Advances in Protograph-Based LDPC Codes and a Rate Allocation Problem

PhD Defense

Sudarsan Vasista Srinivasan Ranganathan
Advisors: Richard D. Wesel and Dariush Divsalar

Communication Systems Laboratory
University of California, Los Angeles

Outline

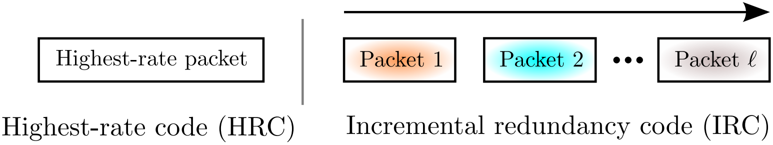
1. Incremental Redundancy (IR) and LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
5. A Rate Allocation Problem

Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes
5. A Rate Allocation Problem

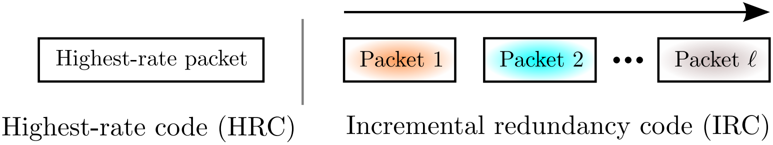
UCLA Samueli School of Engineering

Rate-Compatibility (RC)



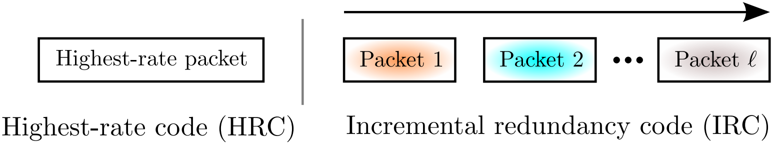
- Useful in many communication applications

Rate-Compatibility (RC)



- Useful in many communication applications
- RC codes most recently proposed for the **5G wireless standard**

Rate-Compatibility (RC)



- Useful in many communication applications
- RC codes most recently proposed for the **5G wireless standard**
- **Challenge today:** short block-lengths, ever-growing throughput requirements, new applications, handling rate-compatibility

Linear Codes $\Longleftrightarrow$ Tanner Graphs

H of size 6×9

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Linear Codes ⟺ Tanner Graphs

H of size 6×9

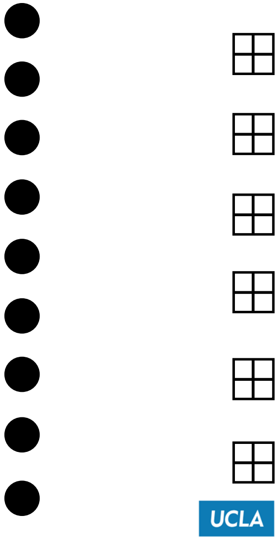
$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{array} \\
H = & \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

Variable nodes

Linear Codes $\Longleftrightarrow$ Tanner Graphs

H of size 6×9

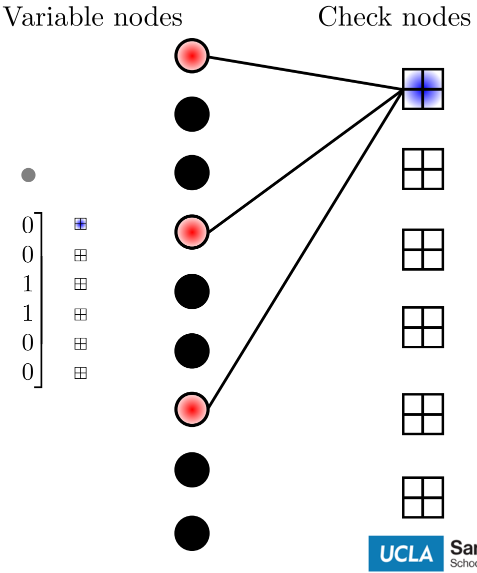
$$H = \begin{array}{ccccccccc|c} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & \boxplus \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \boxplus \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \boxplus \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \boxplus \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \boxplus \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & \boxplus \end{array}$$



Linear Codes $\Longleftrightarrow$ Tanner Graphs

H of size 6×9

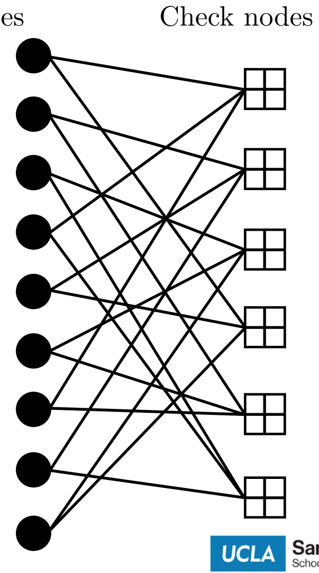
$$H = \begin{array}{cc} \begin{array}{ccccccccc} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{array} & \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} \boxplus \\ \boxplus \\ \boxplus \\ \boxplus \\ \boxplus \\ \boxplus \end{array} \end{array}$$



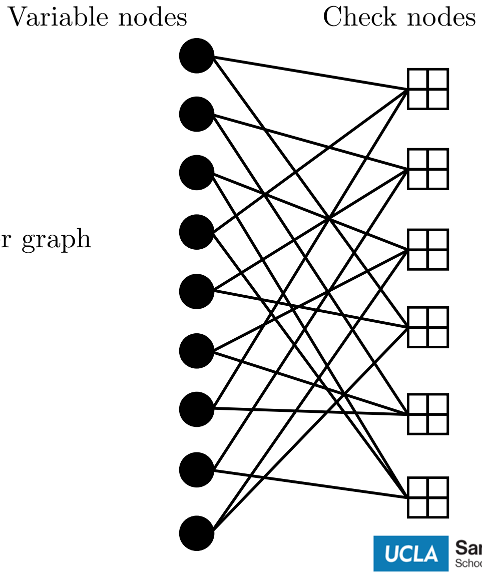
Linear Codes $\Longleftrightarrow$ Tanner Graphs

H of size 6×9

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

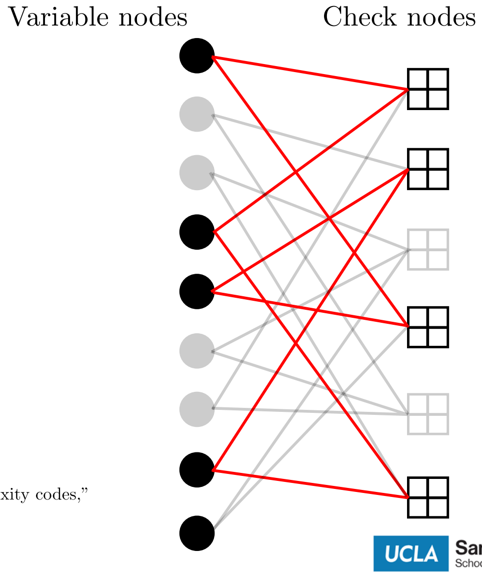


Linear Codes $\Longleftrightarrow$ Tanner Graphs



“Regular” bipartite Tanner graph
Variable node degree 2
Check node degree 3

Linear Codes $\Longleftrightarrow$ Tanner Graphs



Girth:
Length of smallest cycle in graph

R. Tanner, ”A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, Sep. 1981.

Low-Density Parity-Check Codes (LDPC Codes)

- Most important class of codes treated as graphs are **LDPC codes**

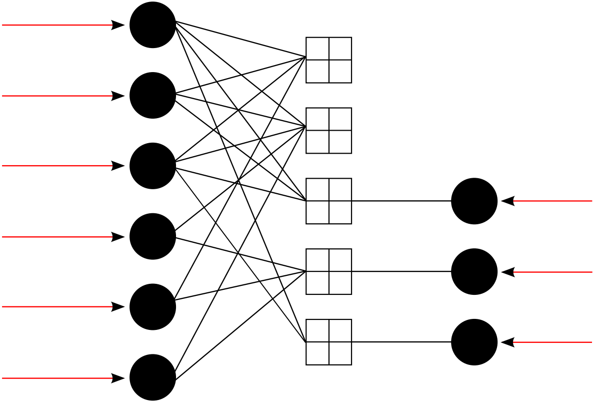
Robert Gray Gallager, “Low-Density Parity-Check Codes,” *MIT Press*, 1963.

Low-Density Parity-Check Codes (LDPC Codes)

- Most important class of codes treated as graphs are **LDPC codes**
- LDPC: parity-check matrix is of “low density”

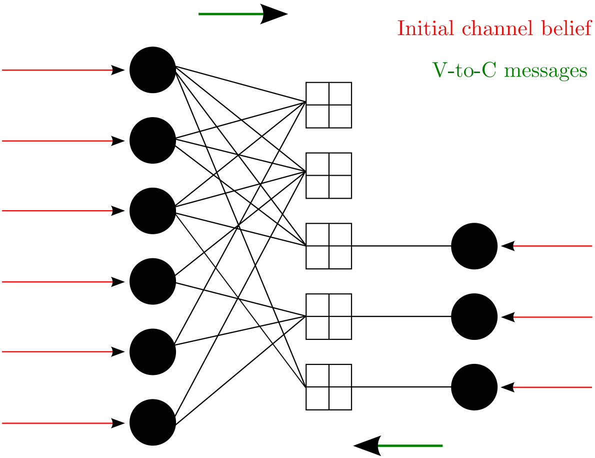
Robert Gray Gallager, “Low-Density Parity-Check Codes,” *MIT Press*, 1963.

Iterative Decoding of (LDPC) Codes



Initial channel belief

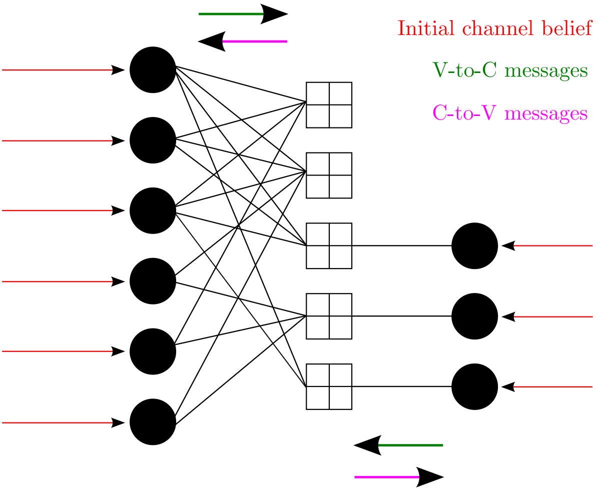
Iterative Decoding of (LDPC) Codes



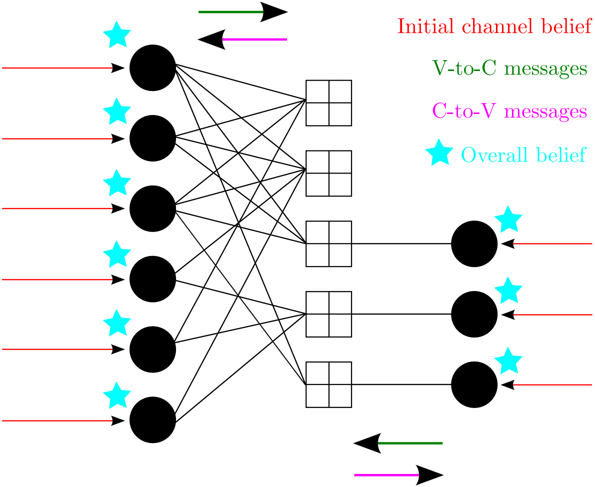
Initial channel belief

V-to-C messages

Iterative Decoding of (LDPC) Codes



Iterative Decoding of (LDPC) Codes



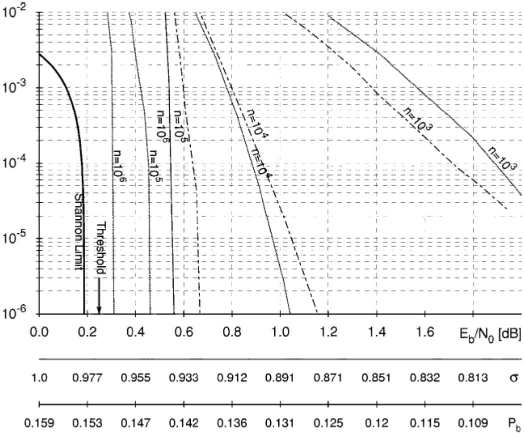
Initial channel belief

V-to-C messages

C-to-V messages

★ Overall belief

Iterative Decoding Threshold



Reproduced from T. J. Richardson and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, Feb. 2001.

Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes
5. A Rate Allocation Problem

Circulant Permutation Matrix (CPM)

Definition (Circulant permutation matrix (CPM))

An $N \times N$ circulant permutation matrix (CPM), x-shifted $(0 \leqslant x \leqslant N-1)$ is

- the identity matrix cyclically shifted by x positions

Circulant Permutation Matrix (CPM)

Definition (Circulant permutation matrix (CPM))

An $N \times N$ circulant permutation matrix (CPM), x-shifted $(0 \leqslant x \leqslant N-1)$ is

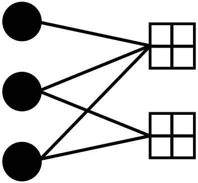
- the identity matrix cyclically shifted by x positions

Example, when $x = 2, N = 4$, right shift:

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

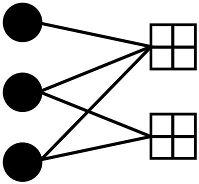
Protograph and Protomatrix

Protograph
A small Tanner graph



Protograph and Protomatrix

Protograph
A small Tanner graph



Corresponding
Protomatrix

$$P = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$n_c = 2, n_v = 3$$

$$R = \frac{n_v - n_c}{n_v - n_p}$$

Jeremy Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN-PR 42-154*, JPL, Aug. 2003.

Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Protomatrix of complete protograph

Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

$$\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \longrightarrow \left[\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1 & 0 & 1\\ \hline 1 & 0 & 1 & 0 & 0 & 1\\ 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right]$$

Protomatrix of
complete protograph

CPM blocks $N \times N$ in
place of each 1

Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

CPM blocks $N \times N$ in place of each 1

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \xrightarrow{\hspace{3cm}} \left[\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right]$$

Protomatrix of complete protograph

Parity-check matrix of complete protograph QC-LDPC code

Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

CPM blocks $N \times N$ in place of each 1

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Lifting}} \left[\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right]$$

Protomatrix of complete protograph

Parity-check matrix of complete protograph QC-LDPC code

Protograph Quasi-Cyclic LDPC (Protograph QC-LDPC) Codes

CPM blocks $N \times N$ in place of each 1

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Lifting}} \left[\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{array}\right]$$

Protomatrix of complete protograph

Parity-check matrix of complete protograph QC-LDPC code

Lifting factor N

Outline

1. Incremental Redundancy (IR) and LDPC Codes
    - Rate-Compatible Codes, Protograph LDPC Codes
    - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
    - Protograph-Based Raptor-Like LDPC (PBRL) Codes
    - Permanent-Bound-Based Design (PBD) Method
    - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
    - UIO-PBRL Codes

5. A Rate Allocation Problem

UCLA Samueli School of Engineering

Based on. . .

S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Design of improved quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths," In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2017.

S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths," Under revision, *IEEE Trans. Inf. Theory*.

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far

Tsung-Yi Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, May 2015.

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication

Tsung-Yi Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, May 2015.

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
    - The code family in 5G for data communication
- General structure of a PBRL protomatrix

$$P = \begin{bmatrix} P_{\mathrm{HRC}} & 0 \\ P_{\mathrm{IRC}} & I \end{bmatrix}_{n_c \times n_v} \tag{1}$$

Tsung-Yi Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, May 2015.

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication
- General structure of a PBRL protomatrix

$$P = \begin{bmatrix} P_{\text{HRC}} & 0 \\ P_{\text{IRC}} & I \end{bmatrix}_{n_c \times n_v} \tag{1}$$

- P_{HRC} = HRC (highest-rate code) part

Tsung-Yi Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, May 2015.

UCLA Samueli School of Engineering

Protograph-Based Raptor-Like (PBRL) Codes

- Best performing RC-LDPC codes known so far
  - The code family in 5G for data communication
- General structure of a PBRL protomatrix

$$P = \begin{bmatrix} P_{\mathrm{HRC}} & 0 \\ P_{\mathrm{IRC}} & I \end{bmatrix}_{n_c \times n_v} \tag{1}$$

- $P_{\mathrm{HRC}} =$ HRC (highest-rate code) part
- $P_{\mathrm{IRC}} =$ IRC (incremental redundancy code) part

Tsung-Yi Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, May 2015.

UCLA Samueli School of Engineering

PBRL Protomatrix

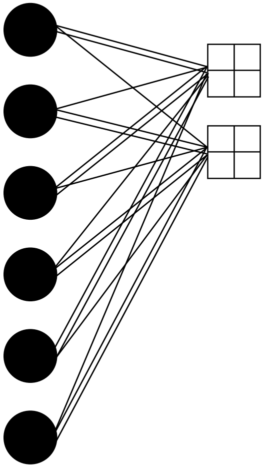
$$n_{c_H} = 2, n_{v_H} = 6$$

$$P_{\mathsf{HRC}}$$

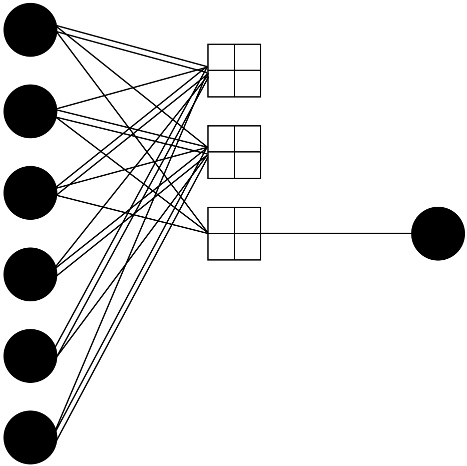
$$P = \left[\begin{array}{cccccc|ccc} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$P_{\mathsf{IRC}} \qquad n_c = 5, n_v = 9$$

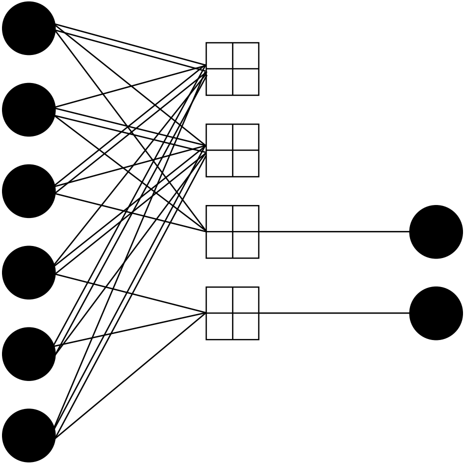
PBRL Protograph



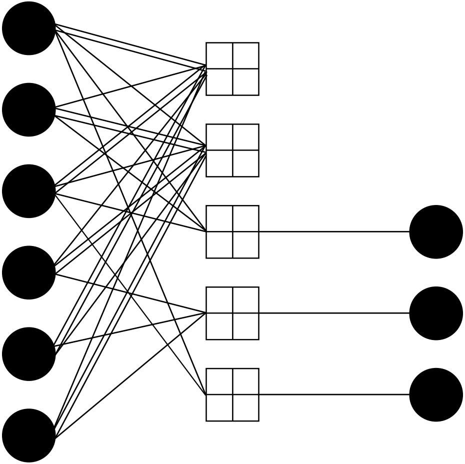
PBRL Protograph



PBRL Protograph



PBRL Protograph



Original PBRL Design Procedure

$$P_{\mathsf{HRC}}$$

$$P = \left[\begin{array}{cccccccc|c} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{array}\right] \text{Rate } 6/7, \text{ Threshold} = 3.07\text{dB}$$

$$P_{\mathsf{IRC}}$$

Original PBRL Design Procedure

$$P_{\mathsf{HRC}}$$

$$P = \left[\begin{array}{cccccccc|c}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{array}\right]$$

Rate 6/7, Threshold = 3.07dB

Rate 6/8, Threshold = 2.11dB

$$P_{\mathsf{IRC}}$$

Original PBRL Design Procedure

$$P_{\text{HRC}}$$

$$P = \left[\begin{array}{cccccccc|cc} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ & & & & & & & & & \\ & & & & & & & & & \end{array}\right] \begin{array}{l} \\ \text{Rate } 6/7, \text{ Threshold} = 3.07\text{dB} \\ \text{Rate } 6/8, \text{ Threshold} = 2.11\text{dB} \\ \text{Rate } 6/9, \text{ Threshold} = 1.61\text{dB} \\ \\ \\ \end{array}$$

$$P_{\text{IRC}}$$

Original PBRL Design Procedure

$$P_{\mathsf{HRC}}$$

$$P = \left[\begin{array}{cccccccc|ccc}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
 & & & & & & & & & & \\
 & & & & & & & & & &
\end{array}\right]
\begin{array}{r}
\\
\text{Rate } 6/7\text{, Threshold} = 3.07\text{dB} \\
\text{Rate } 6/8\text{, Threshold} = 2.11\text{dB} \\
\text{Rate } 6/9\text{, Threshold} = 1.61\text{dB} \\
6/10\text{, Threshold} = 1.17\text{dB} \\
\\
\\
\end{array}$$

$$P_{\mathsf{IRC}}$$

Original PBRL Design Procedure

P_{HRC}

$$P = \left[\begin{array}{cccccccc|ccccc} 2&1&2&1&2&1&2&1&0&0&0&0&0\\ 1&2&1&2&1&2&1&2&0&0&0&0&0\\ \hline 1&0&1&0&1&0&0&0&1&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&1&0&0&0\\ 1&1&1&0&1&0&1&0&0&0&1&0&0\\ &&&\vdots&&&&&&&\vdots&& \end{array}\right] \begin{array}{r} \\ \text{Rate } 6/7, \text{ Threshold} = 3.07\text{dB}\\ \text{Rate } 6/8, \text{ Threshold} = 2.11\text{dB}\\ \text{Rate } 6/9, \text{ Threshold} = 1.61\text{dB}\\ 6/10, \text{ Threshold} = 1.17\text{dB}\\ \\ \end{array}$$

P_{IRC}

Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes

2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results

3. A Property of PBRL Decoding

4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes

5. A Rate Allocation Problem

UCLA Samueli School of Engineering

Why a New Design Method?

- Threshold $\iff$ long-block-length property

Why a New Design Method?

- Threshold $\iff$ long-block-length property
  - Cannot be used for **low FER requirements at short block-lengths**

Why a New Design Method?

- Threshold $\iff$ long-block-length property
  - Cannot be used for **low FER requirements at short block-lengths**
- No explicit design method known previously for short RC-LDPC codes

Why a New Design Method?

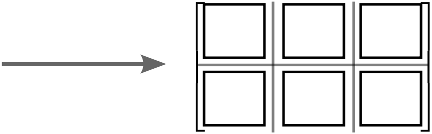
- Threshold $\iff$ long-block-length property
  - Cannot be used for **low FER requirements at short block-lengths**
- No explicit design method known previously for short RC-LDPC codes
  - Short convolutional codes are pretty much the only class of codes used so far for short block-length rate-compatibility

Why a New Design Method?

- Threshold $\iff$ long-block-length property
  - Cannot be used for **low FER requirements at short block-lengths**
- No explicit design method known previously for short RC-LDPC codes
  - Short convolutional codes are pretty much the only class of codes used so far for short block-length rate-compatibility
- Because we can do better...

A Minimum Distance Upper Bound

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

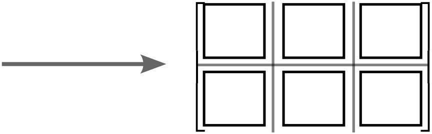


CPM blocks $N \times N$ in place of each 1

A Minimum Distance Upper Bound

CPM blocks $N \times N$ in place of each 1

$$\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$



Theorem (Permanent Bound)

The minimum distance of a protograph QC-LDPC code is upper bounded by a constant that does not depend upon N and that depends only upon the protograph.

R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum hamming distance upper bounds," *IEEE Trans. Inf. Theory*, Feb. 2012.

Complexity of Computing Permanent Bound

Theorem (Complexity of Permanent Bound)

For a protomatrix of size $n_c \times n_v$, the complexity of computing this upper bound is $\Theta\left(\binom{n_v}{n_c+1}(n_c+1)\cdot n_c 2^{n_c}\right)$.

R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum hamming distance upper bounds," *IEEE Trans. Inf. Theory*, Feb. 2012.

Complexity of Computing Permanent Bound

Theorem (Complexity of Permanent Bound)

For a protomatrix of size $n_c \times n_v$, the complexity of computing this upper bound is $\Theta\left(\binom{n_v}{n_c+1}(n_c+1)\cdot n_c 2^{n_c}\right)$.

Note the dependence on n_c, n_v (the size of the whole protomatrix)

R. Smarandache and P. O. Vontobel, "Quasi-cyclic LDPC codes: Influence of proto- and tanner-graph structure on minimum hamming distance upper bounds," *IEEE Trans. Inf. Theory*, Feb. 2012.

Permanent Bound Design (PBD) Method

$$P = \underset{P_{\mathsf{IRC}}}{\overset{P_{\mathsf{HRC}}}{\left[\begin{array}{cccccccc|c} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & \\ \hline & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{array}\right]}}$$

Rate 6/7, Min. Dist. Upper Bound = 8

Rate 6/8, Min. Dist. Upper Bound = 12

Permanent Bound Design (PBD) Method

$$
\begin{array}{c}
P_{\mathsf{HRC}} \\
P = \left[\begin{array}{cccccccc|c}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 \\
\hline
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
\end{array}\right] \\
P_{\mathsf{IRC}}
\end{array}
$$

Rate 6/7, Min. Dist. Upper Bound $= 8$

Rate 6/8, Min. Dist. Upper Bound $= 12$

Permanent Bound Design (PBD) Method

$$\begin{array}{cl}
\qquad P_{\mathsf{HRC}} & \\
P = \left[\begin{array}{cccccccc|cc}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 \\
\hline
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
 & & & & & & & & & \\
 & & & & & & & & & \\
\end{array}\right] & \begin{array}{l}
\\
\text{Rate } 6/7\text{, Min. Dist. Upper Bound} = 8 \\
\text{Rate } 6/8\text{, Min. Dist. Upper Bound} = 12 \\
\text{Rate } 6/9\text{, Min. Dist. Upper Bound} = 14 \\
\\
\\
\end{array} \\
\qquad P_{\mathsf{IRC}} &
\end{array}$$

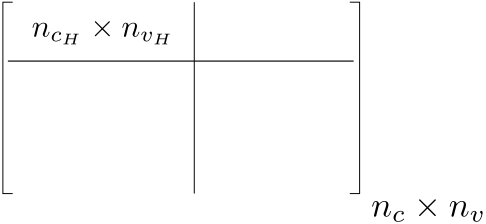
Permanent Bound Design (PBD) Method

$$P_{\mathsf{HRC}}$$

$$P = \left[\begin{array}{cccccccc|ccccc} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & \vdots & & & & & & & \vdots & & \\ & & & & & & & & & & & & \end{array}\right] \begin{array}{l} \\ \text{Rate } 6/7\text{, Min. Dist. Upper Bound} = 8 \\ \text{Rate } 6/8\text{, Min. Dist. Upper Bound} = 12 \\ \text{Rate } 6/9\text{, Min. Dist. Upper Bound} = 14 \\ \\ \\ \end{array}$$

$$P_{\mathsf{IRC}}$$

Complexity Reduction for PBRL Protomatrices



Theorem (Permanent Bound Complexity for PBRL Protomatrices)

For a PBRL protomatrix of size $n_c \times n_v$, the complexity of computing the permanent upper bound is $\Theta\left(\binom{n_{v_H}}{n_{c_H}+1}(n_c+1)\cdot(n_{c_H}+1)\,2^{(n_{c_H}+1)}\right)$.

Complexity Reduction for PBRL Protomatrices

$$\left[\begin{array}{c|c} n_{c_H} \times n_{v_H} & \\ \hline & \\ & \end{array}\right]_{n_c \times n_v}$$

Theorem (Permanent Bound Complexity for PBRL Protomatrices)

For a PBRL protomatrix of size $n_c \times n_v$, the complexity of computing the permanent upper bound is $\Theta\left(\binom{n_{v_H}}{n_{c_H}+1}(n_c+1)\cdot(n_{c_H}+1)\,2^{(n_{c_H}+1)}\right)$.

Recall, it was previously $\Theta\left(\binom{n_v}{n_c+1}(n_c+1)\cdot n_c 2^{n_c}\right)$

Reduction in Complexity of Design Algorithm

P_{HRC}

$$P = \left[\begin{array}{cccccccc|c} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & \\ \hline & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{array}\right]$$

P_{IRC}

Rate 6/7, Min. Dist. Upper Bound = 8

Rate 6/8, Min. Dist. Upper Bound = 12

Reduction in Complexity of Design Algorithm

$$
P = \begin{array}{c} P_{\mathsf{HRC}} \\ \left[\begin{array}{cccccccc|c}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 \\
\hline
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{array}\right] \\ P_{\mathsf{IRC}} \end{array}
$$

Rate 6/7, Min. Dist. Upper Bound $= 8$

Rate 6/8, Min. Dist. Upper Bound $= 12$

Reduction in Complexity of Design Algorithm

$$\begin{array}{c} P_{\mathsf{HRC}} \\ P = \left[\begin{array}{cccccccc|cc} 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ & & & & & & & & & \\ & & & & & & & & & \end{array}\right] \begin{array}{l} \\ \text{Rate } 6/7\text{, Min. Dist. Upper Bound} = 8 \\ \text{Rate } 6/8\text{, Min. Dist. Upper Bound} = 12 \\ \text{Rate } 6/9\text{, Min. Dist. Upper Bound} = 14 \\ \\ \\ \end{array} \\ P_{\mathsf{IRC}} \end{array}$$

Reduction in Complexity of Design Algorithm

$$
\begin{array}{c}
P_{\mathsf{HRC}} \\
P = \left[\begin{array}{cccccccc|ccccc}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & & \vdots & & & & & & & \vdots & & \\
 & & & & & & & & & & & &
\end{array}\right]
\begin{array}{l}
\\
\text{Rate } 6/7\text{, Min. Dist. Upper Bound} = 8 \\
\text{Rate } 6/8\text{, Min. Dist. Upper Bound} = 12 \\
\text{Rate } 6/9\text{, Min. Dist. Upper Bound} = 14 \\
\\
\\
\end{array} \\
P_{\mathsf{IRC}}
\end{array}
$$

Reduction in Complexity of Design Algorithm

Turns out, we can do better!

Reduction in Complexity of Design Algorithm

Theorem (Reduced Complexity Design Algorithm)

Reduction in Complexity of Design Algorithm

Theorem (Reduced Complexity Design Algorithm)

- *The complexity of a "pre-compute" step is* $\Theta\left(\binom{n_{v_H}}{n_{c_H}+1}\left(n_{c_H}+1\right)\cdot n_{c_H}2^{n_{c_H}}+\binom{n_{v_H}}{n_{c_H}+1}\cdot\left(n_{c_H}+1\right)2^{\left(n_{c_H}+1\right)}\right)$.

Reduction in Complexity of Design Algorithm

Theorem (Reduced Complexity Design Algorithm)

- *The complexity of a "pre-compute" step is* $\Theta\left(\binom{n_{v_H}}{n_{c_H}+1}(n_{c_H}+1)\cdot n_{c_H}2^{n_{c_H}}+\binom{n_{v_H}}{n_{c_H}+1}\cdot(n_{c_H}+1)\,2^{\left(n_{c_H}+1\right)}\right).$
- *For the design rows, the complexity is* $O\left(\binom{n_{v_H}}{n_{c_H}+1}n_{v_H}\right).$

The Design is An ILP

Theorem (Design of a Row $\iff$ ILP)

The design of one row of the IRC *part according to the PBD method is an integer linear program with* $\mathrm{dom} = \{\textit{candidate rows}\}$.

The LP Relaxation

Theorem (ILP $\neq$ LP relaxation)

The relaxation of the ILP is not exact.

The LP Relaxation is Still Useful

$$\left[\begin{array}{c|c} n_{c_H} \times n_{v_H} & \\ \hline & \\ & \end{array}\right]_{n_c \times n_v}$$

Theorem (New Upper Bound for PBRL Protomatrices)

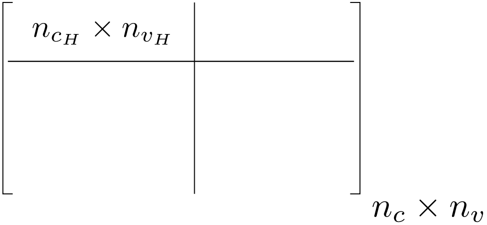
The LP Relaxation is Still Useful

$$\left[\begin{array}{c|c} n_{c_H} \times n_{v_H} & \\ \hline & \\ & \end{array}\right]_{n_c \times n_v}$$

Theorem (New Upper Bound for PBRL Protomatrices)

- *For a given* HRC *part, the LP relaxation provides a new set of upper bounds at all lower design rates.*

The LP Relaxation is Still Useful



Theorem (New Upper Bound for PBRL Protomatrices)

- *For a given* HRC *part, the LP relaxation provides a new set of upper bounds at all lower design rates.*
- *These upper bounds can be obtained without even having to go through the actual design procedure.*

Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes
5. A Rate Allocation Problem

UCLA Samueli School of Engineering

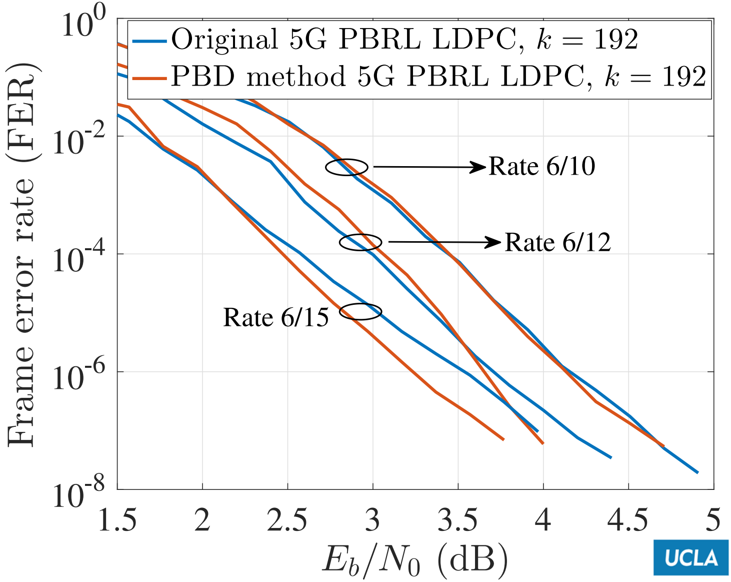
New PBRL Protograph for 5G at 192 Information Bits

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

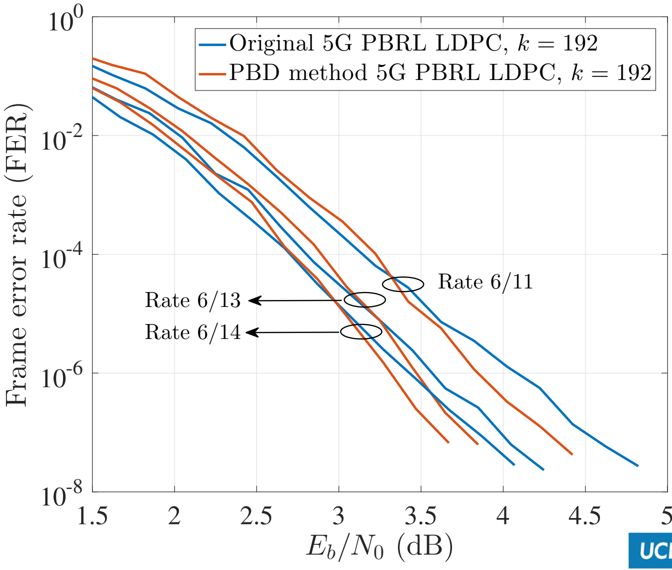
New PBRL Protograph for 5G at 192 Information Bits

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

New Codes for 5G at 192 Information Bits



New Codes for 5G at 192 Information Bits



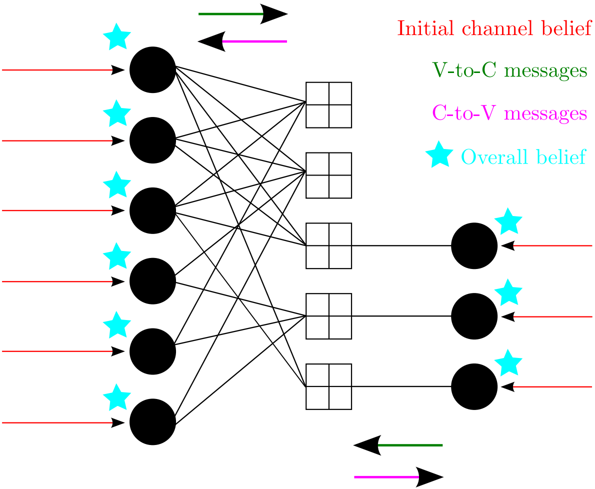
Outline

1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes
5. A Rate Allocation Problem

Based on. . .

S. V. S. Ranganathan, R. D. Wesel, and D. Divsalar, "Linear rate-compatible codes with degree-1 extending variable nodes under iterative decoding," In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2018.

Recall LDPC Decoding. . .



Initial channel belief

V-to-C messages

C-to-V messages

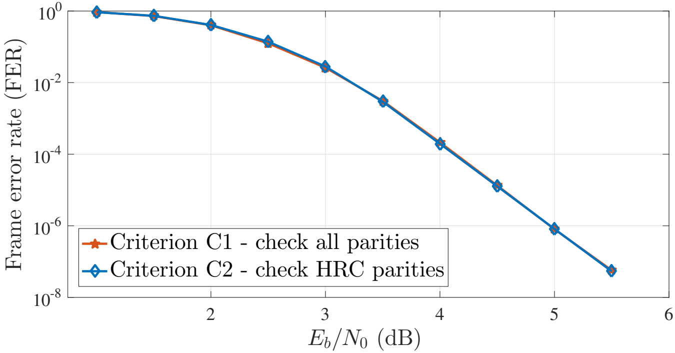
★ Overall belief

Early Stopping in PBRL Decoding

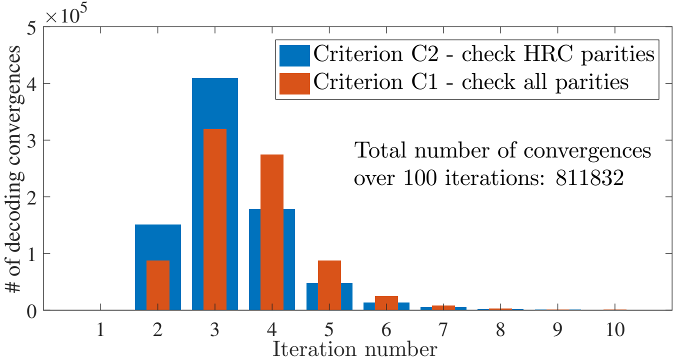
Theorem

It is sufficient to check whether the HRC variable nodes have converged to a codeword.

Checking Only HRC Parities – No Penalty



Checking Only HRC Parities – Faster Convergence



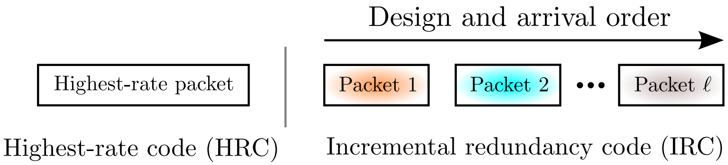
Outline

1. Incremental Redundancy (IR) and LDPC Codes
    - Rate-Compatible Codes, Protograph LDPC Codes
    - Protograph QC-LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
    - Protograph-Based Raptor-Like LDPC (PBRL) Codes
    - Permanent-Bound-Based Design (PBD) Method
    - Simulation Results
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
    - UIO-PBRL Codes
5. A Rate Allocation Problem

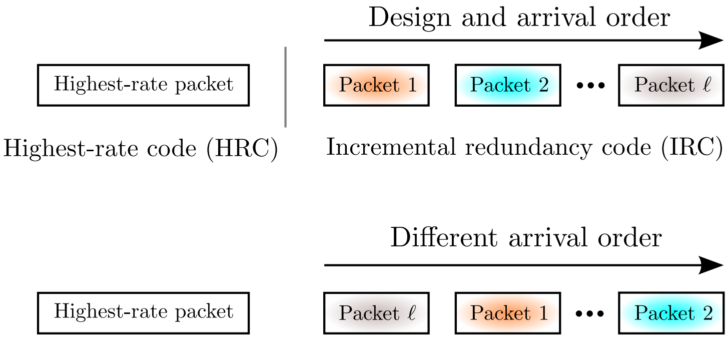
Based on. . .

S. V. S. Ranganathan, K. Vakilinia, D. Divsalar, and R. D. Wesel, “Universal rate-compatible LDPC code families for any increment ordering,” In *Proc. 9th Int. Symp. Turbo Codes & Iterative Inf. Processing (ISTC)*, Sep. 2016.

Increment Arrival Order



Increment Arrival Order



Objective

- **Metric 1**: Require every order of arrival have same performance

Objective

- **Metric 1**: Require every order of arrival have same performance
  - Inspired by requirement of inter-frame coding

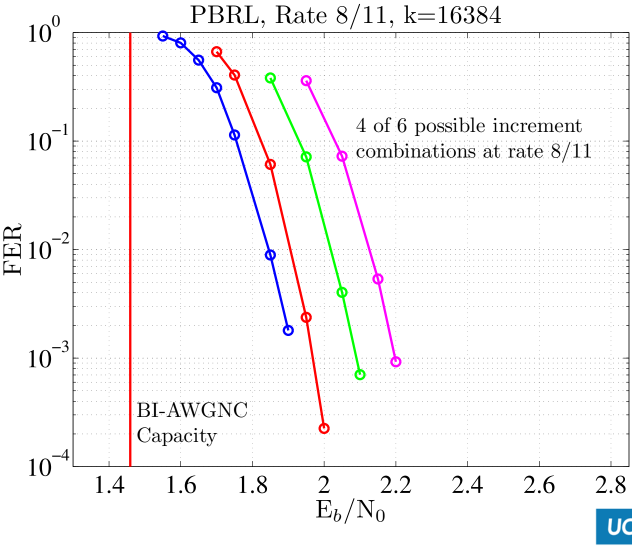
H. Wang, **S. V. S. Ranganathan**, and R. D. Wesel, “Approaching capacity using incremental redundancy without feedback,” In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2017.

Objective

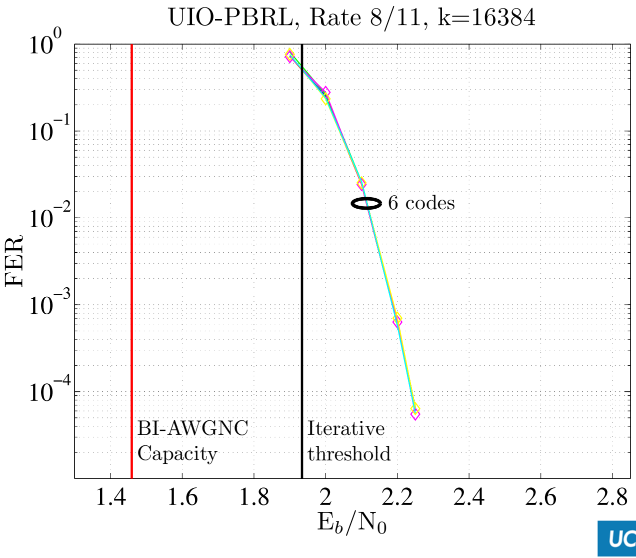
- **Metric 1**: Require every order of arrival have same performance
  - Inspired by requirement of inter-frame coding
- **Metric 2**: Require best throughput as you decode in a feedback system

H. Wang, **S. V. S. Ranganathan**, and R. D. Wesel, "Approaching capacity using incremental redundancy without feedback," In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2017.

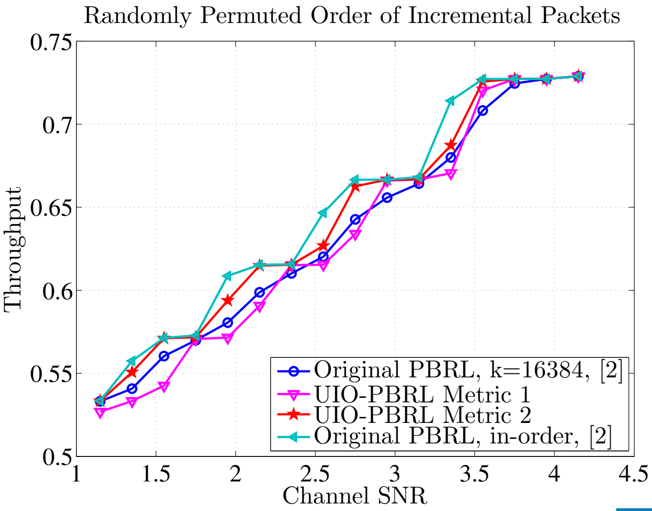
Original PBRL Code – Different Arrival Orderings



UIO-PBRL Code Designed for Metric 1



UIO-PBRL Code Designed for Metric 2



Outline

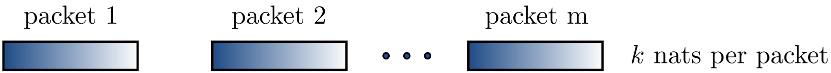
1. Incremental Redundancy (IR) and LDPC Codes
   - Rate-Compatible Codes, Protograph LDPC Codes
   - Protograph QC-LDPC Codes
2. Quasi-Cyclic PBRL Design for Short Block-Lengths
   - Protograph-Based Raptor-Like LDPC (PBRL) Codes
   - Permanent-Bound-Based Design (PBD) Method
   - Simulation Results
3. A Property of PBRL Decoding
4. PBRL Codes for Universal Increment Ordering
   - UIO-PBRL Codes
5. A Rate Allocation Problem

UCLA Samueli School of Engineering

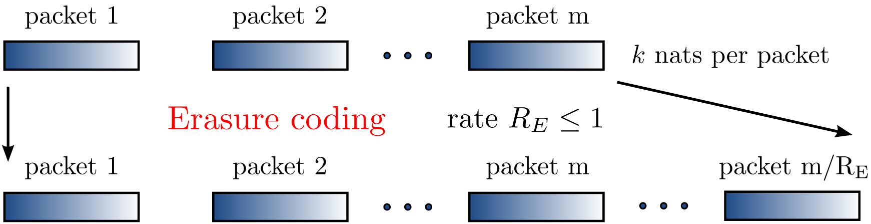
Based on. . .

S. V. S. Ranganathan, T. Mu, and R. D. Wesel, “Allocating redundancy between erasure coding and channel coding when fading channel diversity grows with codeword length,” *IEEE Trans. Commun.*, Aug. 2017.

Problem Setup



Problem Setup



packet 1 packet 2 ... packet m

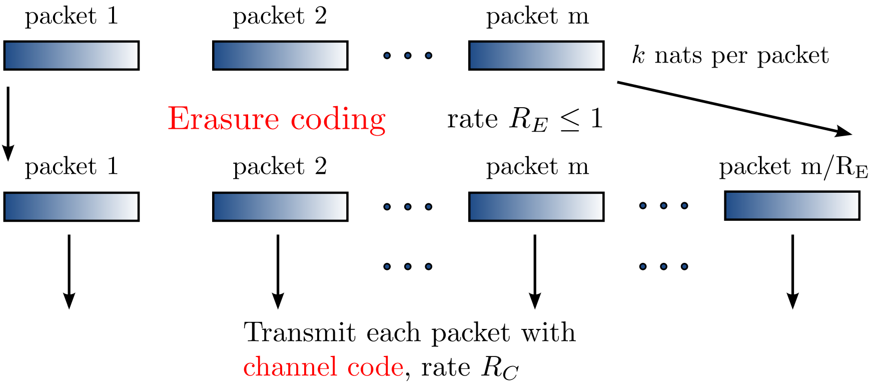
k nats per packet

Erasure coding

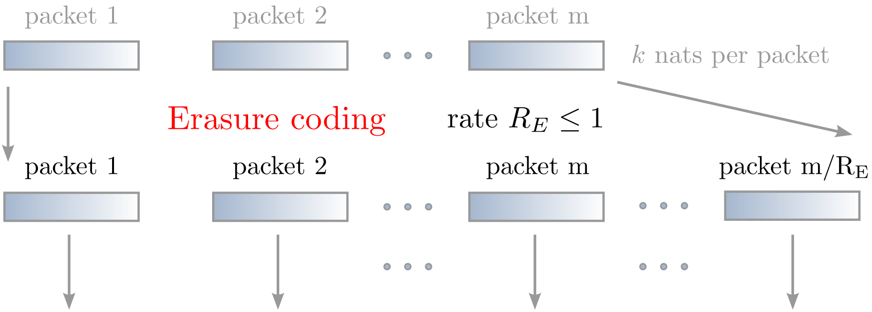
rate $R_E \leq 1$

packet 1 packet 2 ... packet m ... packet $\mathrm{m/R_E}$

Problem Setup



Problem Setup



packet 1 packet 2 ⋯ packet m

k nats per packet

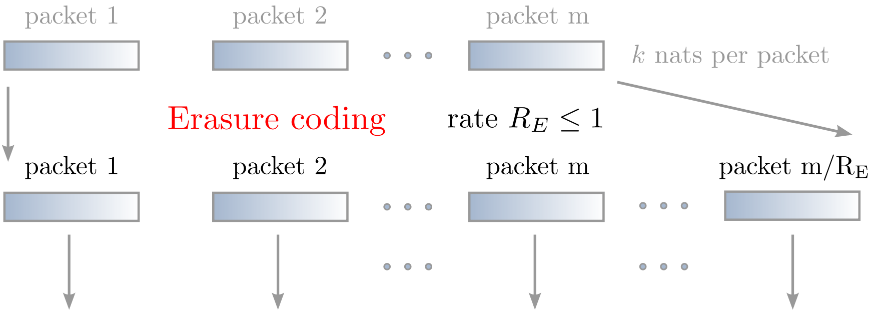
Erasure coding rate $R_E \leq 1$

packet 1 packet 2 ⋯ packet m ⋯ packet m/R$_\mathrm{E}$

Transmit each packet with
channel code, rate R_C

Overall T channel
time units to transmit

Problem Setup



packet 1 packet 2 packet m

k nats per packet

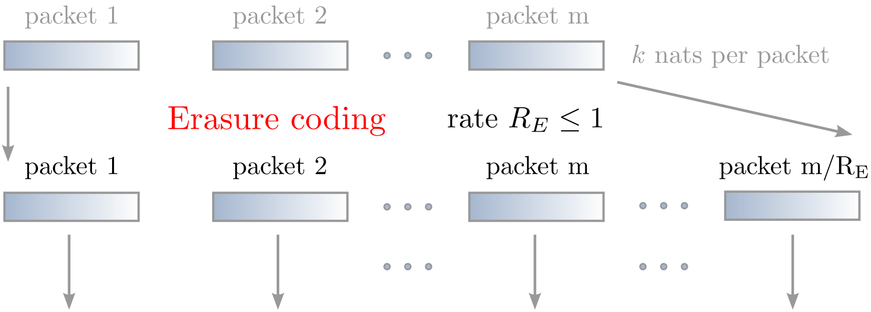
Erasure coding rate $R_E \leq 1$

packet 1 packet 2 packet m packet $\mathrm{m/R_E}$

Transmit each packet with channel code, rate R_C

Overall T channel time units to transmit $\Longrightarrow$ $\frac{k}{R_C}$ channel symbols per packet

Problem Setup



packet 1 packet 2 ⋯ packet m

k nats per packet

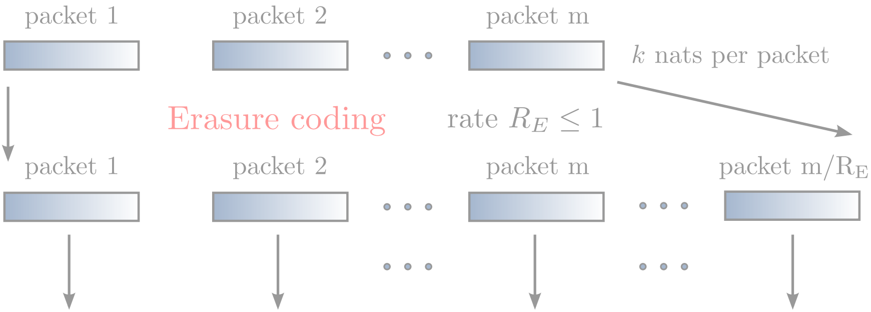
Erasure coding rate $R_E \leq 1$

packet 1 packet 2 ⋯ packet m ⋯ packet m/$\mathrm{R_E}$

Transmit each packet with channel code, rate R_C

Overall T channel time units to transmit $\Longrightarrow$ $\frac{k}{R_C} \cdot \frac{m}{R_E} = T$

Problem Setup



packet 1 packet 2 ⋯ packet m

k nats per packet

Erasure coding rate $R_E \le 1$

packet 1 packet 2 ⋯ packet m ⋯ packet m/R$_E$

Transmit each packet with channel code, rate R_C

"Overall" rate constraint

Overall T channel time units to transmit $\Longrightarrow$ $\frac{k}{R_C} \cdot \frac{m}{R_E} = T$ $\Longrightarrow$ $\frac{mk}{T} = R_E R_C$

UCLA Samueli School of Engineering

Questions

- Fixing a target probability of failure λ and T, what is the minimum operating SNR at PHY

Questions

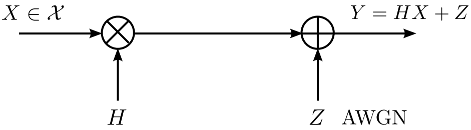
- Fixing a target probability of failure λ and T, what is the minimum operating SNR at PHY
- For the fixed T, what should be R_C^*, R_E^*?

Questions

- Fixing a target probability of failure λ and T, what is the minimum operating SNR at PHY
- For the fixed T, what should be R_C^*, R_E^*?
- How do R_C^*, R_E^* behave as a function of T, as $T \rightarrow \infty$?

Fading Model

Rayleigh fading channel
Average SNR $= P$



$X \in \mathcal{X}$

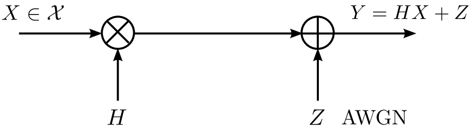
$Y = HX + Z$

H

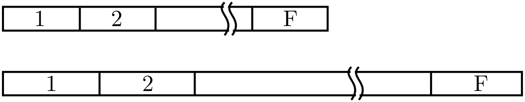
Z AWGN

Fading Model

Rayleigh fading channel
Average SNR = P



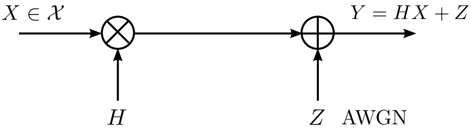
Courtade and Wesel
Block fading



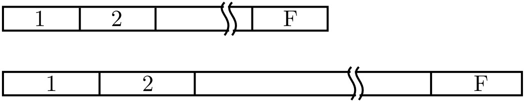
Channel codeword for
two values of R_C

Fading Model

Rayleigh fading channel
Average SNR = P

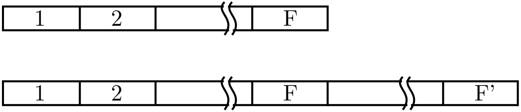


Courtade and Wesel
Block fading

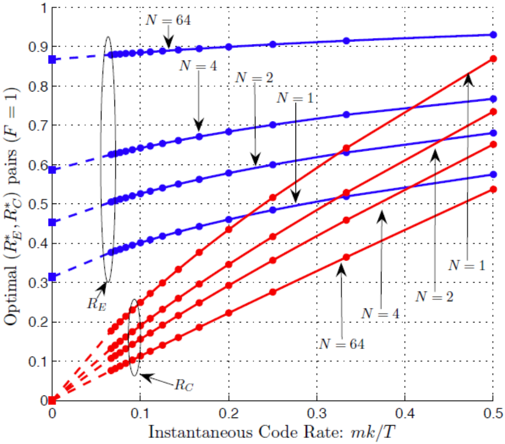


Channel codeword for
two values of R_C

Proportional-diversity
block fading
(PD block fading)

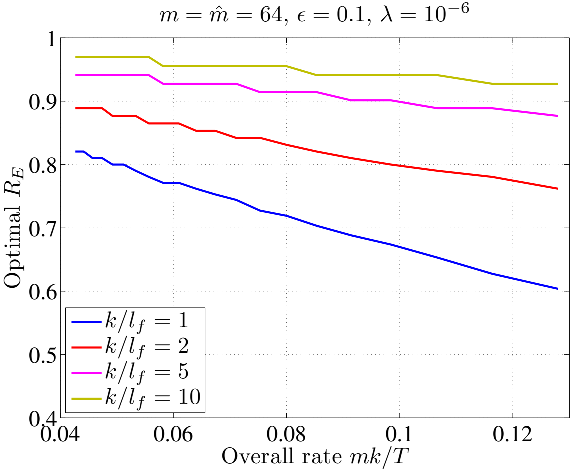


Courtade and Wesel – Block Fading



T. A. Courtade and R. D. Wesel, "Optimal allocation of redundancy between packet-level erasure coding and physical-layer channel coding in fading channels," *IEEE Trans. Commun.*, Aug. 2011.

Proportional-Diversity Block Fading



Theoretical Result

Theorem

Let the coding scheme use an arbitrary erasure code capable of producing any number of packets and a "very good" channel code. Let us assume a Rayleigh proportional-diversity block-fading channel.

Theoretical Result

Theorem

Let the coding scheme use an arbitrary erasure code capable of producing any number of packets and a "very good" channel code. Let us assume a Rayleigh proportional-diversity block-fading channel.
Then, for any sufficiently large T the optimal value of R_E is equal to its highest possible value.

Conclusions

- We designed new PBRL codes for short block-lengths

Conclusions

- We designed new PBRL codes for short block-lengths
  - Demonstrated the effectiveness using a design for 5G

Conclusions

- We designed new PBRL codes for short block-lengths
    - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes

Conclusions

- We designed new PBRL codes for short block-lengths
    - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes
- We looked at an application of PBRL codes for universal increment ordering

Conclusions

- We designed new PBRL codes for short block-lengths
    - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes
- We looked at an application of PBRL codes for universal increment ordering
- We studied a cross-layer rate allocation problem

Conclusions

- We designed new PBRL codes for short block-lengths
    - Demonstrated the effectiveness using a design for 5G
- We studied a decoding property of PBRL codes
- We looked at an application of PBRL codes for universal increment ordering
- We studied a cross-layer rate allocation problem
    - We showed that erasure coding is unnecessary if there is enough diversity at the PHY layer

Acknowledgement

I would like to thank. . .

- Rick & Dariush

Acknowledgement

I would like to thank. . .

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang

Acknowledgement

I would like to thank...

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek

Acknowledgement

I would like to thank. . .

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu

Acknowledgement

I would like to thank. . .

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
- BZ Shen

I would like to thank. . .

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
- BZ Shen
- National Science Foundation (NSF), Broadcom Foundation, S. A. Photonics – for their funding support

Acknowledgement

I would like to thank. . .

- Rick & Dariush
- Kasra Vakilinia & Haobo Wang
- Prof. Dolecek
- My committee members – Profs. Fragouli, Diggavi, and Lu
- BZ Shen
- National Science Foundation (NSF), Broadcom Foundation, S. A. Photonics – for their funding support
- Engineering Graduate Students Association

Appendix – Permanent Upper Bound

Theorem (Smarandache and Vontobel)

Let a protomatrix P with a positive design rate and no punctured variable nodes be of size $n_c \times n_v$. If $S \subseteq [n_v]$, denote by P_S the sub-matrix of P formed by the columns indexed by elements of S. Then, any QC-LDPC code $\mathcal{C}$ obtained from the protomatrix P has a minimum distance $d_{min}(\mathcal{C})$ that is upper bounded as

$$d_{min}(\mathcal{C}) \leqslant \min^*_{S\subseteq[n_v],|S|=n_c+1} \sum_{i\in S} \text{perm}\left(P_{S\setminus i}\right), \tag{2}$$

where $|\cdot|$ refers to the cardinality of a set, $S\setminus i$ is shorthand for $S\setminus\{i\}$, and min^* returns the smallest non-zero value in a set of non-negative values with at least one positive value or $+\infty$ if the set is $\{0\}$.

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$\sum_{i \in S} \text{perm}\left(P_{S \setminus i}\right)$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) = 10$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) = 10 + 10$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) = 10 + 10 + 4$$

UCLA Samueli School of Engineering

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$$\sum_{i \in S} \text{perm}\left(P_{S \setminus i}\right) = 10 + 10 + 4 + 4$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) = 10 + 10 + 4 + 4 + 20$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 3, 4, 6, 7, 9\}$$

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) = 68$$

$$\min^* (\{68, \ldots\})$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$\sum_{i \in S} \mathsf{perm}\left(P_{S \setminus i}\right) = 164$$

$$\min{}^{*}\left(\{68, 164, \ldots\}\right)$$

Appendix – Permanent Bound Example

$$P = \begin{bmatrix} 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\binom{n_v}{n_c+1}$ sets of size $n_c + 1$

For each set $n_c + 1$ permanents of size $n_c \times n_c$

$\binom{n_v}{n_c+1}(n_c + 1)$ permanents of size $n_c \times n_c$

Appendix – Permanent

- A a square matrix, perm$(A) = \sum_\sigma \prod_j A(\sigma(j), j)$

Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

$$\text{perm}(A) = 8+$$

Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

$$\mathsf{perm}(A) = 8 + 8$$

Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

$$\text{perm}(A) = 8 + 8 + 2$$

Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

$$\mathsf{perm}(A) = 8 + 8 + 2 + 0$$

Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

$$\mathsf{perm}(A) = 8 + 8 + 2 + 0 + 8$$

UCLA Samueli School of Engineering

Appendix – Permanent

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix} \in \mathbb{Z}^{3\times 3}$$

$$\mathsf{perm}(A) = 26$$

Appendix – Permanent

Best algorithm (Ryser) is of complexity $\Theta\left(\ell \cdot 2^{\ell}\right)$ for matrix of size $\ell \times \ell$