</td>
<td>500 ms</td>
<td>$10^{-6}$</td>
</tr>
</tbody>
</table>
Multimedia Streaming

Packet-Loss Analysis

- Burst Losses - High Performance Degradation
- FEC vs Retransmission
Real-Time Streaming Communication

Streaming
Error Correction

Collaborators:
Ahmed Badr (Toronto)
John Apostolopoulos (Cisco)
Wai-Tian Tan (Cisco)

Streaming Compression

Collaborators:
Farrokh Etezadi (Toronto)
Mitchell Trott
Real-Time Streaming Communication

Streaming Error Correction

Collaborators:
Ahmed Badr (Toronto)
John Apostolopoulos (Cisco)
Wai-Tian Tan (Cisco)

Streaming Compression

Collaborators:
Farrokh Etezadi (Toronto)
Mitchell Trott

Physical Layer Security (Time Permitting)
Real-Time Communication System

Source stream
\( s[i] \in F_q^k \)

\( \Rightarrow \quad s[0] \)

Causal encoder: \( x[i] = f_i(s[0], s[1], \ldots, s[i]) \in F_q^n \)

Rate = \( \frac{k}{n} \)
Real-Time Communication System

Source stream
$s[i] \in F_q^k$

\[ x[i] = f_i(s[0], s[1], \ldots, s[i]) \in F_q^n \]
Rate = $\frac{k}{n}$
Real-Time Communication System

Source stream
\( s[i] \in \mathbb{F}_q^k \)

\[ x[i] = f_i(s[0], s[1], \ldots, s[i]) \in \mathbb{F}_q^n \]

Rate = \( \frac{k}{n} \)
Causal encoder: $x[i] = f_i(s[0], s[1], \ldots, s[i]) \in F_q^n$
Rate $= \frac{k}{n}$
Causal encoder: \( x[i] = f_i(s[0], s[1], \cdots, s[i]) \in F_q^n \)
Rate = \( \frac{k}{n} \)

Source stream
\( s[i] \in F_q^k \)
Real-Time Communication System

Source stream:
\[ s[i] \in F_q^k \]

Time:

Causal encoder:
\[ x[i] = f_i(s[0], s[1], \ldots, s[i]) \in F_q^n \]
Rate:
\[ \frac{k}{n} \]
Real-Time Communication System

Source stream: \( s[i] \in \mathbb{F}_q^k \)

Causal encoder: \( x[i] = f_i(s[0], s[1], \ldots, s[i]) \in \mathbb{F}_q^n \)

Rate: \( \frac{k}{n} \)

Time

\( s[0] \quad s[1] \quad s[2] \quad \ldots \quad s[i] \quad \ldots \quad s[T-1] \quad s[T] \quad s[T+1] \quad \ldots \)

\( x[0] \quad x[1] \quad x[2] \quad \ldots \quad x[i] \quad \ldots \quad x[T-1] \quad x[T] \quad x[T+1] \quad \ldots \)
Real-Time Communication System

Source stream
\( s[i] \in \mathbb{F}_q^k \)

\( \Rightarrow \)

\( s[0] \quad s[1] \quad s[2] \quad \cdots \quad s[i] \quad \cdots \quad s[T-1] \quad s[T] \quad s[T+1] \quad \cdots \)

\text{Causal encoder: } x[i] = f_i(s[0], s[1], \ldots, s[i]) \in \mathbb{F}_q^n

Rate = \frac{k}{n}

\( x[0] \quad x[1] \quad x[2] \quad \cdots \quad x[i] \quad \cdots \quad x[T-1] \quad x[T] \quad x[T+1] \quad \cdots \)

Packet Erasure Channel
\( y[i] = \ast \) for packet erasure; otherwise \( y[i] = x[i] \)

\( y[0] \quad y[1] \quad y[2] \quad \cdots \quad y[i] \quad \cdots \quad y[T-1] \quad y[T] \quad y[T+1] \quad \cdots \)
Real-Time Communication System

Source stream
$s[i] \in \mathbf{F}_q^k$

$\Rightarrow$

\[ s[0] \quad s[1] \quad s[2] \quad \ldots \quad s[i] \quad \ldots \quad s[T-1] \quad s[T] \quad s[T+1] \quad \ldots \]

Causal encoder: \[ x[i] = f_i(s[0], s[1], \ldots, s[i]) \in \mathbf{F}_q^n \]
Rate: \[ \frac{k}{n} \]

\[ x[0] \quad x[1] \quad x[2] \quad \ldots \quad x[i] \quad \ldots \quad x[T-1] \quad x[T] \quad x[T+1] \quad \ldots \]

Packet Erasure Channel
\[ y[i] = \star \text{ for packet erasure; otherwise } y[i] = x[i] \]

\[ y[0] \quad y[1] \quad y[2] \quad \ldots \quad y[i] \quad \ldots \quad y[T-1] \quad y[T] \quad y[T+1] \quad \ldots \]

Delay constrained decoder: \[ s[i] = g_i(y[0], y[1], \ldots, y[i+T]) \]
Real-Time Communication System

Source stream $s[i] \in \mathbb{F}_q^k$

$\Rightarrow s[0] \quad s[1] \quad s[2] \quad \cdots \quad s[i] \quad \cdots \quad s[T-1] \quad s[T] \quad s[T+1] \quad \cdots$

Causal encoder: $x[i] = f_i(s[0], s[1], \cdots, s[i]) \in \mathbb{F}_q^n$

Rate $= \frac{k}{n}$

Packet Erasure Channel
$y[i] = \star$ for packet erasure; otherwise $y[i] = x[i]$

$\Rightarrow y[0] \quad y[1] \quad y[2] \quad \cdots \quad y[i] \quad \cdots \quad y[T-1] \quad y[T] \quad y[T+1] \quad \cdots$

Delay constrained decoder: $s[i] = g_i(y[0], y[1], \cdots, y[i+T])$

Delay $T$
Real-Time Communication System

Source stream \( s[i] \in F_q \)

\( \rightarrow \)

\( s[0], s[1], s[2], \ldots, s[i], \ldots, s[T-1], s[T], s[T+1], \ldots \)

\( \Rightarrow \)

Causal encoder: \( x[i] = f_i(s[0], s[1], \ldots, s[i]) \in F_q^n \)
Rate: \( \frac{k}{n} \)

\( \xrightarrow{\rightarrow} \)

\( x[0], x[1], x[2], \ldots, x[i], \ldots, x[T-1], x[T], x[T+1], \ldots \)

Packet Erasure Channel

\( y[i] = * \) for packet erasure; otherwise \( y[i] = x[i] \)

\( \xrightarrow{\rightarrow} \)

\( y[0], y[1], y[2], \ldots, y[i], \ldots, y[T-1], y[T], y[T+1], \ldots \)

Delay constrained decoder: \( s[i] = g_i(y[0], y[1], \ldots, y[i+T]) \)

\( \xrightarrow{\rightarrow} \)

\( s[0], s[1], \ldots \)

Delay \( T \)

Delay \( T \)
Proposed Channel Model

Gilbert-Elliott Model

- Structural Properties
- Performance Gains
Proposed Channel Model

Gilbert-Elliott Model

Sliding Window Erasure Channel: $C(N, B, W)$

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

W = 6
N = 2
Sliding Window Erasure Channel: Remarks

\[(N, B, W) = (2, 3, 6)\]

\[W = 6\]

\[N = 2\]
(N,B,W) = (2,3,6)

W = 6
N = 2
Sliding Window Erasure Channel: Remarks

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

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

W = 6
B = 3

• $C(N = 1, B, W)$: Burst-Erasure Channel
(N,B,W) = (2,3,6)

W = 6
B = 3

\[ C(N = 1, B, W) \]: Burst-Erasure Channel
(N,B,W) = (2,3,6)

W = 6
B = 3

- C(N = 1, B, W): Burst-Erasure Channel
- C(N, B, W): “Worst-Case”
Problem Setup - Sliding Window Erasure Channel Model

- **Source Model**: i.i.d. sequence $s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\}$
- **Streaming Encoder**: $x[t] = f_t(s[0], \ldots, s[t]), x[t] \in (\mathbb{F}_q)^n$
- **Erasure Channel**: (Sliding Window Model)
- **Delay-Constrained Decoder**: $\hat{s}[t] = g_t(y[0], \ldots, y[t + T])$
- **Rate** $R = \frac{k}{n}$

### Streaming Capacity

A rate $R$ is achievable over the $C(N, B, W)$ channel, if there is a sequence of encoding and decoding functions, $f_t(\cdot)$ and $g_t(\cdot)$ respectively over a sufficiently large field $\mathbb{F}_q$, with delay $T$ and rate $R = \frac{k}{n}$. The supremum of achievable rates is the streaming capacity.
Problem Setup - Sliding Window Erasure Channel Model

- Source Model: i.i.d. sequence \( s[t] \sim p_s(\cdot) = \text{Unif}\{(\mathbb{F}_q)^k\} \)
- Streaming Encoder: \( x[t] = f_t(s[0], \ldots, s[t]), x[t] \in (\mathbb{F}_q)^n \)
- Erasure Channel: (Sliding Window Model)
- Delay-Constrained Decoder: \( \hat{s}[t] = g_t(y[0], \ldots, y[t+T]) \)
- Rate \( R = \frac{k}{n} \)

### Streaming Capacity

A rate \( R \) is achievable over the \( C(N, B, W) \) channel, if there is a sequence of encoding and decoding functions, \( f_t(\cdot) \) and \( g_t(\cdot) \) respectively over a sufficiently large field \( \mathbb{F}_q \), with delay \( T \) and rate \( R = \frac{k}{n} \). The supremum of achievable rates is the streaming capacity.

- Worst Case Definition
- Arbitrarily large field size
Consider the $C(N, B, W)$ 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.
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k} \]

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin’88, Gluesing-Luerssen’06)
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k} \]

Erasure Codes:

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

\[ \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} \]

\[ \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{bmatrix} \]

\[ \{ x_i \} \]
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k} \]

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin’88, Gluesing-Luerssen’06)
Baseline Codes - Full Rank Condition

\[
p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k}
\]

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin’88, Gluesing-Luerssen’06)
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k} \]

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin’88, Gluesing-Luerssen’06)
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_{q}^{k \times n-k} \]

Erasure Codes:

- Random Linear Codes
- Strongly-MDS Codes (Gabidulin’88, Gluesing-Luerssen’06)
Baseline Codes - Full Rank Condition

- Erasure Codes:
  \[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M \]

- Random Linear Codes

- Strongly-MDS Codes (Gabidulin’88, Gluesing-Luerssen’06)

Recover
\[ s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7 \]
\[ p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7 \]
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k} \]

Erasure Codes:

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

Recover \( s_0, s_1, s_2, s_3 \)
Baseline Codes - Full Rank Condition

\[ p_i = s_i \cdot H_0 + s_{i-1} \cdot H_1 + \ldots + s_{i-M} \cdot H_M, \quad H_i \in \mathbb{F}_q^{k \times n-k} \]

Erasure Codes:

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

\[
\begin{bmatrix}
p_4 \\
p_5 \\
p_6 \\
p_7 \\
\end{bmatrix} = \begin{bmatrix}
H_4 & H_3 & H_2 & H_1 \\
H_5 & H_4 & H_3 & H_2 \\
0 & H_5 & H_4 & H_3 \\
0 & 0 & H_5 & H_4 \\
\end{bmatrix} \begin{bmatrix}
s_0 \\
s_1 \\
s_2 \\
s_3 \\
\end{bmatrix}
\]
Streaming Code - Example

\[
B = 4, \quad T = 8, \quad R = \frac{T}{T+B} = \frac{2}{3}
\]

Rate 1/2 Baseline Erasure Codes, \( T = 7 \)

\[
\begin{align*}
\text{v}_0, \text{v}_1, \text{v}_2, \text{v}_3, \text{v}_4, \text{v}_5, \text{v}_6, \text{v}_7, \text{v}_8, \text{v}_9, \text{v}_{10}, \text{v}_{11} \end{align*}
\]

\[
\text{x}_i
\]

\[
\text{v}_0, \text{v}_1, \text{v}_2, \text{v}_3
\]
Streaming Code - Example

\( B = 4, \ T = 8, \ R = \frac{T}{T+B} = \frac{2}{3} \)

Rate 1/2 Baseline Erasure Codes, \( T = 7 \)

Rate 1/2 Repetition Code, \( T = 8 \)
Streaming Code - Example

\[ B = 4, \ T = 8, \ R = \frac{T}{T+B} = \frac{2}{3} \]
Streaming Code - Example

\[ B = 4, \ T = 8, \ R = \frac{T}{T+8} = \frac{2}{3} \]

\[
\begin{array}{cccccccccccc}
  u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} \\
  v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} \\
  p_0 & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 & p_{10} & p_{11} \\
  u_{-8} & u_{-7} & u_{-6} & u_{-5} & u_{-4} & u_{-3} & u_{-2} & u_{-1} & u_0 & u_1 & u_2 & u_3 \\
\end{array}
\]

\[ R = \frac{u+v}{3u+v} = \frac{1}{2} \]
Streaming Code - Example

\[ B = 4, \ T = 8, \ R = \frac{T}{T+B} = \frac{2}{3} \]

\[ R = \frac{u+v}{2u+v} = \frac{2}{3} \]
Streaming Code - Example

\[ B = 4, \ T = 8, \ R = \frac{T}{T+B} = \frac{2}{3} \]

\[
\begin{array}{cccccccccccc}
\text{\(s_i\)} & u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & u_{10} & u_{11} \\
\text{\(p_0\)} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} & v_{11} \\
\end{array}
\]

\[ R = \frac{u + v}{2u + v} \]

Encoding:

1. **Source Splitting**: \( s_i = (u_i, v_i) \), \( u_i \in \mathbb{F}_q^B \), \( v_i \in \mathbb{F}_q^{T-B} \)
2. **Erasure Code on \( v_i \)**: Generate \( v_i \rightarrow (v_i, p_i) \) where \( p_i \in \mathbb{F}_q^B \) is obtained from a Strongly-MDS code.
3. **Repetition Code on \( u_i \)**: Repeat the \( u_i \) symbols with a shift of \( T \)
4. **Merging**: Combine the repeated \( u_i \)'s with the \( p_i \)'s
5. **Rate**: \( R = \frac{T}{T+B} \)
Streaming Code - Example

\[ B = 4, \ T = 8, \ R = \frac{T}{T+B} = \frac{2}{3} \]

\[ R = \frac{u+v}{2u+v} = \frac{2}{3} \]

\[ v_0, v_1, v_2, v_3 \]

Encoding:

1. Source Splitting: \( s_i = (u_i, v_i) \), \( u_i \in \mathbb{F}_q^B \), \( v_i \in \mathbb{F}_q^{T-B} \)

2. Erasure Code on \( v_i \): Generate \( v_i \rightarrow (v_i, p_i) \) where \( p_i \in \mathbb{F}_q^B \) is obtained from a Strongly-MDS code.

3. Repetition Code on \( u_i \): Repeat the \( u_i \) symbols with a shift of \( T \)

4. Merging: Combine the repeated \( u_i \)'s with the \( p_i \)'s

5. Rate: \( R = \frac{T}{T+B} \)
Streaming Code - Example

$B = 4$, $T = 8$, $R = \frac{T}{T+B} = \frac{2}{3}$

Encoding:

1. **Source Splitting**: $s_i = (u_i, v_i)$, $u_i \in \mathbb{F}_q^B$, $v_i \in \mathbb{F}_q^{T-B}$
2. **Erasure Code on $v_i$**: Generate $v_i \rightarrow (v_i, p_i)$ where $p_i \in \mathbb{F}_q^B$ is obtained from a Strongly-MDS code.
3. **Repetition Code on $u_i$**: Repeat the $u_i$ symbols with a shift of $T$
4. **Merging**: Combine the repeated $u_i$'s with the $p_i$'s
5. **Rate**: $R = \frac{T}{T+B}$
Streaming Code - Example

\[ B = 4, \quad T = 8, \quad R = \frac{T}{T+B} = \frac{2}{3} \]

\[ R = \frac{u + v}{2u + v} = \frac{2}{3} \]

Encoding:

1. **Source Splitting**: \( s_i = (u_i, v_i) \), \( u_i \in \mathbb{F}_q^B \), \( v_i \in \mathbb{F}_q^{T-B} \)
2. **Erasure Code on** \( v_i \): Generate \( v_i \to (v_i, p_i) \) where \( p_i \in \mathbb{F}_q^B \) is obtained from a Strongly-MDS code.
3. **Repetition Code on** \( u_i \): Repeat the \( u_i \) symbols with a shift of \( T \)
4. **Merging**: Combine the repeated \( u_i \)’s with the \( p_i \)’s
5. **Rate**: \( R = \frac{T}{T+B} \)
Streaming Code - Example

\[ B = 4, \ T = 8, \ R = \frac{T}{T+B} = \frac{2}{3} \]

\[ R = \frac{u+v}{2u+v} = \frac{2}{3} \]

Encoding:

1. **Source Splitting**: \( s_i = (u_i, v_i) \), \( u_i \in \mathbb{F}_q^B \), \( v_i \in \mathbb{F}_q^{T-B} \)
2. **Erasure Code on \( v_i \)**: Generate \( v_i \rightarrow (v_i, p_i) \) where \( p_i \in \mathbb{F}_q^B \) is obtained from a Strongly-MDS code.
3. **Repetition Code on \( u_i \)**: Repeat the \( u_i \) symbols with a shift of \( T \)
4. **Merging**: Combine the repeated \( u_i \)'s with the \( p_i \)'s
5. **Rate**: \( R = \frac{T}{T+B} \)
**Source Splitting**: \( s_i = (u_i, v_i), u_i \in \mathbb{F}_q^B, v_i \in \mathbb{F}_q^{T-B} \)

**Erasure Code on** \( v_i \): Generate \( v_i \rightarrow (v_i, p_i) \) where \( p_i \in \mathbb{F}_q^B \) is obtained from a Strongly-MDS code.

**Repetition Code on** \( u_i \): Repeat the \( u_i \) symbols with a shift of \( T \)

**Merging**: Combine the repeated \( u_i \)'s with the \( p_i \)'s
Robust Extension: $\mathcal{C}(N, B, W)$ Channel

Layered Code Design

- **Burst-Erasure Streaming Code**: $(u_i, v_i, p_i + u_{i-T})$
- **Erasure Code**: $q_i = \sum_{t=1}^{M} u_{i-t} \cdot H_t^u, \quad q_i \in \mathbb{F}_q^k$
- **Concatenation**: $(u_i, v_i, p_i + u_{i-T}, q_i)$

$$R = \frac{T}{T + B + k}$$

- Attains the lower bound
Distance and Span Properties

Consider \((n, k, m)\) Convolutional code: \(x_i = \sum_{j=0}^{m} s_{i-j} G_j\)

Trellis Diagram

Column Distance in \([0,3]\)

Column Span in \([0,3]\)

Trellis Diagram – Free Distance
Consider \((n, k, m)\) Convolutional code: 
\[ x_i = \sum_{j=0}^{m} s_{i-j} G_j \]
Distance and Span Properties

Consider \((n, k, m)\) Convolutional code: \(x_i = \sum_{j=0}^{m} s_{i-j} G_j\)

Column Distance: \(d_T\)

\[
d_T = \min_{[s_0, \ldots, s_T]} \text{wt} \begin{bmatrix} s_0 & \ldots & s_T \end{bmatrix} \begin{bmatrix} G_0 & G_1 & \ldots & G_T \\ 0 & G_0 & \ldots & G_{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & G_0 \end{bmatrix}
\]
Consider \((n, k, m)\) Convolutional code: \(x_i = \sum_{j=0}^{m} s_{i-j}G_j\)

**Column Distance:** \(d_T\)

\[
d_T = \min_{[s_0, \ldots, s_T] \atop s_0 \neq 0} \text{wt} \begin{pmatrix} s_0 & \ldots & s_T \end{pmatrix} \begin{bmatrix} G_0 & G_1 & \cdots & G_T \\ 0 & G_0 & \cdots & G_{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & G_0 \end{bmatrix}
\]
Consider \((n, k, m)\) Convolutional code: \(x_i = \sum_{j=0}^{m} s_{i-j} G_j\)

Column Distance: \(d_T\)

\[
d_T = \min_{[s_0, \ldots, s_T] \neq 0} \text{wt} \left( \begin{bmatrix} s_0 & \ldots & s_T \end{bmatrix} \begin{bmatrix} G_0 & G_1 & \ldots & G_T \\ 0 & G_0 & \ldots & G_{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & G_0 \end{bmatrix} \right) \]

Column Span in \([0,3]\)
Consider \((n, k, m)\) Convolutional code: 
\[ x_i = \sum_{j=0}^{m} s_{i-j} G_j \]

\textbf{Column Distance: } \(d_T\)

\[ d_T = \min_{\left[ s_0, \ldots, s_T \right]} \text{wt} \begin{pmatrix} s_0 & \ldots & s_T \end{pmatrix} \begin{bmatrix} G_0 & G_1 & \ldots & G_T \\ 0 & G_0 & \ldots & G_{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & G_0 \end{bmatrix} \]

\textbf{Column Span: } \(c_T\)

\[ c_T = \min_{\left[ s_0, \ldots, s_T \right]} \text{span} \begin{pmatrix} s_0 & \ldots & s_T \end{pmatrix} \begin{bmatrix} G_0 & G_1 & \ldots & G_T \\ 0 & G_0 & \ldots & G_{T-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & G_0 \end{bmatrix} \]
Consider a $C(N, B, W)$ 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

Consider a $C(N, B, W)$ 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:

$$\left( \frac{R}{1 - R} \right) c_T + d_T \geq T + \frac{1}{1 - R}$$
Simulation Result
Gilbert-Elliott Channel \((\alpha, \beta) = (5 \times 10^{-4}, 0.5)\), \(T = 12\) and \(R \approx 0.5\)

**Gilbert Elliott Channel**

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

Gilbert Channel – \((\alpha, \beta) = (5 \times 10^{-4}, 0.5)\) – Simulation Length = \(10^7\)

![Gilbert Elliott Channel Diagram]
Simulation Results

Gilbert-Elliott Channel \((\alpha, \beta) = (5 \times 10^{-4}, 0.5), T = 12\) and \(R \approx 0.5\)

<table>
<thead>
<tr>
<th>Code</th>
<th>N</th>
<th>B</th>
<th>Code</th>
<th>N</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly MDS</td>
<td>6</td>
<td>6</td>
<td>MiDAS</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>Burst-Erasure</td>
<td>1</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Simulation Results-II

Fritchman Channel \((\alpha, \beta) = (1e-5, 0.5)\) and \(T = 40\) and \(R = 40/79\), 9 states

\[ \alpha = 1e-5 \]
\[ \beta = 0.5 \]
Simulation Results-II

Fritchman Channel \((\alpha, \beta) = (1e^{-5}, 0.5)\) and \(T = 40\) and \(R = 40/79\), 9 states

<table>
<thead>
<tr>
<th>Code</th>
<th>(N)</th>
<th>(B)</th>
<th>Code</th>
<th>(N)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly MDS</td>
<td>20</td>
<td>20</td>
<td>MiDAS-I</td>
<td>8</td>
<td>31</td>
</tr>
<tr>
<td>Burst Erasure</td>
<td>1</td>
<td>39</td>
<td>MiDAS-II</td>
<td>4</td>
<td>35</td>
</tr>
</tbody>
</table>
Multicast Streaming Codes

Motivation
- $B_1 < B_2$
- Receiver 1: Good Channel State
- Receiver 2: Weaker Channel State
- Delay adapts to Channel State
Multicast Streaming Codes

Capacity Function

- Capacity function $C(T_1, T_2, B_1, B_2)$
- Single User Upper Bound: $C \leq \min \left( \frac{T_1}{T_1+B_1}, \frac{T_2}{T_2+B_2} \right)$
- Concatenation Lower Bound: $C \geq \frac{1}{1+\frac{B_1}{T_1} + \frac{B_2}{T_2}}$
Multicast Streaming Setup

Capacity Function

- Capacity function $C(T_1, T_2, B_1, B_2)$
- Single User Upper Bound: $C \leq \min \left( \frac{T_1}{T_1+B_1}, \frac{T_2}{T_2+B_2} \right)$
- Concatenation Lower Bound: $C \geq \frac{1}{1+\frac{B_1}{T_1} + \frac{B_2}{T_2}}$
Assume w.l.o.g. \( B_2 \geq B_1 \)

\[
T_2 = \frac{B_2}{B_1} T_1 + B_1
\]

<table>
<thead>
<tr>
<th>Region</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( \frac{T_2}{T_2 + B_2} )</td>
</tr>
<tr>
<td>B</td>
<td>( \frac{T_1}{T_1 + B_1} )</td>
</tr>
<tr>
<td>C</td>
<td>( \frac{T_2 - B_1}{T_2 - B_1 + B_2} )</td>
</tr>
<tr>
<td>D</td>
<td>( \frac{T_1}{T_1 + B_2} )</td>
</tr>
<tr>
<td>E</td>
<td>Partial Characterization</td>
</tr>
</tbody>
</table>
Other Extensions

- **Mismatched Streaming Codes** (Patil-Badr-Khisti-Tan Asilomar 2013)
- **Partial Recovery Streaming Codes** (Badr-Khisti-Tan-Apostolopoulos JSTSP 2014)
- **Multiple Erasure Bursts** (Li-Khisti-Girod Asilomar 2011) - Interleaved Low-Delay Codes
- **Multiple Links** (Lui-Badr-Khisti CWIT 2011) - Layered coding for burst erasure channels
- **Multiple Source Streams with Different Decoding Delays** (Lui (Unpublished) 2011) - Embedded Codes

Other Results

- **Burst Erasure Channels**: Martinian and Sundberg (IT-2004)
- **Other Recent Results**: Leong-Ho (ISIT 2012), Leong-Qureshi-Ho (ISIT 2013)
Real-Time Streaming Communication

Streaming Error Correction

Collaborators:
Ahmed Badr (Toronto)
John Apostolopoulos (Cisco)
Wai-Tian Tan (Cisco)

Streaming Compression

Collaborators:
Farrokh Etezadi (Toronto)
Mitchell Trott
Compression Vs Error Propagation

GOP Picture Structure\(^1\)

<table>
<thead>
<tr>
<th></th>
<th>Compression</th>
<th>Error Propagation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictive Coding</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>Still Image Coding</td>
<td>×</td>
<td>√</td>
</tr>
</tbody>
</table>

- Interleaving Approach
- Error Control Coding

\(^1\)Source: http://www.networkwebcams.com
Motivation - Video Streaming

- Compression Efficiency
- Robustness to Error Propagation
- Predictive Coding
- Still Image Compression

Diagram:
- Compression Efficiency axis
- Robustness to Error Propagation axis
- Points for Predictive Coding and Still Image Compression

October 9, 2014 University of Michigan, Ann Arbor
Motivation - Video Streaming

- Compression
- Efficiency
- Robustness to Error Propagation
- Predictive Coding
- Still Image Compression
- MPEG

Graph showing the trade-off between Compression Efficiency and Robustness to Error Propagation, with Predictive Coding, MPEG, and Still Image Compression as points on the graph.
Motivation - Video Streaming

- Compression
- Efficiency
- Robustness to Error Propagation
- Predictive Coding
- Still Image Compression
- MPEG

Fundamental Tradeoff
Information Theoretic Model

System Parameters:

- Compression Rate: $R$
- Erasure Burst Length: $B$
- Recovery Window: $W$
- Lossless or Lossy Recovery
Problem Setup

- **Source Model**: Sequence of vectors — Temporally Markov, Spatially i.i.d. (see Viswanathan and Berger ('00), Wang and Xu ('10), Song-Chen-Wang-Liu ('13), Ma-Ishwar ('11))

\[
\Pr(s^n_i | s^n_{i-1}, s^n_{i-2}, \ldots) = \prod_{j=1}^{n} \Pr(s_{ij} | s_{i-1,j})
\]

- **Channel Model**: Burst Erasure Model

\[
g_i = \begin{cases} 
  f_i, & i \notin \{j, j + 1, \ldots, j + B - 1\} \\
  *, & \text{otherwise}
\end{cases}
\]
Problem Setup

- **Source Model**: Sequence of vectors — Temporally Markov, Spatially i.i.d. (see Viswanathan and Berger ('00), Wang and Xu ('10), Song-Chen-Wang-Liu ('13), Ma-Ishwar ('11))

  \[
  \Pr(s^n_i|s^n_{i-1}, s^n_{i-2}, \ldots) = \prod_{j=1}^{n} \Pr(s_{ij}|s_{i-1,j})
  \]

- **Channel Model**: Burst Erasure Model

  \[
  g_i = \begin{cases} 
  f_i, & i \notin \{j, j+1, \ldots, j+B-1\} \\
  \ast, & \text{otherwise}
  \end{cases}
  \]

- **Encoder**: \( \mathcal{F}_i : \{s^n_0, \ldots, s^n_i\} \rightarrow f_i \in \{1, 2, \ldots, 2^{nR}\} \).

- **Decoder**: \( \mathcal{G}_i : \{g_0, \ldots, g_i\} \rightarrow \hat{s}_i^n \)

  Lossless Recovery:

  \[
  \Pr(s_i^n \neq \hat{s}_i^n) \leq \varepsilon_n
  \]

  except for \( i \in [j, \ldots, j+B+W-1] \).
Definition (Rate-Recovery Function)

The minimum compression rate $R$ that is achieved when:

- Burst-Erasure Length $= B$
- Recovery Window $= W$
- Lossless or Lossy Recovery

\[
\begin{align*}
    s^n_0 & \rightarrow s^n_1 \rightarrow s^n_2 \rightarrow \cdots s^n_{j-1} \rightarrow s^n_j \rightarrow s^n_{j+1} \cdots s^n_{j+B-1} \rightarrow s^n_{j+B} \rightarrow \cdots \rightarrow s^n_{j+B+W-1} \rightarrow s^n_{j+B+W} \\
    f_0 & \downarrow f_1 \downarrow f_2 \downarrow f_{j-1} \downarrow f_j \downarrow f_{j+1} \downarrow f_{j+B-1} \downarrow f_{j+B} \downarrow f_{j+B+W-1} \downarrow f_{j+B+W} \\
    s^n & \downarrow s^n \downarrow s^n \downarrow s^n \downarrow s^n \downarrow \cdots \downarrow \cdots \downarrow \cdots \downarrow \cdots \downarrow \cdots \downarrow s^n \\
    \hat{s}^n & \downarrow \hat{s}^n \downarrow \hat{s}^n \downarrow \hat{s}^n \downarrow \cdots \downarrow \cdots \downarrow \cdots \downarrow \cdots \downarrow \cdots \downarrow \hat{s}^n \\
    \text{Erased} & \quad \text{Not to be recovered} \\
    \text{Error Propagation Window}
\end{align*}
\]
Theorem (Upper and Lower Bounds - Lossless Case)

\[
R^+(B, W) = H(s_1|s_0) + \frac{1}{W + 1} I(s_B; s_{B-1}|s_{-1})
\]

\[
R^-(B, W) = H(s_1|s_0) + \frac{1}{W + 1} I(s_{B+W}; s_{B-1}|s_{-1})
\]

- Upper bound: Binning based scheme.
- Upper and Lower Bounds Coincide: \( W = 0 \) and \( W \to \infty \).
Let $W = 1$. Encoding of $s^n_j, s^n_{j+1}$
Let $W = 1$. Encoding of $s^n_j, s^n_{j+1}$
Let $W = 1$. Encoding of $s^n_j, s^n_{j+1}$

Lower Bound: $R_j + R_{j+1} \geq H(s_j|s_{j-1}, s_{j+1}) + H(s_{j+1}|s_{j-B-1})$. 

Stationary Gauss-Markov Source, $s_i \sim \mathcal{N}(0, 1)$.

\[ s_i = \rho s_{i-1} + z_i, \quad z_i \sim \mathcal{N}(0, 1 - \rho^2) \perp \{s_j\}_{j<i} \]
Stationary Gauss-Markov Source, $s_i \sim \mathcal{N}(0, 1)$.

$$s_i = \rho s_{i-1} + z_i, \quad z_i \sim \mathcal{N}(0, 1 - \rho^2) \perp \{s_j\}_{j<i}$$

Quadratic Distortion Measure:

$$E \left[ \frac{1}{n} \sum_{k=1}^{n} (s_i(k) - \hat{s}_i(k))^2 \right] \leq D,$$
Stationary Gauss-Markov Source, $s_i \sim \mathcal{N}(0, 1)$.

$$s_i = \rho s_{i-1} + z_i, \quad z_i \sim \mathcal{N}(0, 1 - \rho^2) \perp \{s_j\}_{j < i}$$

Quadratic Distortion Measure:

$$E \left[ \frac{1}{n} \sum_{k=1}^{n} (s_i(k) - \hat{s}_i(k))^2 \right] \leq D,$$

No Recovery Period, $W = 0$. 

\[ s_0 \rightarrow s_1 \rightarrow \cdots s_{j-1} \rightarrow s_j \rightarrow s_{j+1} \cdots s_{j+B-1} \rightarrow s_{j+B} \]

\[ f_0 \quad f_1 \quad f_{j-1} \quad f_j \quad f_{j+1} \quad f_{j+B-1} \quad f_{j+B} \]

\[ \hat{s}_0 \quad \hat{s}_1 \quad \hat{s}_{j-1} \quad \hat{s}_j \quad \hat{s}_{j+B} \]

Erased
Stationary Gauss-Markov Source, \( s_i \sim \mathcal{N}(0, 1) \).

\[
s_i = \rho s_{i-1} + z_i, \quad z_i \sim \mathcal{N}(0, 1 - \rho^2) \perp \{s_j\}_{j<i}
\]

Quadratic Distortion Measure:

\[
E \left[ \frac{1}{n} \sum_{k=1}^{n} (s_i(k) - \hat{s}_i(k))^2 \right] \leq D,
\]

No Recovery Period, \( W = 0 \)

- Lossy Rate Recovery Function \( R(B, D) \)
- High Resolution Limit \( R(B, D) = \frac{1}{2} \log \left( \frac{1 - \rho^2(B+1)}{D} \right) + o_D(1) \).
- Upper and Lower bounds.
- \( W > 0 \): Novel Hybrid Coding Schemes
Gauss-Markov Sources

$W > 0$ (In Progress)

- **Predictive Coding:** $u_i = s_i - \hat{s}_i(u_{i-\infty}^{i-1}) + n_i$
- **Memoryless Quantize and Binning:** $u_i = s_i + n_i$
- **Truncated Prediction (Hybrid):** $u_i = s_i - \hat{s}_i(u_{i-W}^i) + n_i$
Gauss-Markov Sources

\( W > 0 \) (In Progress)

- **Predictive Coding:** \( u_i = s_i - \hat{s}_i(u_{i-1}^i) + n_i \)
- **Memoryless Quantize and Binning:** \( u_i = s_i + n_i \)
- **Truncated Prediction (Hybrid):** \( u_i = s_i - \hat{s}_i(u_{i-W}^i) + n_i \)

![Graph](image-url)  
(a) \( B = W = 1 \)
Block Fading Channels

\[ y_i = h_i x_i + z_i, \quad i = 1, 2, \ldots \]

- Block Fading Channels: \( n \) symbols per block
- Source Packet: One in each coherence block \( nR \) bits
- Decoding Delay: \( T \) coherence blocks
- Quasi-Static Model: \( T = 1 \)
- Ergodic Model \( T \to \infty \)
Theorem (Khisti-Draper, IT-Trans To Appear, 2014)

The diversity multiplexing tradeoff for streaming source with a delay of $T$ coherence blocks and a block-fading channel model is

$$d(r) = T d_1(r)$$

where $d_1(r)$ is the quasi-static DMT.

- Coding Scheme: Time-Sharing
- Upper Bound: Outage Amplification Argument
Achievability: Fading Channels

$T = 2$

- $T$— parallel channel code
- Divide each coherence block into $T$ sub-blocks
- Apply parallel channel code across $T$ sub-blocks
Achievability: Fading Channels

\( T = 2 \)

- \( T \) parallel channel code
- Divide each coherence block into \( T \) sub-blocks
- Apply parallel channel code across \( T \) sub-blocks
Achievability: Fading Channels

\( T = 2 \)

- \( T \) parallel channel code
- Divide each coherence block into \( T \) sub-blocks
- Apply parallel channel code across \( T \) sub-blocks
Achievability: Fading Channels

$T = 2$

- $T$— parallel channel code
- Divide each coherence block into $T$ sub-blocks
- Apply parallel channel code across $T$ sub-blocks

DMT: $d(r) = 2 - 2r$
Physical Layer Security
Two-way block-fading channel

Coherence Period: $T$

Channel Reciprocity

Non Coherent Main-Channel, Perfect CSI for eavesdropper

Interactive Communication
Secret-Key Generation over Fading Channels
Khisti, IT-Trans Submitted Aug. 2013

High SNR Capacity:

\[ C \approx \frac{1}{T} I(h_{AB}; h_{BA}) + E \left[ \log \left( 1 + \frac{|h_{AB}|^2}{|g_A|^2} \right) \right] + E \left[ \log \left( 1 + \frac{|h_{BA}|^2}{|g_B|^2} \right) \right] \]

Optimal Scheme: Training + Interactive Communication

Practical Codes for MIMO Wiretap Channel (A. Khina and Y. Kochman ISIT 2014)
  - V-Blast Type Decomposition
  - Scalar AWGN Wiretap Codebooks

MIMO Arbitrarily Varying Wiretap Channel (A. Yener and X. He, IT-Trans 2013, T-COMM 2014)

Fading Wiretap Channel with Imperfect CSI (Z. Rezki and M. S. Alouini, T-COMM 2014, T-Wireless 2014)

Private Broadcasting (T. Liu IT-Trans 2014)

Secure Broadcasting with Independent Secret Keys (R. Schaefer, CISS 2014)
Real-Time Streaming Communication

- **Part I: Channel Coding**
  - Channels with Burst and Isolated Erasures
  - Explicit Codes
  - New Distance and Span Metrics

- **Part II: Source Coding**
  - Tradeoff between compression rate and error propagation
  - Lossless Recovery
  - Gaussian Sources with Quadratic Distortion