IEEE GLOBECOM 2018, Abu-Dhabi
December 2018

Short-packet communications: fundamentals and practical coding schemes

Giuseppe Durisi, durisi@chalmers.se
Gianluigi Liva, gianluigi.liva@dlr.de
Fabian Steiner, fabian.steiner@tum.de

Outline

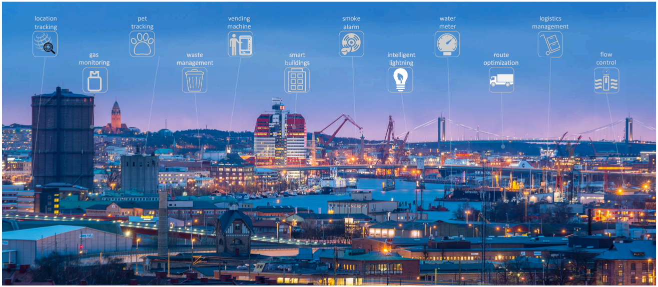
- Motivations
- Finite-blocklength performance bounds
- Applications
- Efficient Short Channel Codes
- Higher-Order Modulation

Outline

- Motivations
- Finite-blocklength performance bounds
- Applications
- Efficient Short Channel Codes
- Higher-Order Modulation

Machine-type communications (MTC)

Key enabler of future autonomous systems

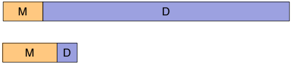


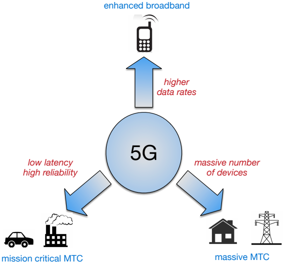
source: IoTpool

- 5G ⇒ massive MTC; ultra-reliable, low-latency comm.
- Low-power wireless-area networks ⇒ LoRa-WAN, SigFox,. . .

MTC traffic has unique characteristics: how to support it?

Unique characteristics of MTC traffic





- massive number of connected terminals
- transmitters are often idle
- short data packets
- low latency, high reliability
- high energy efficiency

Example

Long-term evolution (4G)

- Long packets (500 bytes)
- Packet error probability of 10^{-1} at 5ms latency
- High reliability through retransmissions (HARQ)

MTC for factory automation

- Short packets: 100 bits of payload
- maximum delay of $100\,\mu$s
- packet error probability in the range $[10^{-5}, 10^{-9}]$

Example

Long-term evolution (4G)

- Long packets (500 bytes)
- Packet error probability of 10^{-1} at 5ms latency
- High reliability through retransmissions (HARQ)

MTC for factory automation

- Short packets: 100 bits of payload
- maximum delay of $100\,\mu$s
- packet error probability in the range $[10^{-5}, 10^{-9}]$

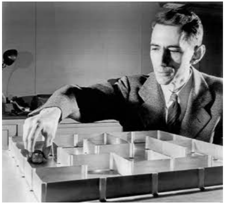
- We need a fundamental paradigm shift in the design of wireless communication
- This tutorial: new fundamental tools & new practical coding schemes

Outline

- Motivations
- Finite-blocklength performance bounds
- Applications
- Efficient Short Channel Codes
- Higher-Order Modulation

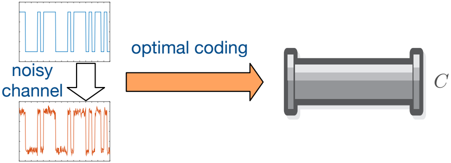
A new toolbox: finite-blocklength information theory

The old toolbox: asymptotic information theory

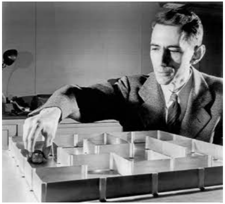


Claude E. Shannon
(1916–2001)

The bit-pipe approximation

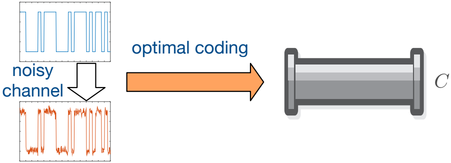


The old toolbox: asymptotic information theory



Claude E. Shannon
(1916–2001)

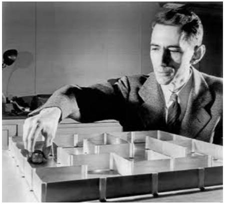
The bit-pipe approximation



$\log(1 + \text{sinr})$ formula used everywhere beyond PHY

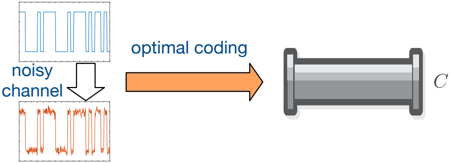
- resource allocation & user scheduling
- delay analyses at the network level

The old toolbox: asymptotic information theory



Claude E. Shannon
(1916–2001)

The bit-pipe approximation

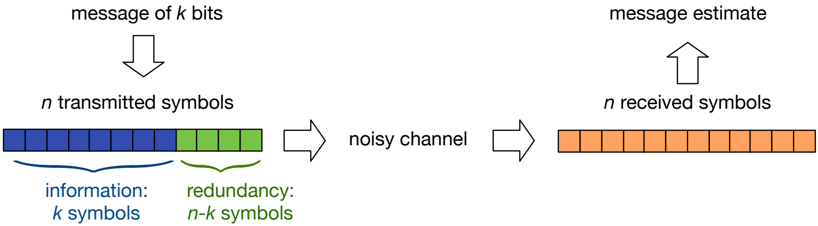


$\log(1+\text{sinr})$ formula used everywhere beyond PHY

- resource allocation & user scheduling
- delay analyses at the network level

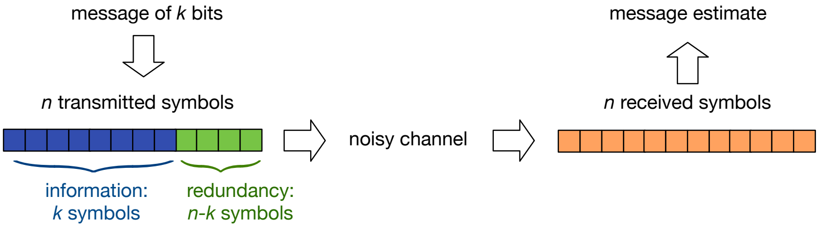
If packet are shorts, bit-pipe approximation is not accurate!

Channel-coding problem



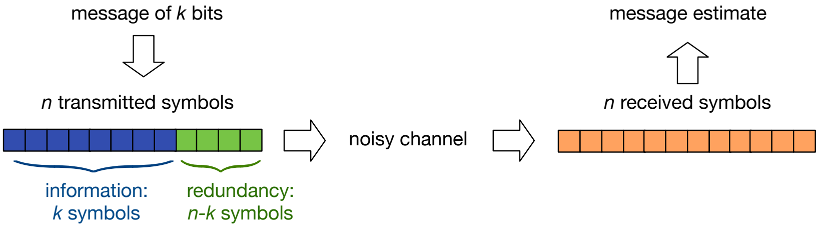
- More redundancy $\Rightarrow$ lower packet error probability $\epsilon \ldots$

Channel-coding problem



- More redundancy $\Rightarrow$ lower packet error probability $\epsilon\ldots$
- ... but also lower transmission rate $R = k/n$

Channel-coding problem



- More redundancy $\Rightarrow$ lower packet error probability $\epsilon\ldots$
- $\ldots$but also lower transmission rate $R = k/n$

Which triplets (k, n, ϵ) are possible?

An unsolvable problem?

Which triplets (k, n, ϵ) are possible?

An unsolvable problem?

Which triplets (k, n, ϵ) are possible?

- Smallest blocklength $n^*(k, \epsilon)$
- Largest number of bits $k^*(n, \epsilon)$
- Largest rate $R^*(n, \epsilon) = k^*(n, \epsilon)/n$
- Smallest error probability $\epsilon^*(k, n)$

An unsolvable problem?

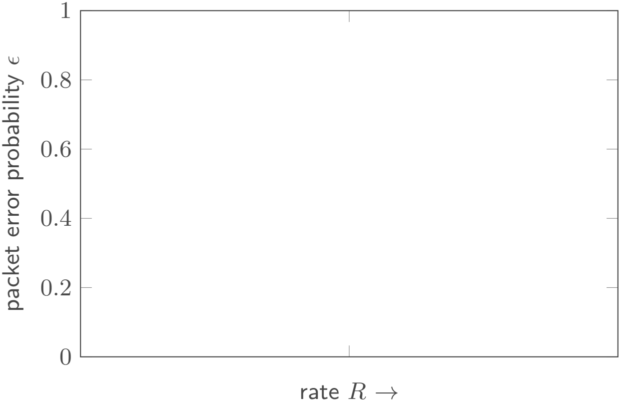
Which triplets (k, n, ϵ) are possible?

- Smallest blocklength $n^*(k, \epsilon)$
- Largest number of bits $k^*(n, \epsilon)$
- Largest rate $R^*(n, \epsilon) = k^*(n, \epsilon)/n$
- Smallest error probability $\epsilon^*(k, n)$

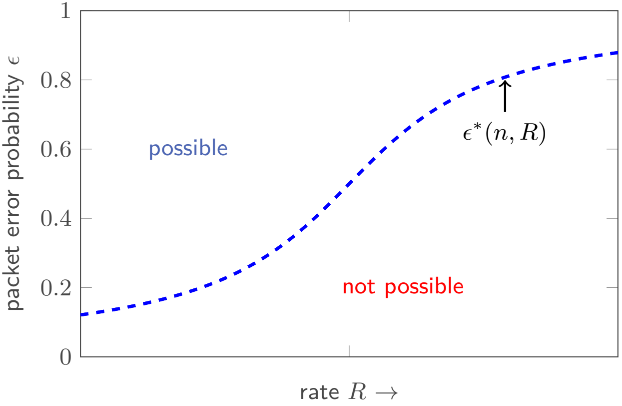
A very hard problem even for binary-input channels!

- Exhaustive search over $\binom{2^n}{2^k}$ codes
- Example: $k = 5$, $n = 10 \Rightarrow 5 \times 10^{60}$ codes!!

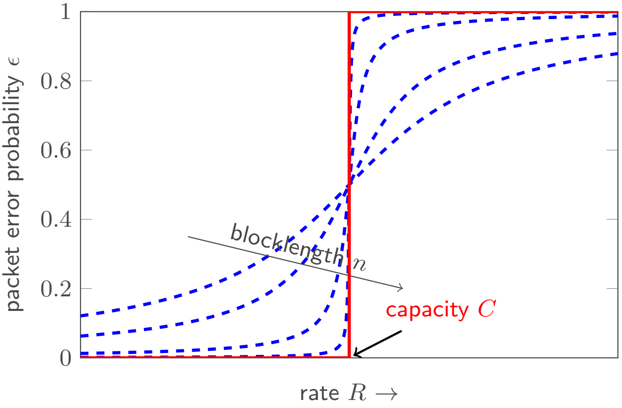
To infinity and back. . .



To infinity and back. . .

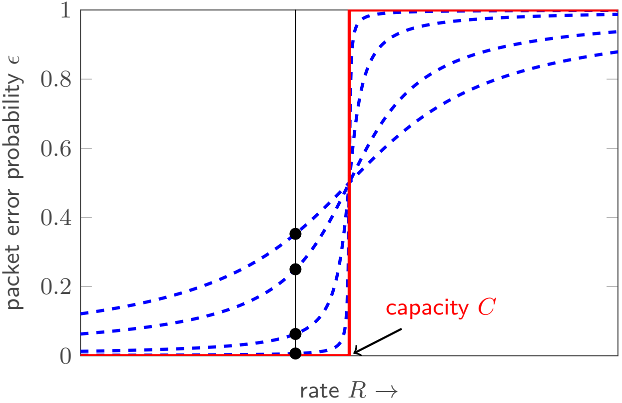


To infinity and back...



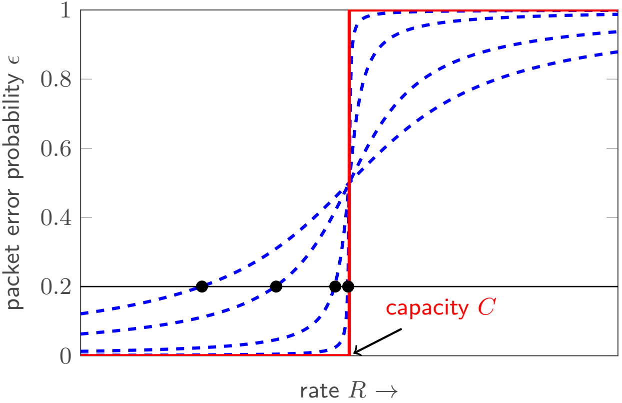
1948: Shannon, channel capacity

To infinity and back...



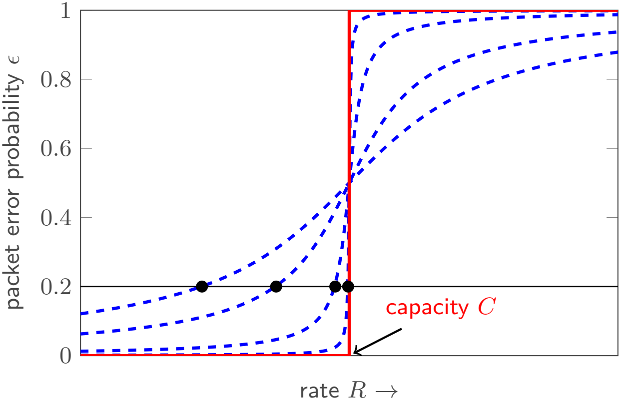
Vertical asymptotics ⇒ error exponent
(Gallager,...)

To infinity and back...



Horizontal asymptotics $\Rightarrow$ strong converse, fixed-error asymptotics
(Wolfowitz, Strassen,. . .)

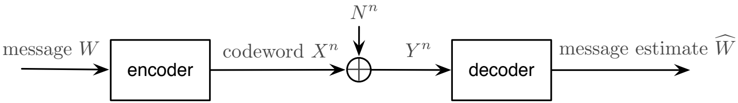
To infinity and back...



Today: tight computationally-feasible bounds and accurate approximations

(Hayashi 2009, Polyanskiy *et al.* 2010,...)

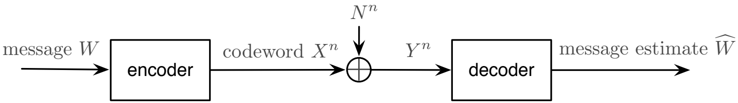
The bi-AWGN channel



$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \ldots, n$$

- $W \in \{1, \ldots, 2^k\}$

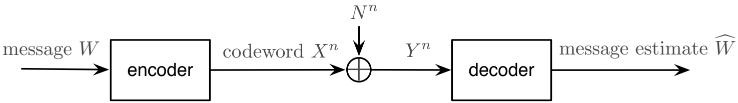
The bi-AWGN channel



$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \dots, n$$

- $W \in \{1, \dots, 2^k\}$
- $X^n = [X_1, \dots, X_n]$ with $X_j \in \{-1, 1\}$, $j = 1, \dots, n$

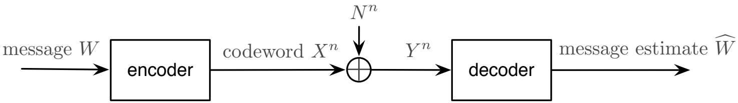
The bi-AWGN channel



$$Y_j = \sqrt{\mathrm{snr}}X_j + N_j, \quad j = 1, \dots, n$$

- $W \in \{1, \dots, 2^k\}$
- $X^n = [X_1, \dots, X_n]$ with $X_j \in \{-1, 1\}$, $j = 1, \dots, n$
- $N^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$

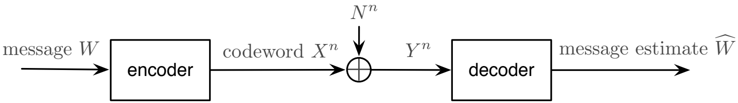
The bi-AWGN channel



$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \ldots, n$$

- $W \in \{1, \ldots, 2^k\}$
- $X^n = [X_1, \ldots, X_n]$ with $X_j \in \{-1, 1\}$, $j = 1, \ldots, n$
- $N^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
- n: blocklength (size of coded packet)

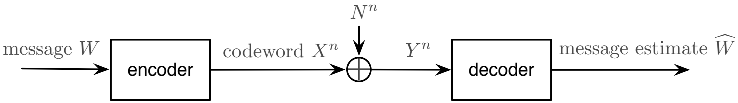
The bi-AWGN channel



$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \dots, n$$

- $W \in \{1, \dots, 2^k\}$
- $X^n = [X_1, \dots, X_n]$ with $X_j \in \{-1, 1\}$, $j = 1, \dots, n$
- $N^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
- n: blocklength (size of coded packet)
- $\epsilon = \mathbb{P}[\widehat{W} \neq W]$: packet error probability

The bi-AWGN channel



$$Y_j = \sqrt{\text{snr}} X_j + N_j, \quad j = 1, \dots, n$$

- $W \in \{1, \dots, 2^k\}$
- $X^n = [X_1, \dots, X_n]$ with $X_j \in \{-1, 1\}$, $j = 1, \dots, n$
- $N^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
- n: blocklength (size of coded packet)
- $\epsilon = \mathbb{P}[\widehat{W} \neq W]$: packet error probability
- $R = k/n$: rate [bits/channel use]

snr **vs.** E_s/N_0 **vs.** E_b/N_0

Real-valued AWGN channel

$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \dots, n$$

- $N^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
- $X_j \in \mathbb{R}, \quad j = 1, \dots, n$
- $\mathbb{E}\left[|X_j|^2\right] = 1$
- $\dfrac{E_s}{N_0} = \mathrm{snr}$
- $\dfrac{E_b}{N_0} = \dfrac{\mathrm{snr}}{2R}$

snr **vs.** E_s/N_0 **vs.** E_b/N_0

Real-valued AWGN channel

$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \dots, n$$

- $N^n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
- $X_j \in \mathbb{R}, \quad j = 1, \dots, n$
- $\mathbb{E}\left[|X_j|^2\right] = 1$
- $\dfrac{E_s}{N_0} = \mathrm{snr}$
- $\dfrac{E_b}{N_0} = \dfrac{\mathrm{snr}}{2R}$

Complex-valued AWGN channel

$$Y_j = \sqrt{\mathrm{snr}} X_j + N_j, \quad j = 1, \dots, n$$

- $N^n \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_n)$
- $X_j \in \mathbb{C}, \quad j = 1, \dots, n$
- $\mathbb{E}\left[|X_j|^2\right] = 1$
- $\dfrac{E_s}{N_0} = \mathrm{snr}$
- $\dfrac{E_b}{N_0} = \dfrac{\mathrm{snr}}{R}$

Shannon's capacity of bi-AWGN

Shannon's capacity:

Largest rate of reliable communication in the large n limit

$$C = \lim_{\epsilon \to 0} \lim_{n \to \infty} R^{\star}(n, \epsilon)$$

Shannon's coding theorem

The capacity of the bi-AWGN $P_{Y|X}$ is

$$C = \sup_{P_X} I(X;Y)$$

Shannon's capacity of bi-AWGN

Shannon's capacity:

Largest rate of reliable communication in the large n limit

$$C = \lim_{\epsilon \to 0} \lim_{n \to \infty} R^{\star}(n, \epsilon)$$

Shannon's coding theorem

The capacity of the bi-AWGN $P_{Y|X}$ is

$$C = \sup_{P_X} I(X;Y)$$

Mutual information

$$I(X;Y) = \mathbb{E}\left[\log \frac{P_{Y|X}(Y|X)}{P_Y(Y)}\right] = D(P_{Y\,|\,X} P_X \,||\, P_Y P_X)$$

where

$$P_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y - \sqrt{\mathrm{snr}}x)^2}{2}\right)$$

Information density

Mutual information

$$I(X;Y) = \mathbb{E}\left[\log \frac{P_{Y|X}(Y|X)}{P_Y(Y)}\right] = D(P_X P_{Y\,|\,X} \,||\, P_X P_Y)$$

Information density

$$\imath(x;y) = \log \frac{P_{Y|X}(y|x)}{P_Y(y)}$$

- asymptotic IT: mean of $\imath(X;Y)$
- FBL-IT: tail distribution of $\imath(X^n;Y^n) = \sum_{j=1}^{n} \imath(X_j;Y_j)$

Computing capacity

$$C = \sup_{P_X} I(X;Y)$$

Computing capacity

$$C = \sup_{P_X} I(X;Y)$$

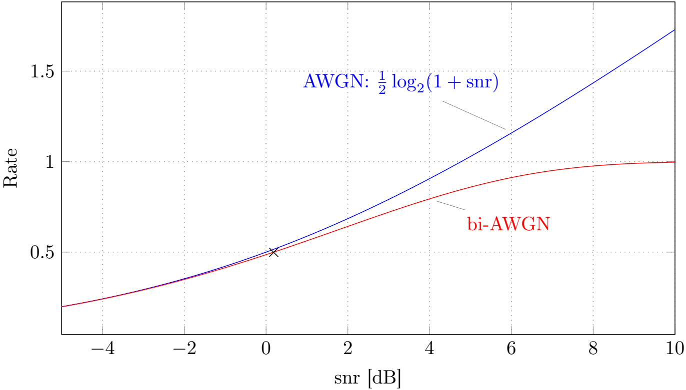
- P_X^* uniform over $\{-1, 1\}$
- $P_Y^* = (1/2)\mathcal{N}(-\sqrt{\mathrm{snr}}, 1) + (1/2)\mathcal{N}(\sqrt{\mathrm{snr}}, 1)$
- Information density

$$\imath(x;y) = \log \frac{P_{Y|X}(y|x)}{P_Y^*(y)} = \log 2 - \log\left(1 + \exp(-2xy\sqrt{\mathrm{snr}})\right)$$

- Capacity

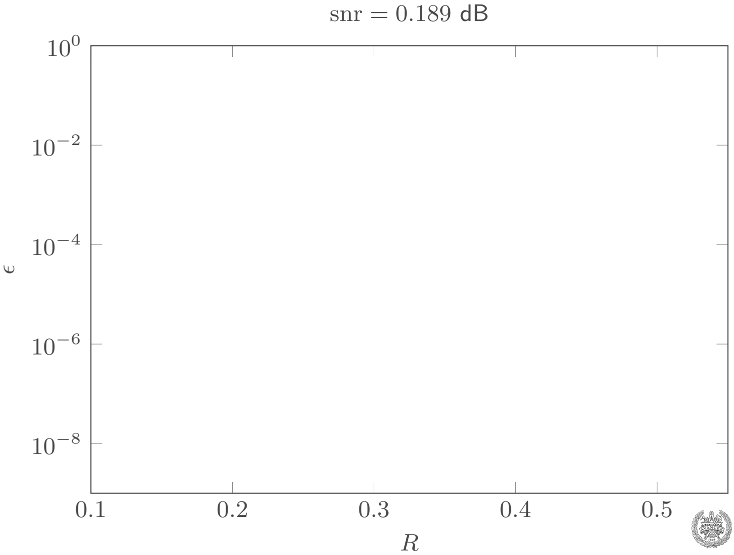
$$C = \frac{1}{\sqrt{2\pi}} \int e^{-z^2/2} \left(\log 2 - \log\left(1 + e^{-2\mathrm{snr} - 2z\sqrt{\mathrm{snr}}}\right)\right) \mathrm{d}z$$

Capacity of bi-AWGN channel

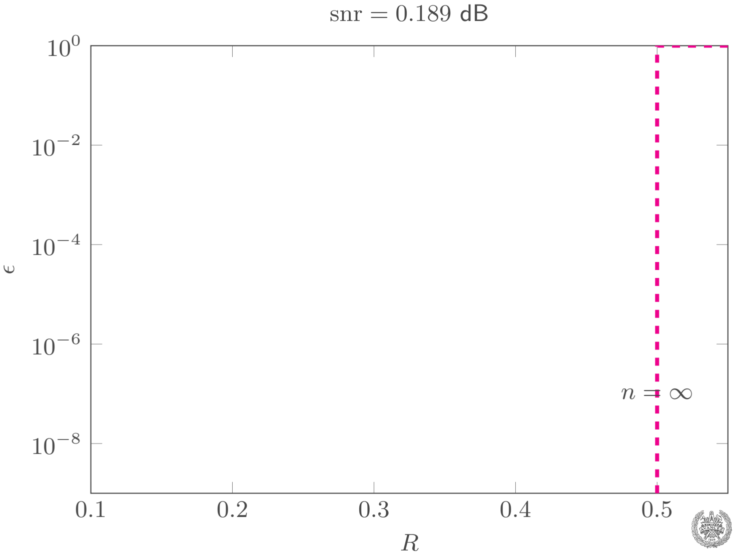


Note: $R = 0.5$ at $\mathrm{snr} = 0.189$ dB

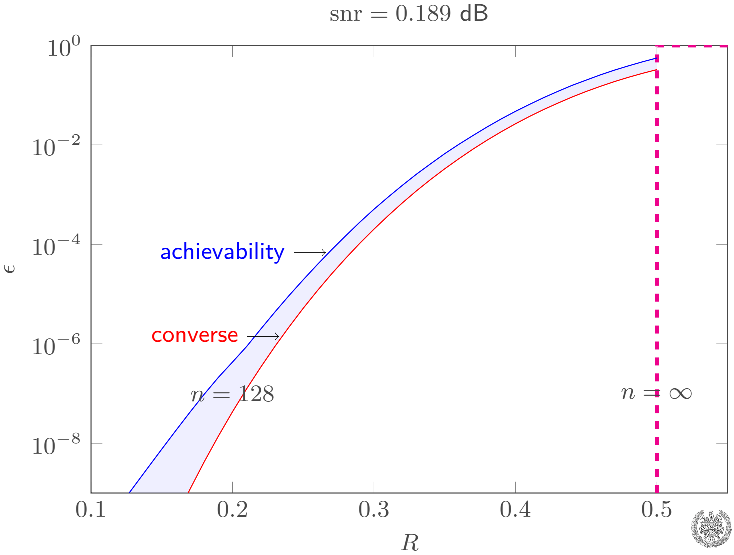
Finite blocklength: R vs. ϵ



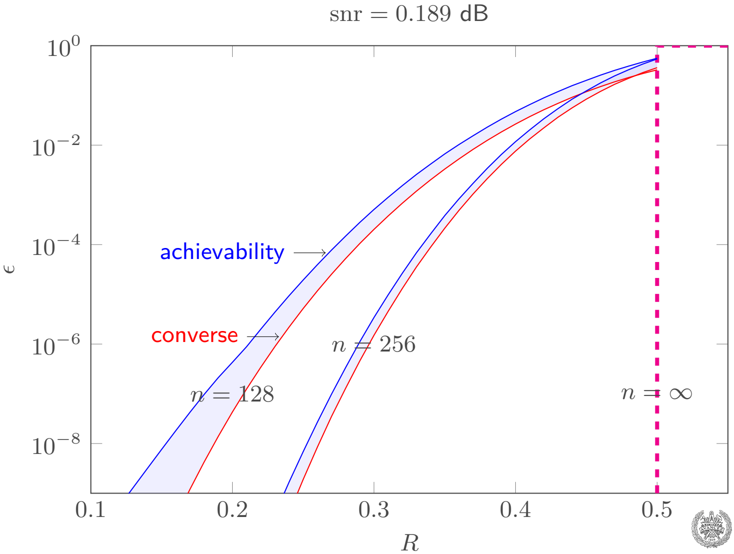
Finite blocklength: R vs. ϵ



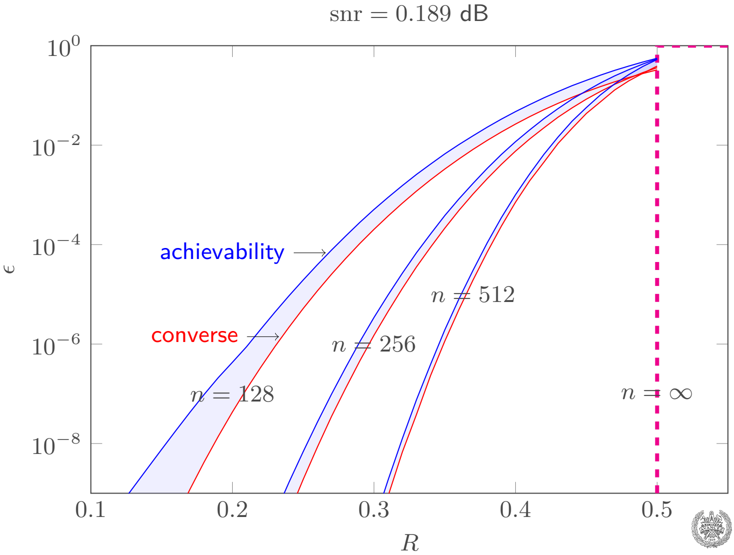
Finite blocklength: R vs. ϵ



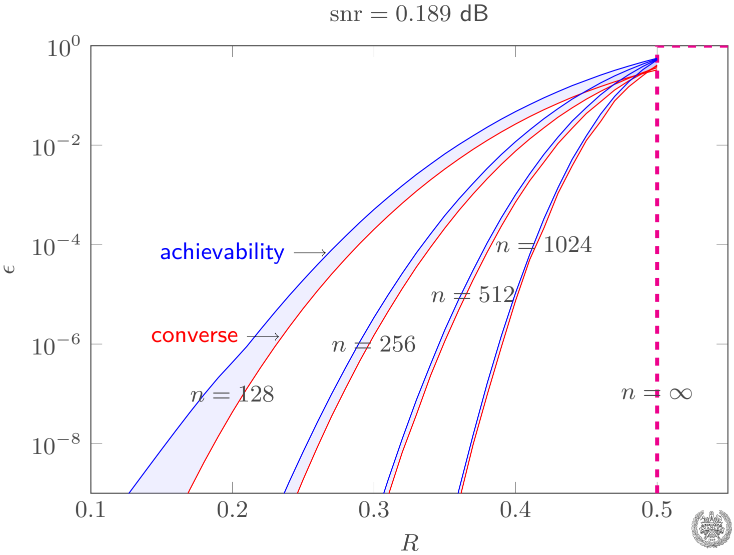
Finite blocklength: R vs. ϵ



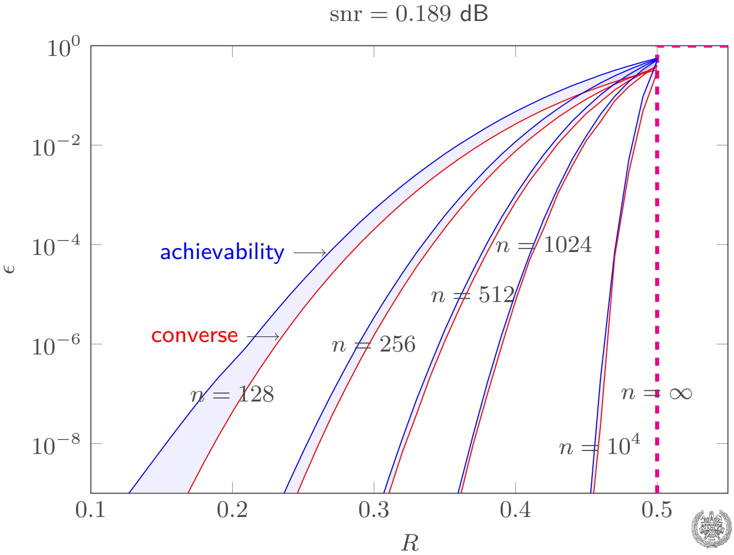
Finite blocklength: R vs. ϵ



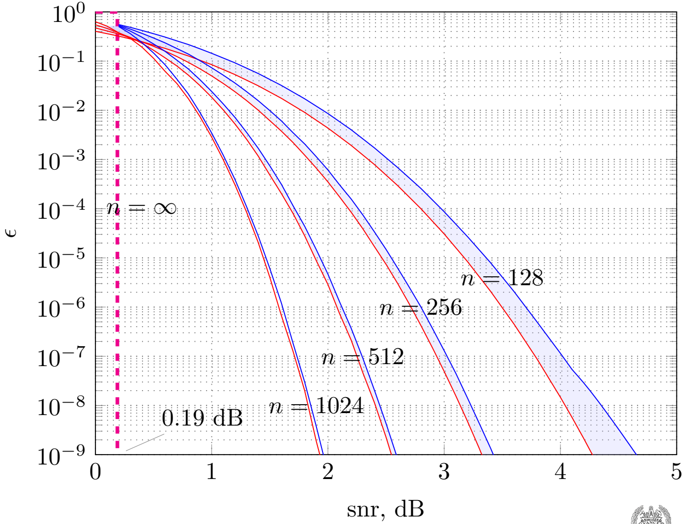
Finite blocklength: R vs. ϵ



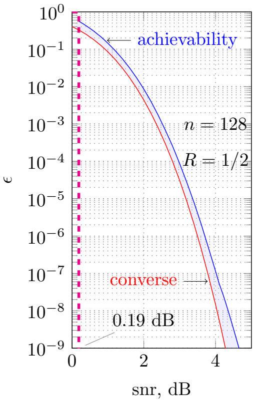
Finite blocklength: R vs. ϵ



A different perspective: snr vs ϵ at $R = 0.5$



Converse bound: a preview



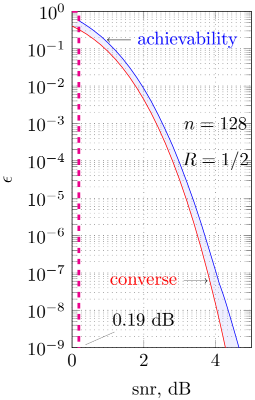
The converse bound

- Based on metaconverse (MC) theorem[1]
- Recovers all previously known converse bounds
- Relies on binary hypothesis testing
- Requires choosing wisely an auxiliary probability distribution[2]

[1] Y. Polyanskiy et al., "Channel coding rate in the finite blocklength regime", IEEE Trans. Inf. Theory (2010)

[2] G. Vazquez-Vilar et al., "Saddlepoint approximation of the error probability of binary hypothesis testing" in Proc. IEEE Int. Symp. Inf. Theory (ISIT) (July 2018)

Achievability bound: a preview



The achievability bound

- Based on the random coding union bound (RCU)[3]
- Not constructive
- Relies on maximum likelihood detection
- Generalizes naturally to arbitrary (mismatched) decoding metrics
- Tight in both normal and error-exponent regimes

[3] Y. Polyanskiy et al., "Channel coding rate in the finite blocklength regime", IEEE Trans. Inf. Theory (2010)

Five facts about the bounds that are good to know

- **Fact 1:** the bounds are tight

Five facts about the bounds that are good to know

- **Fact 1:** the bounds are tight
- **Fact 2:** the bounds are general

Five facts about the bounds that are good to know

- **Fact 1:** the bounds are tight
- **Fact 2:** the bounds are general
- **Fact 3:** the bounds can be computed efficiently

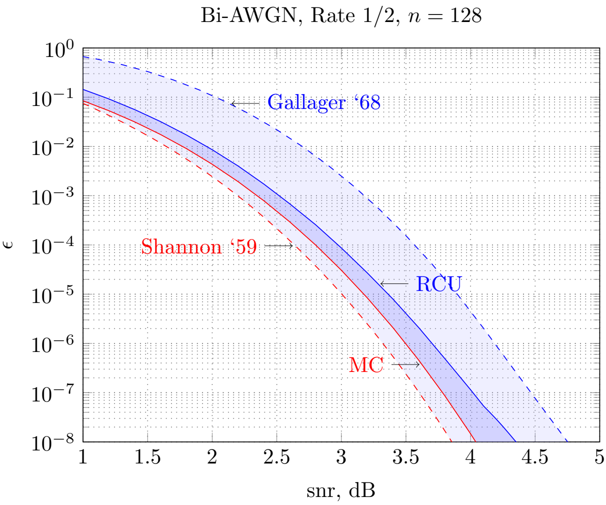
Five facts about the bounds that are good to know

- **Fact 1:** the bounds are tight
- **Fact 2:** the bounds are general
- **Fact 3:** the bounds can be computed efficiently
- **Fact 4:** the bounds can be approximated accurately using simple mathematical expressions

Five facts about the bounds that are good to know

- **Fact 1:** the bounds are tight
- **Fact 2:** the bounds are general
- **Fact 3:** the bounds can be computed efficiently
- **Fact 4:** the bounds can be approximated accurately using simple mathematical expressions
- **Fact 5:** numerical implementations of these bounds are available online

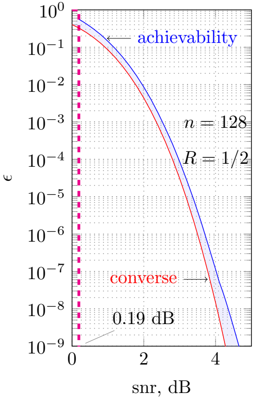
Fact 1: the bounds are tight



Fact 2: the bounds are general

- Discrete memoryless channels: BSC, BEC
- AWGN, bi-AWGN, coded modulation
- Fading channels under various CSI assumptions: from no CSI to full CSI at TX and Rx
- Pilot-assisted transmission, MIMO
- ARQ, HARQ, full feedback
- Joint coding and queuing analyses
- Erasure and list decoding
- Interference
- ...

Fact 3: The bounds can be computed efficiently



- Key problem: compute efficiently

$$\mathbb{P}[\imath(X^n; Y^n) \leq \gamma]$$

- Can be done using the saddlepoint method[4]
- Accurate results for blocklengths as small as 20
- Computational time for bi-AWGN: few seconds on a laptop computer

[4] A. Martinez and A. Guillén i Fàbregas, "Saddlepoint approximation of random–coding bounds", in Proc. Inf. Theory Applicat. Workshop (ITA) (2011)

Fact 4: the bounds are easy to approximate

Normal approximation

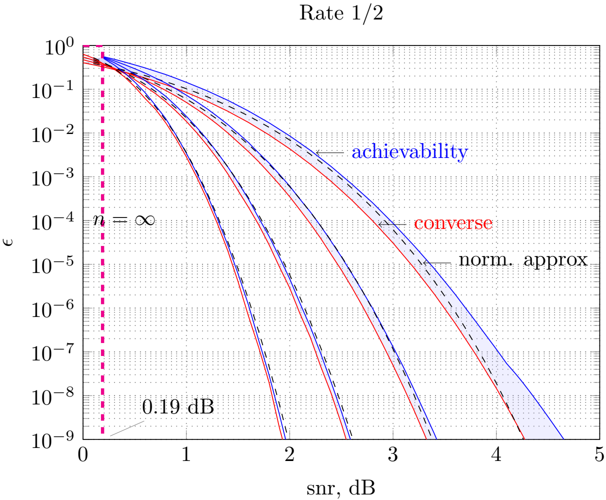
Metaconverse and RCUs expansions for fixed ϵ and $n \to \infty$ match up to third order for many channels!

$$R^*(n,\epsilon) = C - \sqrt{\frac{V}{n}}\mathrm{Q}^{-1}(\epsilon) + \frac{1}{2n}\log n + O\left(\frac{1}{n}\right)$$

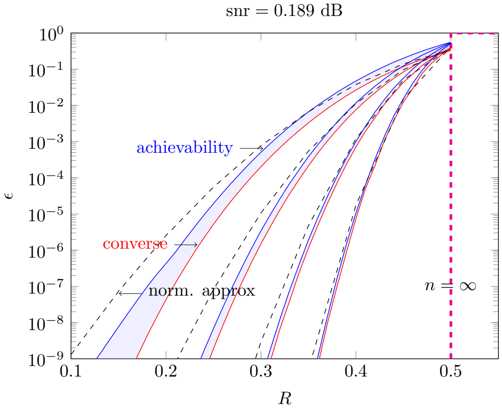
- C: capacity $\Rightarrow$ mean of $\imath(X;Y)$
- V: dispersion $\Rightarrow$ variance of $\imath(X;Y)$
- Proof via Berry-Esseen central limit theorem
- Useful approximation

$$\epsilon^*(k,n) \approx Q\left(\frac{nC - k + 0.5\log_2(n)}{\sqrt{nV}}\right)$$

Normal approximation is accurate for medium rates...



. . . but inaccurate for low rates and low error probabilities



Unsuitable for URLLC?

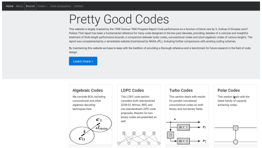
Fact 5: Numerical implementation of these bounds (and more) are available online

Spectre: github.com/yp-mit/spectre

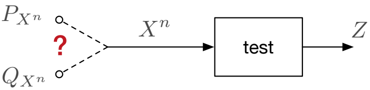
- Collection of numerical routines in finite-blocklength information theory
- Authors: Chalmers, MIT, Caltech, Padova, Technion, Princeton

pretty-good-codes.org

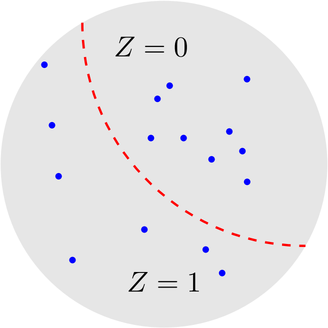
- Repository of channel coding schemes
- G. Liva (DLR) & F. Steiner (TUM)



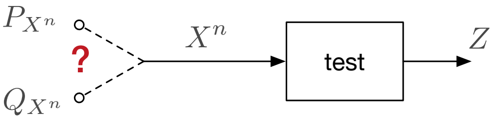
A closer look at the converse bound: binary hypothesis testing



- $Z \in \{0, 1\}$
- $Z = 0 \Rightarrow$ test chooses P_{X^n}



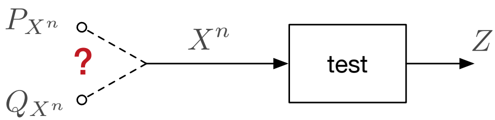
Optimal test



Neyman-Pearson β function

- Optimal test $P^*_{Z\,|\,X^n}$ minimizes error prob. under Q_{X^n} given a constraint on the success prob. under P_{X^n}

Optimal test



Neyman-Pearson β function

- Optimal test $P^*_{Z\,|\,X^n}$ minimizes error prob. under Q_{X^n} given a constraint on the success prob. under P_{X^n}
- $\beta_\alpha(P_{X^n}, Q_{X^n}) = \inf_{P_{Z\,|\,X^n} \,:\, P_{X^n}[Z=0] \geq \alpha} Q_{X^n}[Z = 0]$

Neyman-Pearson & Stein Lemmas

Neyman-Pearson Lemma

- The optimal test involves thresholding log-likelihood ratios
- $\beta_\alpha(P_{X^n}, Q_{X^n}) = Q_{X^n}\left[\log \frac{P_{X^n}}{Q_{X^n}}(X^n) \geq \gamma\right]$
- where $\gamma\,:\, P_{X^n}\left[\log \frac{P_{X^n}}{Q_{X^n}}(X^n) \geq \gamma\right] = \alpha$

Neyman-Pearson & Stein Lemmas

Neyman-Pearson Lemma

- The optimal test involves thresholding log-likelihood ratios
- $\beta_\alpha(P_{X^n}, Q_{X^n}) = Q_{X^n}\left[\log \frac{P_{X^n}}{Q_{X^n}}(X^n) \geq \gamma\right]$
- where $\gamma\ :\ P_{X^n}\left[\log \frac{P_{X^n}}{Q_{X^n}}(X^n) \geq \gamma\right] = \alpha$

Stein's Lemma

Assume that X^n has i.i.d. entries. Then $\beta_\alpha(P_{X^n}, Q_{X^n})$ decays to zero exponentially fast in n

$$\lim_{n\to\infty} \frac{1}{n} \log \beta_\alpha(P_{X^n}, Q_{X^n}) = -D(P_X \,||\, Q_X)$$

Neyman-Pearson & Stein Lemmas

Neyman-Pearson Lemma

- The optimal test involves thresholding log-likelihood ratios
- $\beta_\alpha(P_{X^n}, Q_{X^n}) = Q_{X^n}\left[\log \frac{P_{X^n}}{Q_{X^n}}(X^n) \geq \gamma\right]$
- where γ : $P_{X^n}\left[\log \frac{P_{X^n}}{Q_{X^n}}(X^n) \geq \gamma\right] = \alpha$

Stein's Lemma

Assume that X^n has i.i.d. entries. Then $\beta_\alpha(P_{X^n}, Q_{X^n})$ decays to zero exponentially fast in n

$$\lim_{n\to\infty} \frac{1}{n} \log \beta_\alpha(P_{X^n}, Q_{X^n}) = -D(P_X \,||\, Q_X)$$

But mutual information is a relative entropy; what is the underlying binary test?

The metaconverse framework

Min-max converse theorem

Fix an arbitrary Q_{Y^n}. Every (k, n, ϵ)–code satisfies

$$k \le \sup_{P_{X^n}} \left\{ -\log_2 \beta_{1-\epsilon}(P_{X^n} P_{Y^n \mid X^n}, P_{X^n} Q_{Y^n}) \right\}$$

[5] G. Vazquez-Vilar et al., "Saddlepoint approximation of the error probability of binary hypothesis testing", in Proc. IEEE Int. Symp. Inf. Theory (ISIT) (July 2018)

The metaconverse framework

Min-max converse theorem

Fix an arbitrary Q_{Y^n}. Every (k, n, ϵ)–code satisfies

$$k \leq \sup_{P_{X^n}} \left\{ -\log_2 \beta_{1-\epsilon}(P_{X^n} P_{Y^n \mid X^n}, P_{X^n} Q_{Y^n}) \right\}$$

Evaluation of the bound

- If one chooses $Q_{Y^n} = (P_Y^*)^n$,

$$\beta_{1-\epsilon}(P_{X^n} P_{Y^n \mid X^n}, P_{X^n} Q_{Y^n}) = \beta_{1-\epsilon}(P_{Y^n \mid X^n = \bar{x}}, Q_{Y^n})$$

  where $\bar{x} = [1, 1, \dots, 1]$

- Better choice: error-exponent-achieving output distribution[5]

[5] G. Vazquez-Vilar et al., "Saddlepoint approximation of the error probability of binary hypothesis testing", in Proc. IEEE Int. Symp. Inf. Theory (ISIT) (July 2018)

A closer look at the achievability bound

Random-coding union bound (RCU)

For every input distribution P_{X^n}, there exists a (k, n, ϵ) code satisfying

$$\epsilon \leq \mathbb{E}\left[\min\left\{1, (2^k - 1)\mathbb{P}[\imath(\bar{X}^n, Y^n) \geq \imath(X^n, Y^n)] \,|\, X^n, Y^n\right\}\right]$$

where $P_{X^n, \bar{X}^n, Y^n}(x^n, \bar{x}^n, y^n) = P_{Y^n \,|\, X^n}(y^n \,|\, x^n) P_{X^n}(x^n) P_{\bar{X}^n}(\bar{x}^n)$

- Proof: error probability under random coding and ML decoding + union bound

[6] J. Font-Segura et al., "Saddlepoint approximations of lower and upper bounds to the error probability in channel coding", in Proc. Conf. Inf. Sci. Sys. (CISS) (2018)

A closer look at the achievability bound

Random-coding union bound (RCU)

For every input distribution P_{X^n}, there exists a (k, n, ϵ) code satisfying

$$\epsilon \leq \mathbb{E}\left[\min\left\{1, (2^k - 1)\mathbb{P}[\imath(\bar{X}^n, Y^n) \geq \imath(X^n, Y^n)] \,|\, X^n, Y^n\right\}\right]$$

where $P_{X^n, \bar{X}^n, Y^n}(x^n, \bar{x}^n, y^n) = P_{Y^n \,|\, X^n}(y^n \,|\, x^n) P_{X^n}(x^n) P_{\bar{X}^n}(\bar{x}^n)$

- Proof: error probability under random coding and ML decoding + union bound
- Similar to derivation of Gallager's random coding error exponent

[6] J. Font-Segura et al., "Saddlepoint approximations of lower and upper bounds to the error probability in channel coding", in Proc. Conf. Inf. Sci. Sys. (CISS) (2018)

A closer look at the achievability bound

Random-coding union bound (RCU)

For every input distribution P_{X^n}, there exists a (k, n, ϵ) code satisfying

$$\epsilon \leq \mathbb{E}\left[\min\left\{1, (2^k - 1)\mathbb{P}[\imath(\bar{X}^n, Y^n) \geq \imath(X^n, Y^n)] \,|\, X^n, Y^n\right\}\right]$$

where $P_{X^n, \bar{X}^n, Y^n}(x^n, \bar{x}^n, y^n) = P_{Y^n \,|\, X^n}(y^n \,|\, x^n) P_{X^n}(x^n) P_{\bar{X}^n}(\bar{x}^n)$

- Proof: error probability under random coding and ML decoding + union bound
- Similar to derivation of Gallager's random coding error exponent
- $\imath(x^n, y^n)$ can be replaced by arbitrary mismatched metric

[6] J. Font-Segura et al., "Saddlepoint approximations of lower and upper bounds to the error probability in channel coding", in Proc. Conf. Inf. Sci. Sys. (CISS) (2018)

A closer look at the achievability bound

Random-coding union bound (RCU)

For every input distribution P_{X^n}, there exists a (k, n, ϵ) code satisfying

$$\epsilon \leq \mathbb{E}\left[\min\left\{1, (2^k - 1)\mathbb{P}[\imath(\bar{X}^n, Y^n) \geq \imath(X^n, Y^n)] \,|\, X^n, Y^n\right\}\right]$$

where $P_{X^n,\bar{X}^n,Y^n}(x^n, \bar{x}^n, y^n) = P_{Y^n \,|\, X^n}(y^n \,|\, x^n)P_{X^n}(x^n)P_{\bar{X}^n}(\bar{x}^n)$

- Proof: error probability under random coding and ML decoding + union bound
- Similar to derivation of Gallager's random coding error exponent
- $\imath(x^n, y^n)$ can be replaced by arbitrary mismatched metric
- Efficient saddlepoint approximation available[6]

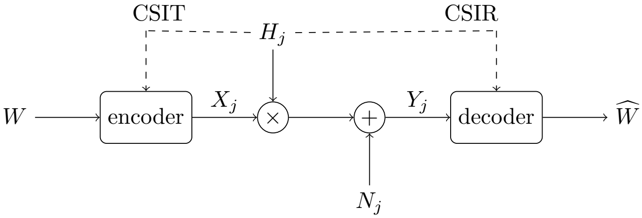
[6] J. Font-Segura et al., "Saddlepoint approximations of lower and upper bounds to the error probability in channel coding", in Proc. Conf. Inf. Sci. Sys. (CISS) (2018)

Outline

- Motivations
- Finite-blocklength performance bounds
- Applications
  - Example 1: short packets over fading channels
  - Example 2: joint queuing and coding analyses
- Efficient Short Channel Codes
- Higher-Order Modulation

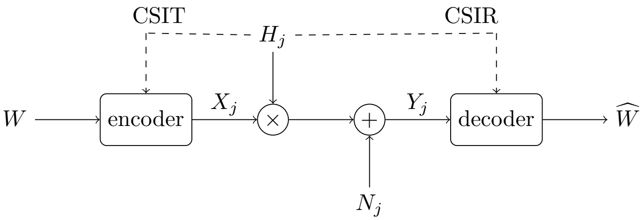
Outline

Enter fading



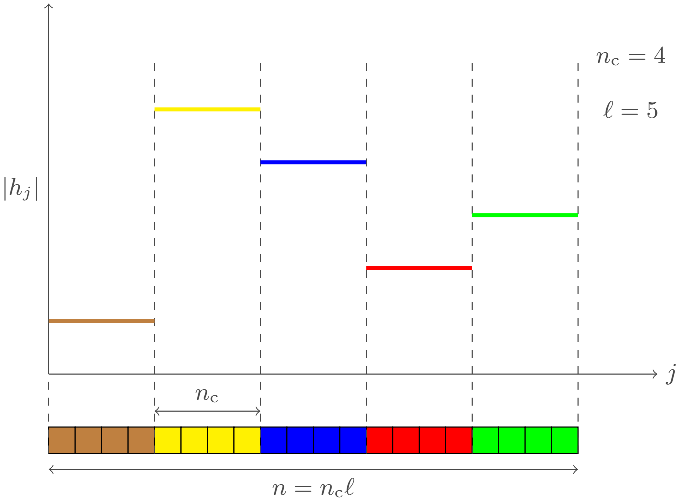
- AWGN channel with fluctuating SNR and mutiple inputs/outputs

Enter fading

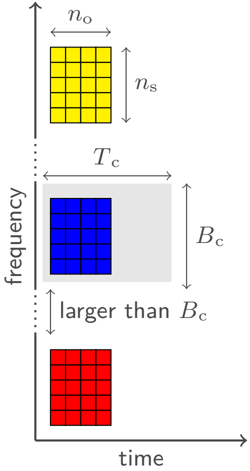


- AWGN channel with fluctuating SNR and mutiple inputs/outputs
- Performance limits depend on:
  - How $\{H_j\}$ varies within the packet
  - Fading knowledge: noCSI ,CSIR, CSIT, CSIRT

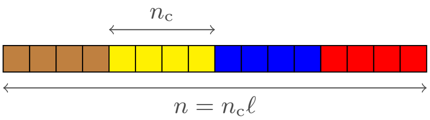
The memoryless block-fading model



Relevance to 5G



Two notions of capacity



Outage capacity

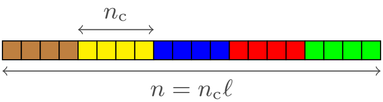
- $n_c \to \infty$, ℓ fixed
- Fading process stays "constant" over the packet
- ✗ Does not capture the "cost" of learning the channel at the receiver

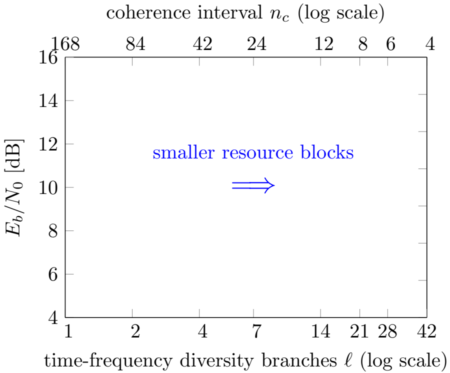
Ergodic capacity

- $\ell \to \infty$, n_c fixed
- Fading process varies rapidly over the packet
- ✗ Requires coding over many coherence intervals
- ✗ Does not depend on ϵ

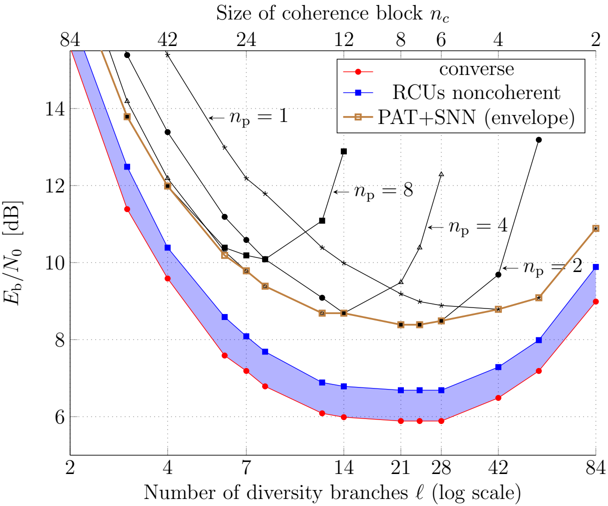
A 5G design problem

- information bits: $k = 81$
- packet size: $n = 168$ symbols
- 14 OFDM symbols, 12 tones per symbol
- Packet error prob.: $\epsilon = 10^{-3}$





SISO case[7]: $n = 168$, $k = 81$, $\epsilon = 10^{-3}$



[7] J. Östman et al., "Short packets over block-memoryless fading channels: pilot-assisted or noncoherent transmission?", IEEE Trans. Commun. (2018)

Outline

Beyond PHY analyses

Extend theory to include

- Queuing delay
- Random arrival of information packets

Beyond PHY analyses

Extend theory to include

- Queuing delay
- Random arrival of information packets

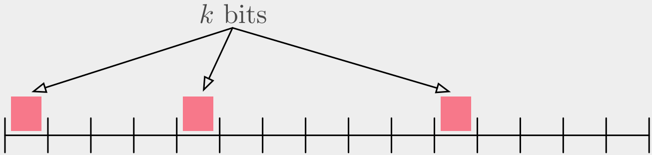
Performance metric: Steady-state delay violation probability

$$\mathbb{P}\left\{\text{packet delay} \geq \text{threshold}\right\}$$

Our setup

Random packet arrival and queue

- Packet arrival: i.i.d. Bernoulli process with parameter λ over channel uses

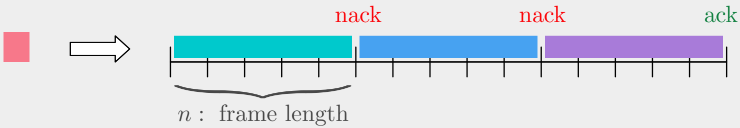


- Packets stored in a single-server FCFS queue

Service process

Service process

- simple ARQ with perfect error detection

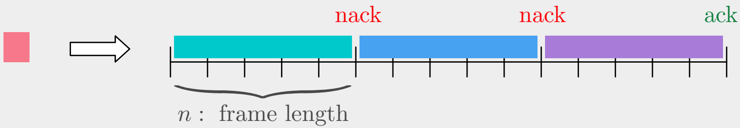


- AWGN channel, error-free, instantaneous 1-bit feedback.
- τ: number of frames after which ack is sent

Service process

Service process

- simple ARQ with perfect error detection



- AWGN channel, error-free, instantaneous 1-bit feedback.
- τ: number of frames after which ack is sent

How should one choose n to minimize the delay-violation probability for a given information packet arrival rate λ?

Steady-state delay-violation probability

- D_m: waiting time + service time of mth packet
- Probability that delay exceeds d_0 at steady state

$$P_{\mathrm{dv}}(d_0) = \limsup_{m\to\infty} \mathbb{P}[D_m \geq d_0]$$

[7] R. Devassy et al., "Delay and peak-age violation probability in short-packet transmission", in Proc. IEEE Int. Symp. Inf. Theory (ISIT) (June 2018)

Steady-state delay-violation probability

- D_m: waiting time + service time of mth packet
- Probability that delay exceeds d_0 at steady state

$$P_{\mathrm{dv}}(d_0) = \limsup_{m\to\infty} \mathbb{P}[D_m \geq d_0]$$

Theorem[8]

For every coding scheme satisfying $\lambda n \mathbb{E}[\tau] < 1$, the probability generating function $G_D(s)$ of D is

$$G_D(s) = F(s, \lambda, \mathbb{E}[\tau], G_\tau(s))$$

[7] R. Devassy et al., "Delay and peak-age violation probability in short-packet transmission", in Proc. IEEE Int. Symp. Inf. Theory (ISIT) (June 2018)

Steady-state delay-violation probability

- D_m: waiting time + service time of mth packet
- Probability that delay exceeds d_0 at steady state

$$P_{\mathrm{dv}}(d_0) = \limsup_{m\to\infty} \mathbb{P}[D_m \geq d_0]$$

Theorem[8]

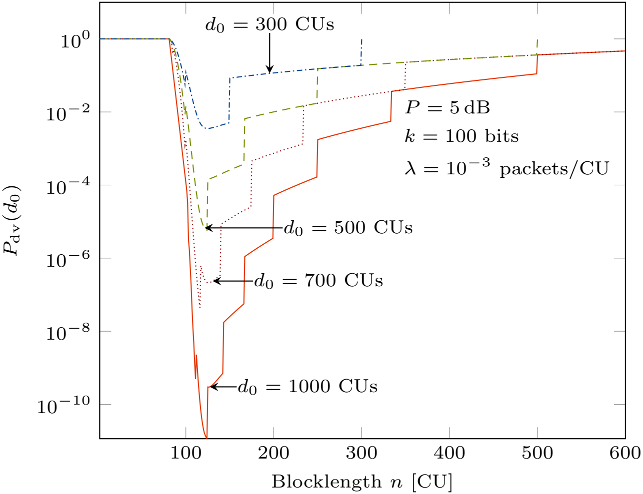
For every coding scheme satisfying $\lambda n \mathbb{E}[\tau] < 1$, the probability generating function $G_D(s)$ of D is

$$G_D(s) = F(s, \lambda, \mathbb{E}[\tau], G_\tau(s))$$

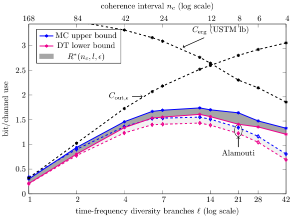
We can use FBL-IT to characterize $G_\tau(s)$ and $\mathbb{E}[\tau]$

[7] R. Devassy et al., "Delay and peak-age violation probability in short-packet transmission", in Proc. IEEE Int. Symp. Inf. Theory (ISIT) (June 2018)

Delay-violation probability vs blocklength (ARQ)



Conclusions



Finite-blocklength inf. theory

- ✓ Elegant theory
- ✓ Tight bounds for short-packet transmissions (including queues)
- ✓ Many engineering insights for the design LP-WAN, 5G, and beyond

Additional material: gdurisi.github.io/tags/#fbl-tutorial

Outline

Efficient Short Channel Codes

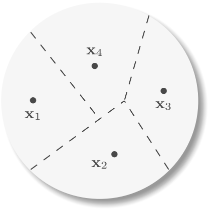
Classical

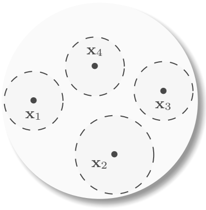
- Algebraic codes (BCH, Reed-Solomon, etc.)
- (Tail-biting) convolutional codes

Modern

- Turbo codes (parallel concatenation)
- Low-density parity-check (LDPC) codes, binary and non-binary
- Polar codes

Decoder Types
Complete vs. Incomplete[8]





Complete:

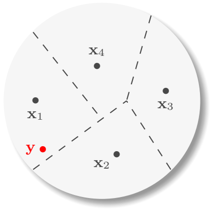
- maximum-likelihood
- ordered statistics
- successive cancellation etc.

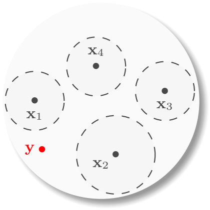
Incomplete:

- bounded distance*
- belief propagation* etc.

[8] G Forney, "Exponential error bounds for erasure, list, and decision feedback schemes", IEEE Trans. Inf. Theory (1968)

Decoder Types
Complete vs. Incomplete[8]





Complete:

- maximum-likelihood
- ordered statistics
- successive cancellation etc.
- all errors are undetected

Incomplete:

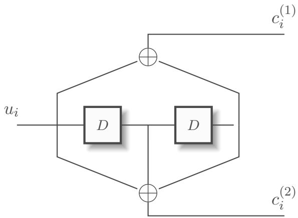
- bounded distance*
- belief propagation* etc.
- error detection capability

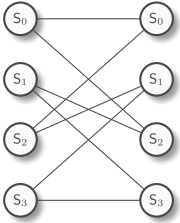
[8] G Forney, "Exponential error bounds for erasure, list, and decision feedback schemes", IEEE Trans. Inf. Theory (1968)

Outline

- Motivations
- Finite-blocklength performance bounds
- Applications
- Efficient Short Channel Codes
  - Efficient Short Classical Codes: Tail-Biting Convolutional Codes
  - Efficient Short Modern Codes: Turbo Codes
  - Efficient Short Modern Codes: Binary Low-Density Parity-Check Codes
  - Efficient Short Modern Codes: Polar Codes
  - Two Case Studies
- Higher-Order Modulation

Convolutional Codes
Definitions by Example (Binary-Input Only)

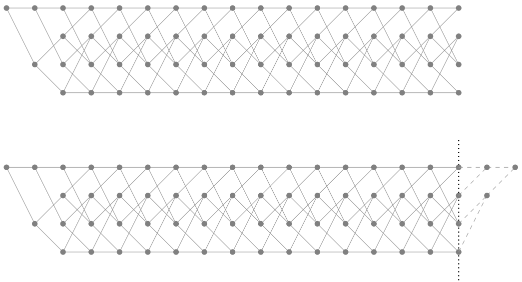




- $k_0 = 1$ inputs and $n_0 = 2$ outputs per clock
- Nominal rate $R_0 = k_0/n_0 = 1/2$
- Memory $m = 2$
- 2^m states per section
- $2^{k_0} = 2$ edges leaving each state

Trellises
Termination Strategies for Convolutional Codes

Convolutional codes to block codes: Run the encoder for k/k_0 clocks, then stop

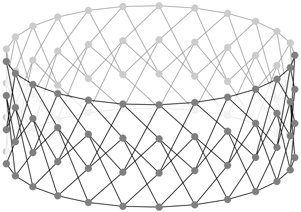


Truncation: Block error probability rises to the last bits

Zero-tail: Improved block error probability BUT rate loss

$$R = \frac{k}{k+m} R_0$$

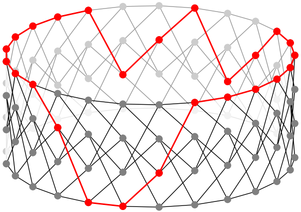
Trellises
Termination Strategies for Convolutional Codes



Tail-biting:

- Force initial = final state

Trellises
Termination Strategies for Convolutional Codes

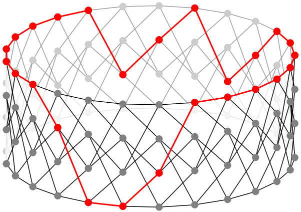


Tail-biting:

- Force initial = final state
- Codewords ≡ circular paths

Trellises

Termination Strategies for Convolutional Codes

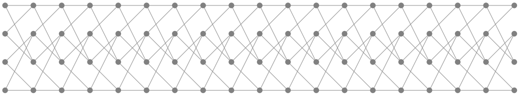


Tail-biting:

- Force initial = final state
- Codewords ≡ circular paths
- No rate loss, but decoding gets more complex...

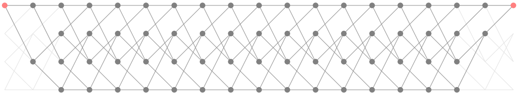
Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

- Unroll the tail-biting trellis



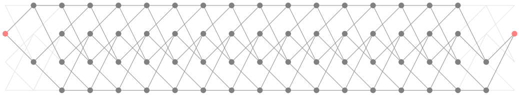
Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

- Unroll the tail-biting trellis
- Run 2^m instances of the Viterbi algorithm, one per initial/final state hypothesis



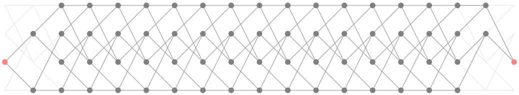
Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

- Unroll the tail-biting trellis
- Run 2^m instances of the Viterbi algorithm, one per initial/final state hypothesis



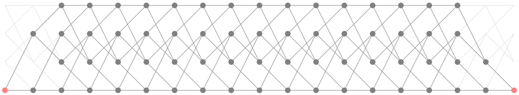
Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

- Unroll the tail-biting trellis
- Run 2^m instances of the Viterbi algorithm, one per initial/final state hypothesis



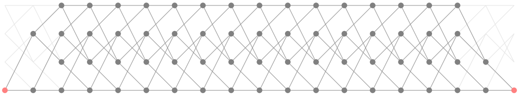
Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

- Unroll the tail-biting trellis
- Run 2^m instances of the Viterbi algorithm, one per initial/final state hypothesis



Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

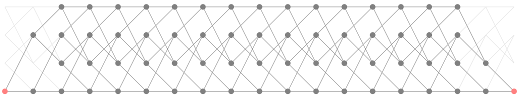
- Unroll the tail-biting trellis
- Run 2^m instances of the Viterbi algorithm, one per initial/final state hypothesis



- Each decoder produces a decision (path): List of 2^m codewords
- Select the most likely codeword in the list

Tail-Biting Convolutional Codes
Maximum-Likelihood Decoding

- Unroll the tail-biting trellis
- Run 2^m instances of the Viterbi algorithm, one per initial/final state hypothesis



- Each decoder produces a decision (path): List of 2^m codewords
- Select the most likely codeword in the list
- Complexity of (almost) 2^m Viterbi decoders, quadratic in 2^m

Tail-Biting Convolutional Codes

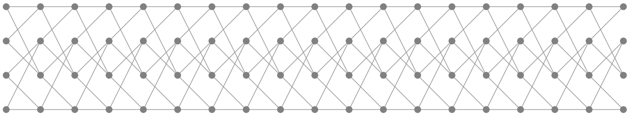
Wrap-Around Viterbi Algorithm (WAVA)[9]

- Runs the Viterbi algorithm successively for more iterations
- Improves the reliability of the decision at each iteration
- Achieves near-optimal performance

[9] R. Y.Shao et al., "Two decoding algorithms for tailbiting codes", IEEE Trans. Commun. (2003)

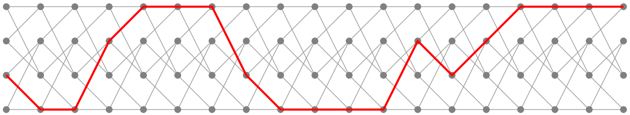
Tail-Biting Convolutional Codes
Wrap-Around Viterbi Algorithm (WAVA)

- Start decoding with equiprobable initial states



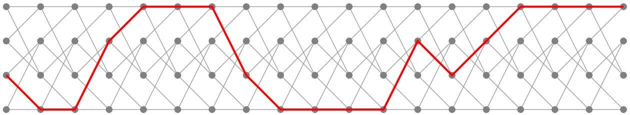
Tail-Biting Convolutional Codes
Wrap-Around Viterbi Algorithm (WAVA)

- Start decoding with equiprobable initial states
- Run a first Viterbi algorithm iteration, and output the most likely path $\mathcal{P}$



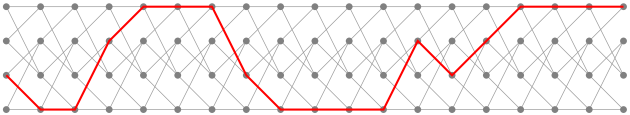
Tail-Biting Convolutional Codes
Wrap-Around Viterbi Algorithm (WAVA)

- Start decoding with equiprobable initial states
- Run a first Viterbi algorithm iteration, and output the most likely path $\mathcal{P}$
- Is $\mathcal{P}$ a tail-biting path?



Tail-Biting Convolutional Codes
Wrap-Around Viterbi Algorithm (WAVA)

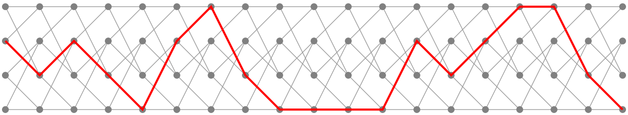
- Start decoding with equiprobable initial states
- Run a first Viterbi algorithm iteration, and output the most likely path $\mathcal{P}$
- Is $\mathcal{P}$ a tail-biting path?
  - YES: stop
  - NO: replace the initial state metrics with the computed final state metrics, and perform another Viterbi algorithm iteration



Tail-Biting Convolutional Codes

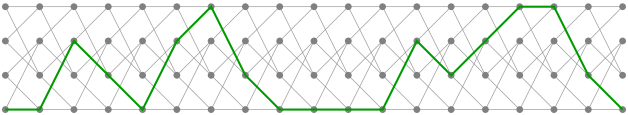
Wrap-Around Viterbi Algorithm (WAVA)

- Start decoding with equiprobable initial states
- Run a first Viterbi algorithm iteration, and output the most likely path $\mathcal{P}$
- Is $\mathcal{P}$ a tail-biting path?
  - YES: stop
  - NO: replace the initial state metrics with the computed final state metrics, and perform another Viterbi algorithm iteration



Tail-Biting Convolutional Codes
Wrap-Around Viterbi Algorithm (WAVA)

- Start decoding with equiprobable initial states
- Run a first Viterbi algorithm iteration, and output the most likely path $\mathcal{P}$
- Is $\mathcal{P}$ a tail-biting path?
  - YES: stop
  - NO: replace the initial state metrics with the computed final state metrics, and perform another Viterbi algorithm iteration
- A maximum number of iterations is allowed (e.g., 4)



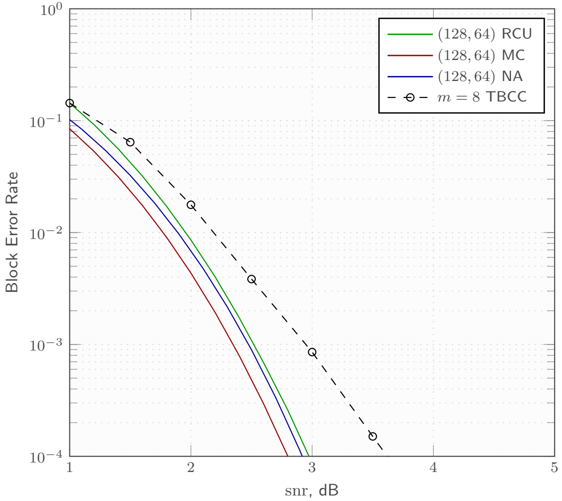
Tail-Biting Convolutional Codes

Examples of Good (Time-Invariant) Tail-Biting Codes[10][11]

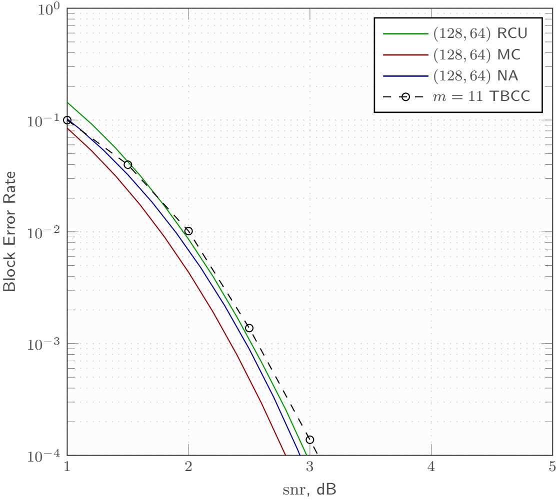
Generators (octal)	m	(n, k)	Minimum Distance
$[515, 677]$	8	$(128, 64)$	12
$[5537, 6131]$	11	$(128, 64)$	14
$[75063, 56711]$	14	$(128, 64)$	16
$[515, 677]$	8	$(256, 128)$	12
$[5537, 6131]$	11	$(256, 128)$	14
$[75063, 56711]$	14	$(256, 128)$	16

[10] P. Stahl et al., "Optimal and near-optimal encoders for short and moderate-length tail-biting trellises", IEEE Trans. Inf. Theory (1999)

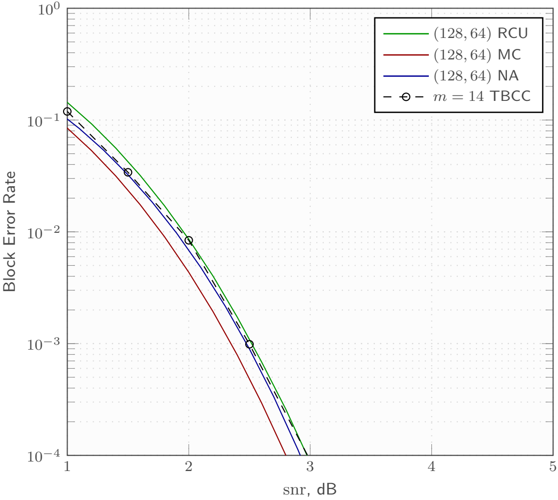
[11] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding*, (John Wiley & Sons, 2015)



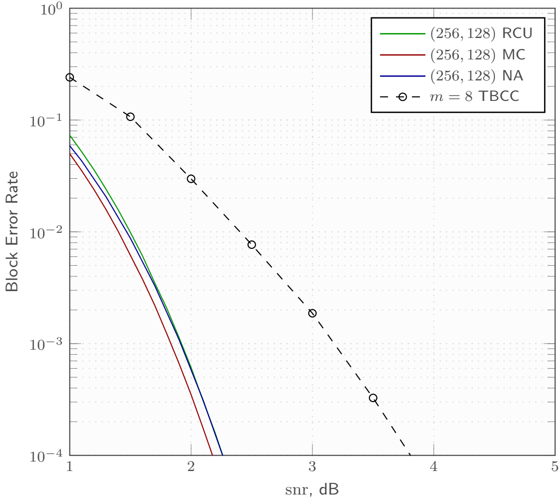
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$m = 8$ TBCC
Block Error Rate
snr, dB
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
1
2
3
4
5

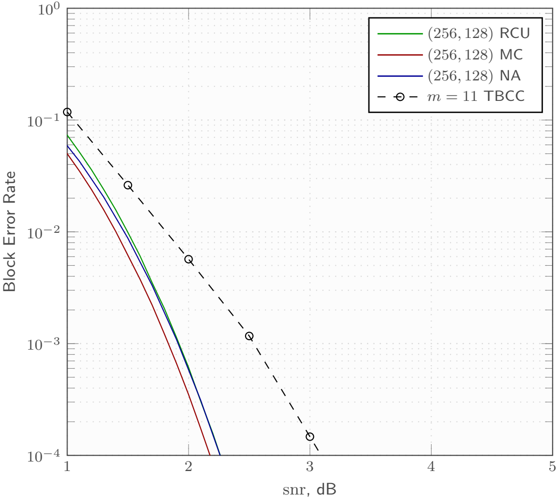


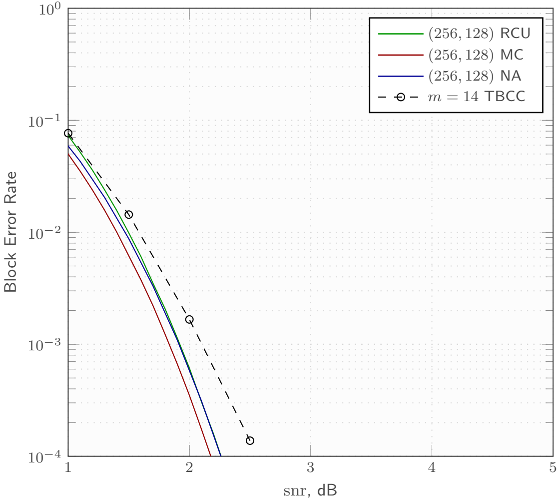
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$m = 11$ TBCC

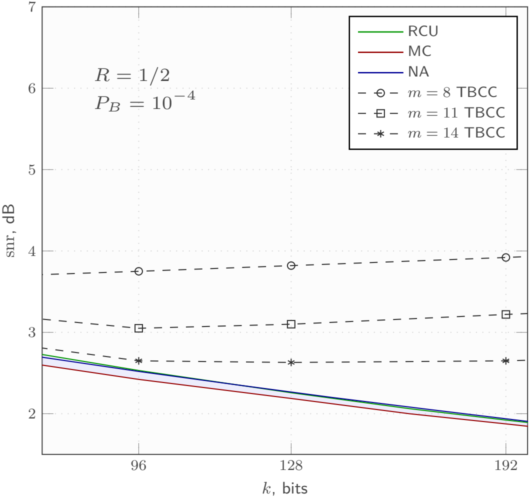


10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$m = 14$ TBCC


(256, 128) RCU
(256, 128) MC
(256, 128) NA
m = 8 TBCC
Block Error Rate
snr, dB
10^0
10^−1
10^−2
10^−3
10^−4
1
2
3
4
5


10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(256, 128)$ RCU
$(256, 128)$ MC
$(256, 128)$ NA
$m = 11$ TBCC


$(256, 128)$ RCU
$(256, 128)$ MC
$(256, 128)$ NA
$m = 14$ TBCC
Block Error Rate
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
1
2
3
4
5
snr, dB



RCU
MC
NA
$m = 8$ TBCC
$m = 11$ TBCC
$m = 14$ TBCC
$R = 1/2$
$P_B = 10^{-4}$
snr, dB
k, bits
7
6
5
4
3
2
96
128
192

Tail-Biting Convolutional Codes
Observations

- Close to optimal at short block lengths ($k \leq 100$ bits)
- Efficient decoding via wrap around Viterbi algorithm (incomplete decoding algorithm)
- For a fixed memory, performance does not improve with the block length
- Shall be employed only at the lowest part of the block length spectrum

Outline

Parallel Concatenated Convolutional Codes

- Turbo codes with 16-states component codes provide the excellent trade-off between minimum distance and decoding threshold[12][13]
- Tail-biting component codes reduce termination overhead[14][15]
- Interleaver design is crucial

FB/FFW Polynomial (Octal)	$(E_b/N_0)^\star$, $R = 1/2$	Notes
27/37	0.56 dB	16-states
23/35	0.62 dB	16-states
15/13	0.70 dB	8-states

[12] C. Berrou et al., "Near Shannon limit error-correcting coding and decoding: turbo-codes", in Proc. ICC (1993)

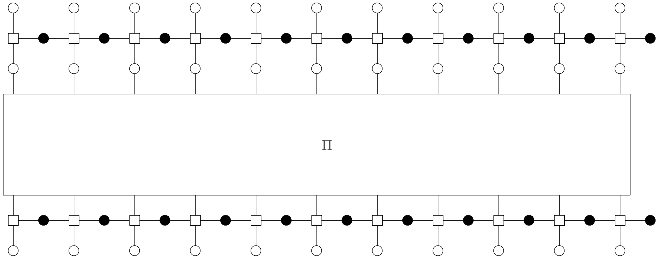
[13] H. El-Gamal and J. Hammons AR., "Analyzing the turbo decoder using the gaussian approximation", IEEE Trans. Inf. Theory (2001)

[14] C. Weiss et al., "Code construction and decoding of parallel concatenated tail-biting codes", IEEE Trans. Inf. Theory (2001)

[15] T. Jerkovits and B. Matuz, "Turbo code design for short blocks", in Proc. 7th Advanced Satellite Mobile Systems Conference (2016)

Parallel Concatenated Convolutional Codes
Factor Graph

Turbo codes factor graphs[16] are characterized by large girth



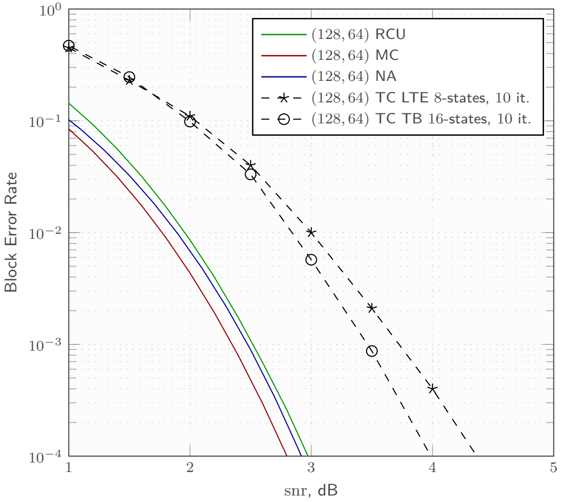
[16] N. Wiberg, "Codes and decoding on general graphs", PhD thesis (Linköping University, 1996)

Parallel Concatenated Convolutional Codes
Interleavers

- The interleaver is the main responsible for large girth and spread (essential for large d_{min})
- Yet, $d_{\text{min}} = \mathcal{O}(\log n)$
- Among the best-known constructions
  - Dithered-Relative-Prime (DRP)[17]
  - Quadratic permutation polynomial (QPP) - LTE [18]

[17] S. Crozier and P. Guinand, "High-performance low-memory interleaver banks for turbo-codes", in Proc. IEEE VTC (2001)

[18] O. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings", IEEE Trans. Inf. Theory (2006)


10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$(128, 64)$ TC LTE 8-states, 10 it.
$(128, 64)$ TC TB 16-states, 10 it.

Turbo Codes
Observations

- Performance within 0.7 dB from RCU bound at moderate error rates
- Decoding can be partially parallelized
- 16-states tail-biting component codes: Good compromise between decoding complexity and performance

Outline

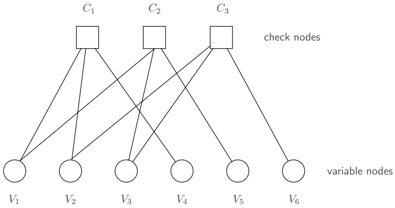
- Motivations
- Finite-blocklength performance bounds
- Applications
- Efficient Short Channel Codes
  - Efficient Short Classical Codes: Tail-Biting Convolutional Codes
  - Efficient Short Modern Codes: Turbo Codes
  - Efficient Short Modern Codes: Binary Low-Density Parity-Check Codes
  - Efficient Short Modern Codes: Polar Codes
  - Two Case Studies
- Higher-Order Modulation

Low-Density Parity-Check Codes

Graphical Representation of the Parity-Check Matrix

- Low-density[19] $\mathbf{H}$ matrix imposing a set of $n-k$ constraints
- Graphical representation via Tanner graphs[20]
  - Codeword bits $\equiv$ variable nodes (VNs)
  - Check equations $\equiv$ check nodes (CNs)

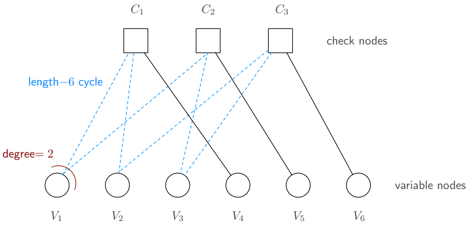
$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$



[19] R. Gallager, *Low-density parity-check codes*, (1963)
[20] M. Tanner, "A recursive approach to low complexity codes", IEEE Trans. Inf. Theory (1981)

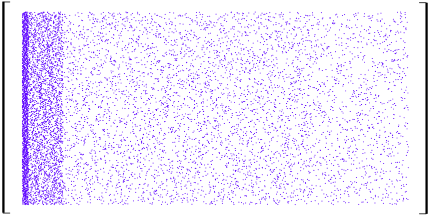
Low-Density Parity-Check Codes
Graphical Representation of the Parity-Check Matrix

- Graphical representation via Tanner graphs (cont'd)



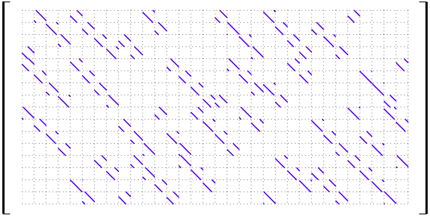
LDPC Codes: Structured Ensembles

$\mathbf{H} =$



Unstructured LDPC Code

$\mathbf{H} =$

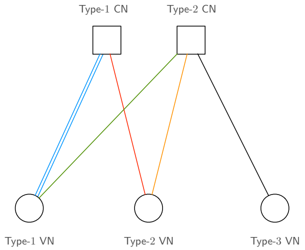


Structured LDPC Code

LDPC Codes: Structured Ensembles

Protograph Codes

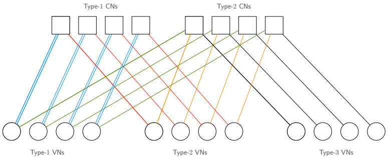
- Protograph: small Tanner graph used as template to build the code graph
- Equivalent representation: base matrix



$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

LDPC Codes: Structured Ensembles
Protograph Codes

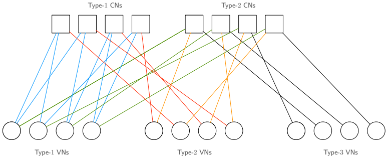
- A protograph can be used to construct a larger Tanner graph by a copy & permute procedure
- The larger Tanner graph defines the code
- First step: Protograph is copied Q times



$$\mathbf{B}' = \begin{pmatrix} 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

LDPC Codes: Structured Ensembles
Protograph Codes

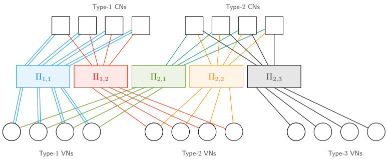
- Second step: Permute edges among the replicas
- Permutations shall avoid parallel edges



$$\mathbf{H} = \begin{pmatrix} 1&1&0&0&0&1&0&0&0&0&0&0\\ 1&0&1&0&0&0&0&1&0&0&0&0\\ 0&0&1&1&0&0&1&0&0&0&0&0\\ 0&1&0&1&1&0&0&0&0&0&0&0\\ 1&0&0&0&1&0&0&0&0&1&0&0\\ 0&0&1&0&0&0&0&1&0&0&1&0\\ 0&1&0&0&0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&0&0&0&0&1 \end{pmatrix}$$

LDPC Codes: Structured Ensembles
Protograph Codes

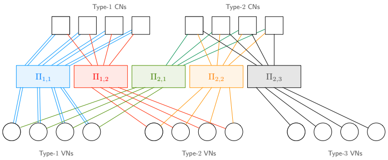
- Second step: Permute edges among the replicas
- Permutations shall avoid parallel edges



$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

LDPC Codes: Structured Ensembles
Protograph Codes

- Second step: Permute edges among the replicas
- Permutations shall avoid parallel edges

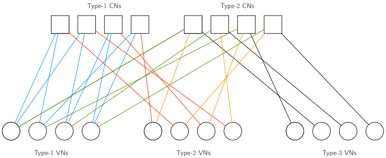


$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- A protograph defines structured LDPC code ensemble: The iterative decoding threshold and distance properties follow from the protograph

LDPC Codes: Structured Ensembles
Protograph Codes

- Depending on code length, the expansion can be done in more steps
- In each step, girth optimization techniques[21] are used
- The final expansion is usually performed by means of circulant permutation matrices (quasi-cyclic code)[22]



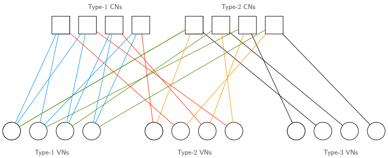
$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

[21] X.-Y. Hu et al., "Regular and irregular progressive edge-growth Tanner graphs", IEEE Trans. Inf. Theory (2005)

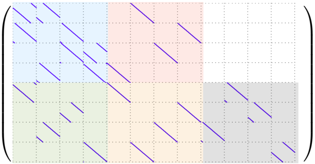
[22] W. Ryan and S. Lin, *Channel codes – Classical and modern*, (Cambridge Univ. Press, 2009)

LDPC Codes: Structured Ensembles
Protograph Codes

- Depending on code length, the expansion can be done in more steps
- In each step, girth optimization techniques[21] are used
- The final expansion is usually performed by means of circulant permutation matrices (quasi-cyclic code)[22]



$\mathbf{H} =$

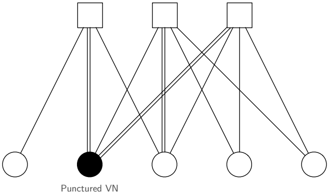


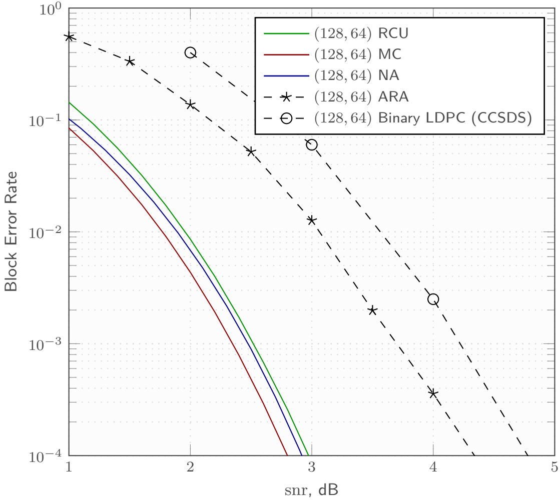
[21] X.-Y. Hu et al., "Regular and irregular progressive edge-growth Tanner graphs", IEEE Trans. Inf. Theory (2005)

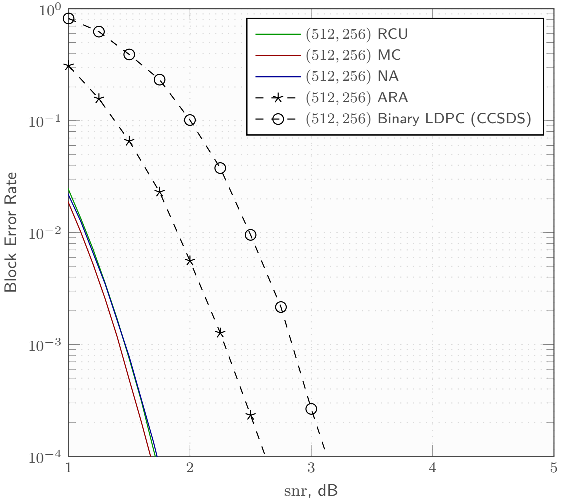
[22] W. Ryan and S. Lin, *Channel codes – Classical and modern*, (Cambridge Univ. Press, 2009)

LDPC Codes: Structured Ensembles
Protograph Codes

- Punctured (state) and degree-1 variable nodes are allowed
- Near-capacity thresholds can be achieved with lower average degrees than unstructured LDPC codes $\rightarrow$ larger girth
- Example: Accumulate-Repeat-3-Accumlate (AR3A), $R = 1/2$, $\mathrm{snr}^{\star} = 0.475$ dB, only 0.3 dB from Shannon limit




$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$(128, 64)$ ARA
$(128, 64)$ Binary LDPC (CCSDS)
Block Error Rate
snr, dB
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
1
2
3
4
5



(512, 256) RCU
(512, 256) MC
(512, 256) NA
(512, 256) ARA
(512, 256) Binary LDPC (CCSDS)
Block Error Rate
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
1
2
3
4
5
snr, dB

Protograph Ensembles: Raptor-like

- Serial concatenation of a high-rate protograph-based outer LDPC code, and a protograph-based LT code[23]

$$\mathbf{B} = \left(\begin{array}{c|c} \mathbf{B}_{\mathsf{o}} & \mathbf{0} \\ \hline \multicolumn{2}{c}{\mathbf{B}_{\mathsf{LT}}} \end{array} \right)$$

[23] T.-Y. Chen et al., "Protograph-Based Raptor-Like LDPC Codes", ArXiv (2014)

Protograph Ensembles: Raptor-like

- Serial concatenation of a high-rate protograph-based outer LDPC code, and a protograph-based LT code[23]

$$\mathbf{B} = \left(\begin{array}{c|c} \mathbf{B}_{\mathrm{o}} & \mathbf{0} \\ \hline \multicolumn{2}{c}{\mathbf{B}_{\mathsf{LT}}} \end{array} \right)$$

- Although the construction targets short block lengths, the outer code parity-check matrix density prevents from obtaining large girths at very short block lengths

[23] T.-Y. Chen et al., "Protograph-Based Raptor-Like LDPC Codes", ArXiv (2014)

Protograph Ensembles: Raptor-like

Large flexibility of rates, with thresholds within 0.5 dB from the Shannon limit

$$\mathbf{B} = \begin{pmatrix}
2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

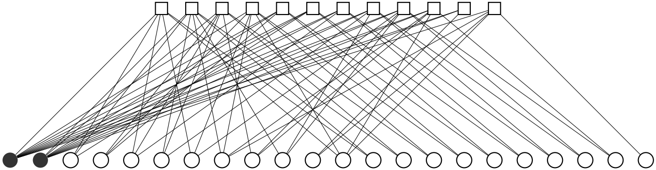
Protograph Ensembles: Raptor-like

Large flexibility of rates, with thresholds within 0.5 dB from the Shannon limit

R	$\mathrm{snr}^\star$	Shannon Limit
$6/7$	3.077 dB	2.625 dB
$6/8$	1.956 dB	1.626 dB
$6/9$	1.392 dB	1.059 dB
$6/10$	1.078 dB	0.679 dB
$6/11$	0.798 dB	0.401 dB
$6/12$	0.484 dB	0.187 dB
$6/13$	0.338 dB	0.018 dB
$6/14$	0.144 dB	-0.122 dB
$6/15$	0.072 dB	-0.238 dB
$6/16$	0.030 dB	-0.337 dB
$6/17$	-0.024 dB	-0.422 dB
$6/18$	-0.150 dB	-0.495 dB

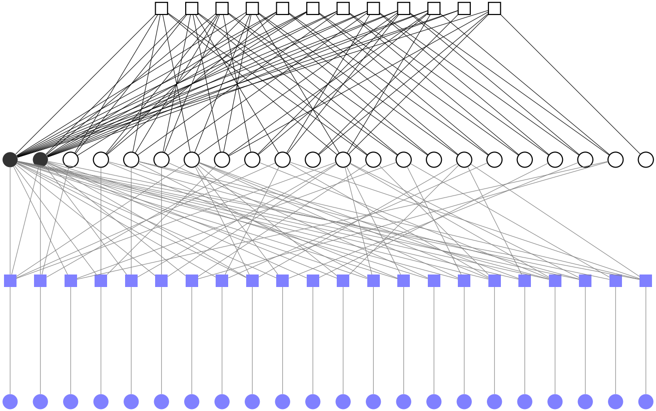
Protograph Ensembles: Raptor-like

- 5G proposal (enhanced mobile broadband)



Protograph Ensembles: Raptor-like

- 5G proposal (enhanced mobile broadband)



LDPC Codes for 5G New Radio (NR)
Introduction

- In 3G and 4G, Turbo codes were used as channel codes.
- For 5G NR enhanced mobile broadband (eMBB), 3GPP opted for LDPC codes[24].
- Requirements for 5G NR:
  1. Support of a wide range of blocklengths and code rates.

[24] *3GPP TS 38.212 V15.0.0: Multiplexing and channel coding*, Dec. 2017

LDPC Codes for 5G New Radio (NR)

Introduction

- In 3G and 4G, Turbo codes were used as channel codes.
- For 5G NR enhanced mobile broadband (eMBB), 3GPP opted for LDPC codes[24].
- Requirements for 5G NR:
    1. Support of a wide range of blocklengths and code rates.
    2. Support for incremental-redundancy hybrid automatic repeat request (ARQ).

[24] *3GPP TS 38.212 V15.0.0: Multiplexing and channel coding*, Dec. 2017

LDPC Codes for 5G New Radio (NR)
Introduction

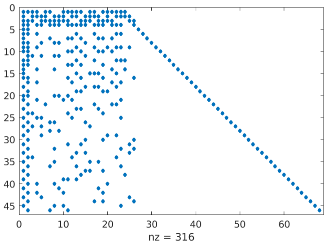
- In 3G and 4G, Turbo codes were used as channel codes.
- For 5G NR enhanced mobile broadband (eMBB), 3GPP opted for LDPC codes[24].
- Requirements for 5G NR:
  1. Support of a wide range of blocklengths and code rates.
  2. Support for incremental-redundancy hybrid automatic repeat request (ARQ).
  3. Hardware-friendly implementation: minimal description complexity, possibility for parallelization.

[24] *3GPP TS 38.212 V15.0.0: Multiplexing and channel coding*, Dec. 2017

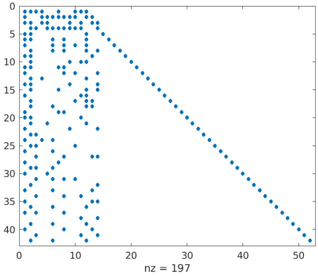
LDPC Codes for 5G New Radio

Base Matrices

Base Matrix 1

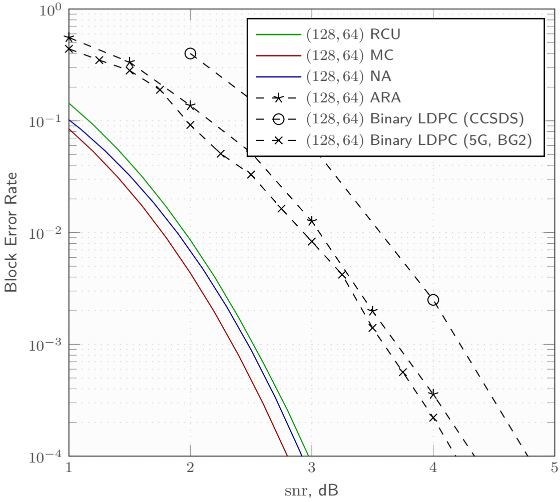


Base Matrix 2

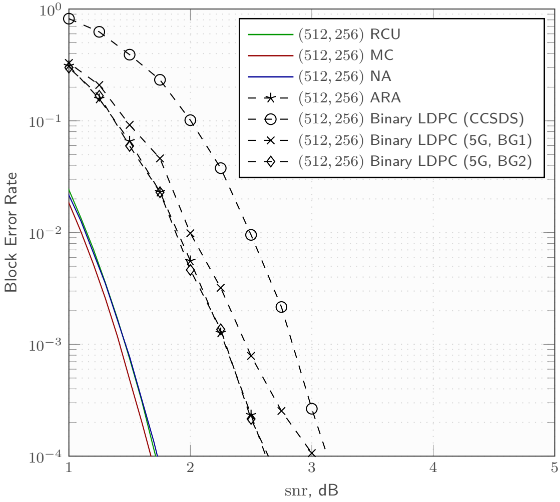


LDPC codes for 5G New Radio
Design Principles

- Introduction of two state, i.e., punctured, variable nodes. Beneficial for lowering the decoding threshold.
- Punctured variable nodes are in the systematic part and have high variable node degrees.
- Connected to at least one degree 1 variable node in the extension part.



10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$(128, 64)$ ARA
$(128, 64)$ Binary LDPC (CCSDS)
$(128, 64)$ Binary LDPC (5G, BG2)



10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
(512, 256) RCU
(512, 256) MC
(512, 256) NA
(512, 256) ARA
(512, 256) Binary LDPC (CCSDS)
(512, 256) Binary LDPC (5G, BG1)
(512, 256) Binary LDPC (5G, BG2)

Binary Low-Density Parity-Check Codes
Observations

- Performance within 1.2 dB from RCU bound at short block lengths
- Protograph construction fundamental to achieve good performance with practical decoders
- Depending on the code design, strong error detection capability

Outline

Polar Codes
Introduction

- Class of provably capacity achieving codes over memoryless binary input output symmetric channels under low-complexity (successive cancellation) decoding[25]

[25] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Trans. Inf. Theory (2009)

[26] I. Tal and A. Vardy, "List decoding of polar codes", IEEE Trans. Inf. Theory (2015)

Polar Codes
Introduction

- Class of provably capacity achieving codes over memoryless binary input output symmetric channels under low-complexity (successive cancellation) decoding[25]
- Their performance at short block lengths is disappointing but...

[25] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Trans. Inf. Theory (2009)
[26] I. Tal and A. Vardy, "List decoding of polar codes", IEEE Trans. Inf. Theory (2015)

Polar Codes
Introduction

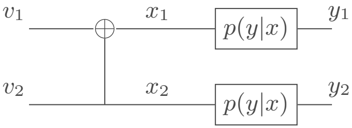
- Class of provably capacity achieving codes over memoryless binary input output symmetric channels under low-complexity (successive cancellation) decoding[25]
- Their performance at short block lengths is disappointing but...

  list decoding with the aid of an outer-high rate code[26] yields one of the best code constructions at short block lengths!

[25] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Trans. Inf. Theory (2009)

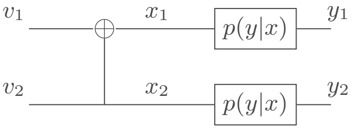
[26] I. Tal and A. Vardy, "List decoding of polar codes", IEEE Trans. Inf. Theory (2015)

Polar Codes



$$\mathbf{x} = \mathbf{v}\mathbf{G}_2 \qquad\qquad \mathbf{G}_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Polar Codes



$$\mathbf{x} = \mathbf{v}\mathbf{G}_2 \qquad \mathbf{G}_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Polar Codes

Denote $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Then

$$\mathbf{x} = \mathbf{u}\mathbf{G}_n$$

with $\mathbf{G}_n$ being a $n \times n$ matrix with structure

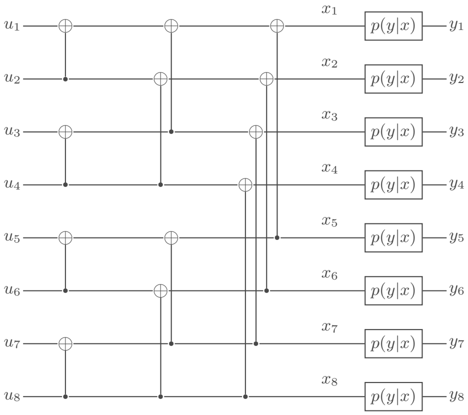
$$\mathbf{G}_n = \mathbf{G}_2 \otimes \mathbf{G}_2 \otimes \ldots \otimes \mathbf{G}_2$$

Polar Codes
Example

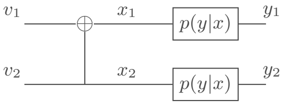
With $n = 8$, $\mathbf{G}_8 = \mathbf{G}_2 \otimes \mathbf{G}_2 \otimes \mathbf{G}_2$

$$\mathbf{G}_8 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

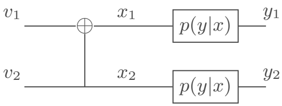
Polar Codes
Example



Polar Codes
Successive Cancellation Decoding

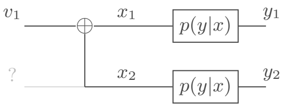


Polar Codes
Successive Cancellation Decoding



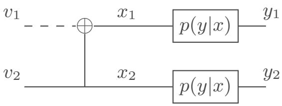
$$p(\mathbf{y}|\mathbf{v}) = p(y_1|v_1 + v_2)p(y_2|v_2)$$

Polar Codes
Successive Cancellation Decoding



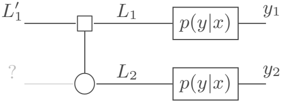
$$p(\mathbf{y}|v_1) = \sum_{v_2} p(\mathbf{y}, v_2|v_1) = \frac{1}{2}\sum_{v_2} p(y_1|v_1 + v_2)p(y_2|v_2)$$

Polar Codes
Successive Cancellation Decoding



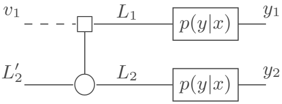
$$p(\mathbf{y}, v_1|v_2) = p(\mathbf{y}|v_1, v_2)p(v_1) = \frac{1}{2}p(y_1|v_1 + v_2)p(y_2|v_2)$$

Polar Codes
Successive Cancellation Decoding



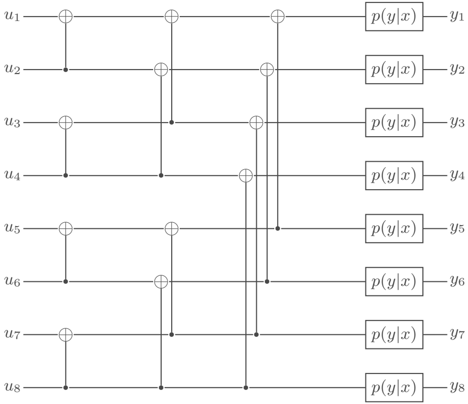
$$L_1' = 2\tanh^{-1}\left(\tanh\left(\frac{L_1}{2}\right)\tanh\left(\frac{L_2}{2}\right)\right) \qquad \text{with} \qquad L_i = \log\frac{p(y_i|0)}{p(y_i|1)}$$

Polar Codes
Successive Cancellation Decoding



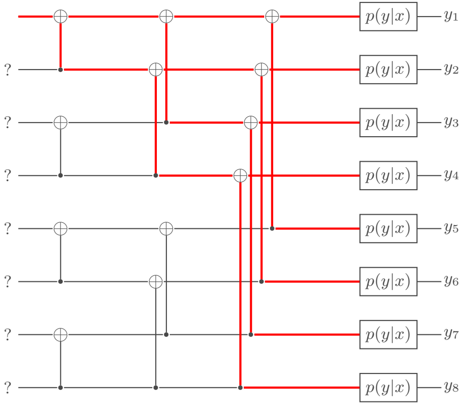
$$L_2' = L_2 + (-1)^{v_1} L_1$$

Polar Codes
Successive Cancellation Decoding



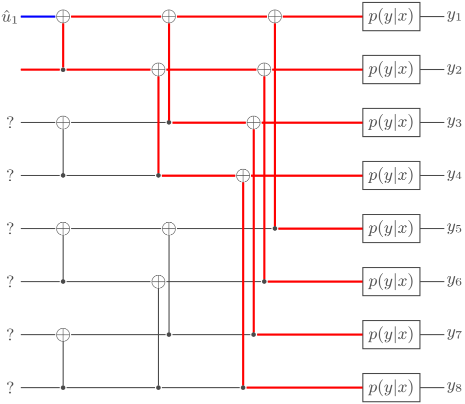
Polar Codes
Successive Cancellation Decoding

$$\hat{u}_1 = \arg\max_{u_1} p(\mathbf{y}|u_1)$$



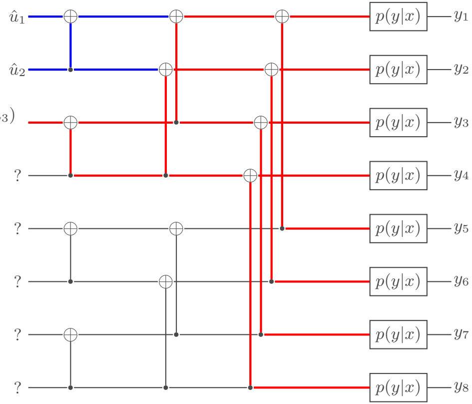
Polar Codes
Successive Cancellation Decoding

$$\hat{u}_2 = \arg\max_{u_2} p(\mathbf{y}, \hat{u}_1 | u_2)$$

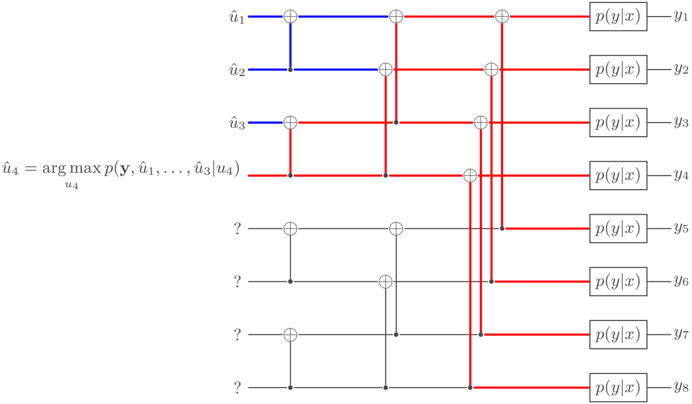


Polar Codes
Successive Cancellation Decoding

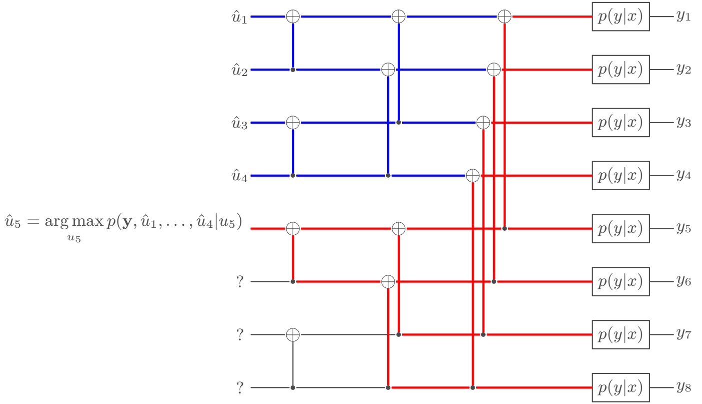
$$\hat{u}_3 = \arg\max_{u_3} p(\mathbf{y}, \hat{u}_1, \hat{u}_2 | u_3)$$



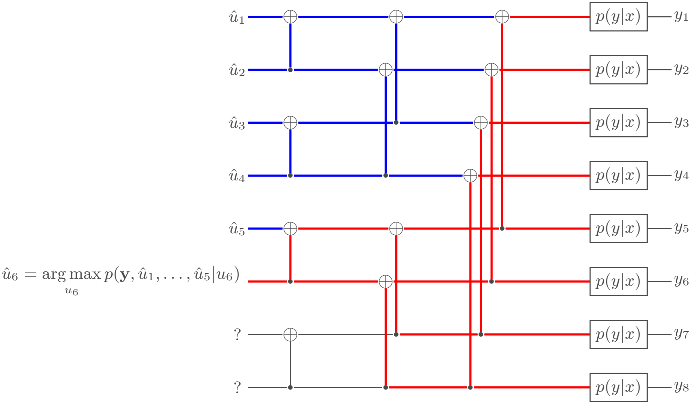
Polar Codes
Successive Cancellation Decoding



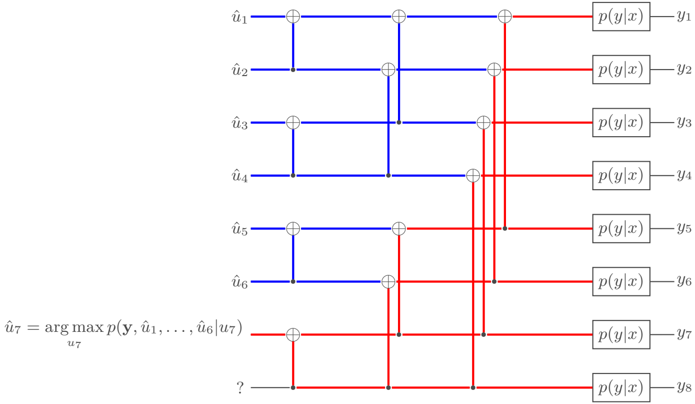
Polar Codes
Successive Cancellation Decoding



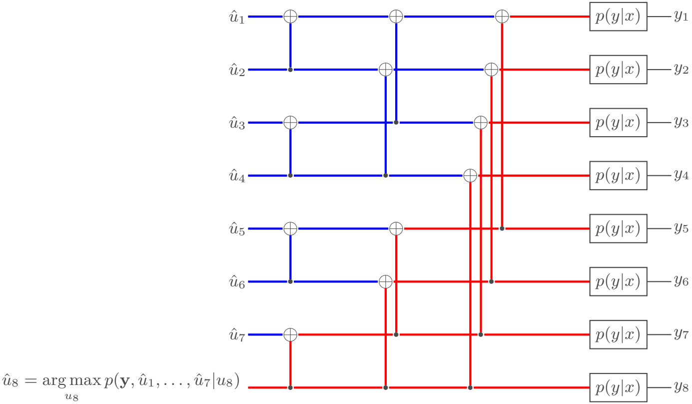
Polar Codes
Successive Cancellation Decoding



Polar Codes
Successive Cancellation Decoding



Polar Codes
Successive Cancellation Decoding



Polar Codes
Code Design

- (n, k) polar code: $\mathcal{A} =$ set of k indexed in $\{1, 2, \ldots, n\}$
- Map the k information bits on u_i, $i \in \mathcal{A}$

[27] N. Stolte, "Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung", PhD thesis (TU Darmstadt, 2002)

[28] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Trans. Inf. Theory (2009)

Polar Codes
Code Design

- (n, k) polar code: $\mathcal{A} =$ set of k indexed in $\{1, 2, \ldots, n\}$
- Map the k information bits on u_i, $i \in \mathcal{A}$
- Set the remaining elements of $\mathbf{u}$ to 0 (frozen bits)

[27] N. Stolte, "Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung", PhD thesis (TU Darmstadt, 2002)

[28] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Trans. Inf. Theory (2009)

Polar Codes
Code Design

- (n, k) polar code: $\mathcal{A} =$ set of k indexed in $\{1, 2, \dots, n\}$
- Map the k information bits on u_i, $i \in \mathcal{A}$
- Set the remaining elements of $\mathbf{u}$ to 0 (frozen bits)
- Selection of the frozen bits: For the target channel, find the least $n-k$ reliable bits in $\mathbf{u}$ under successive cancellation decoding[27][28]

[27] N. Stolte, "Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung", PhD thesis (TU Darmstadt, 2002)

[28] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", IEEE Trans. Inf. Theory (2009)

Polar Codes
Example

- $(8, 4)$ polar code: $\mathcal{A} = \{4, 6, 7, 8\}$

$$\mathbf{G}_8 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Polar Codes
Example

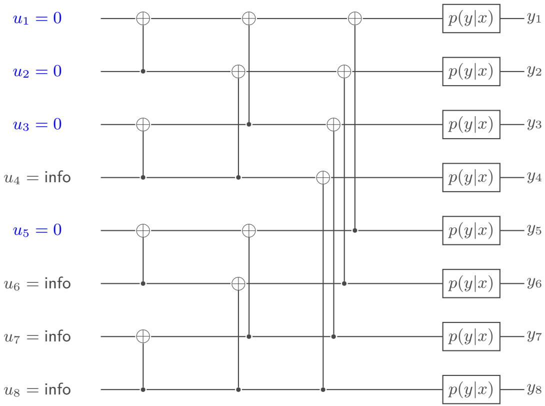
- $(8,4)$ polar code: $\mathcal{A} = \{4,6,7,8\}$

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

- $\mathbf{G}$: generator matrix of the $(8,4)$ polar code

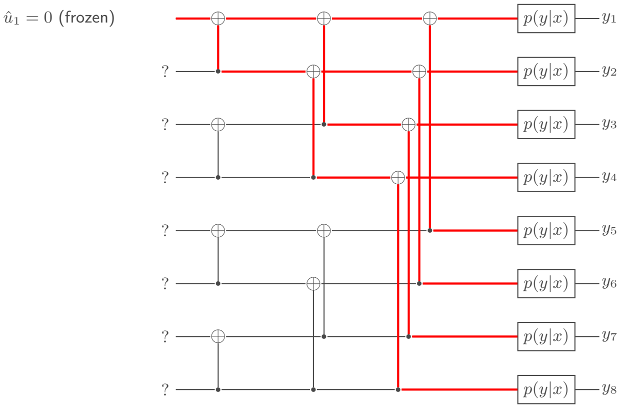
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



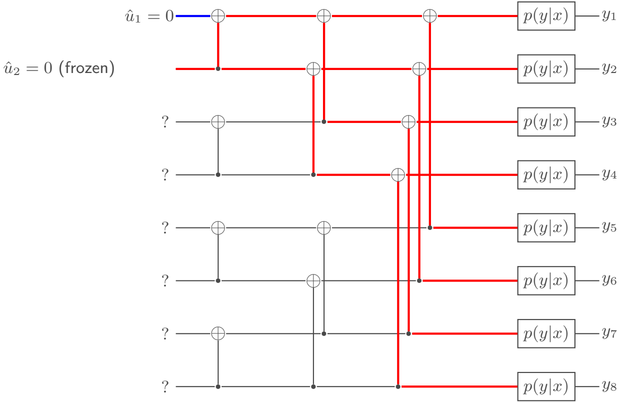
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



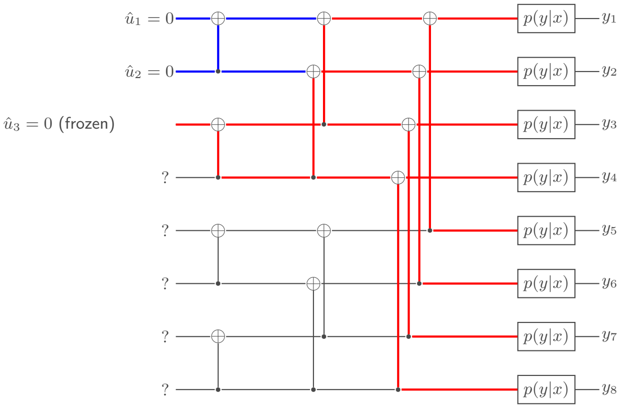
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



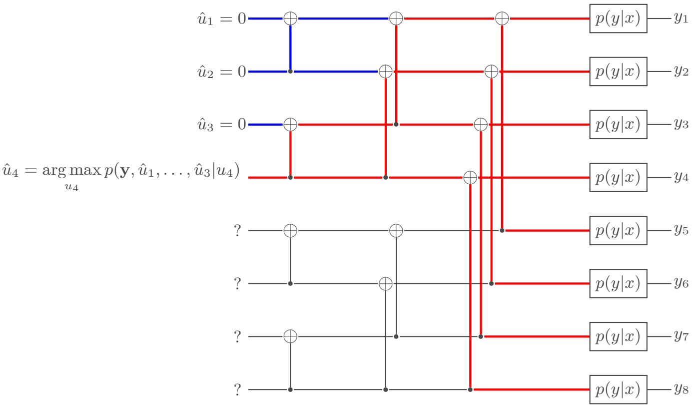
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



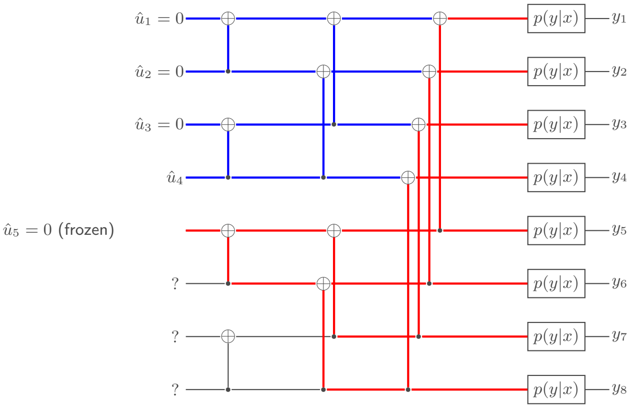
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



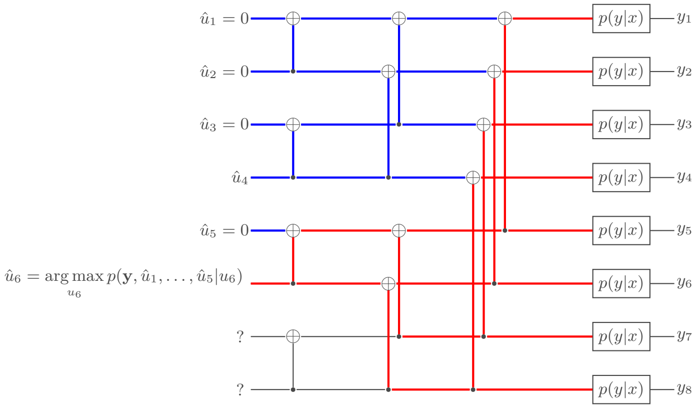
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



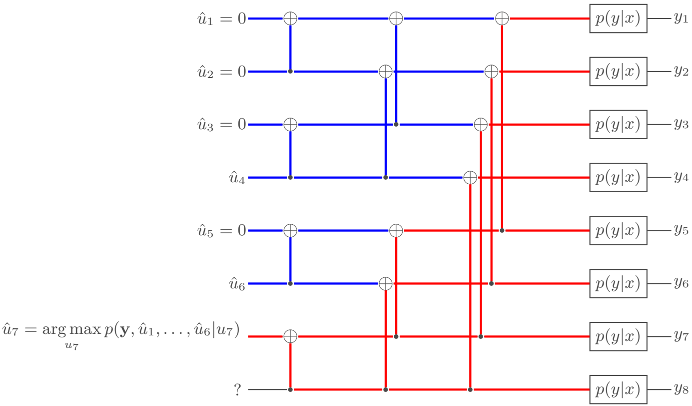
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



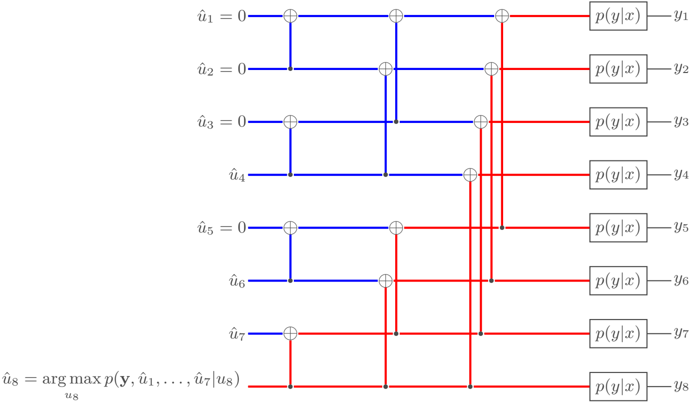
Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)

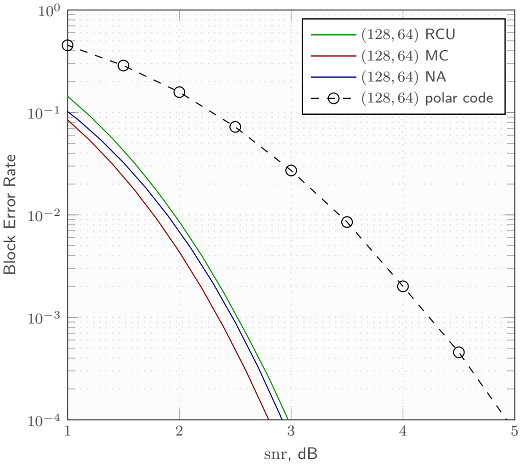


Polar Codes

Example: $u_1 = u_2 = u_3 = u_5 = 0$ (frozen bits)



Polar Codes: Shortcomings



Albeit capacity-achieving (for large n), at moderate-short block lengths polar codes under successive cancellation decoding perform poorly

Polar Codes
List Decoding: Principle

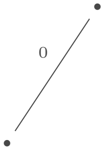
- List decoding: Exploit the serial bit decision process to improve the SC decoder performance

List size $L = 4$

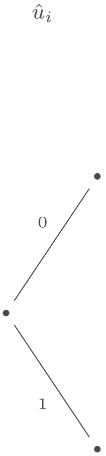
•

$\hat{u}_i$

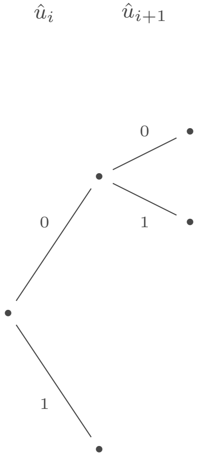
List size $L = 4$



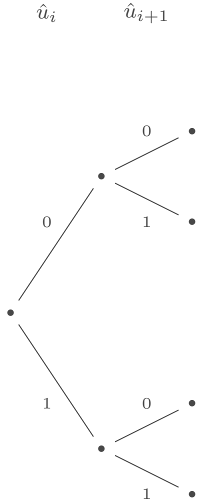
List size $L = 4$



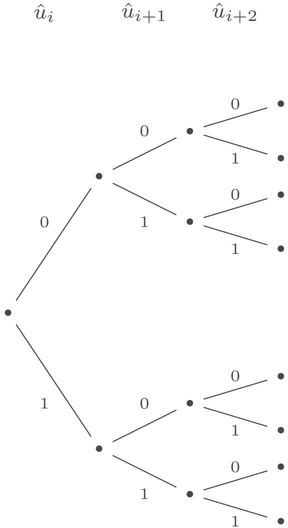
List size $L = 4$



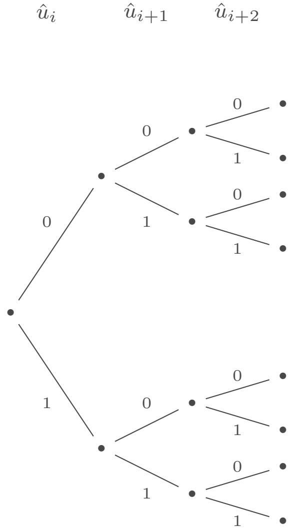
List size $L = 4$



List size $L = 4$

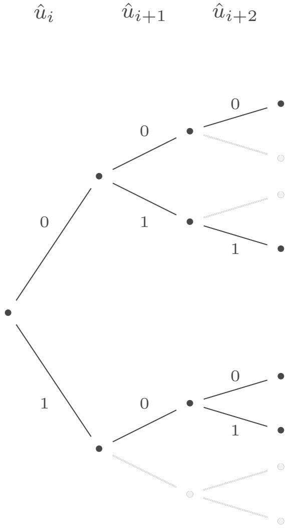


List size $L = 4$



Rank based on

$$p\left(\mathbf{y}|\hat{u}_1, \ldots, \hat{u}_{i+2}\right)$$

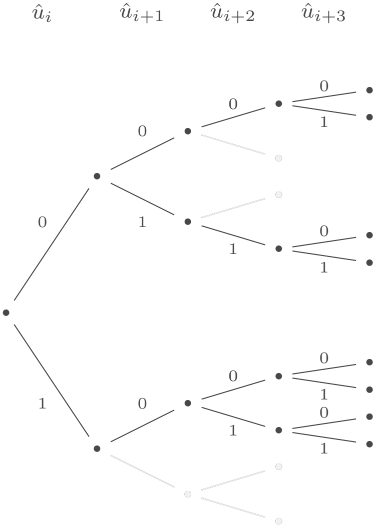


List size $L = 4$

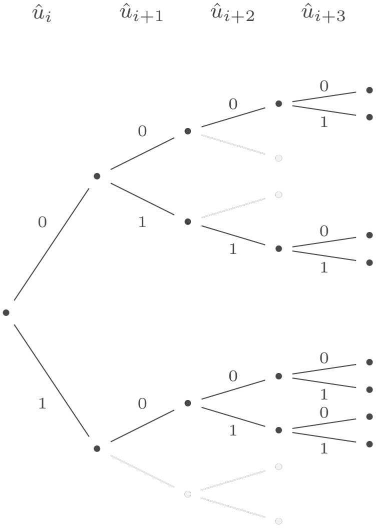
Discard the $L/2$

least likely paths

List size $L = 4$

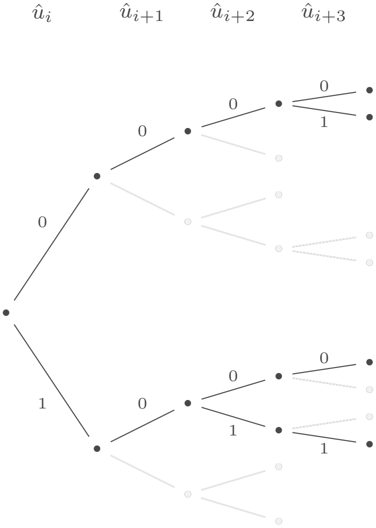


List size $L = 4$



Rank based on

$p\left(\mathbf{y}|\hat{u}_1,\ldots,\hat{u}_{i+3}\right)$



List size $L = 4$

Discard the $L/2$

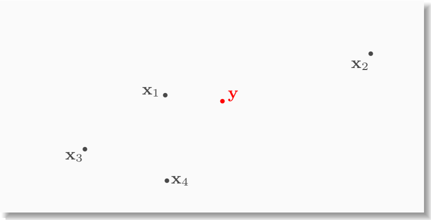
least likely paths

Polar Codes
List Decoding: Principle

- After k steps, L codewords in the list $\mathcal{L}$

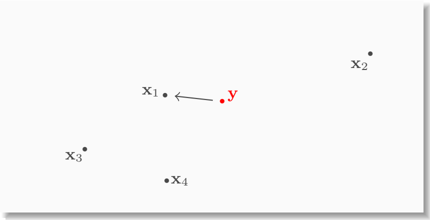
Polar Codes
List Decoding: Principle

- After k steps, L codewords in the list $\mathcal{L}$



Polar Codes
List Decoding: Principle

- After k steps, L codewords in the list $\mathcal{L}$

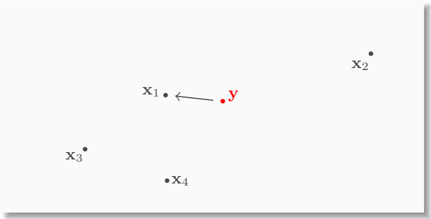


- Pick the codeword in $\mathcal{L}$ maximizing the likelihood

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{L}} p\left(\mathbf{y}|\mathbf{x}\right)$$

Polar Codes
List Decoding: Principle

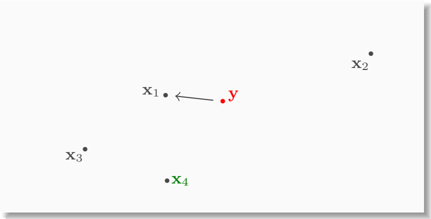
- Two error events:



- The correct codeword $\mathbf{x}$ is not in the list

Polar Codes
List Decoding: Principle

- Two error events:

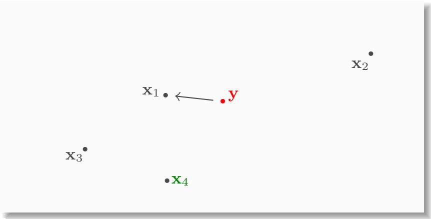


- The correct codeword $\mathbf{x}$ is not in the list
- The correct codeword $\mathbf{x}$ is in the list but $\exists \mathbf{x}' \in \mathcal{L}$ s.t. $p(\mathbf{y}|\mathbf{x}') > p(\mathbf{y}|\mathbf{x})$

Polar Codes

List Decoding: Principle

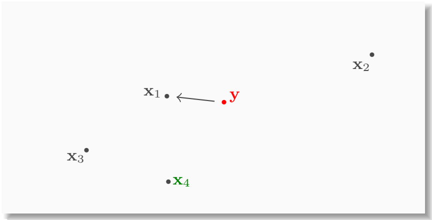
- Two error events:



- The correct codeword $\mathbf{x}$ is not in the list
- The correct codeword $\mathbf{x}$ is in the list but $\exists \mathbf{x}' \in \mathcal{L}$ s.t. $p\left(\mathbf{y}|\mathbf{x}'\right) > p\left(\mathbf{y}|\mathbf{x}\right)$
  The error would take place even with ML decoding...

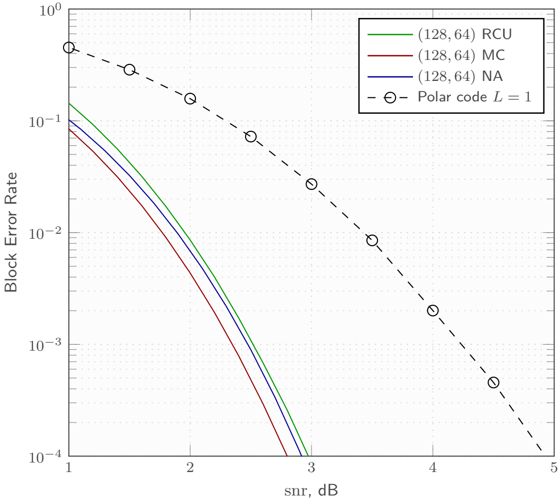
Polar Codes
List Decoding: Principle

- Two error events:

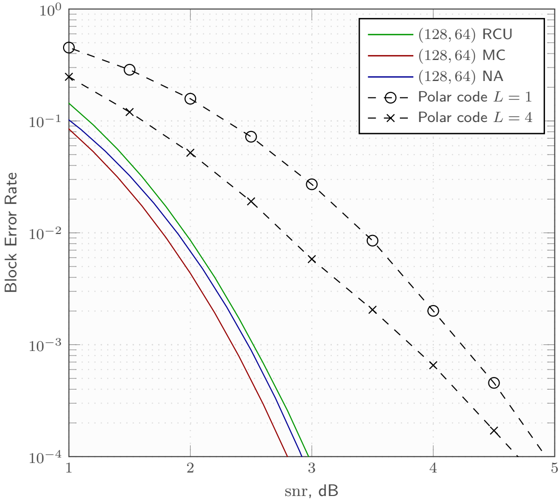


- The correct codeword $\mathbf{x}$ is not in the list
- The correct codeword $\mathbf{x}$ is in the list but $\exists \mathbf{x}' \in \mathcal{L}$ s.t. $p(\mathbf{y}|\mathbf{x}') > p(\mathbf{y}|\mathbf{x})$
  The error would take place even with ML decoding...

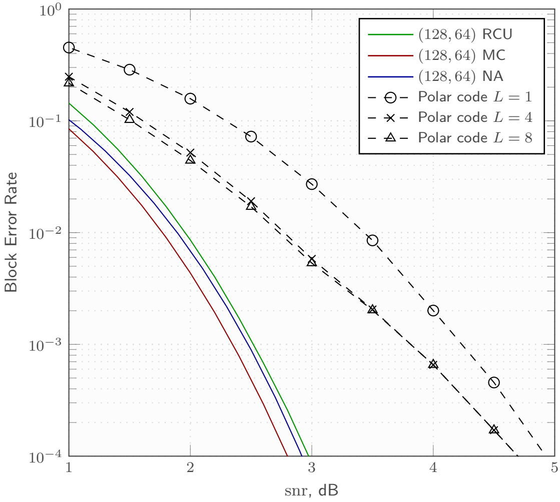
Performance limited by distance spectrum



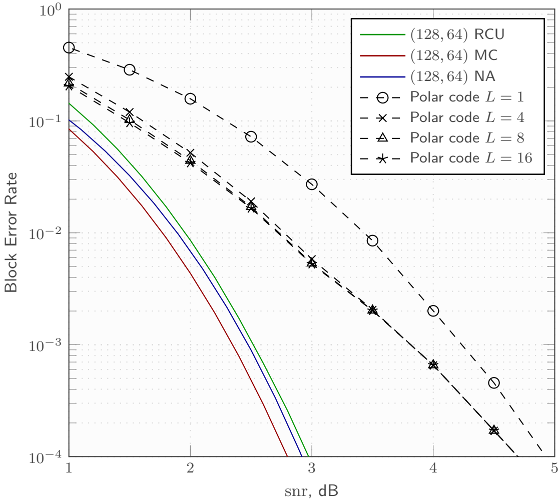
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
Polar code $L = 1$

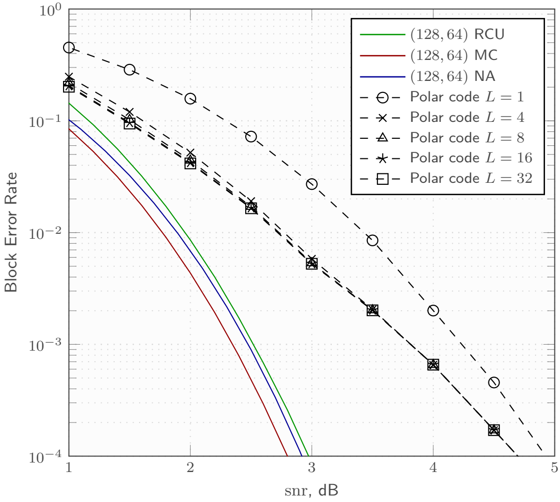


10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
Polar code $L = 1$
Polar code $L = 4$

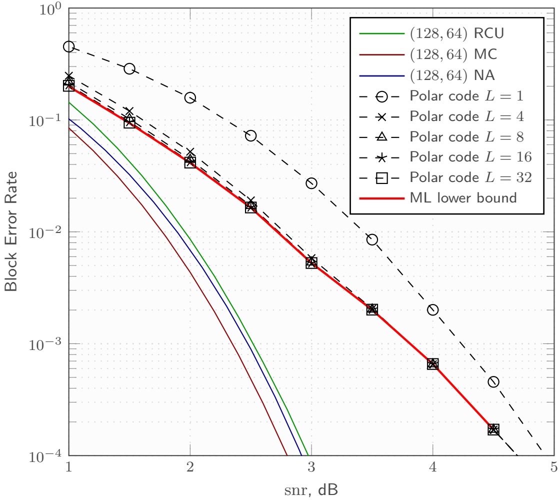


10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
Polar code $L = 1$
Polar code $L = 4$
Polar code $L = 8$


10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
Polar code $L = 1$
Polar code $L = 4$
Polar code $L = 8$
Polar code $L = 16$



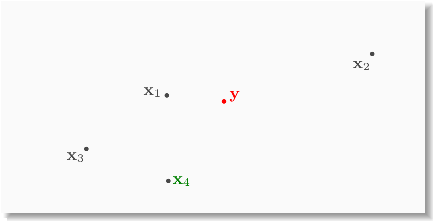
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
Polar code $L = 1$
Polar code $L = 4$
Polar code $L = 8$
Polar code $L = 16$
Polar code $L = 32$



10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
Polar code $L = 1$
Polar code $L = 4$
Polar code $L = 8$
Polar code $L = 16$
Polar code $L = 32$
ML lower bound

Polar Codes
List Decoding: Principle

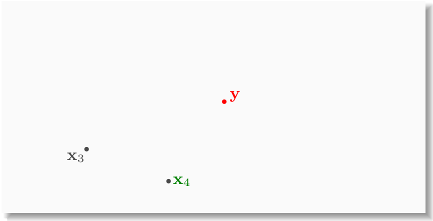
- Concatenation with an outer code to improve distance spectrum



- List decoding (inner code), followed by syndrome check with outer code

Polar Codes
List Decoding: Principle

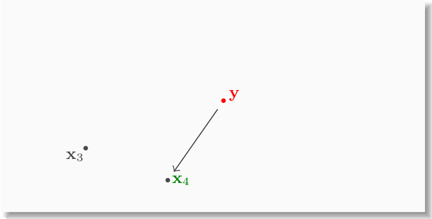
- Concatenation with an outer code to improve distance spectrum



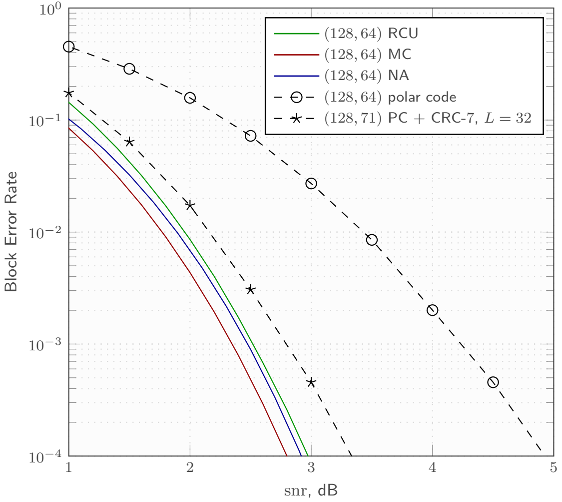
- List decoding (inner code), followed by syndrome check with outer code
- Expurgated list: all codewords not satisfying the check are removed

Polar Codes
List Decoding: Principle

- Concatenation with an outer code to improve distance spectrum



- List decoding (inner code), followed by syndrome check with outer code
- Expurgated list: all codewords not satisfying the check are removed
- Selection within the remaining codewords based on likelihood



10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Block Error Rate
1
2
3
4
5
snr, dB
$(128, 64)$ RCU
$(128, 64)$ MC
$(128, 64)$ NA
$(128, 64)$ polar code
$(128, 71)$ PC + CRC-7, $L = 32$

Polar Codes
Observations

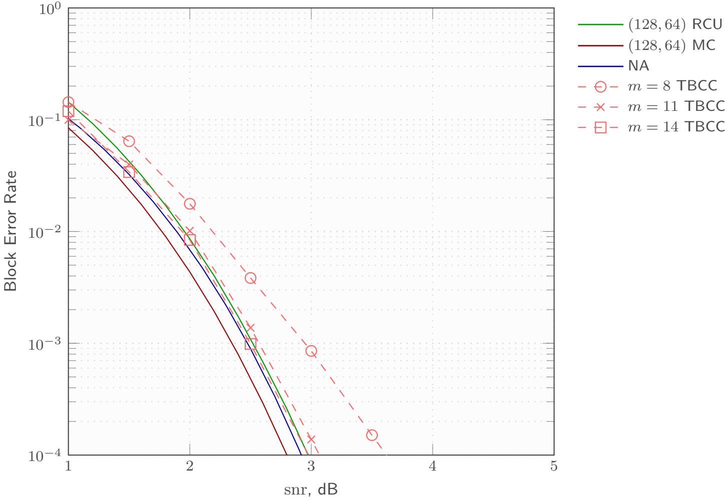
- With successive cancellation + list decoding and the aid of an outer code, consistently close to the normal approximation
- Complexity growing with the list size L
- Large list size: close to maximum-likelihood performance (but large complexity)
- Error floor behavior only partially addressed[29]
- Good trade-off between decoding complexity and performance

[29] G. Ricciutelli et al., "On the error probability of short concatenated polar and cyclic codes with interleaving", arXiv preprint arXiv:1701.07262 (2017)

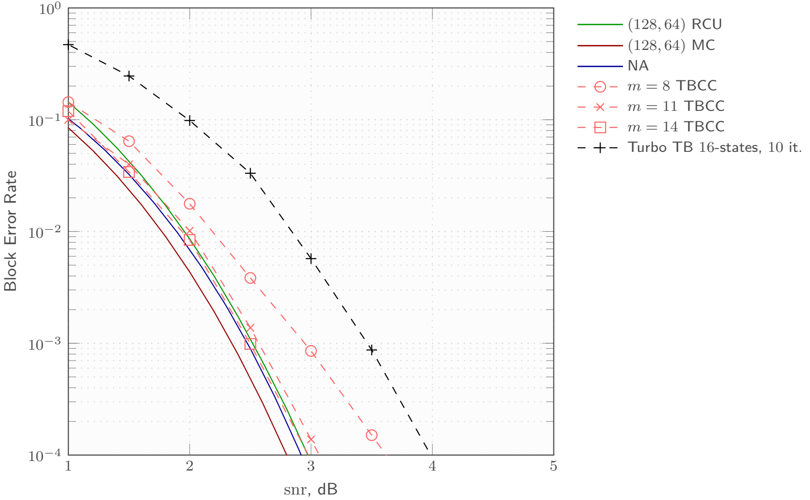
Outline

- Motivations
- Finite-blocklength performance bounds
- Applications
- Efficient Short Channel Codes
  - Efficient Short Classical Codes: Tail-Biting Convolutional Codes
  - Efficient Short Modern Codes: Turbo Codes
  - Efficient Short Modern Codes: Binary Low-Density Parity-Check Codes
  - Efficient Short Modern Codes: Polar Codes
  - Two Case Studies
- Higher-Order Modulation

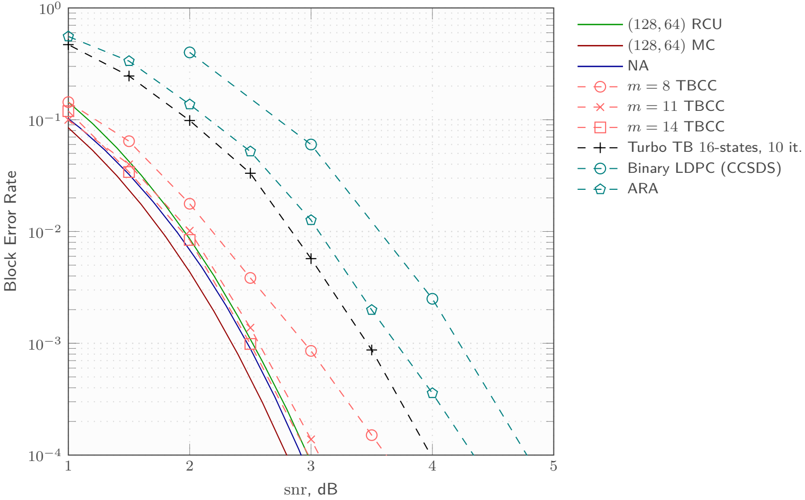
Case 1: $n = 128, k = 64$



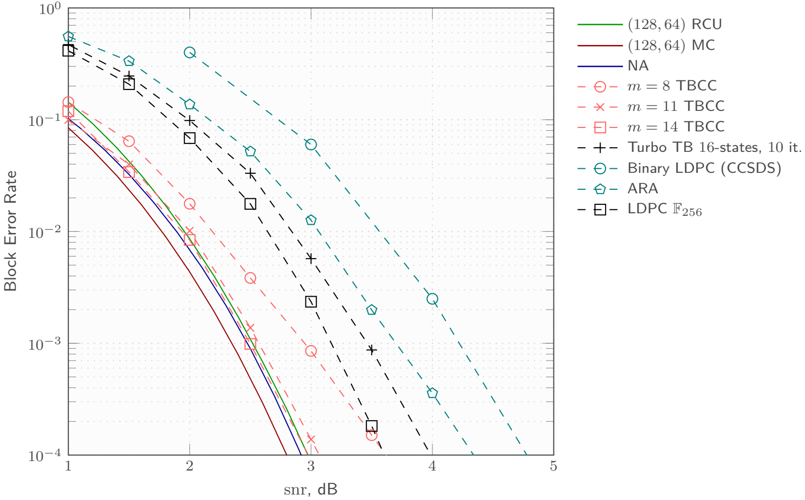
Case 1: $n = 128, k = 64$



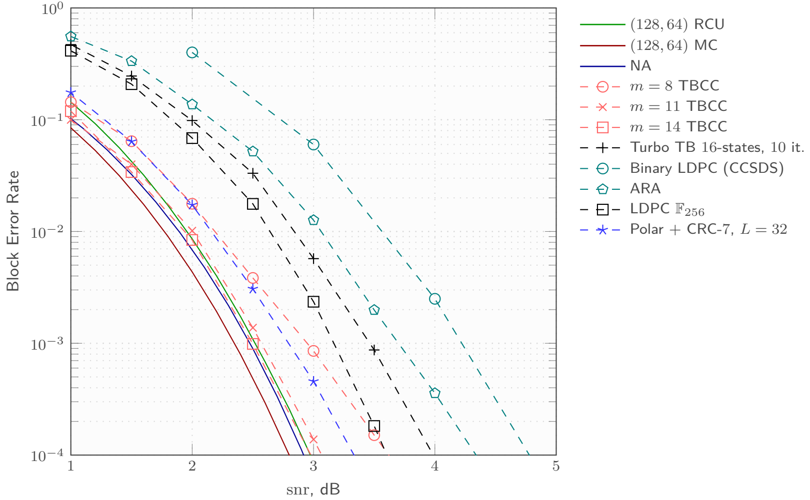
Case 1: $n = 128, k = 64$



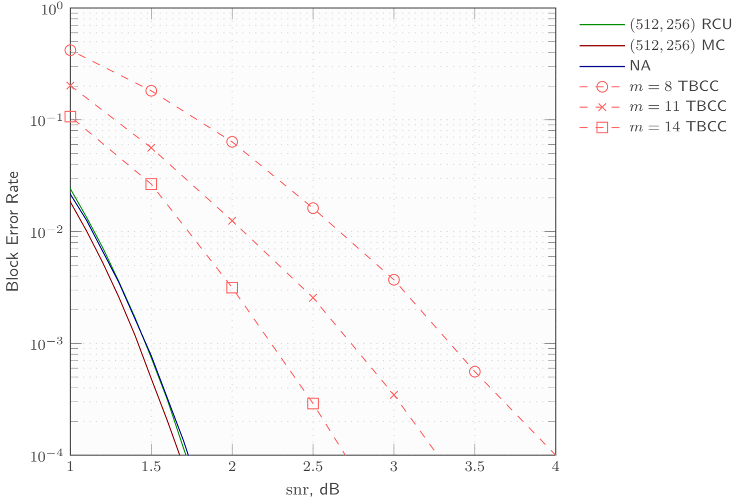
Case 1: $n = 128, k = 64$



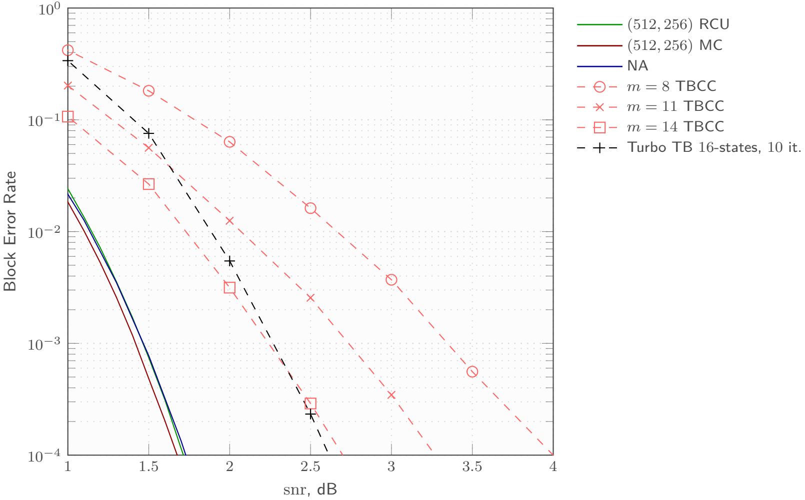
Case 1: $n = 128, k = 64$



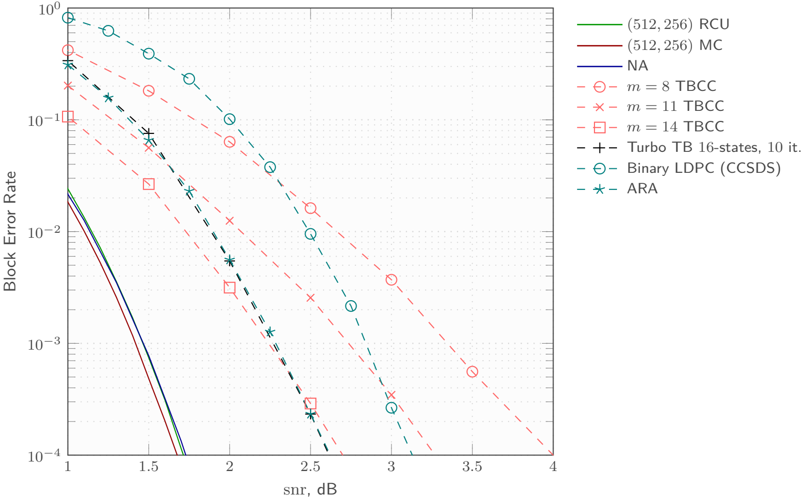
Case 2: $n = 512, k = 256$



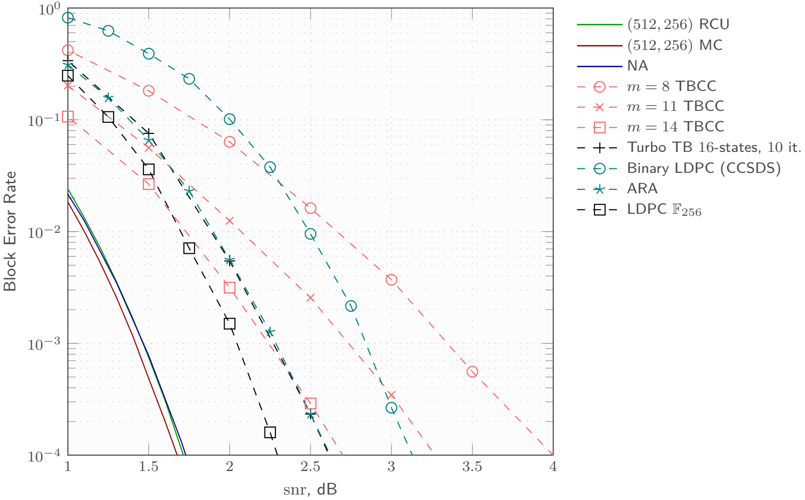
Case 2: $n = 512, k = 256$



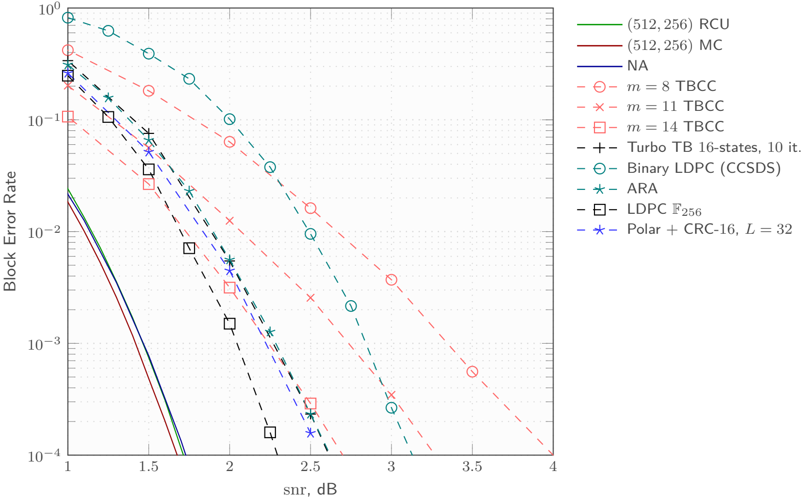
Case 2: $n = 512, k = 256$



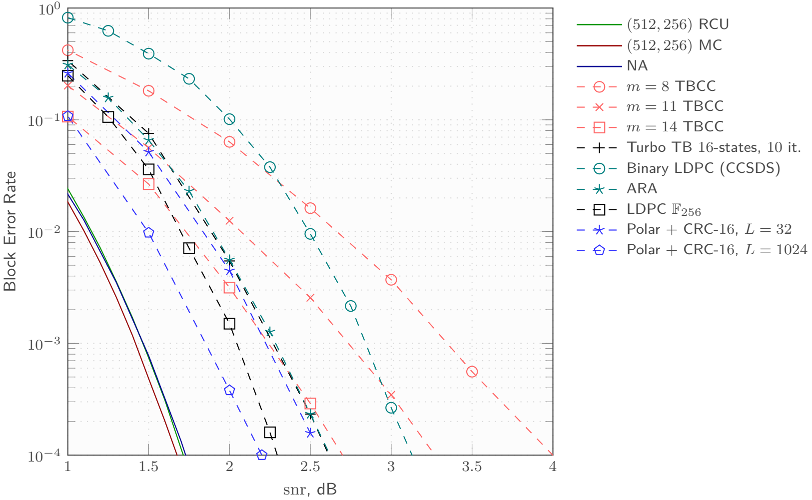
Case 2: $n = 512, k = 256$



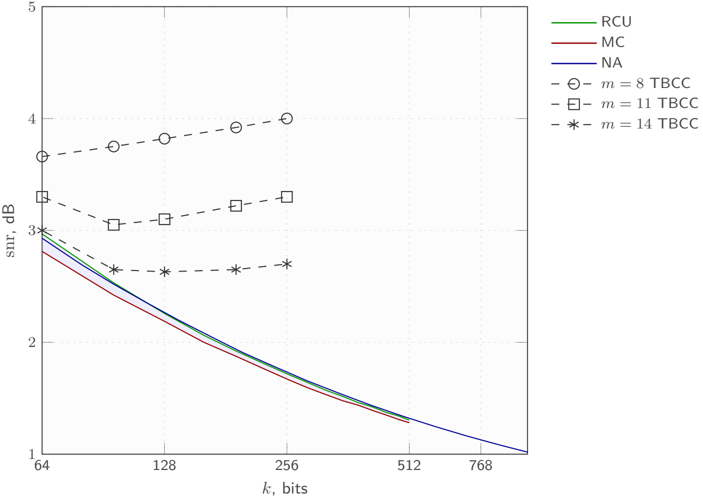
Case 2: $n = 512, k = 256$



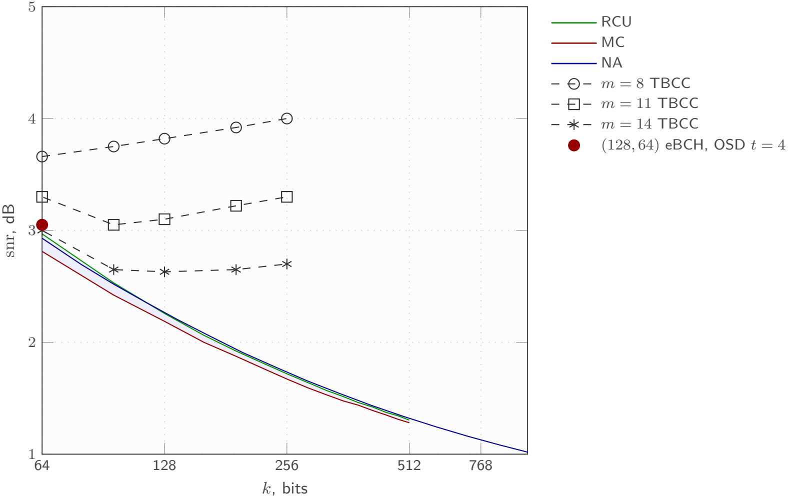
Case 2: $n = 512, k = 256$



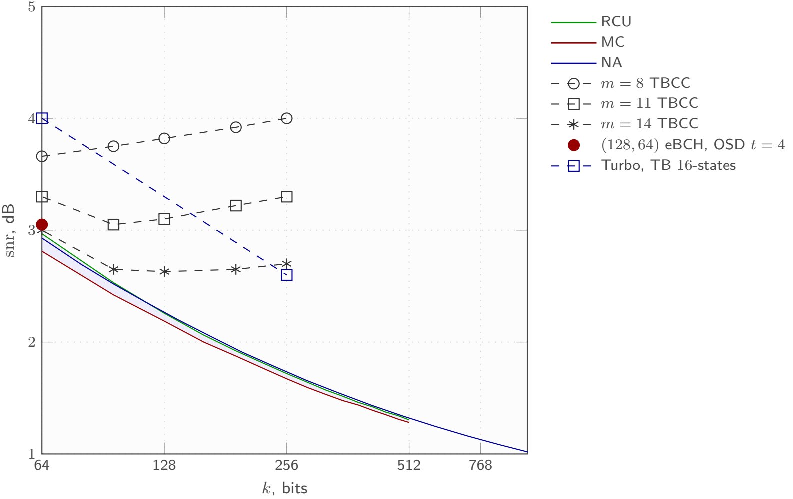
Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



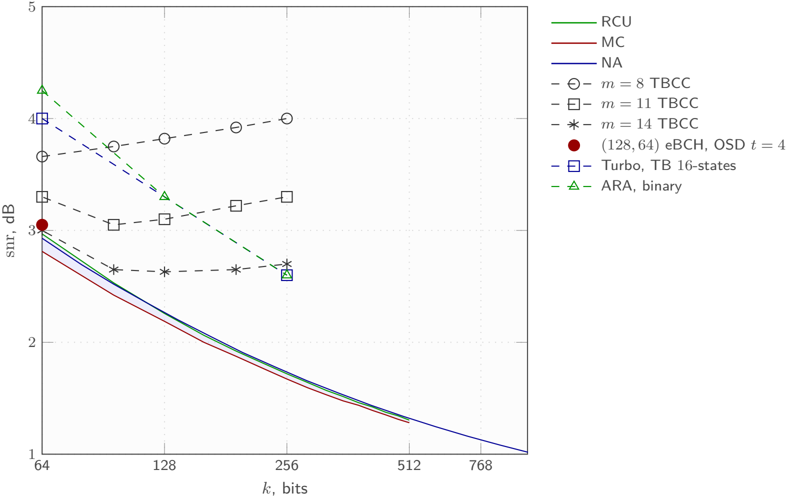
Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



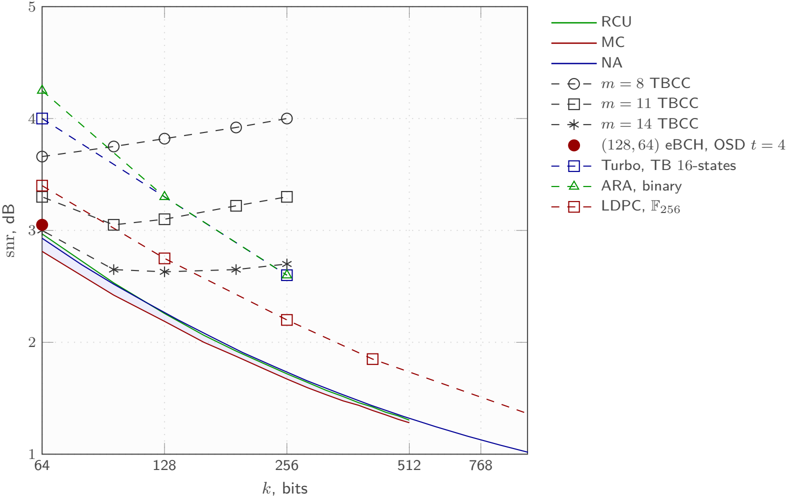
Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



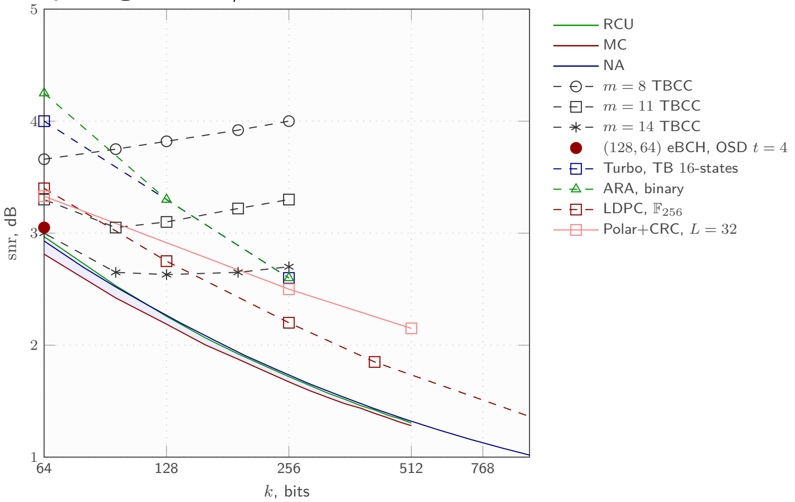
Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



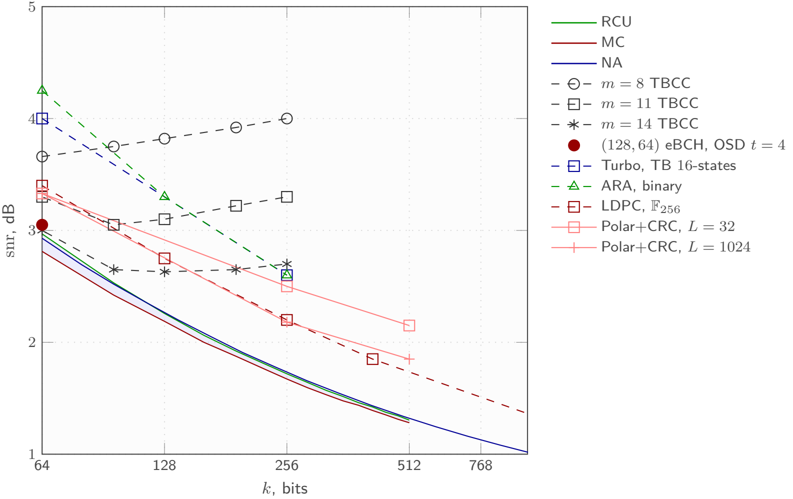
Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



Comparing Rate-$1/2$ Codes at $P_B = 10^{-4}$



Complexity

- We use the model from

O. İşcan et al., "A Comparison of Channel Coding Schemes for 5G Short Message Transmission", in Proc. globecom (2016)

[30] *Channel coding evaluation assumptions - performance and complexity*, tech. rep. (Qualcomm Inc., Nanjing, China, May 2016), 3GPP TSG–RAN WG1 no. 85

Complexity

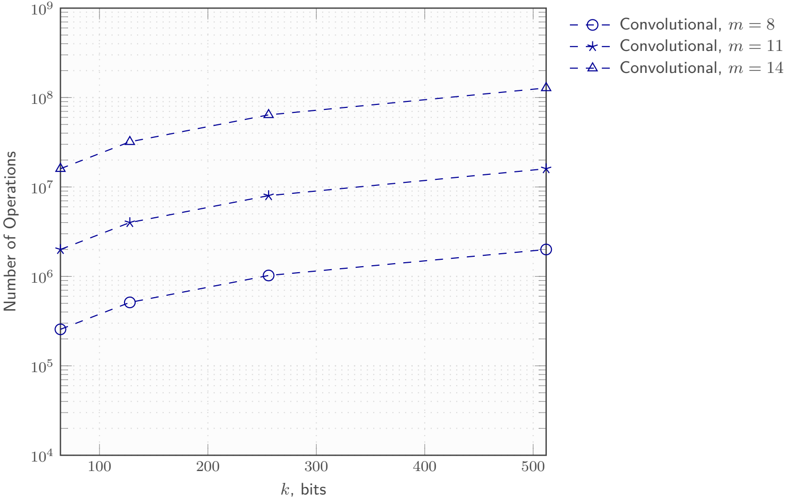
- We use the model from

  O. İșcan et al., "A Comparison of Channel Coding Schemes for 5G Short Message Transmission", in Proc. globecom (2016)

- Another good reference with a detailed comparison of (binary) LDPC, Turbo and Polar codes is [30].

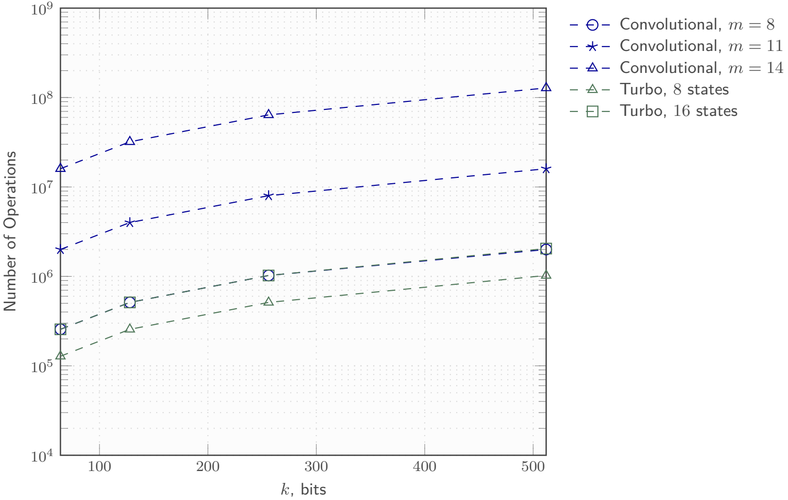
[30] *Channel coding evaluation assumptions - performance and complexity*, tech. rep. (Qualcomm Inc., Nanjing, China, May 2016), 3GPP TSG–RAN WG1 no. 85

Complexity

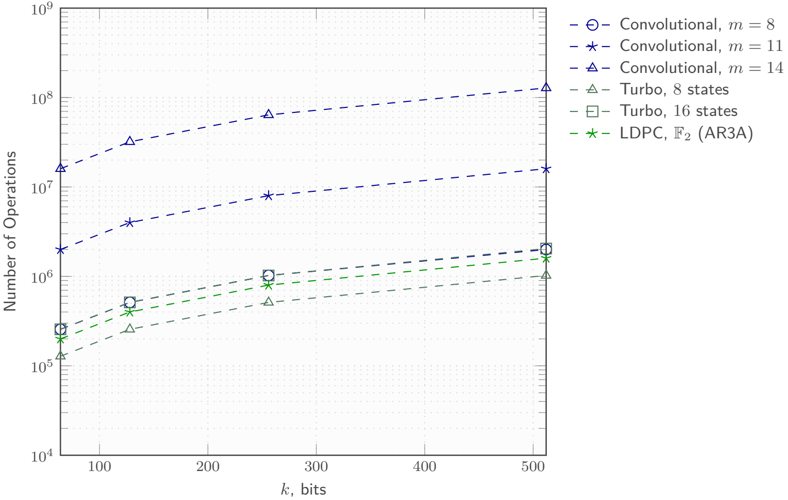


10^9
10^8
10^7
10^6
10^5
10^4
Number of Operations
100
200
300
400
500
k, bits
Convolutional, $m = 8$
Convolutional, $m = 11$
Convolutional, $m = 14$

Complexity

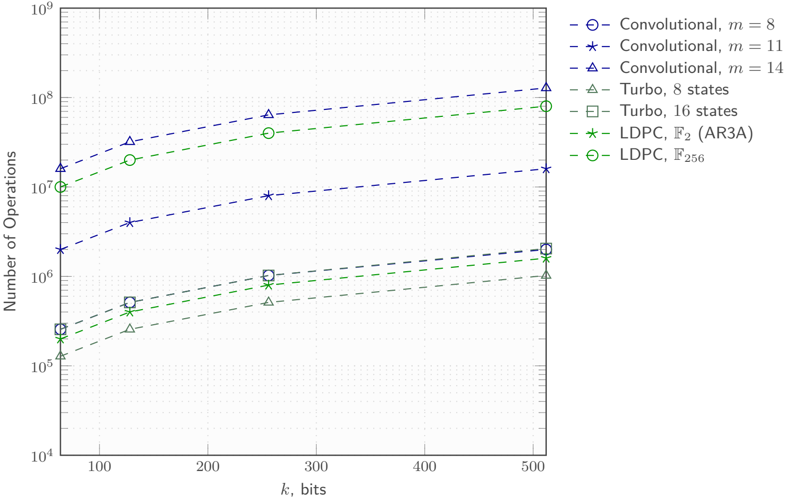

Convolutional, $m = 8$
Convolutional, $m = 11$
Convolutional, $m = 14$
Turbo, 8 states
Turbo, 16 states
Number of Operations
k, bits
10^4
10^5
10^6
10^7
10^8
10^9
100
200
300
400
500

Complexity



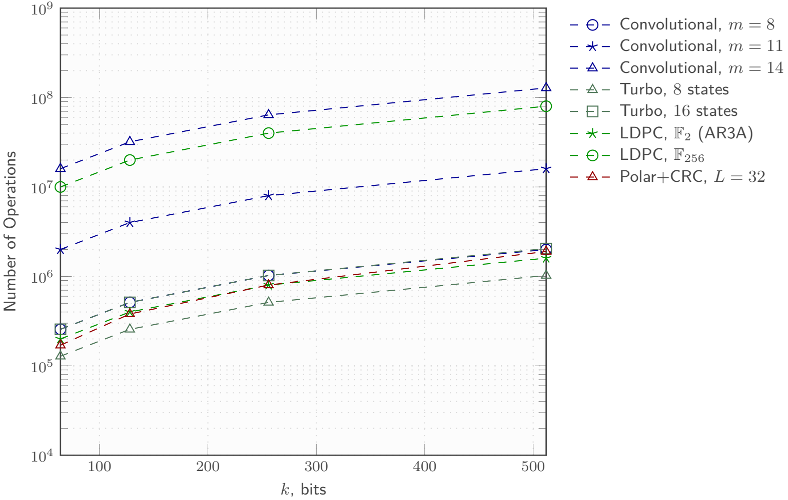
Number of Operations
10^4
10^5
10^6
10^7
10^8
10^9
100
200
300
400
500
k, bits
Convolutional, $m = 8$
Convolutional, $m = 11$
Convolutional, $m = 14$
Turbo, 8 states
Turbo, 16 states
LDPC, $\mathbb{F}_2$ (AR3A)

Complexity



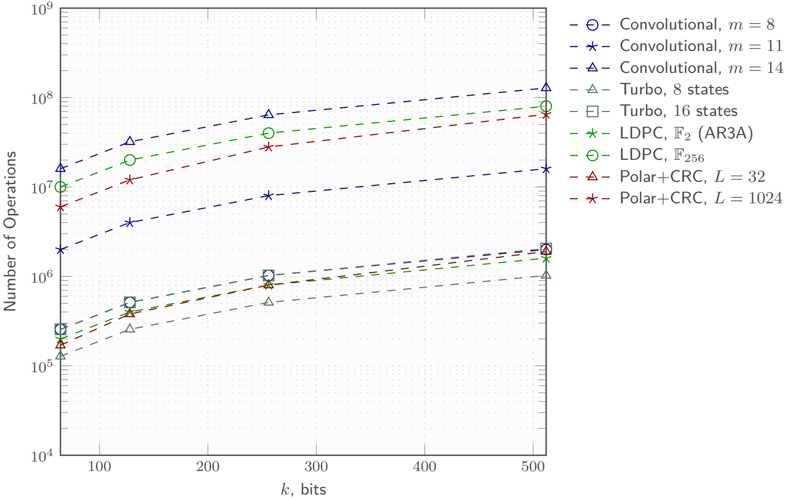
10^9
10^8
10^7
10^6
10^5
10^4
Number of Operations
100
200
300
400
500
k, bits
Convolutional, $m = 8$
Convolutional, $m = 11$
Convolutional, $m = 14$
Turbo, 8 states
Turbo, 16 states
LDPC, $\mathbb{F}_2$ (AR3A)
LDPC, $\mathbb{F}_{256}$

Complexity



10^9
10^8
10^7
10^6
10^5
10^4
Number of Operations
100
200
300
400
500
k, bits
Convolutional, $m = 8$
Convolutional, $m = 11$
Convolutional, $m = 14$
Turbo, 8 states
Turbo, 16 states
LDPC, $\mathbb{F}_2$ (AR3A)
LDPC, $\mathbb{F}_{256}$
Polar+CRC, $L = 32$

Complexity



Number of Operations
10^9
10^8
10^7
10^6
10^5
10^4
100
200
300
400
500
k, bits
Convolutional, $m = 8$
Convolutional, $m = 11$
Convolutional, $m = 14$
Turbo, 8 states
Turbo, 16 states
LDPC, $\mathbb{F}_2$ (AR3A)
LDPC, $\mathbb{F}_{256}$
Polar+CRC, $L = 32$
Polar+CRC, $L = 1024$

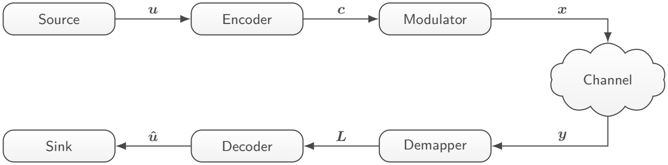
Complete vs. Incomplete: Some Observations on Error Detection

Code Family	Decoding Algorithm	Complete/Incomplete
TBCC	WAVA	“Almost” complete
Linear Block	OSD	Complete
Polar+CRC	List	“Almost” complete for large lists
LDPCC	BP	Incomplete
Turbo	BP	Complete?

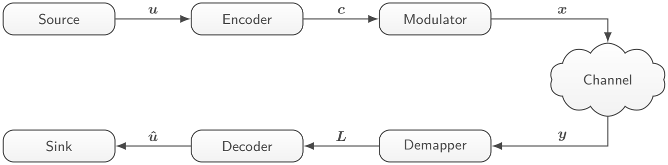
Outline

Outline

Higher-Order Modulation
System Model

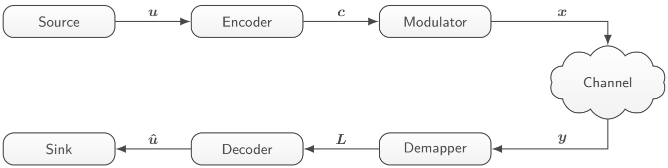


Higher-Order Modulation
System Model



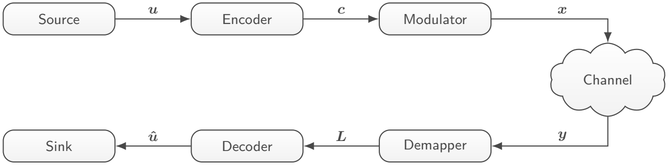
- Channel code with rate $R_\mathsf{c} = k/n_\mathsf{c}$, blocklength n_c bits.

Higher-Order Modulation
System Model



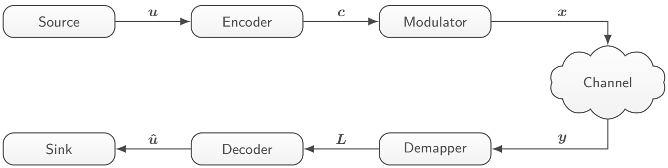
- Channel code with rate $R_c = k/n_c$, blocklength n_c bits.
- Additive White Gaussian Noise (AWGN) channel with $M = 2^m$-ary discrete input.

Higher-Order Modulation
System Model



- Channel code with rate $R_\mathsf{c} = k/n_\mathsf{c}$, blocklength n_c bits.
- Additive White Gaussian Noise (AWGN) channel with $M = 2^m$-ary discrete input.
- Optimal signaling strategy to achieve capacity, i.e., $\log_2(1 + \mathrm{snr})$, requires Gaussian distributed inputs.

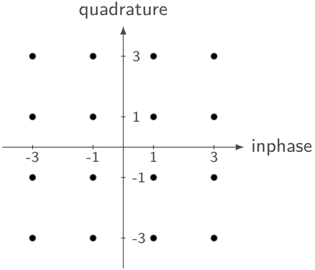
Higher-Order Modulation
System Model



- Channel code with rate $R_c = k/n_c$, blocklength n_c bits.
- Additive White Gaussian Noise (AWGN) channel with $M = 2^m$-ary discrete input.
- Optimal signaling strategy to achieve capacity, i.e., $\log_2(1 + \mathrm{snr})$, requires Gaussian distributed inputs.
- Transmission rate: $R = R_c m$ (bits/per channel use).

Higher-Order Modulation
Discrete Signaling (I)

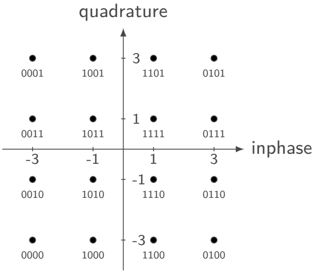
- We use $M = 2^m$-quadrature amplitude (QAM) constellations $\mathcal{X}$.



$$\mathcal{X} = \{-(M-1) - \mathrm{j}(M-1), -(M-1) - \mathrm{j}(M-2), \ldots (M-1) + \mathrm{j}(M-1)\}$$

Higher-Order Modulation
Discrete Signaling (I)

- We use $M = 2^m$-quadrature amplitude (QAM) constellations $\mathcal{X}$.



$$\mathcal{X} = \{-(M-1)-\mathrm{j}(M-1), -(M-1)-\mathrm{j}(M-2), \ldots (M-1)+\mathrm{j}(M-1)\}$$

- Binary labeling $\chi : \mathcal{X} \to \{0,1\}^m$, e.g., Binary Reflected Gray Code (BRGC), $\chi(-3+3\,\mathrm{j}) = 0001$.

Higher-Order Modulation

Decoding Metrics

- The decoder uses a metric $q(\boldsymbol{x}, \boldsymbol{y}) : \mathcal{X}^n \times \mathcal{Y}^n \to \mathbb{R}^+$ to estimate the sent codeword from the observation:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c} \in \mathcal{C}} q(\chi^{-1}(\boldsymbol{c}), \boldsymbol{y})$$

- We distinguish between symbol-metric decoding (SMD) and bit-metric decoding (BMD).
  - SMD:

$$q(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{n} p_{Y|X}(y_j|x_j)$$

  - BMD:

$$q(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{n} \prod_{i=1}^{m} p_{Y|B_i}(y_j|b_{ji})$$

  with $p_{Y|B_i}(y|b) = \sum_{x \in \mathcal{X}_i^b} p_{Y|X}(y|x)$ and $\mathcal{X}_i^b = \{x \in \mathcal{X} \colon [\chi(x)]_i = b\}$.

Higher-Order Modulation
Achievable Rates: Overview

$$q(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{n} p_{Y|X}(y_j | x_j)$$

- Achievable rate is the mutual information:

$$R_{\mathsf{a}} = \mathrm{I}(X; Y).$$

- Relevant metric usually for non-binary codes and multilevel coding.

$$q(\boldsymbol{x}, \boldsymbol{y}) = \prod_{j=1}^{n} \prod_{i=1}^{m} p_{Y|B_i}(y_j | b_{ji})$$

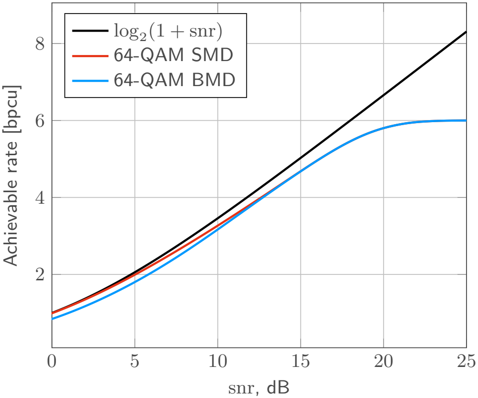
- Achievable rate is the "BICM capacity"

$$R_{\mathsf{a}} = \sum_{i=1}^{m} \mathrm{I}(B_i; Y).$$

- Relevant metric for binary codes, when each bit-level is treated independently at the receiver.

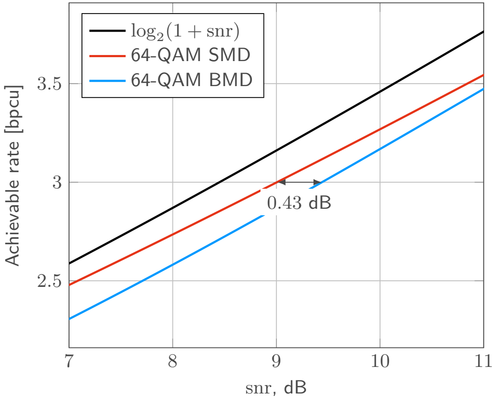
Higher-Order Modulation

Achievable Rates: Numerical Example



Higher-Order Modulation

Achievable Rates: Numerical Example



Outline

Approaching Capacity with Discrete Signaling
The Last dB

- Discrete signaling with M uniformly distributed, equi-spaced constellation points is suboptimal for the AWGN channel under an average power constraint.

Approaching Capacity with Discrete Signaling
The Last dB

- Discrete signaling with M uniformly distributed, equi-spaced constellation points is suboptimal for the AWGN channel under an average power constraint.
- For $M \to \infty$, loss in power efficiency may increase up to 1.53 dB in the high SNR regime.

Approaching Capacity with Discrete Signaling
The Last dB

- Discrete signaling with M uniformly distributed, equi-spaced constellation points is suboptimal for the AWGN channel under an average power constraint.
- For $M \to \infty$, loss in power efficiency may increase up to 1.53 dB in the high SNR regime.
- How to solve this problem? Follow Shannon's 1948 blueprint.

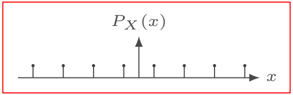
$$C = \max_{p_X} \quad \mathrm{I}(X;Y) \quad \text{subject to “}p_X\text{ is permissible”}$$

Approaching Capacity with Discrete Signaling
The Last dB

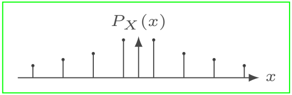
- Discrete signaling with M uniformly distributed, equi-spaced constellation points is suboptimal for the AWGN channel under an average power constraint.
- For $M \to \infty$, loss in power efficiency may increase up to 1.53 dB in the high SNR regime.
- How to solve this problem? Follow Shannon's 1948 blueprint.

$$C = \max_{p_X} \quad \mathrm{I}(X;Y) \quad \text{subject to "} p_X \text{ is permissible"}$$

- Impose non-uniform distribution on the constellation points.



$\Longrightarrow$



Probabilistic Shaping
Integration with FEC

- The difficult aspect of non-uniform signaling is its integration with FEC.

[31] R. G. Gallager, *Information Theory and Reliable Communication*, (John Wiley & Sons, Inc., 1968)
[32] G. D. Forney, "Trellis shaping", IEEE Trans. Inf. Theory **38**, 281–300 (1992)
[33] G. Böcherer et al., "Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation", IEEE Trans. Commun. **63**, 4651–4665 (2015)

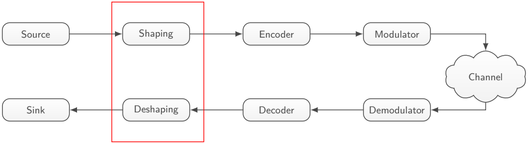
Probabilistic Shaping
Integration with FEC

- The difficult aspect of non-uniform signaling is its integration with FEC.
- Straightforward approaches suffer from error propagation, synchronization issues or complex joint FEC/shaping code decoding[31,32].

[31] R. G. Gallager, *Information Theory and Reliable Communication*, (John Wiley & Sons, Inc., 1968)
[32] G. D. Forney, "Trellis shaping", IEEE Trans. Inf. Theory **38**, 281–300 (1992)
[33] G. Böcherer et al., "Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation", IEEE Trans. Commun. **63**, 4651–4665 (2015)

Probabilistic Shaping
Integration with FEC

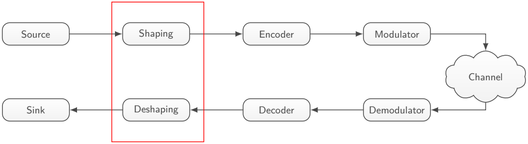
- The difficult aspect of non-uniform signaling is its integration with FEC.
- Straightforward approaches suffer from error propagation, synchronization issues or complex joint FEC/shaping code decoding[31,32].
- Recent approach: Probabilistic amplitude shaping (PAS)[33] uses reverse concatenation.



[31] R. G. Gallager, *Information Theory and Reliable Communication*, (John Wiley & Sons, Inc., 1968)
[32] G. D. Forney, "Trellis shaping", IEEE Trans. Inf. Theory **38**, 281–300 (1992)
[33] G. Böcherer et al., "Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation", IEEE Trans. Commun. **63**, 4651–4665 (2015)

Probabilistic Shaping
Integration with FEC

- The difficult aspect of non-uniform signaling is its integration with FEC.
- Straightforward approaches suffer from error propagation, synchronization issues or complex joint FEC/shaping code decoding[31,32].
- Recent approach: Probabilistic amplitude shaping (PAS)[33] uses reverse concatenation.



- PAS requires: Symmetric input distribution, systematic FEC encoding.

[31] R. G. Gallager, *Information Theory and Reliable Communication*, (John Wiley & Sons, Inc., 1968)
[32] G. D. Forney, "Trellis shaping", IEEE Trans. Inf. Theory **38**, 281–300 (1992)
[33] G. Böcherer et al., "Bandwidth Efficient and Rate-Matched Low-Density Parity-Check Coded Modulation", IEEE Trans. Commun. **63**, 4651–4665 (2015)

Probabilistic Amplitude Shaping (PAS)
Achievable Rates

- An achievable rate for the considered scheme is[34]:

$$R_{\mathsf{a}} = \left[\mathrm{H}(X) - \mathrm{E}\left[-\log_2 \left(\frac{q(X,Y)}{\sum_{x \in \mathcal{X}} q(x,Y)} \right) \right] \right]^+$$

- $q(x,y)$ is the previously introduced decoding metric, e.g.,

[34] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Probabilistic Amplitude Shaping (PAS)
Achievable Rates

- An achievable rate for the considered scheme is[34]:

$$R_{\mathsf{a}} = \left[\mathrm{H}(X) - \mathrm{E}\left[-\log_2 \left(\frac{q(X,Y)}{\sum_{x \in \mathcal{X}} q(x,Y)} \right) \right] \right]^{+}$$

- $q(x,y)$ is the previously introduced decoding metric, e.g.,

SMD: $q(x,y) = P_{X|Y}(x|y)$

$$R_{\mathsf{SMD}} = \mathrm{I}(X;Y)$$

[34] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Probabilistic Amplitude Shaping (PAS)
Achievable Rates

- An achievable rate for the considered scheme is[34]:

$$R_{\mathsf{a}} = \left[\mathrm{H}(X) - \mathrm{E}\left[-\log_2 \left(\frac{q(X,Y)}{\sum_{x\in\mathcal{X}} q(x,Y)} \right) \right] \right]^+$$

- $q(x,y)$ is the previously introduced decoding metric, e.g.,

SMD: $q(x,y) = P_{X|Y}(x|y)$

$$R_{\mathsf{SMD}} = \mathrm{I}(X;Y)$$

BMD: $q(x,y) = \prod_{i=1}^{m} P_{B_i|Y}(b_i|y)$

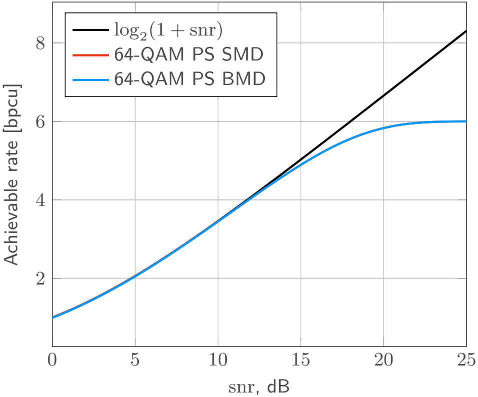
$$R_{\mathsf{BMD}} = \left[\mathrm{H}(X) - \sum_{i=1}^{m} \mathrm{H}(B_i|Y) \right]^+$$

[34] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Probabilistic Amplitude Shaping (PAS)

The shaping gap has vanished

PAS operates at the Shannon limit for SMD and BMD.



Outline

Non-Binary LDPC Codes
Basics with Higher-Order Modulation

- Straightforward approach for higher-order modulation: Use non-binary code over a field $\mathbb{F}_q$ that matches the constellation size, i.e., $M = q$.

Non-Binary LDPC Codes
Basics with Higher-Order Modulation

- Straightforward approach for higher-order modulation: Use non-binary code over a field $\mathbb{F}_q$ that matches the constellation size, i.e., $M = q$.
- For practical implementations the field size q is a binary extension field such that $q = 2^p$.

Non-Binary LDPC Codes
Basics with Higher-Order Modulation

- Straightforward approach for higher-order modulation: Use non-binary code over a field $\mathbb{F}_q$ that matches the constellation size, i.e., $M = q$.
- For practical implementations the field size q is a binary extension field such that $q = 2^p$.
- We can also combine a sequence of ℓ 2^m-QAM symbols with a code over $\mathbb{F}_q$, if

$$\ell \cdot m \stackrel{!}{=} p$$

Examples:

Non-Binary LDPC Codes
Basics with Higher-Order Modulation

- Straightforward approach for higher-order modulation: Use non-binary code over a field $\mathbb{F}_q$ that matches the constellation size, i.e., $M = q$.
- For practical implementations the field size q is a binary extension field such that $q = 2^p$.
- We can also combine a sequence of ℓ 2^m-QAM symbols with a code over $\mathbb{F}_q$, if

$$\ell \cdot m \stackrel{!}{=} p$$

  Examples:
  - $\ell = 2$ 16-QAM symbols with $\mathbb{F}_{256}$.
  - $\ell = 3$ 8-QAM symbols with $\mathbb{F}_{512}$.
  - ...

Non-Binary LDPC Codes

Decoding: Uniform case

- We introduce a mapping $\beta_{\mathcal{X}} : \mathcal{X}^{\ell} \to \mathbb{F}_q$. Its inverse is defined analogously.
- For the i-th variable node, the NB-LDPC decoder is provided with the soft-information vector $\boldsymbol{P}_i = (P_i(0), P_i(1), P_i(\alpha), \ldots, P_i(\alpha^{q-2}))$ where

$$P_i(c) \propto \prod_{j=1}^{\ell} p_{Y|X}(y_j | [\beta_{\mathcal{X}}^{-1}(c)]_j)$$

[35] F. Steiner et al., "Ultra-Sparse Non-Binary LDPC Codes for Probabilistic Amplitude Shaping", in Proc. IEEE Global Telecommun. Conf. (GLOBECOM) (Dec. 2017), pp. 1–5

Non-Binary LDPC Codes

Decoding: Uniform case

- We introduce a mapping $\beta_{\mathcal{X}} : \mathcal{X}^{\ell} \to \mathbb{F}_q$. Its inverse is defined analogously.
- For the i-th variable node, the NB-LDPC decoder is provided with the soft-information vector $\boldsymbol{P}_i = (P_i(0), P_i(1), P_i(\alpha), \ldots, P_i(\alpha^{q-2}))$ where

$$P_i(c) \propto \prod_{j=1}^{\ell} p_{Y|X}(y_j|[\beta_{\mathcal{X}}^{-1}(c)]_j)$$

- For PAS, similar expressions can be derived[35].

[35] F. Steiner et al., "Ultra-Sparse Non-Binary LDPC Codes for Probabilistic Amplitude Shaping", in Proc. IEEE Global Telecommun. Conf. (GLOBECOM) (Dec. 2017), pp. 1–5

Non-Binary LDPC Codes
Decoding: Uniform case

- We introduce a mapping $\beta_{\mathcal{X}} : \mathcal{X}^{\ell} \to \mathbb{F}_q$. Its inverse is defined analogously.
- For the i-th variable node, the NB-LDPC decoder is provided with the soft-information vector $\boldsymbol{P}_i = (P_i(0), P_i(1), P_i(\alpha), \ldots, P_i(\alpha^{q-2}))$ where

$$P_i(c) \propto \prod_{j=1}^{\ell} p_{Y|X}(y_j | [\beta_{\mathcal{X}}^{-1}(c)]_j)$$

- For PAS, similar expressions can be derived[35].
- Achievable rate is the mutual information $\mathrm{I}(X;Y)$.

[35] F. Steiner et al., "Ultra-Sparse Non-Binary LDPC Codes for Probabilistic Amplitude Shaping", in Proc. IEEE Global Telecommun. Conf. (GLOBECOM) (Dec. 2017), pp. 1–5

Outline

Binary LDPC Codes
Basics

- Binary LDPC codes are usually operated in a "BICM fashion" for higher-order modulation, i.e., BMD is performed.

Binary LDPC Codes
Basics

- Binary LDPC codes are usually operated in a "BICM fashion" for higher-order modulation, i.e., BMD is performed.
- The input to the LDPC decoder is the soft-information

$$\ell_{ji} = \log\left(\frac{P_{B_i|X}(0|y_j)}{P_{B_i|X}(1|y_j)}\right) = \log\left(\frac{\sum_{x\in\mathcal{X}_i^0} p_{Y|X}(y_j|x)P_X(x)}{\sum_{x\in\mathcal{X}_j^1} p_{Y|X}(y_j|x)P_X(x)}\right),$$

for $j = 1, \ldots, n$ and $i = 1, \ldots, m$.

Binary LDPC Codes
Basics

- Binary LDPC codes are usually operated in a "BICM fashion" for higher-order modulation, i.e., BMD is performed.
- The input to the LDPC decoder is the soft-information

$$\ell_{ji} = \log\left(\frac{P_{B_i|X}(0|y_j)}{P_{B_i|X}(1|y_j)}\right) = \log\left(\frac{\sum_{x\in\mathcal{X}_i^0} p_{Y|X}(y_j|x)P_X(x)}{\sum_{x\in\mathcal{X}_j^1} p_{Y|X}(y_j|x)P_X(x)}\right),$$

  for $j = 1, \ldots, n$ and $i = 1, \ldots, m$.
- Most practically relevant and standardized LDPC codes are quasi-cyclic and allow a protograph representation.

Binary LDPC Codes
Basics

- Binary LDPC codes are usually operated in a "BICM fashion" for higher-order modulation, i.e., BMD is performed.
- The input to the LDPC decoder is the soft-information

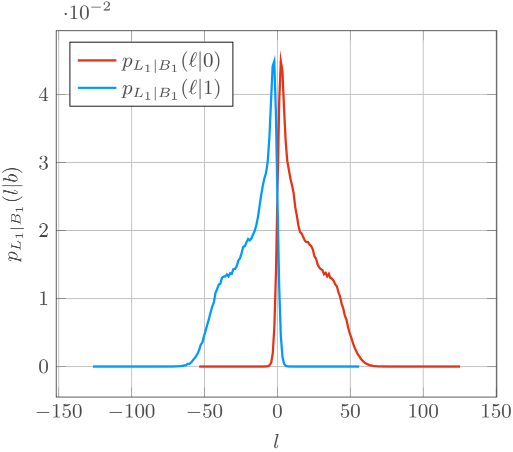
$$\ell_{ji} = \log\left(\frac{P_{B_i|X}(0|y_j)}{P_{B_i|X}(1|y_j)}\right) = \log\left(\frac{\sum_{x\in\mathcal{X}_i^0} p_{Y|X}(y_j|x)P_X(x)}{\sum_{x\in\mathcal{X}_j^1} p_{Y|X}(y_j|x)P_X(x)}\right),$$

for $j = 1, \ldots, n$ and $i = 1, \ldots, m$.

- Most practically relevant and standardized LDPC codes are quasi-cyclic and allow a protograph representation.
- Most codes have a irregular variable node degree profile as they are superior to the regular counterparts.

Binary LDPC Codes
Distribution of the Log-Likelihood Ratios

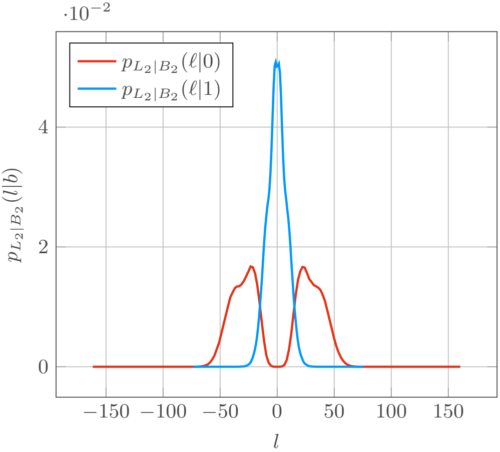
8-ASK uniform, 14 dB, Binary Reflected Gray Code, Bit-Level 1



Binary LDPC Codes

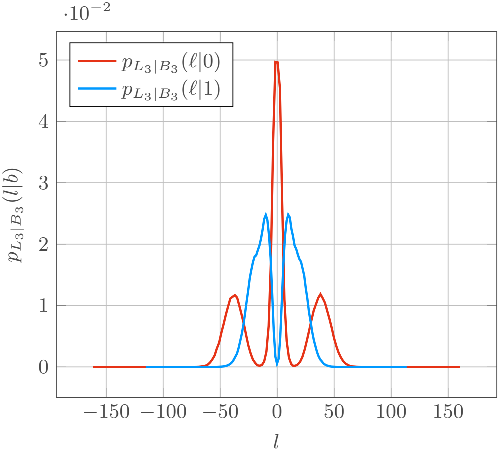
Distribution of the Log-Likelihood Ratios

8-ASK uniform, 14 dB, Binary Reflected Gray Code, Bit-Level 2



Binary LDPC Codes
Distribution of the Log-Likelihood Ratios

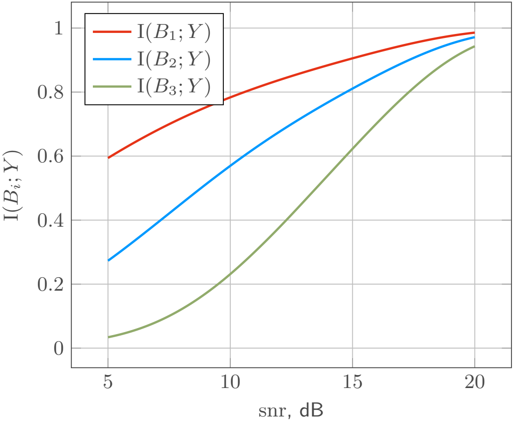
8-ASK uniform, 14 dB, Binary Reflected Gray Code, Bit-Level 3



Binary LDPC Codes

Quality of Bit-Levels: Bitwise mutual information $\mathrm{I}(B_i;Y)$

8-ASK uniform



Binary LDPC Codes
P-EXIT Analysis

- P-EXIT extends traditional EXIT approach[36] to protographs.

[36] G. Liva and M. Chiani, "Protograph LDPC Codes Design Based on EXIT Analysis", in IEEE Global Telecommun. Conf. (GLOBECOM) (Nov. 2007), pp. 3250–3254

Binary LDPC Codes
P-EXIT Analysis

- P-EXIT extends traditional EXIT approach[36] to protographs.
- Consider protograph base matrix $\boldsymbol{B} \in \mathbb{N}^{m_\mathrm{p} \times n_\mathrm{p}}$.

[36] G. Liva and M. Chiani, "Protograph LDPC Codes Design Based on EXIT Analysis", in IEEE Global Telecommun. Conf. (GLOBECOM) (Nov. 2007), pp. 3250–3254

Binary LDPC Codes
P-EXIT Analysis

- P-EXIT extends traditional EXIT approach[36] to protographs.
- Consider protograph base matrix $\boldsymbol{B} \in \mathbb{N}^{m_\mathsf{p} \times n_\mathsf{p}}$.
- Assign to each variable node one of the m different bit levels, i.e., a number of n_p/m variable nodes "see" the same bit channel.

[36] G. Liva and M. Chiani, "Protograph LDPC Codes Design Based on EXIT Analysis", in IEEE Global Telecommun. Conf. (GLOBECOM) (Nov. 2007), pp. 3250–3254

Binary LDPC Codes
P-EXIT Analysis

- P-EXIT extends traditional EXIT approach[36] to protographs.
- Consider protograph base matrix $\boldsymbol{B} \in \mathbb{N}^{m_\mathsf{p} \times n_\mathsf{p}}$.
- Assign to each variable node one of the m different bit levels, i.e., a number of n_p/m variable nodes "see" the same bit channel.
- P-EXIT is used to track the reliability of the exchanged messages.

[36] G. Liva and M. Chiani, "Protograph LDPC Codes Design Based on EXIT Analysis", in IEEE Global Telecommun. Conf. (GLOBECOM) (Nov. 2007), pp. 3250–3254

Binary LDPC Codes
P-EXIT Analysis

- P-EXIT extends traditional EXIT approach[36] to protographs.
- Consider protograph base matrix $\boldsymbol{B} \in \mathbb{N}^{m_\mathsf{p} \times n_\mathsf{p}}$.
- Assign to each variable node one of the m different bit levels, i.e., a number of n_p/m variable nodes "see" the same bit channel.
- P-EXIT is used to track the reliability of the exchanged messages.
- P-EXIT was derived for the BEC and the biAWGN channel. How to use it for our scenario?

[36] G. Liva and M. Chiani, "Protograph LDPC Codes Design Based on EXIT Analysis", in IEEE Global Telecommun. Conf. (GLOBECOM) (Nov. 2007), pp. 3250–3254

Binary LDPC Codes
Surrogate Parameter Design

Obtaining the surrogate parameters

Which information theoretic quantity should be used to relate the real bit channels $p_{L_i|B_i}$ to the surrogate channel parameter?

[37] F. Steiner et al., "Protograph-Based LDPC Code Design for Shaped Bit-Metric Decoding", IEEE J. Sel. Areas Commun. **34**, 397–407 (2016)

[38] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Binary LDPC Codes
Surrogate Parameter Design

Obtaining the surrogate parameters

Which information theoretic quantity should be used to relate the real bit channels $p_{L_i|B_i}$ to the surrogate channel parameter?

- It turns out[37,38]: It's the channel uncertainty.

$$\mathrm{E}\left[-\log_2\left(\frac{q(X,Y)}{\sum_{x\in\mathcal{X}} q(x,Y)}\right)\right]$$

[37] F. Steiner et al., "Protograph-Based LDPC Code Design for Shaped Bit-Metric Decoding", IEEE J. Sel. Areas Commun. **34**, 397–407 (2016)

[38] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Binary LDPC Codes
Surrogate Parameter Design

Obtaining the surrogate parameters

Which information theoretic quantity should be used to relate the real bit channels $p_{L_i|B_i}$ to the surrogate channel parameter?

- It turns out[37,38]: It's the channel uncertainty.

$$\mathrm{E}\left[-\log_2\left(\frac{q(X,Y)}{\sum_{x\in\mathcal{X}} q(x,Y)}\right)\right] \overset{\text{soft BMD}}{=} \sum_{i=1}^{m} \mathrm{H}(B_i|Y)$$

[37] F. Steiner et al., "Protograph-Based LDPC Code Design for Shaped Bit-Metric Decoding", IEEE J. Sel. Areas Commun. **34**, 397–407 (2016)

[38] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Binary LDPC Codes
Surrogate Parameter Design

Obtaining the surrogate parameters

Which information theoretic quantity should be used to relate the real bit channels $p_{L_i|B_i}$ to the surrogate channel parameter?

- It turns out[37,38]: It's the channel uncertainty.

$$\mathrm{E}\left[-\log_2\left(\frac{q(X,Y)}{\sum_{x\in\mathcal{X}} q(x,Y)}\right)\right] \overset{\text{soft BMD}}{=} \sum_{i=1}^{m} \mathrm{H}(B_i|Y)$$

- The biAWGN surrogate parameters are therefore given by

$$\sigma_{\text{ch}_i} : \mathrm{H}(B_i|Y) = \mathrm{H}(\tilde{X}|\tilde{Y}), \text{ where } \tilde{Y} = \tilde{X} + N_i \text{ and } N_i \sim \mathcal{N}(0, \sigma^2_{\text{ch}_i}).$$

[37] F. Steiner et al., "Protograph-Based LDPC Code Design for Shaped Bit-Metric Decoding", IEEE J. Sel. Areas Commun. **34**, 397–407 (2016)

[38] G. Böcherer, "Achievable Rates for Probabilistic Shaping", arXiv:1707.01134v5 (2018)

Outline

Polar Codes
Decoding Metric

- Because of SC decoding, the most "natural" way for higher-order modulation is a multilevel coding/multistage decoding approach[39].

[39] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes", IEEE Trans. Inf. Theory **23**, 371–377 (1977)

Polar Codes
Decoding Metric

- Because of SC decoding, the most "natural" way for higher-order modulation is a multilevel coding/multistage decoding approach[39].
- This builds heavily on using the chain rule of mutual information:

$$
\begin{aligned}
\mathrm{I}(X;Y) = \mathrm{I}(\boldsymbol{B};Y) &= \mathrm{I}(B_1;Y) + \mathrm{I}(B_2;Y|B_1) + \ldots + \mathrm{I}(B_m;Y|B_1 \ldots B_{m-1}) \\
&= \sum_{i=1}^{m} \mathrm{I}(B_i;Y|B_1^{m-1})
\end{aligned}
$$

[39] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes", IEEE Trans. Inf. Theory **23**, 371–377 (1977)

Polar Codes
Decoding Metric

- Because of SC decoding, the most "natural" way for higher-order modulation is a multilevel coding/multistage decoding approach[39].
- This builds heavily on using the chain rule of mutual information:

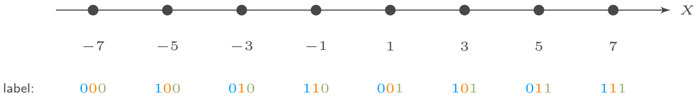
$$\begin{aligned} \mathrm{I}(X;Y) = \mathrm{I}(\boldsymbol{B};Y) &= \mathrm{I}(B_1;Y) + \mathrm{I}(B_2;Y|B_1) + \ldots + \mathrm{I}(B_m;Y|B_1 \ldots B_{m-1}) \\ &= \sum_{i=1}^{m} \mathrm{I}(B_i;Y|B_1^{m-1}) \end{aligned}$$

- Polar codes with multilevel coding/multistage decoding do not suffer from a "BICM loss".

[39] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes", IEEE Trans. Inf. Theory **23**, 371–377 (1977)

Polar Codes
Set-Partition Labeling

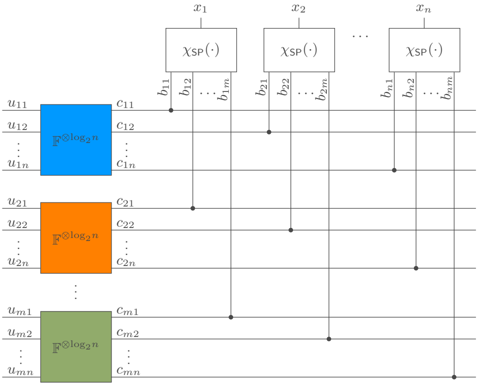
- Seidl *et al.*[40] showed that a set-partition (SP) labeling is best for polar codes and higher-order modulation (improves polarization).



[40] M. Seidl et al., "Polar-Coded Modulation", IEEE Trans. Commun. **61**, 4108–4119 (2013)

Polar Codes
Construction (I)

Example: 8-ASK, $m = 3$.



Polar Codes
Construction (II)

- Construction with Gaussian approximation and biAWGN surrogate channels[41].

[41] G. Böcherer et al., "Efficient Polar Code Construction for Higher-Order Modulation", in IEEE Wireless Commun. Netw. Conf. (WCNC) (Mar. 2017)

Polar Codes
Construction (II)

- Construction with Gaussian approximation and biAWGN surrogate channels[41].
- The variance of the i-th biAWGN surrogate channel is

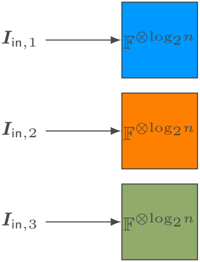
$$\sigma_i^2 = \mathsf{C}_{\mathsf{biAWGN}}^{-1}(\mathrm{I}(B_i; Y|B_1^{i-1})), \text{ where}$$

$$\mathrm{I}(B_i; Y|B_1^{i-1}) = \int_{-\infty}^{\infty} \sum_{b_1 \ldots b_i \in \{0,1\}^i} p_{Y B_1 \ldots B_i}(y, b_1 \ldots b_i) \cdot \log_2 \left(\frac{p_{Y|B_1 \ldots B_i}(y|b_1 \ldots b_i)}{p_{Y|B_1 \ldots B_{i-1}}(y|b_1 \ldots b_{i-1})} \right) \mathrm{d}y$$

[41] G. Böcherer et al., "Efficient Polar Code Construction for Higher-Order Modulation", in IEEE Wireless Commun. Netw. Conf. (WCNC) (Mar. 2017)

Polar Codes
Construction (III)

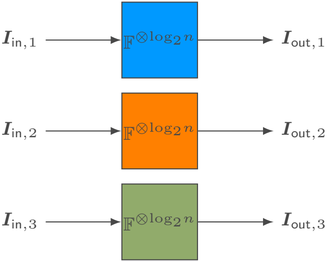
- Run the construction for each sub code separately.



[42] T. Prinz et al., "Polar Coded Probabilistic Amplitude Shaping for Short Packets", in IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC) (2017)

Polar Codes
Construction (III)

- Run the construction for each sub code separately.

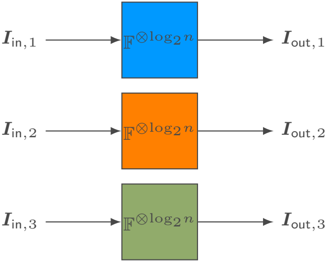


- Stack the reliability vectors in $\boldsymbol{I}_{\text{out}} = (\boldsymbol{I}_{\text{out},1}, \boldsymbol{I}_{\text{out},2}, \boldsymbol{I}_{\text{out},3})$.

[42] T. Prinz et al., "Polar Coded Probabilistic Amplitude Shaping for Short Packets", in IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC) (2017)

Polar Codes
Construction (III)

- Run the construction for each sub code separately.

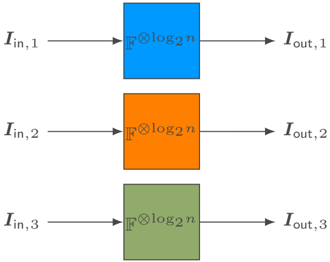


- Stack the reliability vectors in $\boldsymbol{I}_{\text{out}} = (\boldsymbol{I}_{\text{out},1}, \boldsymbol{I}_{\text{out},2}, \boldsymbol{I}_{\text{out},3})$.
- The frozen bit indices are the worst $(n_{\text{c}} - k_{\text{c}})$ bit channels in $\boldsymbol{I}_{\text{out}}$.

[42] T. Prinz et al., "Polar Coded Probabilistic Amplitude Shaping for Short Packets", in IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC) (2017)

Polar Codes
Construction (III)

- Run the construction for each sub code separately.



- Stack the reliability vectors in $\boldsymbol{I}_{\text{out}} = (\boldsymbol{I}_{\text{out},1}, \boldsymbol{I}_{\text{out},2}, \boldsymbol{I}_{\text{out},3})$.
- The frozen bit indices are the worst $(n_{\text{c}} - k_{\text{c}})$ bit channels in $\boldsymbol{I}_{\text{out}}$.
- Construction for PAS is detailed in[42].

[42] T. Prinz et al., "Polar Coded Probabilistic Amplitude Shaping for Short Packets", in IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC) (2017)

Polar Codes
Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$\ell_{j1} = \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right)$$

Polar Codes
Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$\ell_{j1} = \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right) \Longrightarrow \hat{b}_{j1}$$

Polar Codes
Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$\begin{aligned}\ell_{j1} &= \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right) \Longrightarrow \hat{b}_{j1} \\ \ell_{j2} &= \log\left(\frac{P_{B_2|YB_1}(0|y_j,\hat{b}_{j1})}{P_{B_2|YB_1}(1|y_j,\hat{b}_{j1})}\right)\end{aligned}$$

Polar Codes
Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$\begin{aligned} \ell_{j1} &= \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right) \Longrightarrow \hat{b}_{j1} \\ \ell_{j2} &= \log\left(\frac{P_{B_2|YB_1}(0|y_j,\hat{b}_{j1})}{P_{B_2|YB_1}(1|y_j,\hat{b}_{j1})}\right) \Longrightarrow \hat{b}_{j2} \end{aligned}$$

Polar Codes
Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$\begin{aligned}
\ell_{j1} &= \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right) \Longrightarrow \hat{b}_{j1} \\
\ell_{j2} &= \log\left(\frac{P_{B_2|YB_1}(0|y_j,\hat{b}_{j1})}{P_{B_2|YB_1}(1|y_j,\hat{b}_{j1})}\right) \Longrightarrow \hat{b}_{j2} \\
\ell_{j3} &= \log\left(\frac{P_{B_3|YB_1B_2}(0|y_j,\hat{b}_{j1}\hat{b}_{j2})}{P_{B_3|YB_1B_2}(1|y_j,\hat{b}_{j1}\hat{b}_{j2})}\right)
\end{aligned}$$

Polar Codes

Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$
\begin{aligned}
\ell_{j1} &= \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right) \Longrightarrow \hat{b}_{j1} \\
\ell_{j2} &= \log\left(\frac{P_{B_2|YB_1}(0|y_j,\hat{b}_{j1})}{P_{B_2|YB_1}(1|y_j,\hat{b}_{j1})}\right) \Longrightarrow \hat{b}_{j2} \\
\ell_{j3} &= \log\left(\frac{P_{B_3|YB_1B_2}(0|y_j,\hat{b}_{j1}\hat{b}_{j2})}{P_{B_3|YB_1B_2}(1|y_j,\hat{b}_{j1}\hat{b}_{j2})}\right) \Longrightarrow \hat{b}_{j3}
\end{aligned}
$$

Polar Codes
Decoding

- SC Decoding: Each sub-code is decoded one after the other, using the previously calculated hard estimates for the conditioning:

$$
\begin{aligned}
\ell_{j1} &= \log\left(\frac{P_{B_1|Y}(0|y_j)}{P_{B_1|Y}(1|y_j)}\right) \Longrightarrow \hat{b}_{j1} \\
\ell_{j2} &= \log\left(\frac{P_{B_2|YB_1}(0|y_j,\hat{b}_{j1})}{P_{B_2|YB_1}(1|y_j,\hat{b}_{j1})}\right) \Longrightarrow \hat{b}_{j2} \\
\ell_{j3} &= \log\left(\frac{P_{B_3|YB_1B_2}(0|y_j,\hat{b}_{j1}\hat{b}_{j2})}{P_{B_3|YB_1B_2}(1|y_j,\hat{b}_{j1}\hat{b}_{j2})}\right) \Longrightarrow \hat{b}_{j3}
\end{aligned}
$$

- SCL Decoding: As before, but from bit-level 2 on, we additionally pass a list to the next level. CRC is evaluated over all bit-levels.

Outline

Case Study
Summary

We investigate the following codes for a target SE of 3 bpcu with 64-QAM and $n = 32$ channel uses (i.e., $n_c = 192$ bits):

Case Study
Summary

We investigate the following codes for a target SE of 3 bpcu with 64-QAM and $n = 32$ channel uses (i.e., $n_{\mathsf{c}} = 192$ bits):

- Binary LDPC:
  - 5G BG2, $R_c = 1/2$ (uniform)
  - 5G BG1, $R_c = 3/4$ (PAS with CCDM and SMDM)

Case Study
Summary

We investigate the following codes for a target SE of 3 bpcu with 64-QAM and $n = 32$ channel uses (i.e., $n_c = 192$ bits):

- Binary LDPC:
  - 5G BG2, $R_c = 1/2$ (uniform)
  - 5G BG1, $R_c = 3/4$ (PAS with CCDM and SMDM)
- Binary Polar:
  - $R_c = 1/2, L = 32$, 8-CRC (uniform)
  - $R_c = 3/4, L = 32$, 4-CRC (PAS with CCDM and type check)

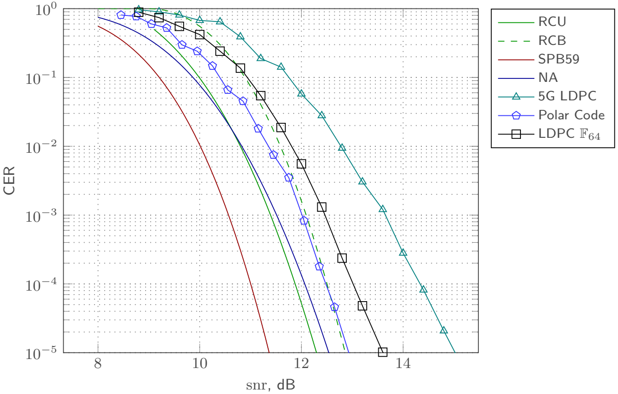
Case Study
Summary

We investigate the following codes for a target SE of 3 bpcu with 64-QAM and $n = 32$ channel uses (i.e., $n_\mathsf{c} = 192$ bits):

- Binary LDPC:
  - 5G BG2, $R_c = 1/2$ (uniform)
  - 5G BG1, $R_c = 3/4$ (PAS with CCDM and SMDM)
- Binary Polar:
  - $R_\mathsf{c} = 1/2, L = 32$, 8-CRC (uniform)
  - $R_\mathsf{c} = 3/4, L = 32$, 4-CRC (PAS with CCDM and type check)
- Non-Binary LDPC
  - Ultra-Sparse GF(64), $R_\mathsf{c} = 1/2$ (uniform)
  - Ultra-Sparse GF(256), $R_\mathsf{c} = 2/3$ (PAS with CCDM and SMDM)

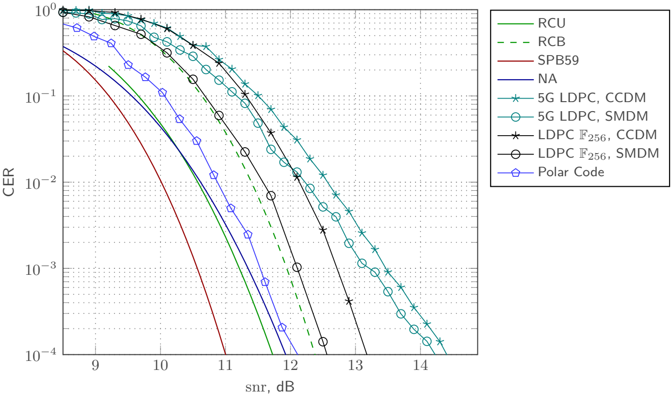
Case Study

64-QAM uniform, $n = 32$, $\eta = 3$ bpcu



Case Study

64-QAM PAS, $n = 32$, $\eta = 3\,\text{bpcu}$



Outline

Conclusion

- Higher-order modulation requires to be aware of the decoding metrics, their (sub-)optimality and the incurred losses.

Conclusion

- Higher-order modulation requires to be aware of the decoding metrics, their (sub-)optimality and the incurred losses.
- Probabilistic Amplitude Shaping (PAS) closes the gap to the Shannon limit, allows flexibility in SE with one modcod and has virtually no BMD loss.

Conclusion

- Higher-order modulation requires to be aware of the decoding metrics, their (sub-)optimality and the incurred losses.
- Probabilistic Amplitude Shaping (PAS) closes the gap to the Shannon limit, allows flexibility in SE with one modcod and has virtually no BMD loss.
- Tailored design for higher-order modulation is possible, but requires adjusted tools.

Conclusion

- Higher-order modulation requires to be aware of the decoding metrics, their (sub-)optimality and the incurred losses.
- Probabilistic Amplitude Shaping (PAS) closes the gap to the Shannon limit, allows flexibility in SE with one modcod and has virtually no BMD loss.
- Tailored design for higher-order modulation is possible, but requires adjusted tools.
- Polar Codes show very good performance in the short blocklength regime also for higher-order modulation.