Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?

Daniel J. Costello, Jr.
Dept. of Electrical Engineering,
University of Notre Dame

University of Michigan, Nov. 17, 2016

Research Collaborators: David Mitchell, Michael Lentmaier, and Ali Pusane
Outline

- **From Shannon to Modern Coding Theory**
  - Channel capacity, structured codes, random codes, LDPC codes

- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions

- **Spatially Coupled LDPC Codes**
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- **Practical Considerations**
  - Window decoding, performance, latency, and complexity comparisons to LDPC block codes, rate-compatibility, implementation aspects
Shannon's Legacy

Claude Elwood Shannon
Father of Information Theory
Shannon's Legacy

Shannon’s Theory Was Invented at Bell Labs

Bell Labs in Murray Hill, New Jersey
Three Great Successes of Information Theory

- Source Coding for Data Compression
- Secret Coding (Cryptography) for Data Security
- Channel Coding for Data Reliability (the focus of this presentation)
Shannon's Legacy

1948

Bandwidth Efficiency, $\eta$ (bits/2D)

Power Efficiency, $E_b/N_0$ (dB)

Capacity Bound
Power efficiency, or signal-to-noise ratio (SNR), represents the quality of the channel and is defined as the ratio of the average information symbol energy $E_b$ to the single-sided power spectral density $N_0$ of the noise. It is usually given in decibels (dB).

$$E_b/N_0 (dB) = 10 \log_{10} \frac{E_b}{N_0}$$
Bandwidth efficiency is normalized to the number of information bits transmitted per 2-dimensional signal (bits/2D).

For example, uncoded 8PSK transmits $\eta = 3$ bits/2D and uncoded 16QAM transmits $\eta = 4$ bits/2D.
Shannon's capacity theorem lower bounds the channel quality for which there exists a code that can achieve arbitrarily reliable transmission. For the AWGN channel, this lower bound is

\[ \frac{E_b}{N_0} > \frac{2^\eta - 1}{\eta} \]
Shannon's Legacy

1948

Coding theory playing field

Bandwidth Efficiency, $\eta$ (bits/2D)

Power Efficiency, $E_b/N_0$ (dB)
Shannon showed that random codes with large block length can achieve capacity, but...
Shannon showed that random codes with large block length can achieve capacity, but...

... code structure (algebraic/topological) is required in order to permit decoding with reasonable complexity.
Shannon showed that random codes with large block length can achieve capacity, but...

... code structure (algebraic/topological) is required in order to permit decoding with reasonable complexity.

“Almost all codes are good...
Shannon showed that random codes with large block length can achieve capacity, but…

… code structure (algebraic/topological) is required in order to permit decoding with reasonable complexity.

“Almost all codes are good... except those we can think of.”
The coding dilemma

Shannon showed that random codes with large block length can achieve capacity, but...

... code structure (algebraic/topological) is required in order to permit decoding with reasonable complexity.

“Almost all codes are good... except those we can think of.”

Solution: Construct random-like codes with just enough structure to allow efficient decoding

→ Modern Coding Theory
LDPC Codes: motivation (for a target BER $10^{-5}$)
LDPC Codes: motivation (for a target BER $10^{-5}$)
**LDPC Codes: motivation**  
(for a target BER $10^{-5}$)

- Capacity Bound
- Cutoff Rate

**Power Efficiency, $E_b/N_0$ (dB)**

**Bandwidth Efficiency, $\eta$ (bits/2D)**

- BPSK/QPSK capacity bounds
- Uncoded BPSK/QPSK
- Hamming (7,4)
LDPC Codes: motivation
(for a target BER $10^{-5}$)

![Graph showing bandwidth efficiency and power efficiency for different codes](image)

- **Capacity Bound**
- **Cutoff Rate**
- **BPSK/QPSK capacity bounds**
- **Uncoded BPSK/QPSK**
- **Hamming (7,4)**
- **Golay (24,12)**
LDPC Codes: motivation
(for a target BER $10^{-5}$)

![Graph showing bandwidth efficiency vs. power efficiency for various codes.](image)

- Uncoded BPSK/QPSK
- BPSK/QPSK capacity bounds
- BCH (255, 123)
- Hamming (7, 4)
- Golay (24, 12)
LDPC Codes: motivation
(for a target BER $10^{-5}$)

![Diagram showing bandwidth efficiency and power efficiency for various codes.]

- BPSK/QPSK capacity bounds
- Uncoded BPSK/QPSK
- BCH (255,123)
- RS (64,32)
- Golay (24,12)
- Hamming (7,4)
LDPC Codes: motivation
(for a target BER $10^{-5}$)
LDPC Codes: motivation
(for a target BER $10^{-5}$)
LDPC Codes: motivation
(for a target BER $10^{-5}$)
Random-like codes (2000s - today)

- Turbo codes use a long pseudorandom interleaver

- 3G and 4G telephony standards HSPA, EV-DO, LTE, satellite DVB-RCS, Mars Reconnaissance Rover, WiMAX, and so on.
Random-like codes (2000s - today)

- Turbo codes use a long pseudorandom interleaver

- 3G and 4G telephony standards HSPA, EV-DO, LTE, satellite DVB-RCS, Mars Reconnaissance Rover, WiMAX, and so on.

- Low-density parity-check (LDPC) codes are defined on a large sparse graph

- DVB-S2, ITU-T G.hn standard (data networking over power lines, phone lines, and coaxial cables), 10GBase-T Ethernet, Wi-Fi standards 802.11, and so on.
Outline

- From Shannon to Modern Coding Theory
  - Channel capacity, structured codes, random codes, LDPC codes

- LDPC Block Codes
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions

- Spatially Coupled LDPC Codes
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- Practical Considerations
  - Window decoding, performance, latency, and complexity comparisons to LDPC block codes, rate-compatibility, implementation aspects
Definition by parity-check matrix: [Gallager, '62]  
Bipartite graph representation: [Tanner, '81]

Code: \( \{ \mathbf{v} \mid \mathbf{v} H^T = 0 \} \)

\( n = 20 \) variable nodes of degree \( J = 3 \)

\( l = 15 \) check nodes of degree \( K = 4 \)

\( (J,K) \)-regular LDPC block code: 
\[ R \geq 1 - \frac{J}{K} \]
Definition by parity-check matrix: [Gallager, '62]
Bipartite graph representation: [Tanner, '81]

\[ H = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \]

Graph-based codes can be decoded iteratively with low complexity by exchanging messages in the graph using Belief Propagation (BP).

\[ n = 20 \text{ variable nodes of degree } J = 3 \]
\[ l = 15 \text{ check nodes of degree } K = 4 \]

\[ R \geq 1 - \frac{J}{K} \]
For an asymptotically good code ensemble, the minimum distance grows linearly with the block length $n$. 

![Graph showing minimum distance growth rates of (J,K)-regular LDPC block code ensembles](image)
Minimum Distance Growth Rates of (J,K)-Regular LDPC Block Code Ensembles

- For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length $n$.

- $(J,K)$-regular block code ensembles are asymptotically good, i.e.,

$$d_{\text{min}} \geq n\delta_{JK}$$

where $\delta_{JK}$ is called the **typical minimum distance ratio**, or minimum distance growth rate.
For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length $n$.

- $(J,K)$-regular block code ensembles are asymptotically good, i.e.,
  $$d_{\text{min}} \geq n\delta_{JK}$$
  where $\delta_{JK}$ is called the **typical minimum distance ratio**, or **minimum distance growth rate**.

- As the density of $(J,K)$-regular ensembles increases, $\delta_{JK}$ approaches the Gilbert-Varshamov bound.
Iterative decoding thresholds can be calculated for $(J,K)$-regular LDPC block code ensembles using density evolution (DE).

<table>
<thead>
<tr>
<th>$J$</th>
<th>$K$</th>
<th>Rate</th>
<th>$\varepsilon^*$</th>
<th>$\varepsilon_{Sh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>0.429</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>0.383</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>0.341</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.517</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>0.506</td>
<td>0.667</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$J$</th>
<th>$K$</th>
<th>Rate</th>
<th>$(E_b/N_0)^*$</th>
<th>$(E_b/N_0)_{Sh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>1.11</td>
<td>0.184</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1.61</td>
<td>0.184</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>2.04</td>
<td>0.184</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.96</td>
<td>$-0.229$</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>1.67</td>
<td>$-0.480$</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>1.00</td>
<td>$-0.790$</td>
</tr>
</tbody>
</table>

Thresholds of $(J,K)$-regular LDPC Block Code Ensembles

- Iterative decoding thresholds can be calculated for $(J,K)$-regular LDPC block code ensembles using density evolution (DE).

<table>
<thead>
<tr>
<th>$J$</th>
<th>$K$</th>
<th>Rate</th>
<th>$\varepsilon^*$</th>
<th>$\varepsilon_{\text{Sh}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>0.429</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>0.383</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>0.341</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.517</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>0.506</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>0.647</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$J$</th>
<th>$K$</th>
<th>Rate</th>
<th>$(E_b/N_0)^*$</th>
<th>$(E_b/N_0)_{\text{Sh}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>1.11</td>
<td>0.184</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1.61</td>
<td>0.184</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>2.04</td>
<td>0.184</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.96</td>
<td>−0.229</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>1.67</td>
<td>−0.480</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>1.00</td>
<td>−0.790</td>
</tr>
</tbody>
</table>

- There exists a relatively large gap to capacity.

Thresholds of \((J,K)\)-regular LDPC Block Code Ensembles

- Iterative decoding thresholds can be calculated for \((J,K)\)-regular LDPC block code ensembles using density evolution (DE).

### Table: BEC thresholds

<table>
<thead>
<tr>
<th>(J)</th>
<th>(K)</th>
<th>Rate</th>
<th>(\varepsilon^*)</th>
<th>(\varepsilon_{\text{Sh}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>0.429</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>0.383</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>0.341</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.517</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>0.506</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>0.647</td>
<td>0.75</td>
</tr>
</tbody>
</table>

### Table: AWGNC thresholds

<table>
<thead>
<tr>
<th>(J)</th>
<th>(K)</th>
<th>Rate</th>
<th>(\left(\frac{E_b}{N_0}\right)^*)</th>
<th>(\left(\frac{E_b}{N_0}\right)_{\text{Sh}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6</td>
<td>0.5</td>
<td>1.11</td>
<td>0.184</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.5</td>
<td>1.61</td>
<td>0.184</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.5</td>
<td>2.04</td>
<td>0.184</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.4</td>
<td>0.96</td>
<td>-0.229</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.333</td>
<td>1.67</td>
<td>-0.480</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>1.00</td>
<td>-0.790</td>
</tr>
</tbody>
</table>

- There exists a relatively large gap to capacity.
- Iterative decoding thresholds get further from capacity as the graph density increases.

Protographs (Matrix Description)

Large LDPC codes can be obtained from a small base parity-check matrix $B$ by replacing each nonzero entry in $B$ with an $M \times M$ permutation matrix, where $M$ is the lifting factor.

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3 \times 6} \quad \rightarrow \quad H = \begin{bmatrix} \Pi_{1,1} & \Pi_{1,2} & \Pi_{1,3} & \Pi_{1,4} & \Pi_{1,5} & 0 \\ \Pi_{2,1} & 0 & \Pi_{2,3} & \Pi_{2,4} & \Pi_{2,5} & \Pi_{2,6} \\ 0 & \Pi_{3,2} & \Pi_{3,3} & 0 & 0 & \Pi_{3,6} \end{bmatrix}_{3M \times 6M}$$
Large LDPC codes can be obtained from a small base parity-check matrix $B$ by replacing each nonzero entry in $B$ with an $M \times M$ permutation matrix, where $M$ is the lifting factor.

Example: Irregular code with $M = 4$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3 \times 6} \quad H = \begin{bmatrix} \Pi_{1,1} & \Pi_{1,2} & \Pi_{1,3} & \Pi_{1,4} & \Pi_{1,5} & 0 \\ \Pi_{2,1} & 0 & \Pi_{2,3} & \Pi_{2,4} & \Pi_{2,5} & \Pi_{2,6} \\ 0 & \Pi_{3,2} & \Pi_{3,3} & 0 & 0 & \Pi_{3,6} \end{bmatrix}_{3M \times 6M}$$

- Length $6M = 24$
- Rate $R = 1/2$
Large LDPC codes can be obtained from a small \textbf{base parity-check matrix} \( B \) by replacing each nonzero entry in \( B \) with an \( M \times M \) \textbf{permutation matrix}, where \( M \) is the \textbf{lifting factor}.

\[
B = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1
\end{bmatrix}_{3 \times 6}
\quad \rightarrow \quad
H = \begin{bmatrix}
\Pi_{1,1} & \Pi_{1,2} & \Pi_{1,3} & \Pi_{1,4} & \Pi_{1,5} & 0 \\
\Pi_{2,1} & 0 & \Pi_{2,3} & \Pi_{2,4} & \Pi_{2,5} & \Pi_{2,6} \\
0 & \Pi_{3,2} & \Pi_{3,3} & 0 & 0 & \Pi_{3,6}
\end{bmatrix}_{3M \times 6M}
\]

\textbf{Example: Irregular code} with \( M = 4 \)

- \textbf{length} \( 6M = 24 \)
- \textbf{rate} \( R = 1/2 \)

\textbf{Irregular codes} have variable \textbf{row and column weights} (check node and variable node degrees)
Protographs (Graphical Description)

Protographs are often represented using a bipartite **Tanner graph**

\[
B = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 \\
\end{bmatrix}_{3 \times 6}
\]

Protographs (Graphical Description)

Protographs are often represented using a bipartite **Tanner graph**

\[
B = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 \\
\end{bmatrix}_{3 \times 6}
\]

3 check nodes

6 variable nodes

The collection of all possible parity-check matrices with lifting factor \(M\) forms a **code ensemble**, where all the codes share a common structure

\[
H = \begin{bmatrix}
\Pi_{1,1} & \Pi_{1,2} & \Pi_{1,3} & \Pi_{1,4} & \Pi_{1,5} & 0 \\
\Pi_{2,1} & 0 & \Pi_{2,3} & \Pi_{2,4} & \Pi_{2,5} & \Pi_{2,6} \\
0 & \Pi_{3,2} & \Pi_{3,3} & 0 & 0 & \Pi_{3,6} \\
\end{bmatrix}
\]

Quasi-Cyclic LDPC Codes

**Quasi-cyclic** (QC) LDPC codes are of great interest in practice, since they have **efficient encoder and decoder implementations**.
**Quasi-Cyclic LDPC Codes**

- **Quasi-cyclic** (QC) LDPC codes are of great interest in practice, since they have **efficient encoder and decoder implementations**

**Example:** protograph construction of a **(2,3)-regular** QC-LDPC block code

\[ B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \in \mathbb{F}_2^{2 \times 3} \]
**Quasi-Cyclic LDPC Codes**

**Quasi-cyclic** (QC) LDPC codes are of great interest in practice, since they have **efficient encoder and decoder implementations**

**Example:** protograph construction of a \((2,3)\)-regular QC-LDPC block code

\[
B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{2 \times 3} \quad \Rightarrow \quad H = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}_{14 \times 21}
\]

\(M = 7\)

\(n = 3M = 21\)

\(R = 1/3\)

For QC codes, the permutation matrices are **shifted identities**
Multi-Edge Protographs

- Protographs can have repeated edges (corresponding to integer values greater than one in $B$)

$$B = \begin{bmatrix} 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 3 & 1 \end{bmatrix}_{3 \times 5}$$

- Note that this makes no sense without lifting

Multi-Edge Protographs

- Protographs can have repeated edges (corresponding to integer values greater than one in $B$)

- Note that this makes no sense without lifting

- Repeated edges in a protograph correspond to using sums of permutation matrices to form LDPC code ensembles

\[
B = \begin{bmatrix}
2 & 2 & 0 & 1 & 1 \\
0 & 0 & 1 & 2 & 0 \\
1 & 1 & 0 & 3 & 1
\end{bmatrix}_{3 \times 5}
\]

\[
H = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
& & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0
\end{bmatrix}_{18 \times 5}
\]

Multi-Edge Protographs

- Protographs can have repeated edges (corresponding to integer values greater than one in $\mathbf{B}$)

$$
\begin{bmatrix}
2 & 2 & 0 & 1 & 1 \\
0 & 0 & 1 & 2 & 0 \\
1 & 1 & 0 & 3 & 1
\end{bmatrix}
$$

- Note that this makes no sense without lifting

- Repeated edges in a protograph correspond to using sums of permutation matrices to form LDPC code ensembles

  denser graphs!

  can also be QC (using circulant matrices)!

'Good' Protographs

- Ensemble average properties can be easily calculated from a protograph, thus simplifying the construction of 'good' code ensembles.
- Iterative decoding thresholds close to capacity for irregular protographs
- Minimum distance growing linearly with block length (asymptotically good) for regular and some irregular protographs

\[ R = \frac{e + 1}{e + 2} \]

<table>
<thead>
<tr>
<th>Rate</th>
<th>Threshold ((E_b/N_0)^*)</th>
<th>Capacity ((E_b/N_0)_{\text{Sh}})</th>
<th>Distance growth rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2</td>
<td>0.628</td>
<td>0.187</td>
<td>0.01450</td>
</tr>
<tr>
<td>2/3</td>
<td>1.450</td>
<td>1.059</td>
<td>0.00582</td>
</tr>
<tr>
<td>3/4</td>
<td>2.005</td>
<td>1.628</td>
<td>0.00323</td>
</tr>
<tr>
<td>4/5</td>
<td>2.413</td>
<td>2.040</td>
<td>0.00207</td>
</tr>
<tr>
<td>5/6</td>
<td>2.733</td>
<td>2.362</td>
<td>0.00145</td>
</tr>
<tr>
<td>6/7</td>
<td>2.993</td>
<td>2.625</td>
<td>0.00108</td>
</tr>
</tbody>
</table>

Regular vs. Irregular LDPC codes
Regular vs. Irregular LDPC codes

- "Regular" LDPC codes:
  - Structure aids implementation
  - Low error floors
  - Some thresholds far from capacity

---

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
“Regular” LDPC codes:
- structure aids implementation
- low error floors
- thresholds far from capacity
- not suitable for severely power constrained applications
“Regular” LDPC codes:
- Structure aids implementation
- Low error floors
- Thresholds far from capacity
- Not suitable for severely power constrained applications

“Irregular” LDPC codes:
- Less desirable structure
- Thresholds close to capacity
- Visible error floors
**Regular vs. Irregular LDPC codes**

- **“Regular”** LDPC codes:
  - ✓ structure aids implementation
  - ✓ low error floors
  - ✗ thresholds far from capacity
  - ➡️ not suitable for severely power constrained applications

- **“Irregular”** LDPC codes:
  - ✗ Less desirable structure
  - ✓ thresholds close to capacity
  - ✗ visible error floors
  - ➡️ not suitable for applications that require very low error rates
Regular vs. Irregular LDPC codes

“Regular” LDPC codes:
- Structure aids implementation
- Low error floors
- Thresholds far from capacity
- Not suitable for severely power constrained applications

“Irregular” LDPC codes:
- Less desirable structure
- Thresholds close to capacity
- Visible error floors
- Not suitable for applications that require very low error rates

Spatially coupled LDPC codes combine all of the positive features!

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Outline

- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions

- **Spatially Coupled LDPC Codes**
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- **Practical Considerations**
  - Window decoding; performance, latency, and complexity comparisons to LDPC block codes; rate-compatibility; implementation aspects
Consider transmission of consecutive blocks (protograph representation):

\[
B = \begin{bmatrix} 3 & 3 \end{bmatrix}_{b_c \times b_v}
\]

\((3,6)\)-regular LDPC-BC base matrix
Spatially Coupled Protographs

- Consider transmission of consecutive blocks (protograph representation):

- Blocks are spatially coupled (introducing memory) by spreading edges over time:

\[
B = \begin{bmatrix}
3 & 3 \\
\end{bmatrix}_{b_c \times b_v}
\]

(3,6)-regular LDPC-BC base matrix

\[
B = \begin{bmatrix}
3 & 3 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
B_0 \\
B_1 \\
B_2 \\
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & 1 \\
1 & 1 \\
\end{bmatrix}
\]
Spatially Coupled Protographs

- Consider transmission of consecutive blocks (protograph representation):

- Blocks are spatially coupled (introducing memory) by spreading edges over time:

- Spreading constraint: $\sum_{i=0}^{m_s} B_i = B$ (where $B_i$ has size $b_c \times b_v$)
Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:
Spatially Coupled Protographs

- Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:

![Convolutional Protograph Diagram]
Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:
Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:
Spa\textit{tially Coupled Protographs}

- Transmission of consecutive spatially coupled (SC) blocks results in a convolutional protograph:

\[
\ldots \quad B_{m_s} \quad \ldots \quad B_1 \quad B_0 \quad \ldots \quad B_{m_s} \quad \ldots
\]

- The bi-infinite convolutional protograph corresponds to a bi-infinite convolutional base matrix:

\[
B_{[-\infty,\infty]} = \begin{bmatrix}
\ddots & \cdots & \cdots & \cdots & \\
& B_{m_s} & \cdots & B_1 & B_0 \\
& \cdots & \ddots & \cdots & \cdots \\
& B_{m_s} & \cdots & B_1 & B_0 \\
& \cdots & \cdots & \ddots & \cdots & \cdots
\end{bmatrix}
\]

\[
R = \frac{b_v - b_c}{b_v}
\]

\[
\nu_s = b_v (m_s + 1)
\]
An ensemble of (3,6)-regular SC-LDPC codes can be created from the convolutional protograph by the graph lifting operation.

\[
B_{[-\infty,\infty]} = \begin{bmatrix}
\ldots & \ldots & \ldots \\
\begin{array}{c|c|c|c}
B_2 & B_1 & B_0 \\
B_2 & B_1 & B_0 \\
B_2 & B_1 & B_0 \\
\end{array} \\
\ldots & \ldots & \ldots 
\end{bmatrix}
\]
An ensemble of (3,6)-regular SC-LDPC codes can be created from the convolutional protograph by the graph lifting operation

\[
B_{[-\infty, \infty]} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
& 1 & 1 & 1 & 1 & 1 \\
& & 1 & 1 & 1 & 1 \\
& & & 1 & 1 & 1 \\
& & & & \ddots & \ddots \\
\end{bmatrix}
\]

\[
B_i = \begin{bmatrix} 1 & 1 \end{bmatrix}
\]

\[
b_c = 1 \\
b_v = 2 \\
m_s = 2
\]
SC-LDPC Code Ensembles

An ensemble of (3,6)-regular SC-LDPC codes can be created from the convolutional protograph by the graph lifting operation.

\[ B_{[-\infty, \infty]} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ \end{bmatrix} \]

**Graph lifting:** \( \Pi_{i,j} \) is an \( M \times M \) permutation matrix

\[ B_i = [1 \ 1] \]
\[ b_c = 1 \]
\[ b_v = 2 \]
\[ m_s = 2 \]

\[ \nu_s = M b_v (m_s + 1) = 6M \]

\[ H_{cc} = \begin{bmatrix} \Pi_{5,t} & \Pi_{4,t} & \Pi_{3,t} & \Pi_{2,t} & \Pi_{1,t} & \Pi_{0,t} \\ \Pi_{5,t+1} & \Pi_{4,t+1} & \Pi_{3,t+1} & \Pi_{2,t+1} & \Pi_{1,t+1} & \Pi_{0,t+1} \\ \Pi_{5,t+2} & \Pi_{4,t+2} & \Pi_{3,t+2} & \Pi_{2,t+2} & \Pi_{1,t+2} & \Pi_{0,t+2} \\ \end{bmatrix} \]
SC-LDPC Code Ensembles

- An ensemble of (3,6)-regular SC-LDPC codes can be created from the **convolutional protograph** by the graph lifting operation

\[
B_{[-\infty, \infty]} = \begin{bmatrix}
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
1 & 1 & 1 & 1 & 1 & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
\]

\[
B_i = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad b_c = 1, \quad b_v = 2, \quad m_s = 2
\]

**Graph lifting:** \( \Pi_{i,j} \) is an \( M \times M \) permutation matrix

\[
\nu_s = M b_v (m_s + 1) = 6M
\]

- If each permutation matrix \( \Pi_{i,j} \) is **circulant**, the codes are **quasi-cyclic**
Terminated Spatially Coupled Codes

Consider terminating $\mathbf{B}_{[-\infty, \infty]}$ to a (block code) base matrix of length $Lb_v$:

$$
\mathbf{B}_{[0, L-1]} = \begin{bmatrix}
\mathbf{B}_0 \\
\vdots \\
\mathbf{B}_{m_s} \\
\mathbf{B}_0 \\
\vdots \\
\mathbf{B}_{m_s}
\end{bmatrix}
$$

$$(L+m_s)b_c \times Lb_v$$

**Code rate:**

$$R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}.$$
**Terminated Spatially Coupled Codes**

- Consider terminating $B_{[-\infty,\infty]}$ to a (block code) **base matrix** of length $L b_v$:

  $$B_{[0,L-1]} = \begin{bmatrix}
  B_0 \\
  \vdots \\
  B_{m_s} \\
  \vdots \\
  B_{m_s}
\end{bmatrix}^{(L+m_s)b_c \times L b_v}$$

- For large $L$, $R_L$ approaches the **unterminated** code rate $R = (b_v - b_c)/b_v$. 

**Code rate:**

$$R_L = \frac{L b_v - (L + m_s)b_c}{L b_v}.$$

---

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?” 25 / 44
Terminated Spatially Coupled Codes

- Consider **terminating** $B_{[-\infty,\infty]}$ to a (block code) base matrix of length $Lb_v$:

\[
B_{[0,L-1]} = \begin{bmatrix}
B_0 \\
\vdots \\
B_{m_s} \\
\vdots \\
B_{m_s}
\end{bmatrix}_{(L+m_s)b_c \times Lb_v}
\]

- **Code rate:**

\[
R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}
\]

- For large $L$, $R_L$ approaches the **unterminated** code rate $R = (b_v - b_c)/b_v$.

- **Example:** $(3,6)$-regular base matrix $B = [3 \ 3]$, $m_s = 2$, $L = 4$, $R_4 = 1/4$

- (check node degrees lower at the ends)
Consider terminating $B_{[-\infty,\infty]}$ to a (block code) base matrix of length $Lb_v$:

$$B_{[0,L-1]} = \begin{bmatrix} B_0 & & \\
& \ddots & \\
& & \ddots & \\
& & & B_0 \\
& & & & \ddots & \\
& & & & & \ddots & \\
& & & & & & \ddots & \\
& & & & & & & B_{m_s} \\
\end{bmatrix}_{(L+m_s)b_c \times Lb_v}$$

**Code rate:**

$$R_L = \frac{Lb_v - (L + m_s)b_c}{Lb_v}.$$ 

For large $L$, $R_L$ approaches the **unterminated** code rate $R = (b_v - b_c)/b_v$.

**Example:** $(3,6)$-regular base matrix $B = [3 \ 3]$, $m_s = 2$, $L = 4$, $R_4 = 1/4$

**Codes can be lifted to different lengths and rates** by varying $M$ and $L$. 

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.
Variable nodes all have the **same degree** as the block code.

- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

**Evolution of message probabilities** \((L = 100)\):
Variable nodes all have the same degree as the block code. Check nodes with lower degrees (at the ends) improve the BP decoder.

Evolution of message probabilities \((L = 100)\):

![Graph showing the evolution of message probabilities over iterations.](image-url)
Variable nodes all have the **same degree** as the block code.

Check nodes with **lower degrees** (at the ends) improve the BP decoder.

**Thresholds of SC-LDPC Codes**

![Graph showing message probabilities evolution](image)

Evolution of message probabilities \( (L = 100) \):
Variable nodes all have the **same degree** as the block code.

Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities \((L = 100)\):
Variable nodes all have the **same degree** as the block code.

- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

**Evolution of message probabilities** ($L = 100$):
Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

**Evolution of message probabilities** \((L = 100)\):

```
1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 1 0 0
300 iterations
```

---

**Thresholds of SC-LDPC Codes**

![Graph showing the evolution of message probabilities](image-url)
Variable nodes all have the **same degree** as the block code.
Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities \((L = 100)\):

"wave-like decoding"
Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities ($L = 100$):

![Graph showing wave-like decoding](image)

- **Note:** the fraction of lower degree nodes tends to zero as $L \to \infty$, i.e., the codes are **asymptotically regular**.
Density evolution can be applied to the protograph-based ensembles with $M \rightarrow \infty$ [Sridharan et al. '04]:

Example: BEC

![Graph showing BEC thresholds and Shannon limits.]
Thresholds of SC-LDPC Codes

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

**Example**: BEC

$L = 4$, $R = 1/4$

$\varepsilon^* = 0.635$, $\varepsilon_{Sh} = 0.75$

![Graph showing BEC thresholds](image-url)
Density evolution can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

**Example:** BEC

$L = 4, R = 1/4$
$\varepsilon^* = 0.635, \varepsilon_{Sh} = 0.75$

$L = 10, R = 2/5$
$\varepsilon^* = 0.505, \varepsilon_{Sh} = 0.6$
Density evolution can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

Example: BEC

$L = 4$, $R = 1/4$
$\varepsilon^* = 0.635$, $\varepsilon_{Sh} = 0.75$

$L = 10$, $R = 2/5$
$\varepsilon^* = 0.505$, $\varepsilon_{Sh} = 0.6$

$L \to \infty$, $R \to 1/2$
$\varepsilon^* = 0.488$, $\varepsilon_{Sh} = 0.5$

(3,6)-regular block code:
$\varepsilon^* = 0.429$
We observe a **significant improvement** in the thresholds of SC-LDPC codes compared to the associated LDPC block codes (LDPC-BCs) due to the lower degree check nodes at the ends of the graph and the wave-like decoding.

We observe a significant improvement in the thresholds of SC-LDPC codes compared to the associated LDPC block codes (LDPC-BCs) due to the lower degree check nodes at the ends of the graph and the wave-like decoding.

In contrast to LDPC-BCs, the iterative decoding thresholds of SC-LDPC codes improve as the graph density increases.

Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.

- The **threshold saturates** (converges) to a fixed value numerically indistinguishable from the **maximum a posteriori** (MAP) threshold of the $(J, K)$-regular LDPC-BC ensemble as $L \to \infty$  [LSCZ10].

When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.

Why are SC-LDPC Codes Better?

- The threshold saturates (converges) to a fixed value numerically indistinguishable from the maximum a posteriori (MAP) threshold of the \((J, K)\)-regular LDPC-BC ensemble as \(L \to \infty\) [LSCZ10].

- For a more random-like ensemble, this has been proven analytically, first for the BEC [KRU11], then for all BMS channels [KRU13].


Threshold Saturation (BEC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

\[ \epsilon \]

\[ B_c \]

\[ (3,6) \]

BP threshold

MAP threshold

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Threshold Saturation (BEC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

BP

MAP

(3,6)

(5,10)
Threshold Saturation (BEC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

$(5,10)$ BC
$(3,6)$ BC

SC-LDPC codes

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

Threshold Saturation (BEC)

optimal decoding performance with a suboptimal iterative algorithm!
Threshold Saturation (AWGNC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

(3,6)-regular block code

~0.5 dB
Threshold Saturation (AWGNC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

(3,6)-regular block code
(4,8)-regular block code

Capacity

$E_b/N_0$ (dB)

$\sim 1.25$ dB
Threshold Saturation (AWGNC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

(4,8)  (3,6)  (3,6)-regular block code  (4,8)-regular block code

Spatially coupled codes
Threshold Saturation (AWGNC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

(4,8) (3,6) (3,6)-regular block code (4,8)-regular block code

Spatially coupled codes

→ **optimal** decoding performance with a **suboptimal** iterative algorithm!
By increasing $J$ and $K$, we obtain capacity achieving $(J,K)$-regular SC-LDPC code ensembles with linear minimum distance growth.

$(J,K)$-regular SC-LDPC codes combine the best features of irregular and regular LDPC-BCs, i.e., capacity approaching thresholds and linear distance growth.
Similar results are obtained for the AWGNC

As $L \to \infty$ the minimum distance growth rates of terminated SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the unterminated ensembles remain constant.
As \( L \to \infty \) the minimum distance growth rates of terminated SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the unterminated ensembles remain constant.

For large \( L \), the strength of unterminated ensembles scales with the constraint length
\[
\nu_s = M(m_s + 1)b_v
\]
and is independent of \( L \).
As \( L \to \infty \) the minimum distance growth rates of terminated SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the unterminated ensembles remain constant.

For large \( L \), the strength of unterminated ensembles scales with the constraint length

\[
\nu_s = M(m_s + 1)b_v
\]

and is independent of \( L \).

An appropriate distance measure for 'convolutional-like' terminated ensembles should be independent of \( L \).
Outline

- LDPC Block Codes
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions

- Spatially Coupled LDPC Codes
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance

- Practical Considerations
  - Window decoding; performance, latency, and complexity comparisons to LDPC block codes; rate-compatibility; implementation aspects
SC-LDPC codes can be decoded with standard iterative decoding schedules.
Block Decoding of SC-LDPC Codes

SC-LDPC codes can be decoded with standard iterative decoding schedules. Reliable messages from the ends propagate through the graph toward the center as iterations proceed.
SC-LDPC codes can be decoded with standard iterative decoding schedules. Reliable messages from the ends propagate through the graph toward the center as iterations proceed.
Block Decoding of SC-LDPC Codes

SC-LDPC codes can be decoded with standard iterative decoding schedules. Reliable messages from the ends propagate through the graph toward the center as iterations proceed.
Block Decoding of SC-LDPC Codes

SC-LDPC codes can be decoded with standard iterative decoding schedules.

Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

The frame error rate (FER) of a terminated graph can be analyzed.

1 frame = $L$ sections

The FER depends on $L$ \( \frac{\text{FER}}{L \to \infty} = 1 \)
Consider LDPC-BCs and SC-LDPC codes with increasing frame length $N$.

As $N$ increases, the LDPC-BC performance approaches $\varepsilon^* = 0.429$. 

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Consider LDPC-BCs and SC-LDPC codes with increasing frame length $N$

As $N$ increases, the LDPC-BC performance approaches $\varepsilon^* = 0.429$
Consider LDPC-BCs and SC-LDPC codes with increasing frame length $N$.

As $N$ increases, the LDPC-BC performance approaches $\varepsilon^* = 0.429$. 

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Consider LDPC-BCs and SC-LDPC codes with increasing frame length $N$.

As $N$ increases, the LDPC-BC performance approaches $\varepsilon^* = 0.429$.

As $N$ increases, the SC-LDPC code performance approaches $\varepsilon^* = 0.488$, outperforming the LDPC-BC for a sufficiently large $N$. 
Consider LDPC-BCs and SC-LDPC codes with increasing frame length $N$.

As $N$ increases, the LDPC-BC performance approaches $\varepsilon^* = 0.429$.

As $N$ increases, the SC-LDPC code performance approaches $\varepsilon^* = 0.488$ and outperforms the LDPC-BC for a sufficiently large $N$. 

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Consider LDPC-BCs and SC-LDPC codes with increasing frame length $N$.

As $N$ increases the LDPC-BC performance approaches

$$\varepsilon^* = 0.429$$

As $N$ increases the SC-LDPC code performance approaches

$$\varepsilon^* = 0.488$$

outperforming the LDPC-BC for a sufficiently large $N$. 

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.
The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

**Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].
The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

Sliding window decoding (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].
The highly localized (convolutional) structure is well-suited for efficient decoding schedules that reduce memory and latency requirements.

Sliding window decoding (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].
Window Decoding of SC-LDPC Codes

The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

**Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].

One block of $cM$ **target symbols** is decoded in each window position.
The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

**Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].

One block of $cM$ **target symbols** is decoded in each window position.

The window then shifts to the right.
Window Decoding Performance

Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

$M_{BC} = 1000$

For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.

Latencies:
- LDPC: $6M_{BC}$
- SC-LDPC: $2M_{SC}W$

Window Decoding Performance

Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.

For equal latency, SC-LDPC codes still display a significant performance gain.

Window Decoding Performance

Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

- For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.
- For equal latency, SC-LDPC codes still display a significant performance gain.

$M_{BC} = 1000$
$M_{SC} = 1000$
$M_{SC} = 500$
$W = 6$
$M_{SC} = 250$
$W = 12$

$E_b/N_0$

For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.

For equal latency, SC-LDPC codes still display a significant performance gain.


D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?” 39 / 44
Window Decoding Performance

Latencies:
LDPC: \(6M_{BC}\)
SC-LDPC: \(2M_{SC}W\)

For equal lifting factors, SC-LDPC codes display a large convolutional gain at the cost of increased latency.

For equal latency, SC-LDPC codes still display a significant performance gain.

Trade-off in \(M\) vs \(W\)

Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

**Latencies:**
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

![Graph showing the comparison of BC, SC, and SC-LDPC latencies.](image)

Latencies:
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

\[
\text{Latencies:} \\
\text{LDPC: } 4M_{BC} \\
\text{SC-LDPC: } 2M_{SC}W
\]

- Decreases as $W$ (and thus the latency) increases.
- Does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

**Latencies:**
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- Decreases as $W$ (and thus the latency) increases.
- Does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

Latencies:
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

Latencies:
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).
Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required $E_b/N_0$ to achieve a BER of $10^{-5}$ as a function of latency:

**Latencies:**
- LDPC: $4M_{BC}$
- SC-LDPC: $2M_{SC}W$

- decreases as $W$ (and thus the latency) increases.
- does not decrease significantly beyond a certain $W$ ($W \approx 10$).

- When choosing parameters:
  - large $M_{SC}$ improves code performance.
  - large $W$ improves decoder performance.
Regular SC-LDPC Codes vs. Irregular LDPC-BCs

Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard.

Ex: $M = 250$
$n = 24M = 6000$
Regular SC-LDPC Codes vs. Irregular LDPC-BCs

- Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard.

\[ M = 250 \]
\[ n = 24M = 6000 \]

- The IRA LDPC-BC ensemble has rate \( R=0.5 \), BEC threshold \( \epsilon^* \approx 0.4489 \), and AWGNC threshold \( (E_b/N_0)^* \approx 0.8216 \) dB.
Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard.

The IRA LDPC-BC ensemble has rate $R=0.5$, BEC threshold $\epsilon^* \approx 0.4489$, and AWGNC threshold $(E_b/N_0)^* \approx 0.8216$ dB.

We compare this to a (3,6)-regular SC-LDPC code ensemble with $L=50$, $R=0.49$, and thresholds $\epsilon^* \approx 0.4881$ and $(E_b/N_0)^* \approx 0.4317$ dB.
Regular SC-LDPC Codes vs. Irregular LDPC-BCs

- Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard.

- The IRA LDPC-BC ensemble has rate $R=0.5$, BEC threshold $\epsilon^* \approx 0.4489$, and AWGNC threshold $(E_b/N_0)^* \approx 0.8216$ dB.

- We compare this to a (3,6)-regular SC-LDPC code ensemble with $L=50$, $R=0.49$, and thresholds $\epsilon^* \approx 0.4881$ and $(E_b/N_0)^* \approx 0.4317$ dB.

- For the SC-LDPC code, we choose $W=6$ and $M=500$ so that the latency of both codes is 6000 bits. (Since a code symbol is present in $W=6$ 'windows', we allow fewer iterations per position for the SC-LDPC window decoder.)
Regular SC-LDPC Codes vs. Irregular LDPC-BCs

- (3,6)-regular SC-LDPC, $I_{\text{max}} = 10$
- IRA LDPC-BC, $I_{\text{max}} = 10$
- IRA LDPC-BC, $I_{\text{max}} = 100$

Gaps to threshold will reduce with increasing latency.

D. J. Costello, Jr., “Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?”
Regular SC-LDPC Codes vs. Irregular LDPC-BCs

- Gaps to threshold will reduce with increasing latency
- The asymptotically good regular SC-LDPC code shows no sign of an error floor

\[ (3,6)\text{-regular SC-LDPC, } I_{\text{max}} = 10 \]
\[ \text{IRA LDPC-BC, } I_{\text{max}} = 10 \]
\[ \text{IRA LDPC-BC, } I_{\text{max}} = 100 \]
The asymptotically good regular SC-LDPC code shows no sign of an error floor.

The regular SC-LDPC code structure has implementation advantages.

Gaps to threshold will reduce with increasing latency.
As a result of their capacity approaching performance and simple structure, regular SC-LDPC codes may be attractive for future coding standards. Several key features will require further investigation:

- Hardware advantages of QC designs obtained by circulant liftings
- Hardware advantages of the 'asymptotically-regular' structure
- Design advantages of flexible frame length and flexible rate obtained by varying M, L, and/or puncturing
Implementation Aspects

- As a result of their capacity approaching performance and simple structure, regular SC-LDPC codes may be attractive for future coding standards. Several key features will require further investigation:
  - Hardware advantages of QC designs obtained by circulant liftings
  - Hardware advantages of the 'asymptotically-regular' structure
  - Design advantages of flexible frame length and flexible rate obtained by varying $M$, $L$, and/or puncturing

- Of particular importance for applications requiring extremely low decoded bit error rates (e.g., optical communication, data storage) is an investigation of error floor issues related to stopping sets, trapping sets, and absorbing sets.
Conclusions

- Spatially coupled LDPC code ensembles achieve **threshold saturation**, i.e., their iterative decoding thresholds (for large $L$ and $M$) approach the MAP decoding thresholds of the underlying LDPC block code ensembles.

- The threshold saturation and linear minimum distance growth properties of $(J,K)$-regular SC-LDPC codes combine the best asymptotic features of both regular and irregular LDPC-BCs.

- With window decoding, SC-LDPC codes also compare favorably to LDPC-BCs in the finite-length regime, providing flexible tradeoffs between BER performance, decoding latency, and decoding complexity.

- SC-LDPC codes can be punctured to achieve robustly good performance over a wide variety of code rates.