Ingegneria delle Telecomunicazioni

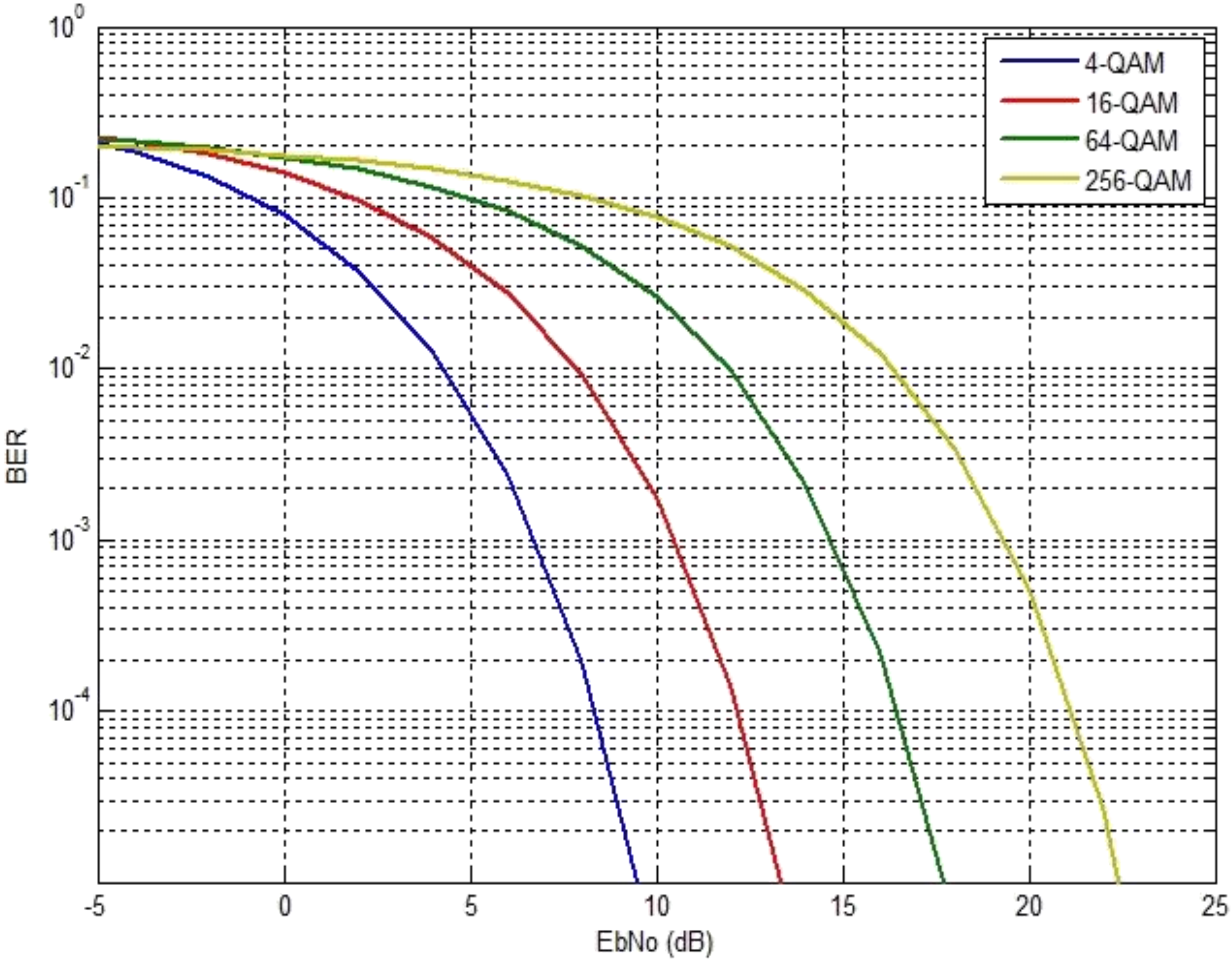
Information Theory @ Digital Communications

Turbo,LDPC, Polar Codes, ACM

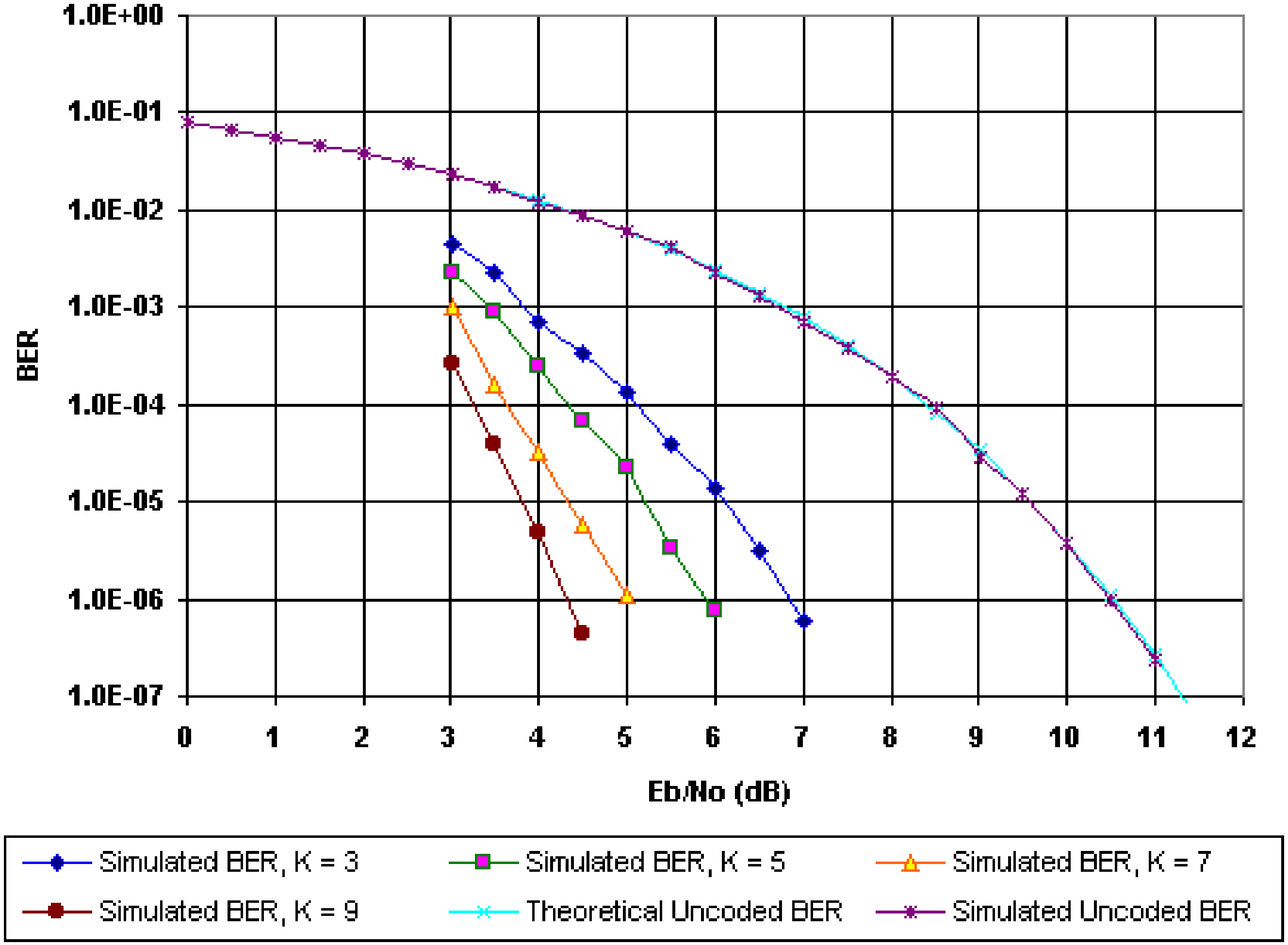
Marco Luise
marco.luise@unipi.it

Dip. Ingegneria dell'Informazione, Univ. Pisa, Pisa, Italy

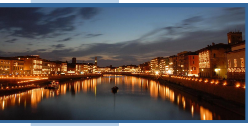
Conventional Modulations

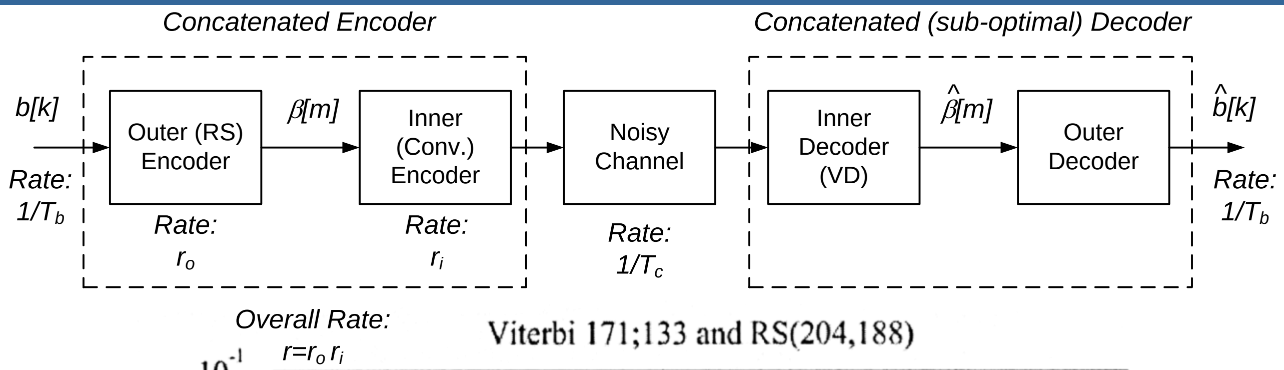


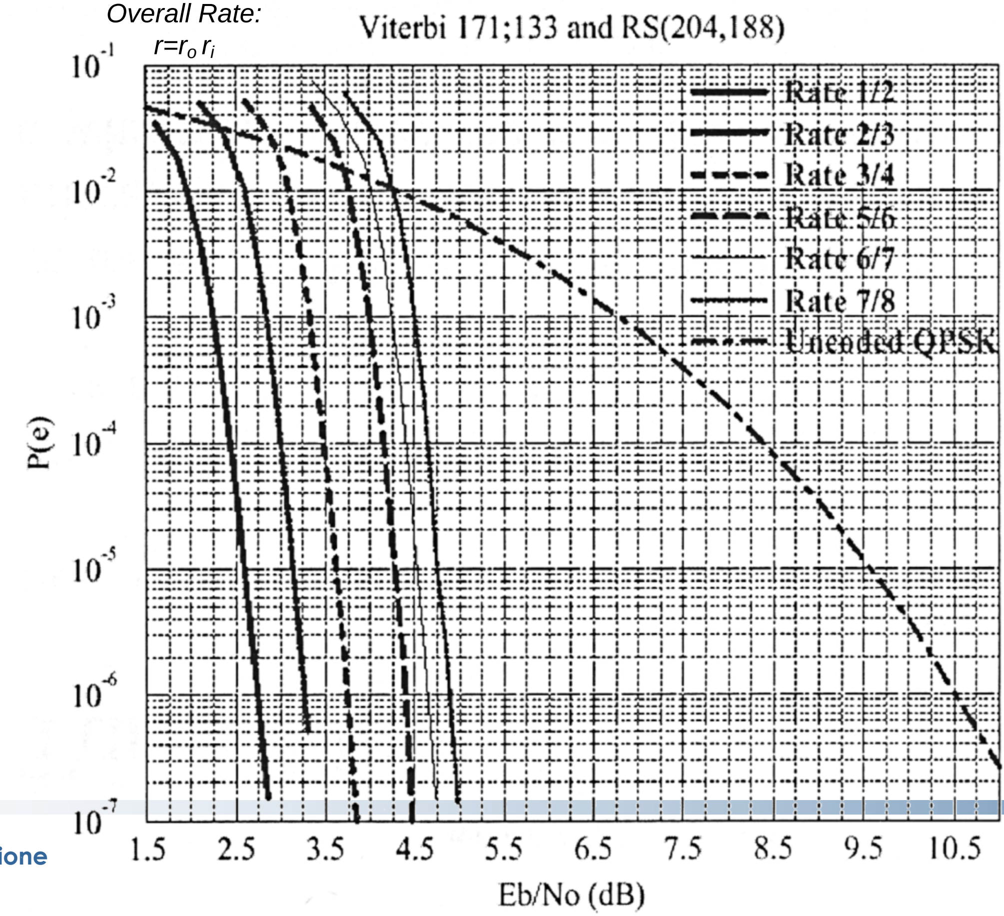
BER of *r*=1/2 Best Conv Codes



R-S & Convolutional Concatenated Coding



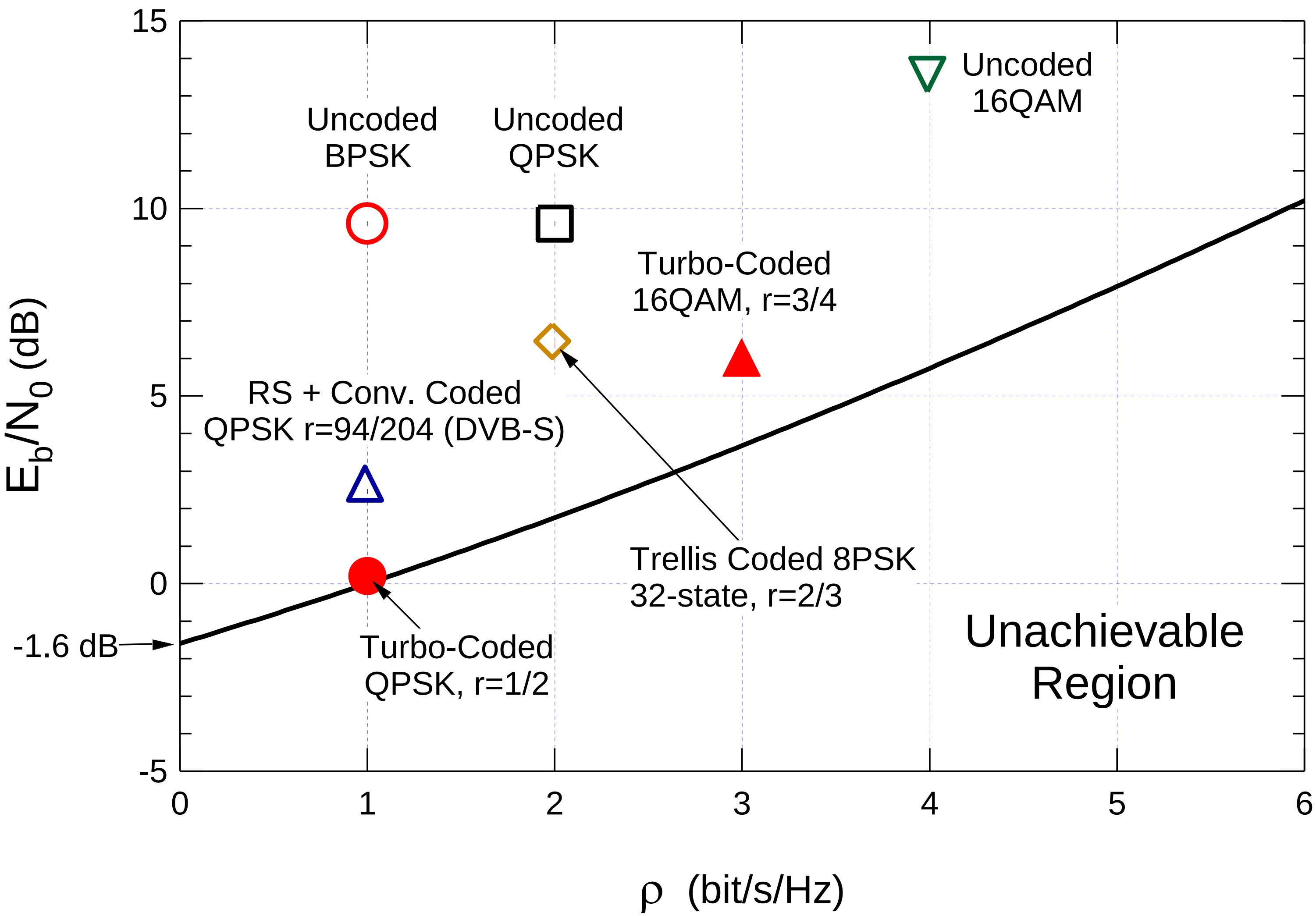




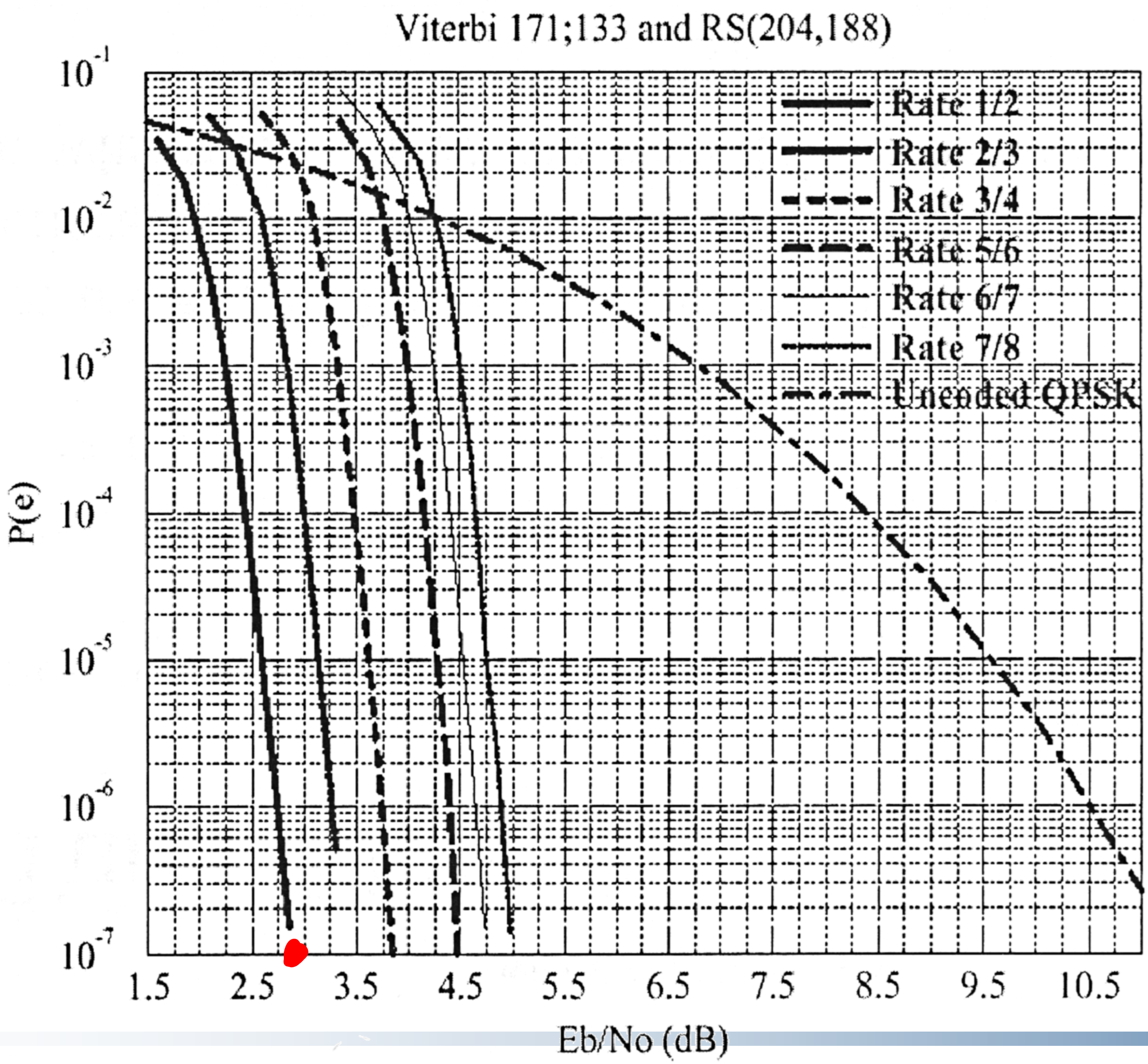
Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

Marco Luise

MOD/COD on the Shannon Plane



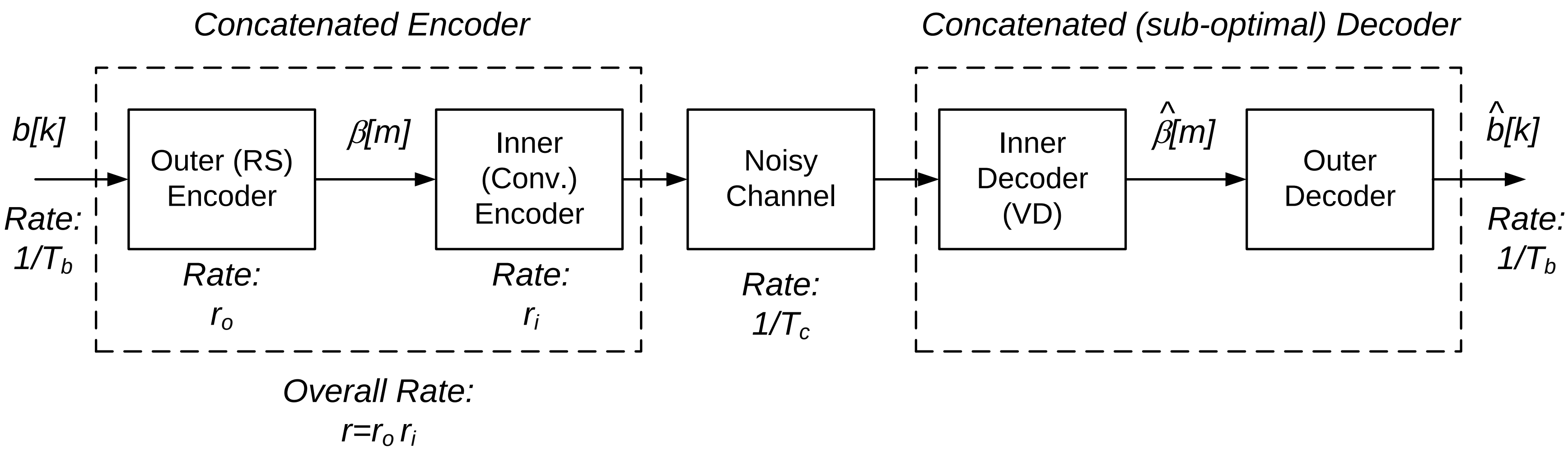
R-S & Convolutional Concatenated Coding



Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

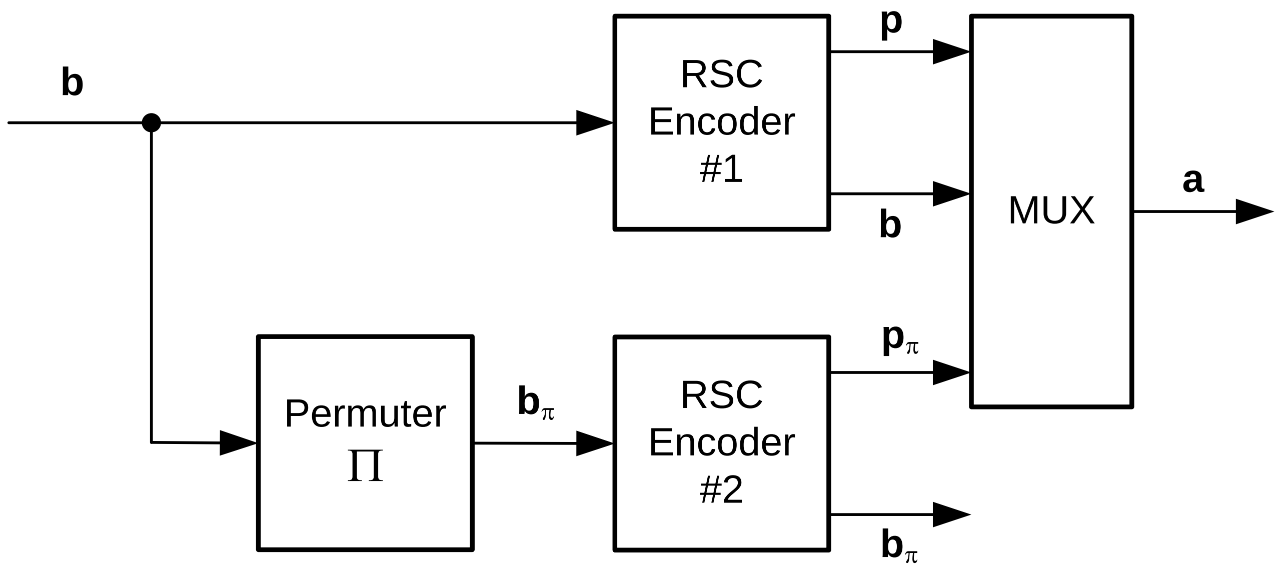
Marco Luise

Sub-optimal Concatenated Decoding

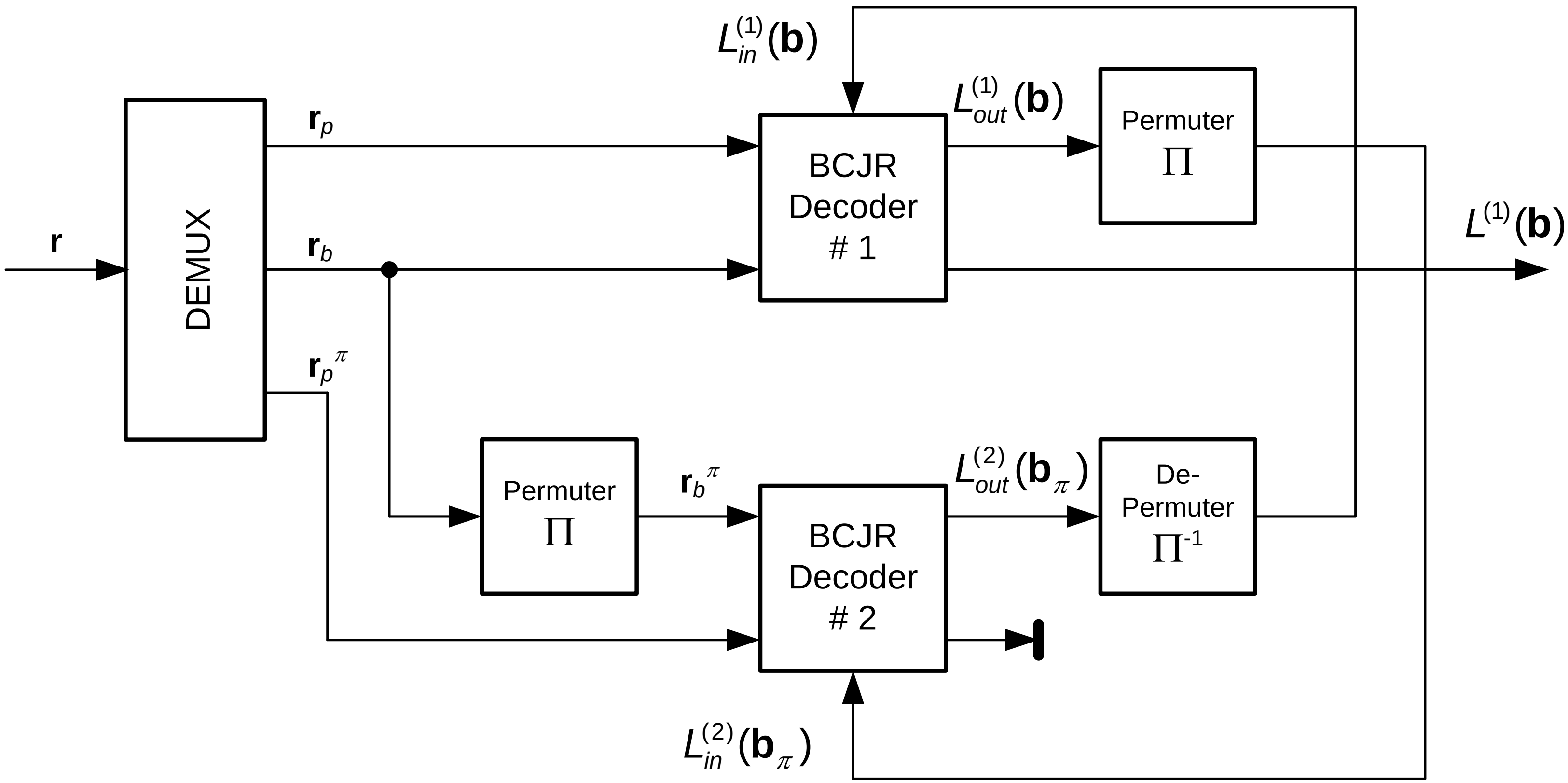


Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy
Marco Luise

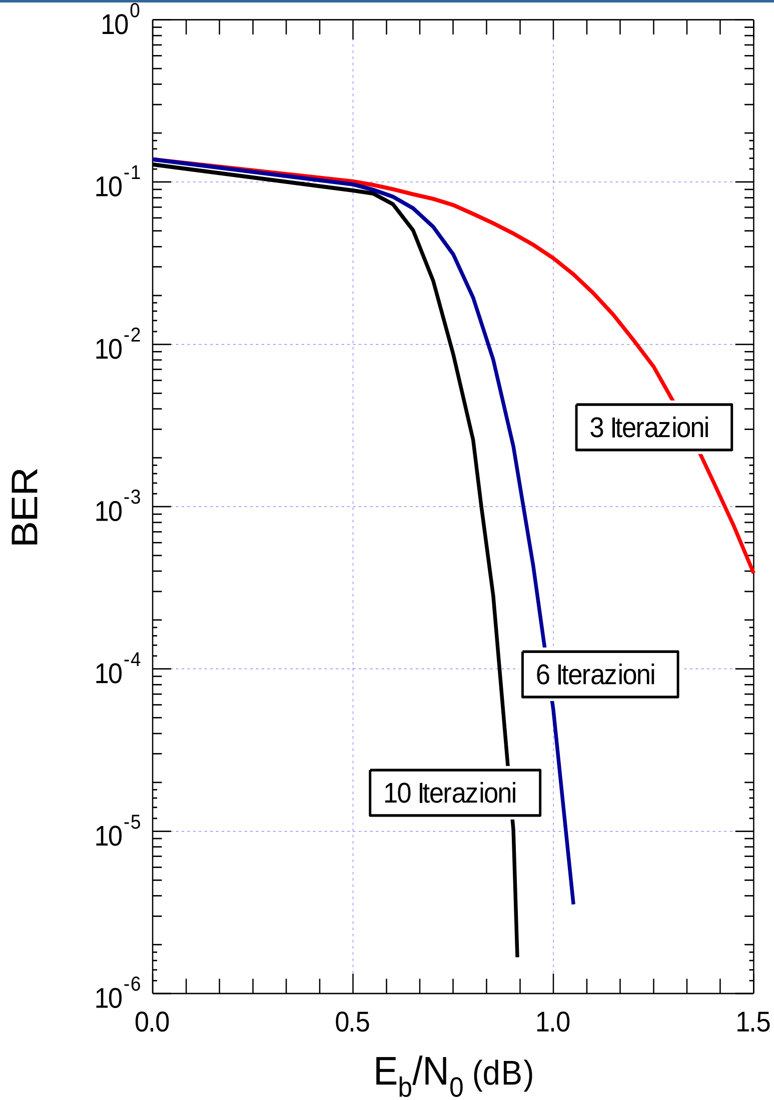
Rate-1/3 Turbo PCCC Encoder



Rate-1/3 Turbo PCCC Decoder

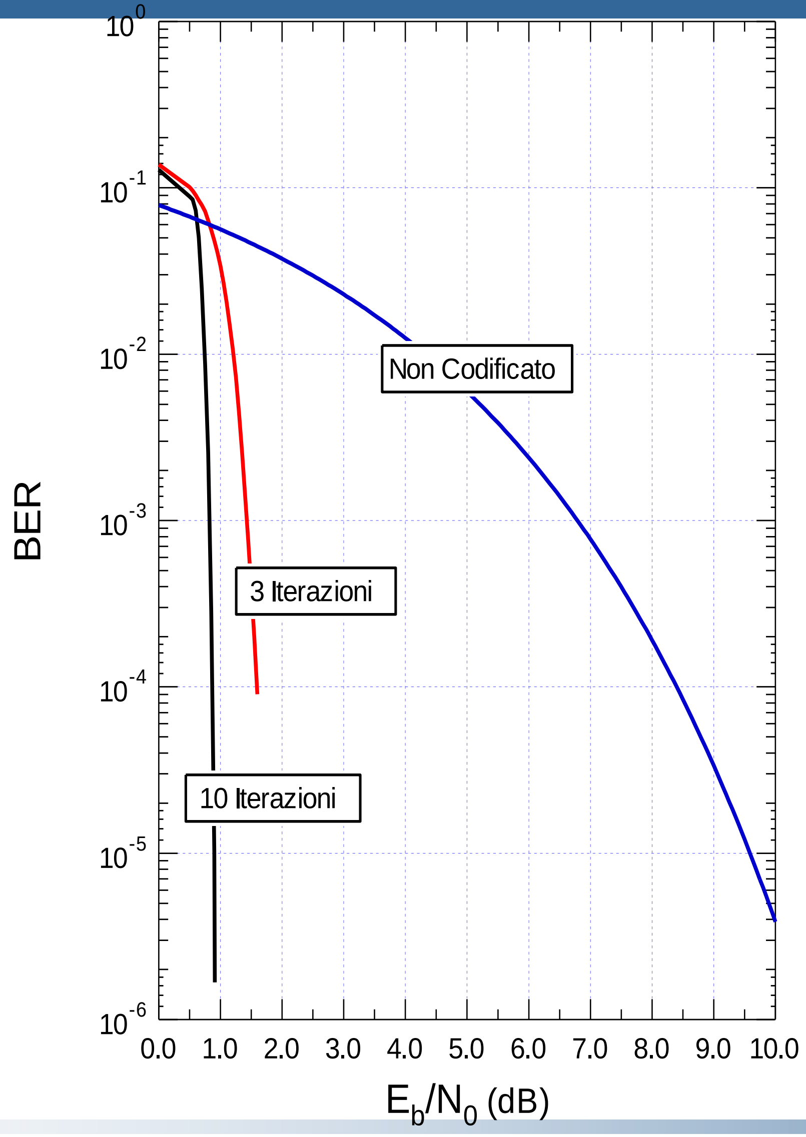


Turbo BER 1/2



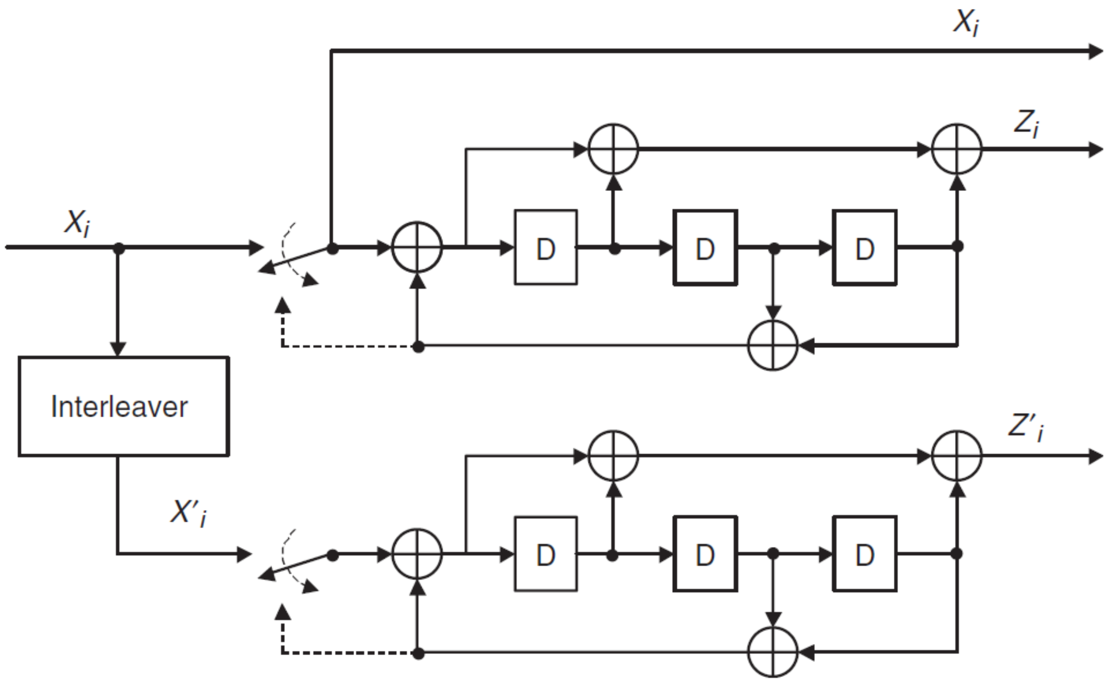
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
BER
0.0
0.5
1.0
1.5
E_b/N_0 (dB)
3 Iterazioni
6 Iterazioni
10 Iterazioni

Turbo BER 2/2



BER
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
E_b/N_0 (dB)
Non Codificato
3 Iterazioni
10 Iterazioni

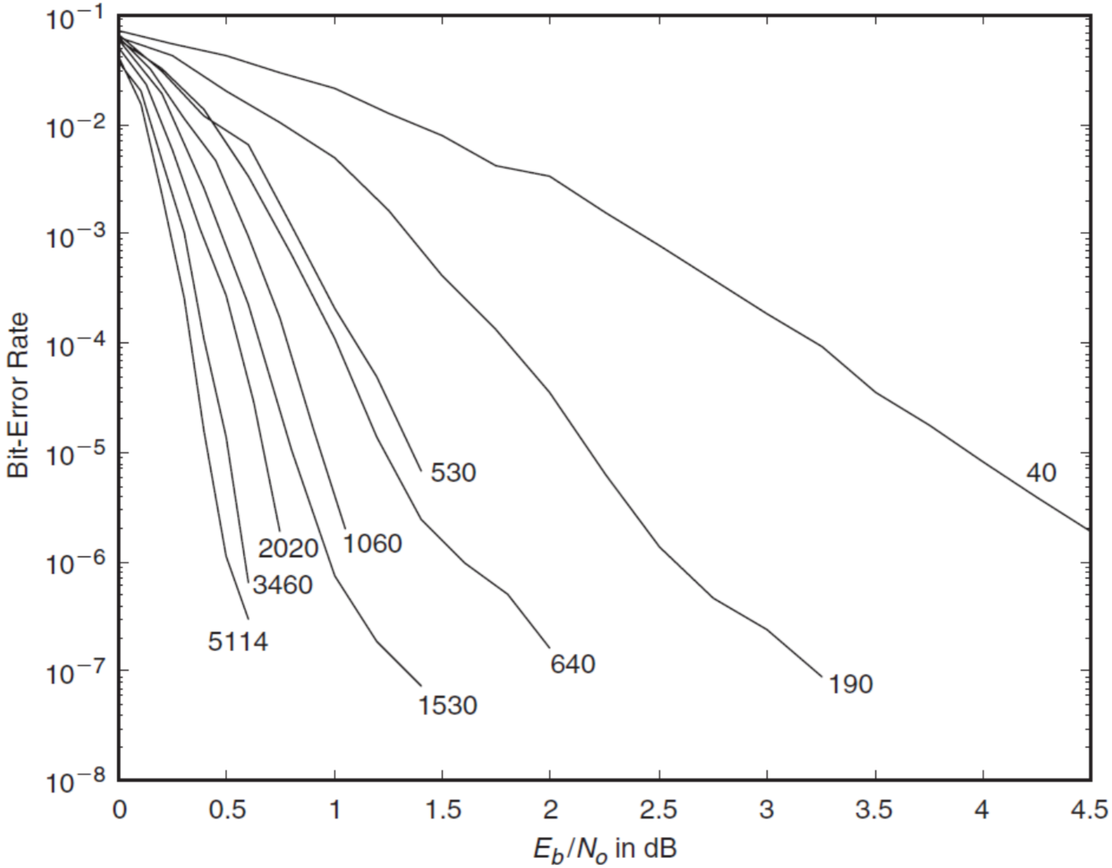
The UMTS (3gpp) rate-1/3 Turbo Encoder



Performance is strongly dependent on the interleaver (packet) size (40 to 5114 bits)

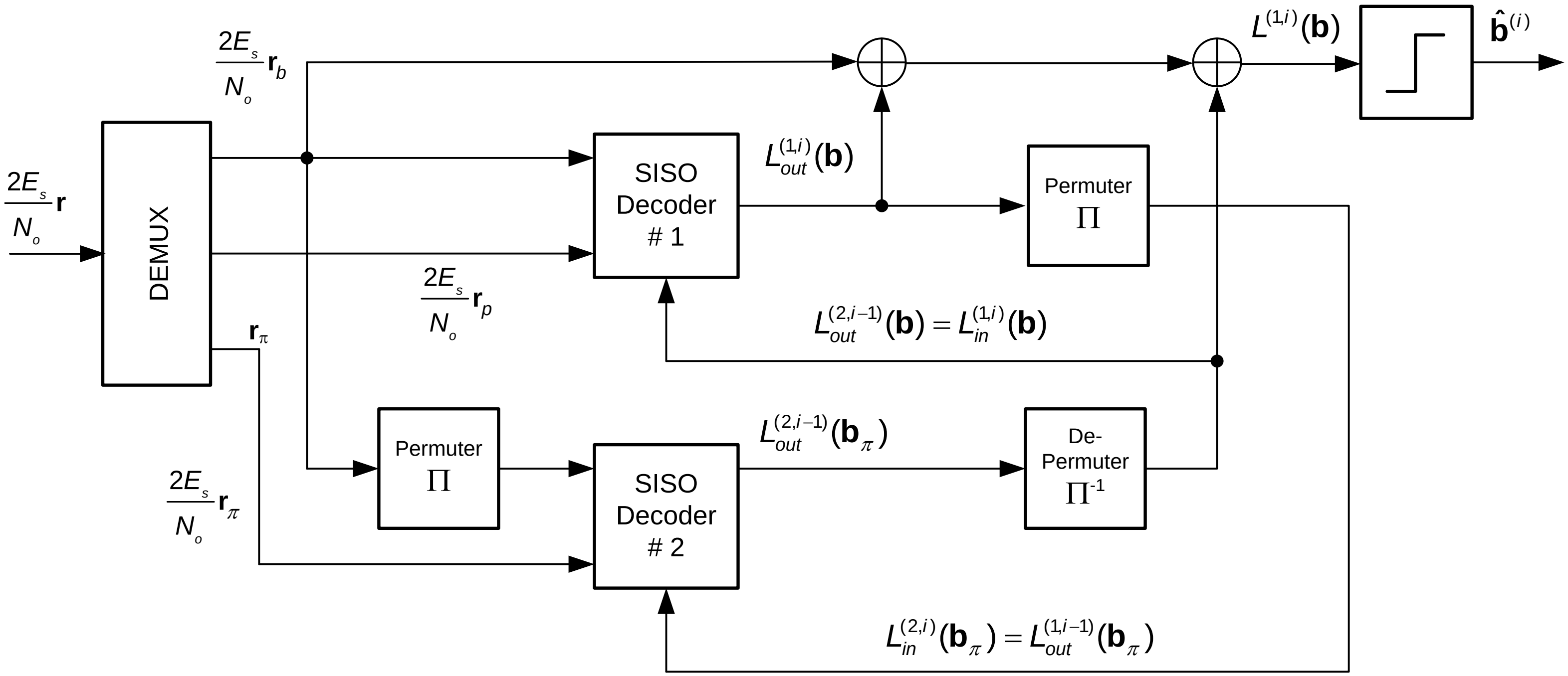
Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

The UMTS (3gpp) rate-1/3 Turbo Encoder



Performance is strongly dependent on the interleaver (packet) size (40 to 5114 bits)

Rate-1/3 Turbo PCCC Decoder with Extrinsic Inf. Flow



Hamming (7,4) Code

$$\mathbf{G} = \left[\underbrace{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{\mathbf{I}_4} \; \underbrace{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{matrix}}_{\mathbf{P}_{4\times 3}}\right]$$

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

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

LDPC

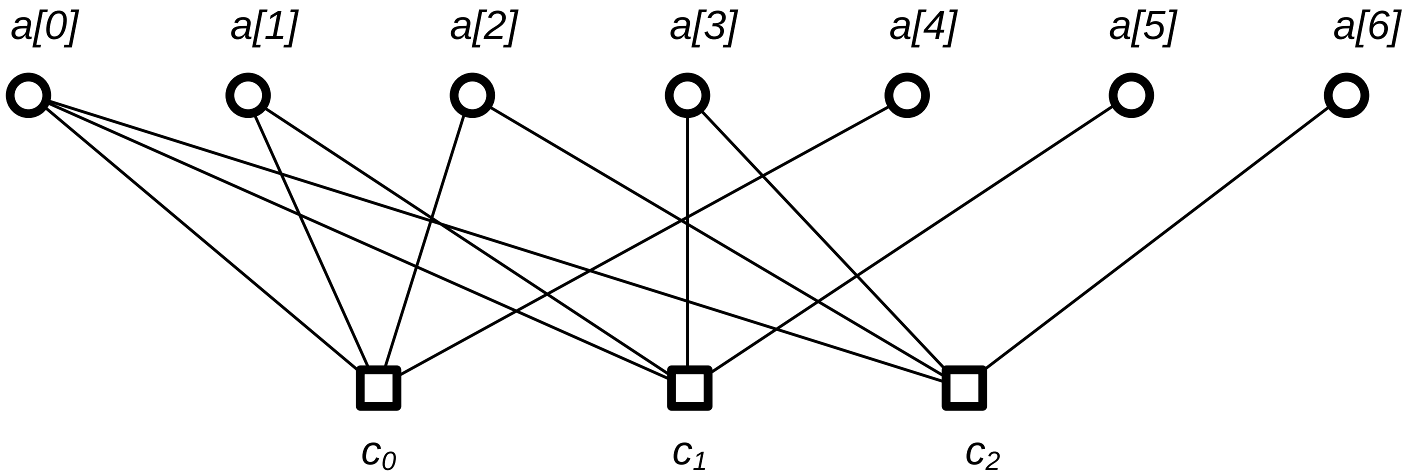
$$
\mathbf{H} = \left[\begin{array}{c}
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 \\
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 \\
0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 \\
0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 \\
1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 \\
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 \\
1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 \\
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 \\
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 \\
0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 \\
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 \\
0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 \\
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 \\
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 \\
1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 \\
0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
\end{array}\right]
$$

(50,25) code

$r=1/2$

Tanner Graph

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



Should not bear *short cycles*

Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

Coding/Decoding Complexity

- ***Regular*** **code: same number of 1's in each row of H , otherwise** ***Irregular*** **code**
  - Irregular as a rule better than regular, harder to encode

- **Decoding complexity**
  - For a regular code, it is proportional to the number of edges in the Tanner graph, equal to $w_R N$ – LINEAR with N thanks to the low-density property of **H** ($w_R \ll N$)
  - Same remark for irregular codes, where the proportionality coefficient of N is $E\{w_R\}$

- **Coding complexity**
  - Surprisingly an issue: if **H** is random Low-density, then after Gaussian elimination **G** is NOT low-density in the parity-generating section **P** and the complexity of encoding is *quadratic* ($N(N-K)$) – needs special attention

Received Block

$$\mathbf{a} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

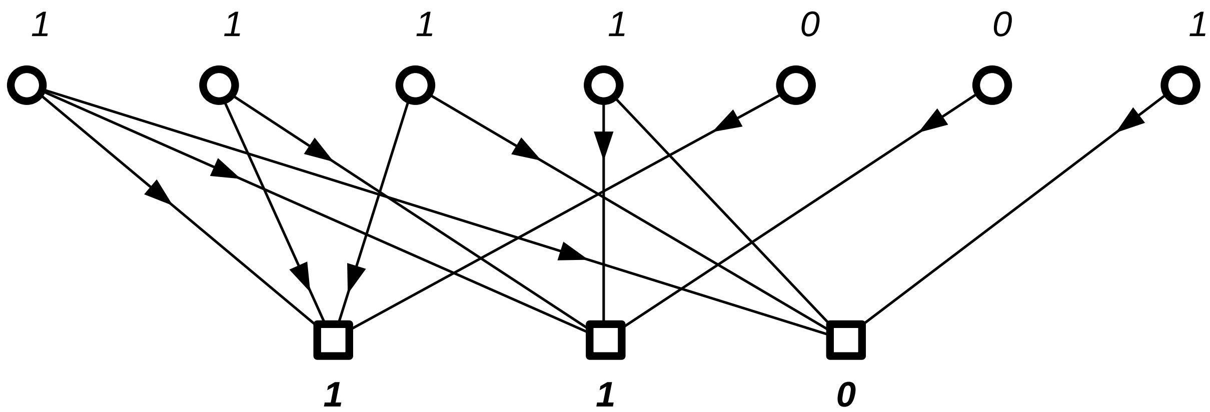
$$\mathbf{H}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} 1 & \mathbf{1} & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{e} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Bit flipping: 1st semi-iteration

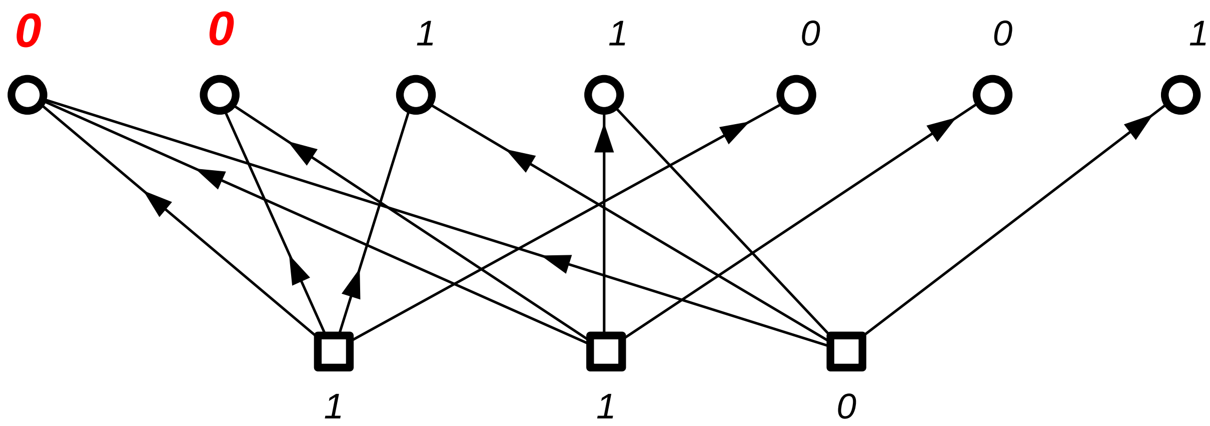
$$\mathbf{d} = \begin{bmatrix} 1 & \mathbf{1} & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$



Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

Bit flipping: 2nd semi-iteration

$$\mathbf{d} = \begin{bmatrix} 1 & \mathbf{1} & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$



Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

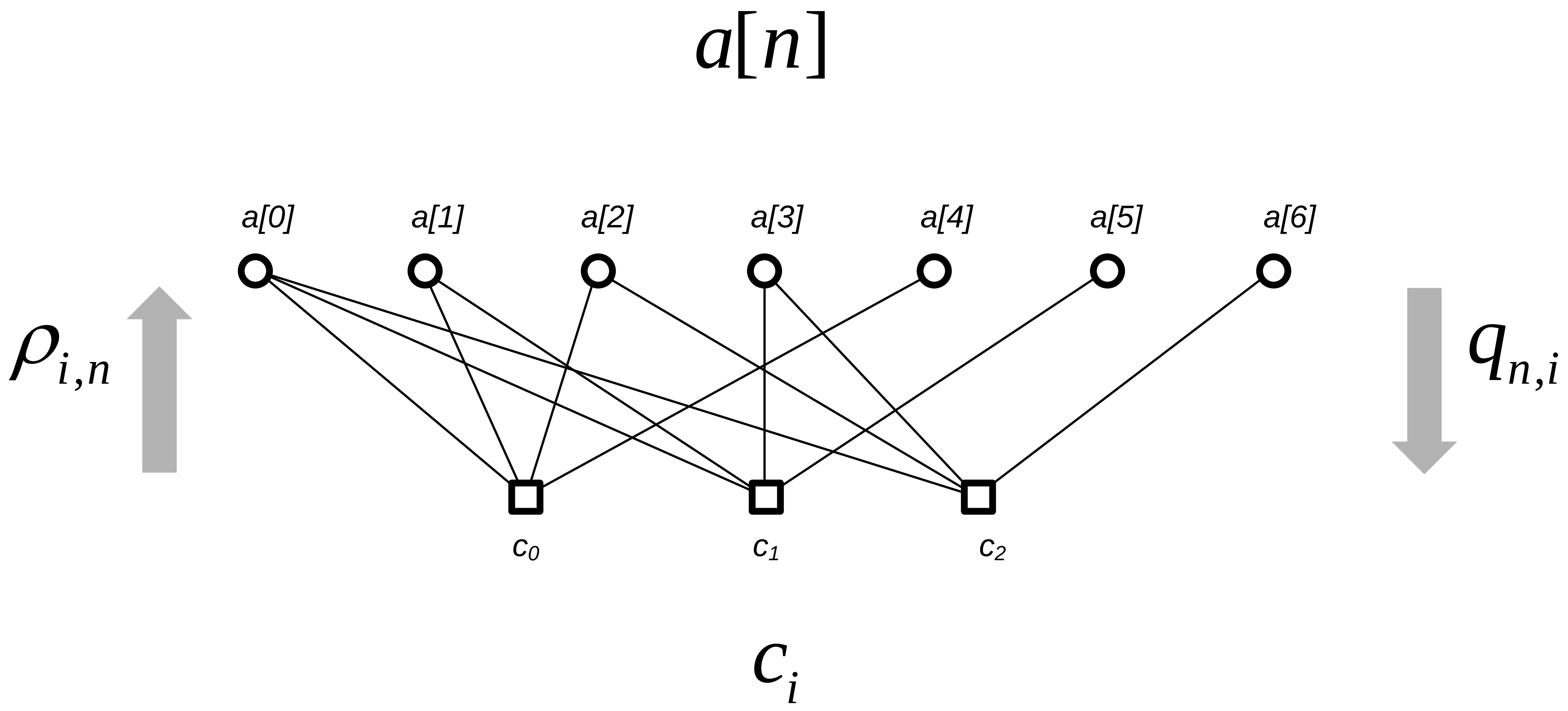
Probabilistic (Soft-Output) Decoding

$$\mathbf{r} = \mathbf{a}_{\pm} + \mathbf{w}$$

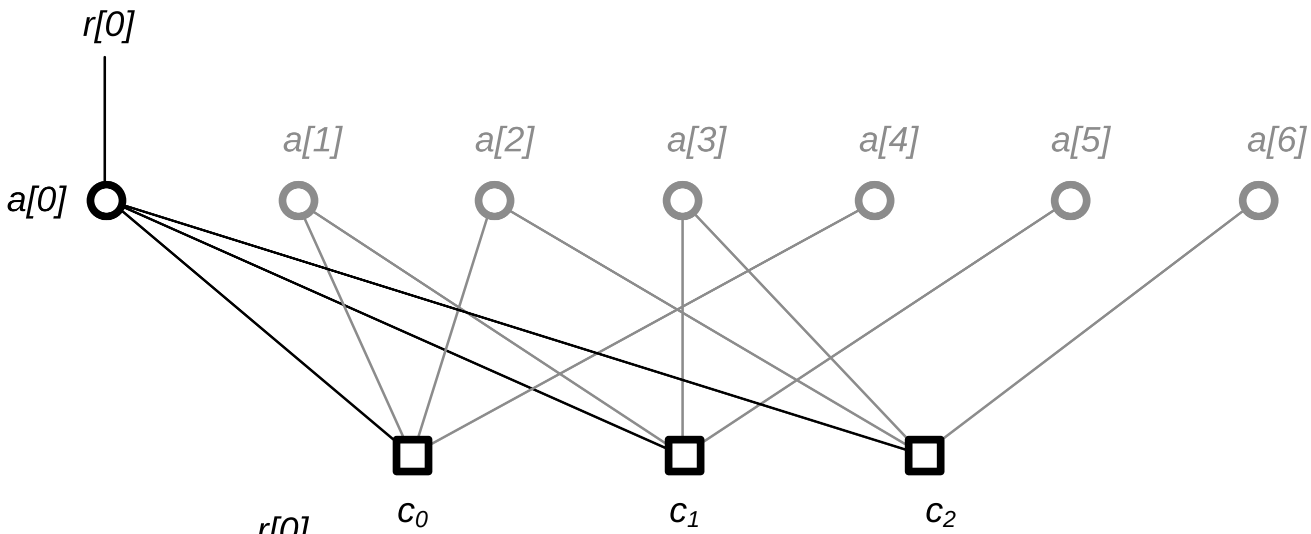
$$\Pr\left\{a[n] = x | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\right\} \quad x = 0, 1$$

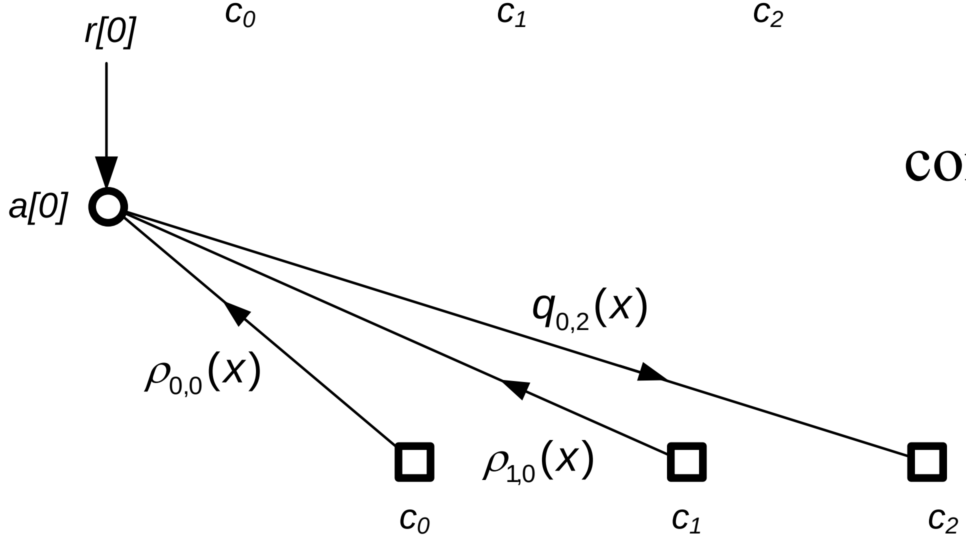
$$L(a[n]) \triangleq \ln\left(\frac{\Pr\left\{a[n] = 1 | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\right\}}{\Pr\left\{a[n] = 0 | \mathbf{r}, \{c_i = 0\}_{i \in C_{i,n}}\right\}}\right)$$

(Soft) Message-Passing Algorithm



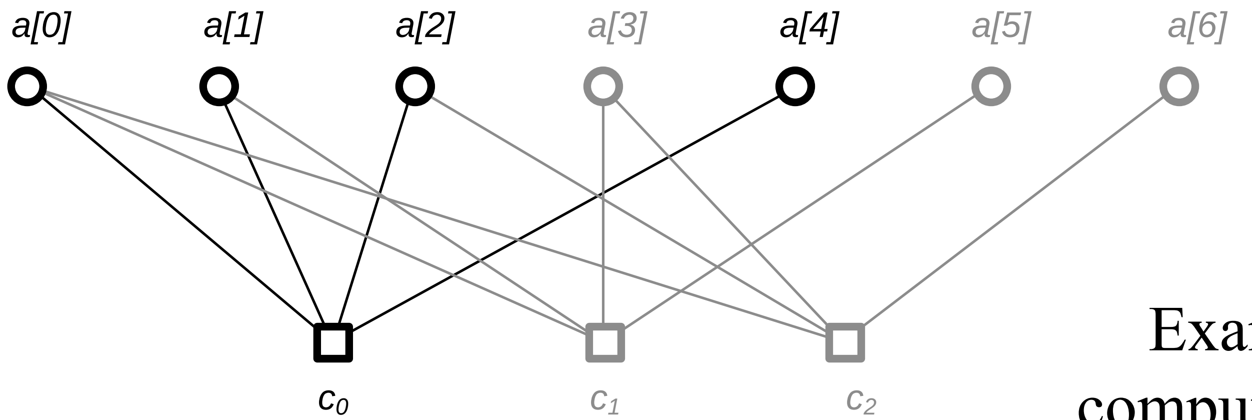
Message Passing: 1st semi-iteration



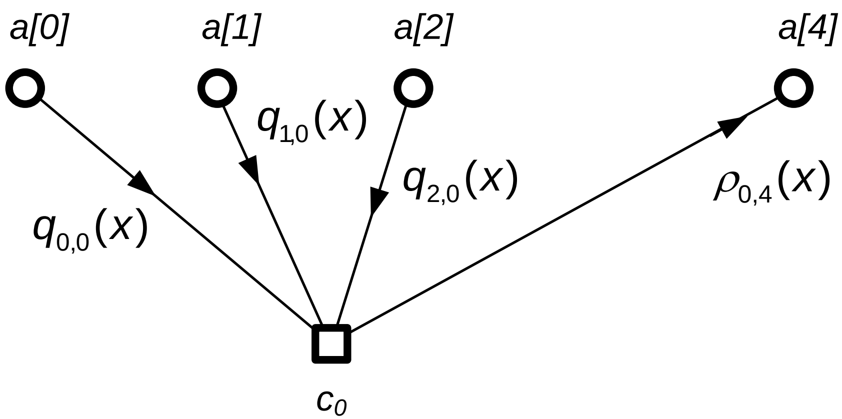


Example: computation of $q_{0,2}$

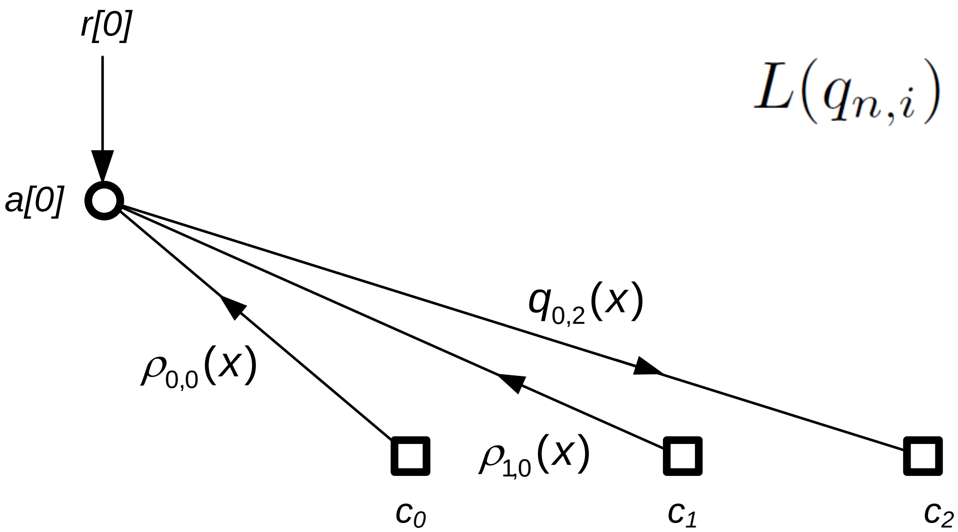
Message Passing: 2nd semi-iteration



Example: computation of $\rho_{0,4}$



Messages 1/2

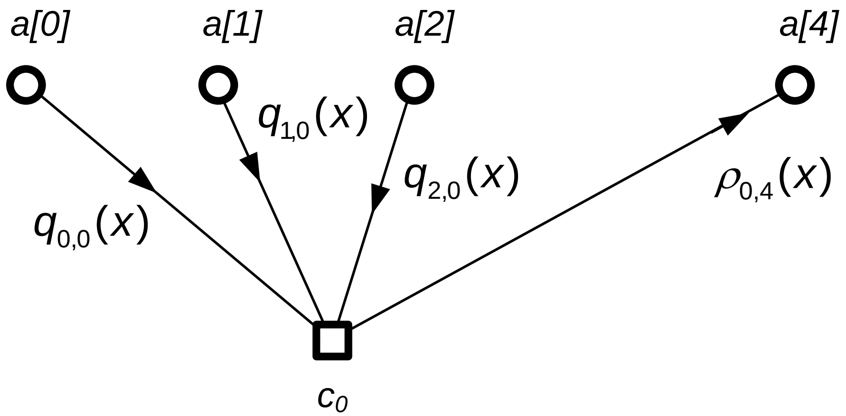


$$L(q_{n,i}) \triangleq \ln \frac{q_{n,i}(1)}{q_{n,i}(0)}$$

$$q_{n,i}\left(x\right) \triangleq \Pr\left\{ a[n] = x \,\middle|\, r[n], \{\rho_{i',n}\}_{i' \neq i} \right\} \quad x = 0, 1$$

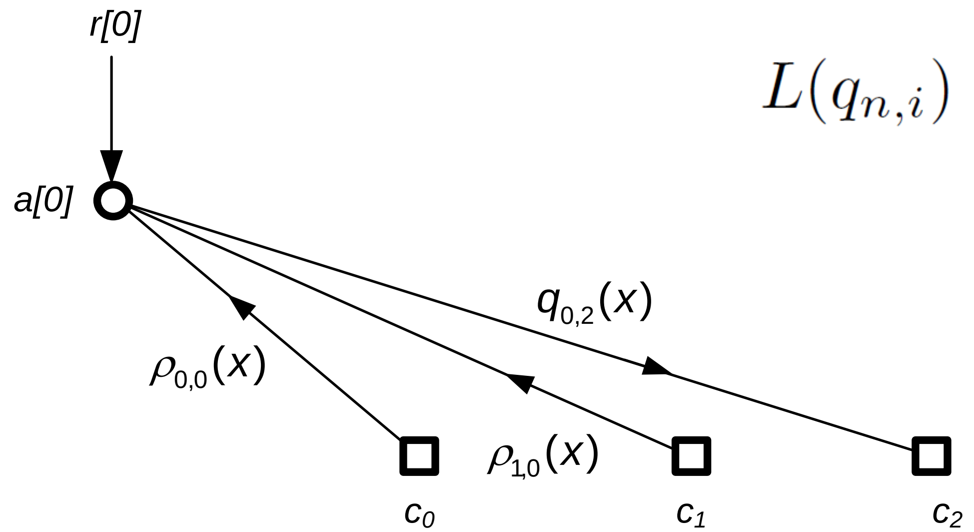
Messages 2/2

$$L(\rho_{i,n}) \triangleq \ln \frac{\rho_{i,n}(1)}{\rho_{i,n}(0)}$$



$$\rho_{i,n}(x) \triangleq \Pr\left\{ c_i = 0 \middle| a[n] = x, \{q_{n',i}\}_{n' \neq n} \right\} \quad x = 0, 1$$

Messages in the log-domain 1/2



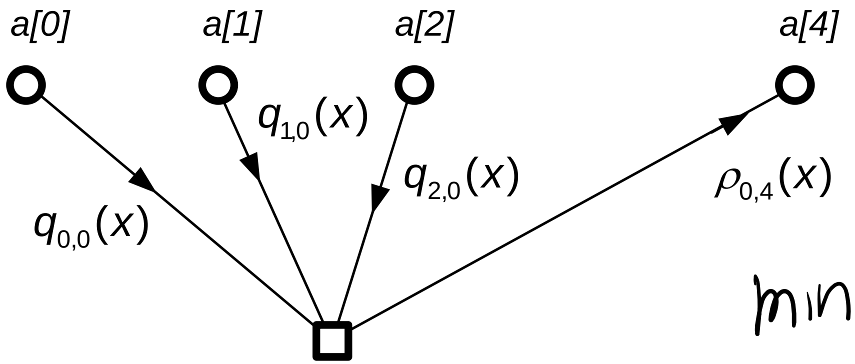
$$L(q_{n,i}) \triangleq \ln \frac{q_{n,i}(1)}{q_{n,i}(0)}$$

$$L^{(m+1)}(q_{n,i}) = \frac{4E_s}{N_0} r[n] + \sum_{i' \in C_{i',n}, i' \neq i} L^{(m)}(\rho_{i',n})$$

Messages in the log-domain 2/2

$\min(\min(a,b), c)$

$$L(\rho_{i,n}) \triangleq \ln \frac{\rho_{i,n}(1)}{\rho_{i,n}(0)}$$



$\min(a, b, c) =$

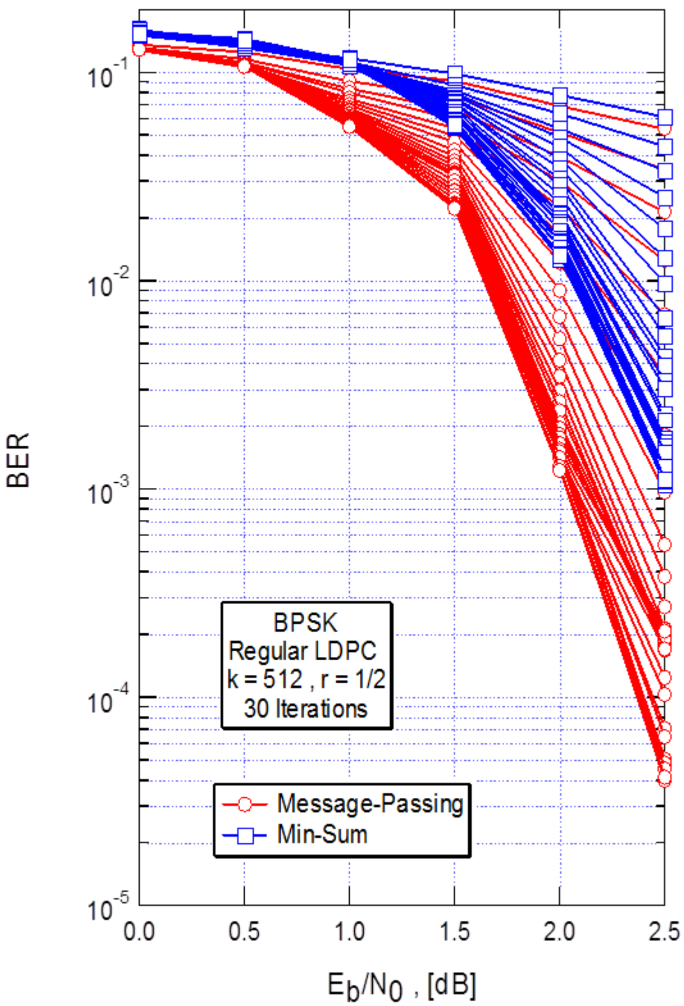
$$L^{(m+1)}(\rho_{i,n}) = -\left\{ \prod_{n' \in R_{i \setminus n}} \text{sgn}[-L^{(m)}(q_{n',i})] \right\} \cdot \min_{n' \in R_{i \setminus n}} |L^{(m)}(q_{n',i})|$$

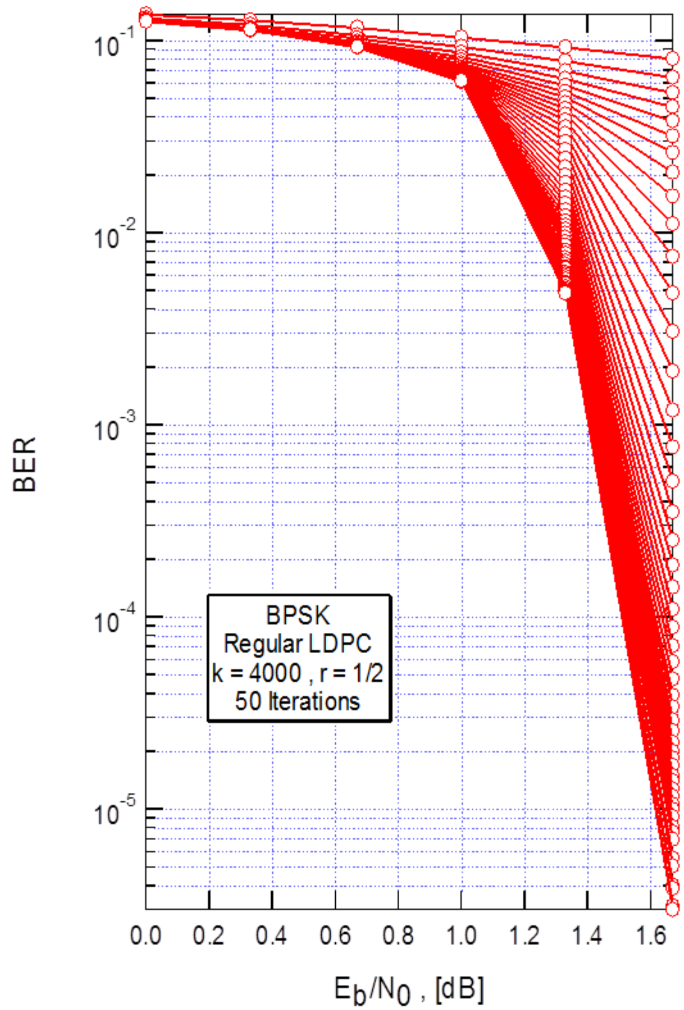
LAPPR of *a*[*n*] & Tentative Detection

$$L^{(m)}(a[n]) = \frac{4E_s}{N_0} r[n] + \sum_{i \in C_{i,n}} L^{(m)}(\rho_{n,i})$$

$$\hat{a}^{(m)}[n] = \text{sgn}\left[L^{(m)}(a[n]) \right]$$

LDPC Performance Sample





LDPC vs. Turbo 1/2

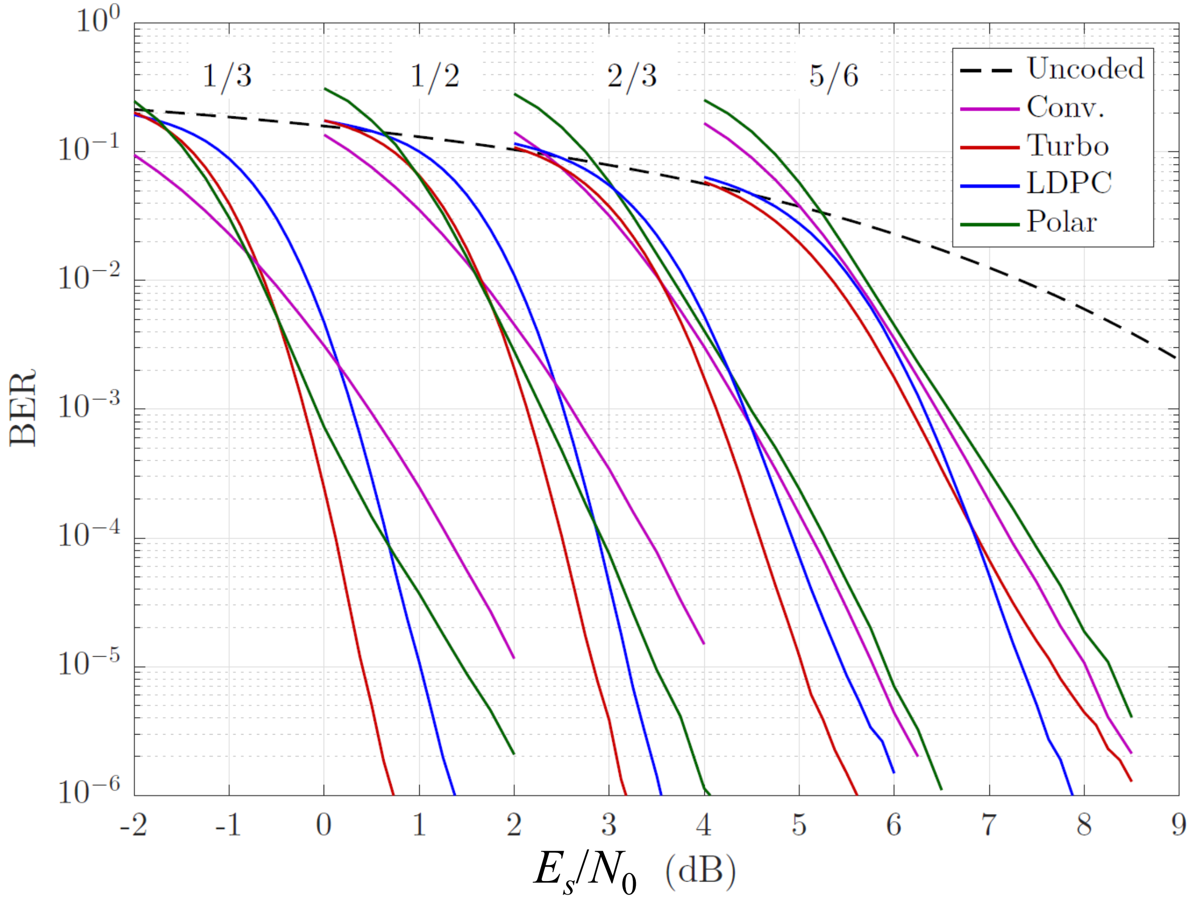


Fig. 9. BER comparison for different code rates, $K = 256$ (For LDPC, $K = 252$ for $R = 1/2$, and $1/3$, and $K = 260$ for $R = 5/6$.)

LDPC vs. Turbo 2/2

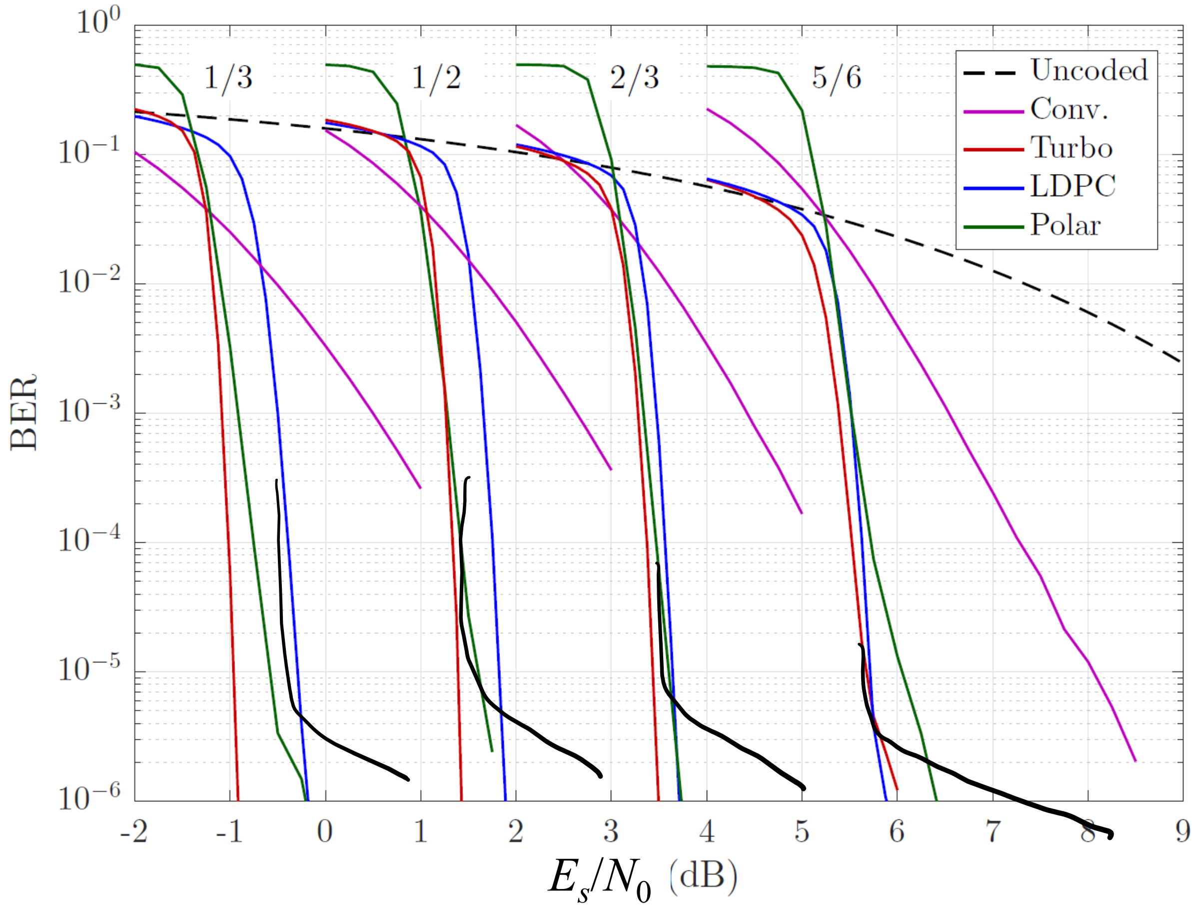
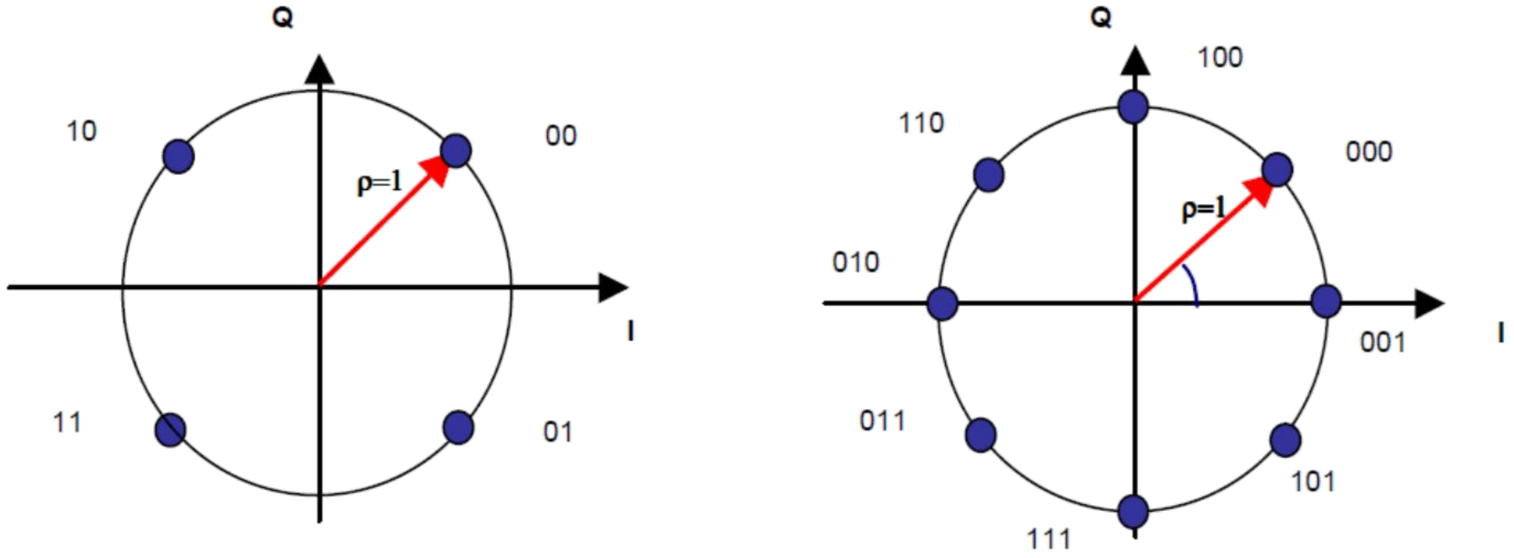


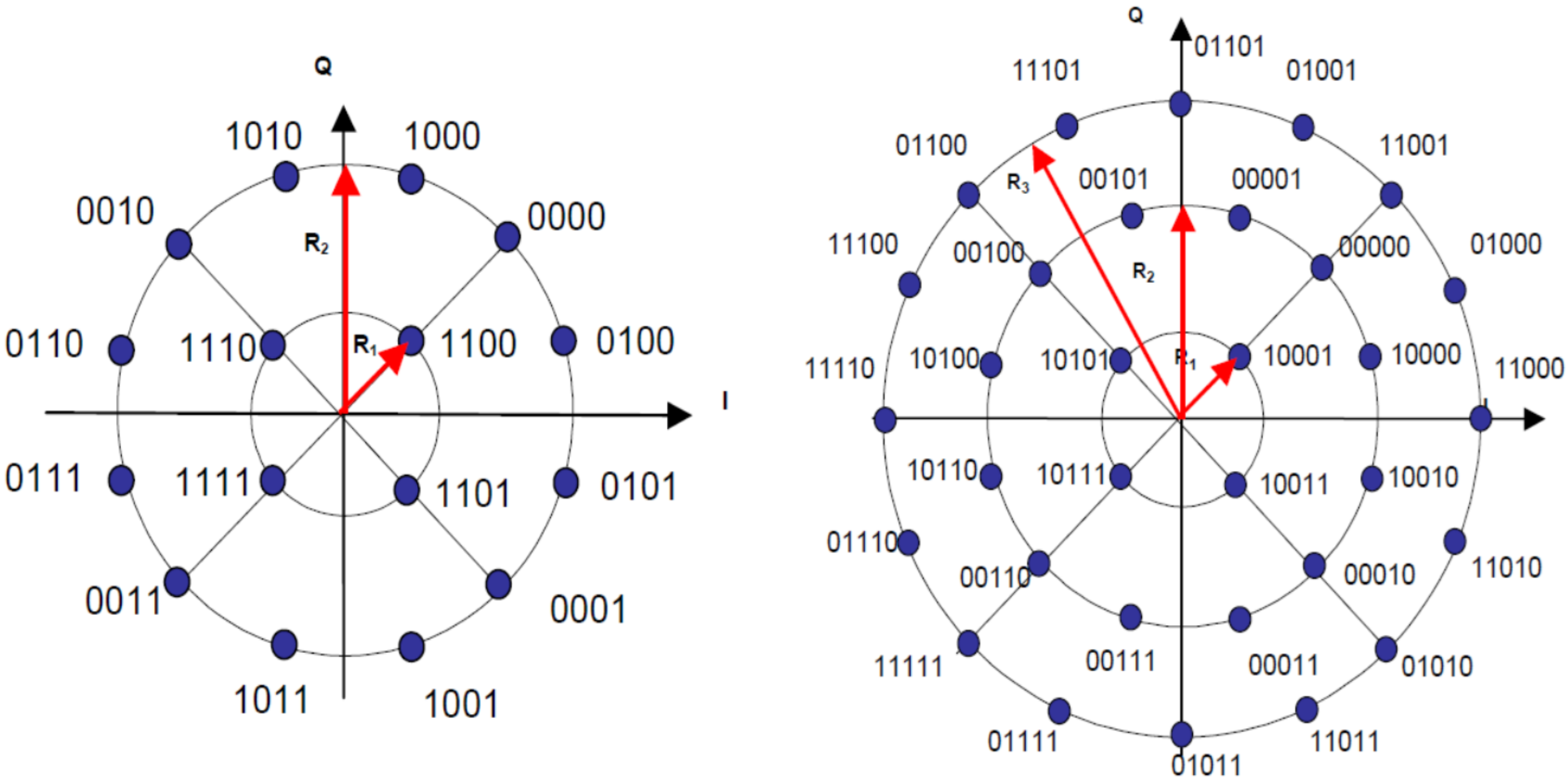
Fig. 14. BER comparison for different code rates, $K = 8192$ (For LDPC, $K = 8196$ for $R = 1/2$, and $1/3$, and $K = 8200$ for $R = 5/6$.)

DVB-S2 Constellations 1/2

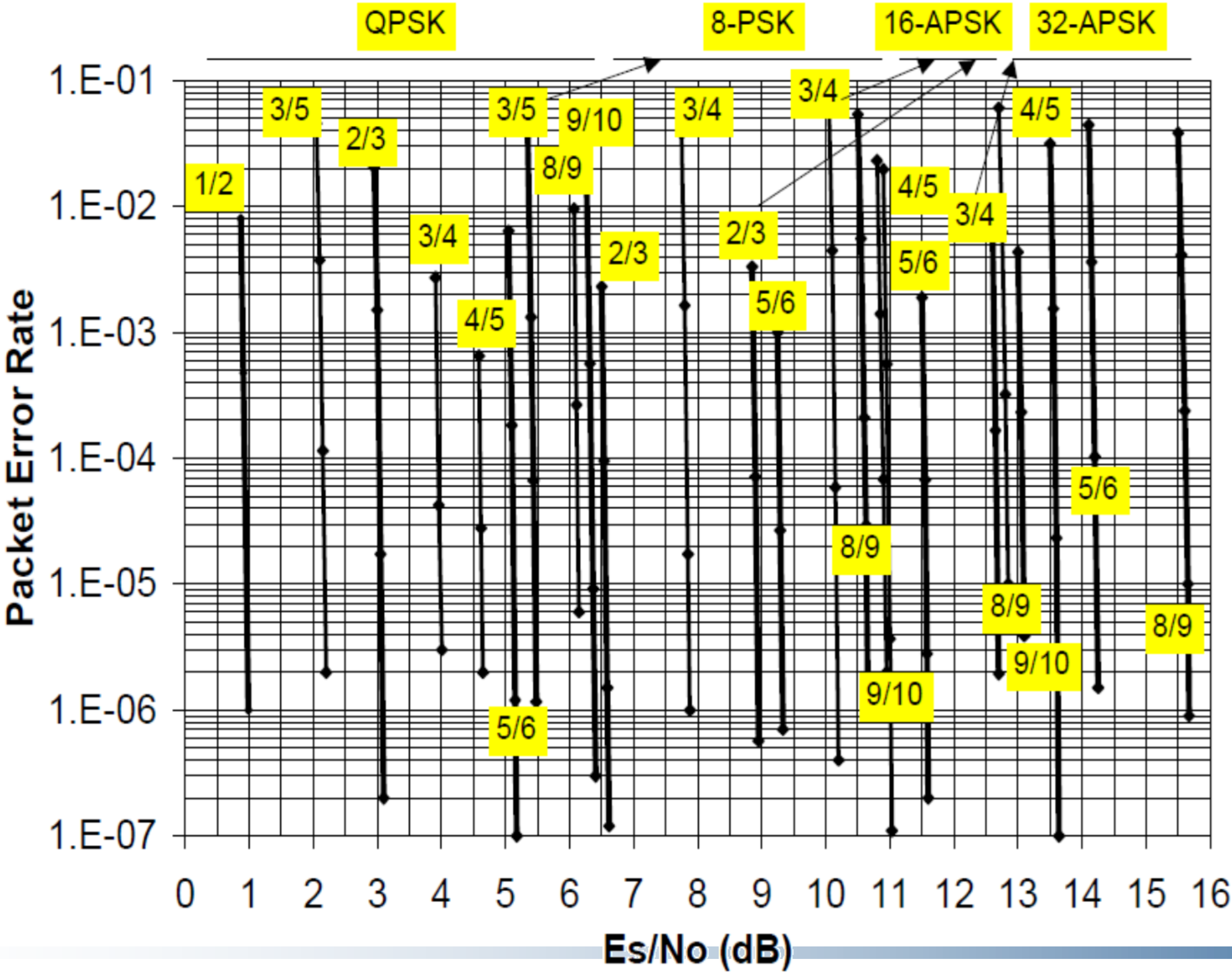


(a) QPSK (b) 8-PSK

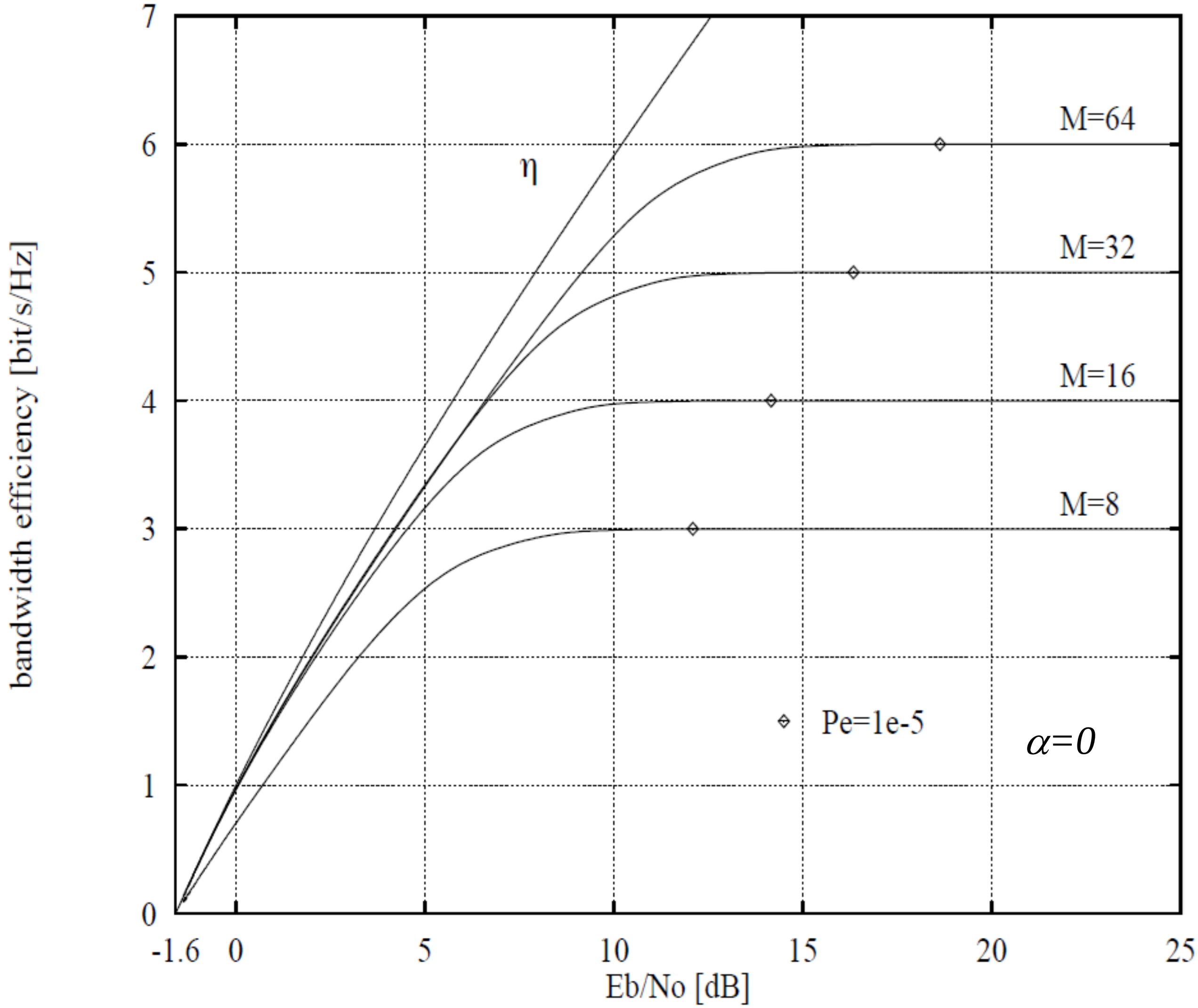
DVB-S2 Constellations 2/2



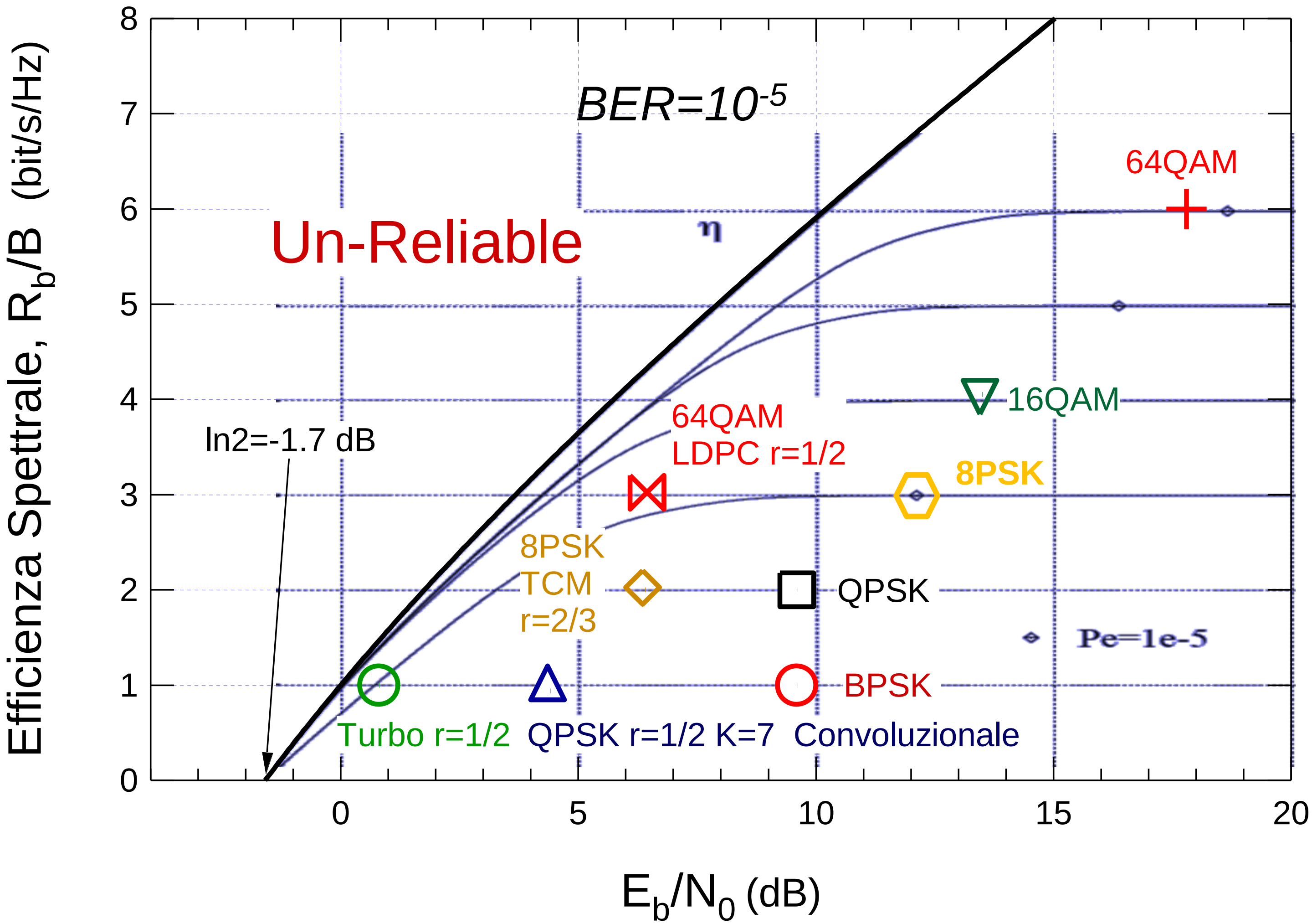
Adaptive Coding and Modulation (ACM): MOD/COD



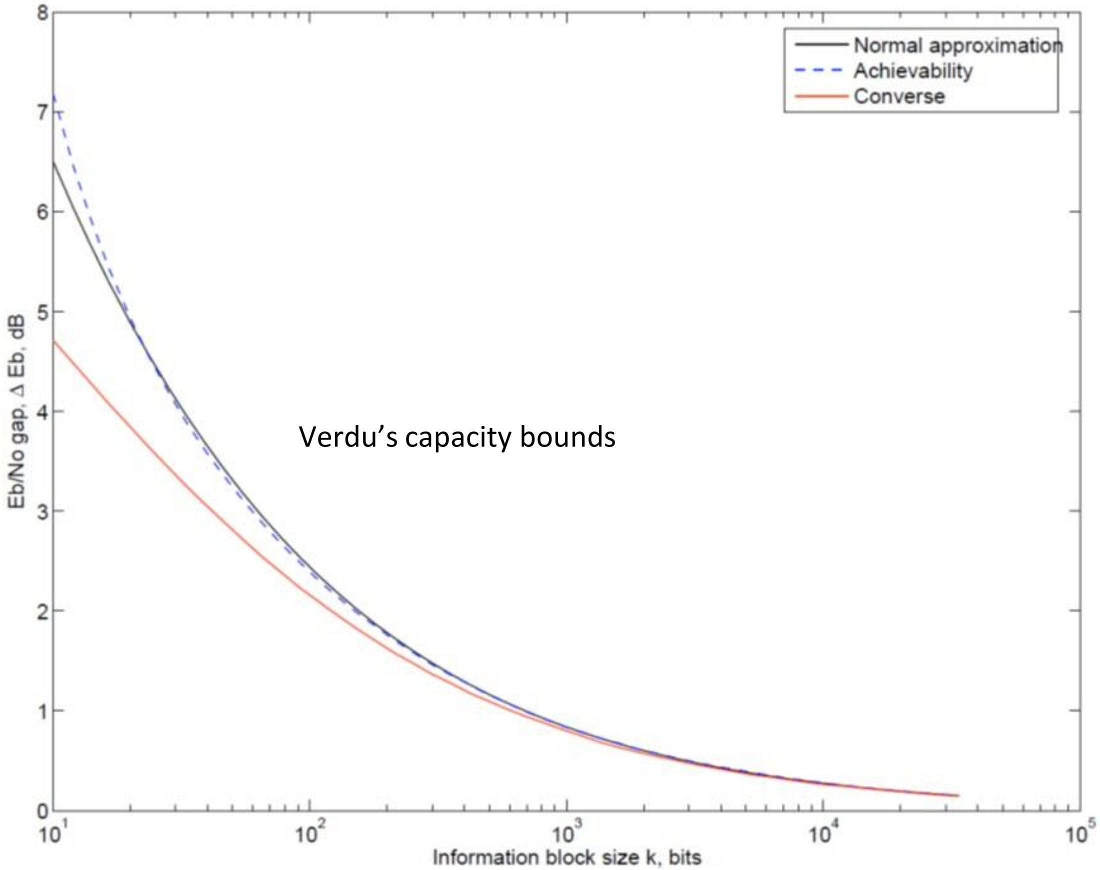
Digital-Input Soft-Output AWGN



MOD/COD Performance

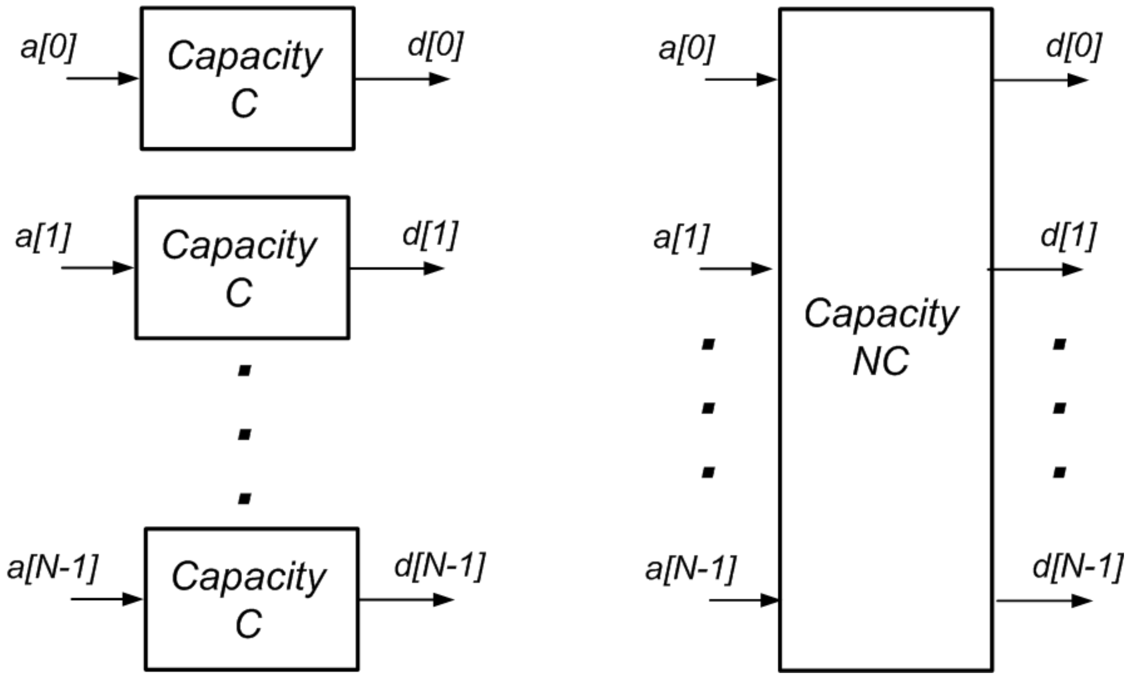


Capacità di Shannon Vincolata



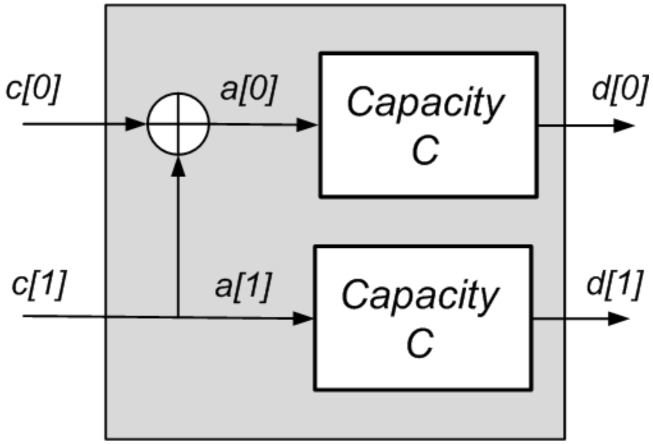
Verdu's capacity bounds

Canali Scalari → Canale Vettoriale



Pre-codifica Polare 2 x 2

Linear Transformation

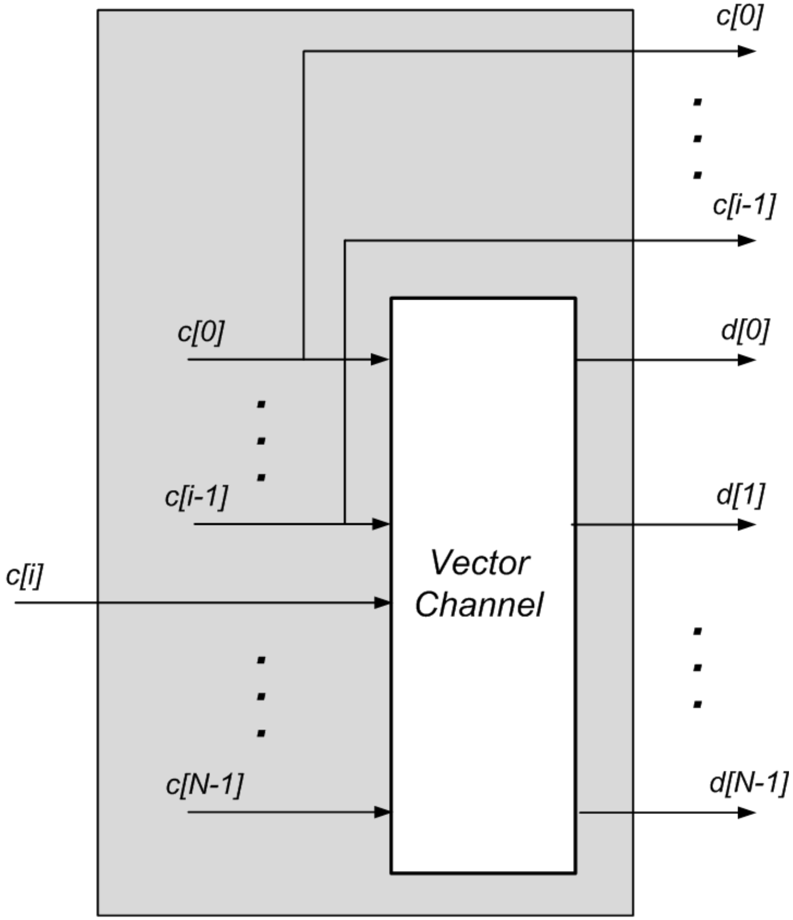


Channel #0: from *c*[0] to [*d*[0],*d*[1]]
Channel #1: from *c*[1] to [*c*[0],*d*[0],*d*[1]]

Starting from two scalar BEC(p_{CE}) channels:

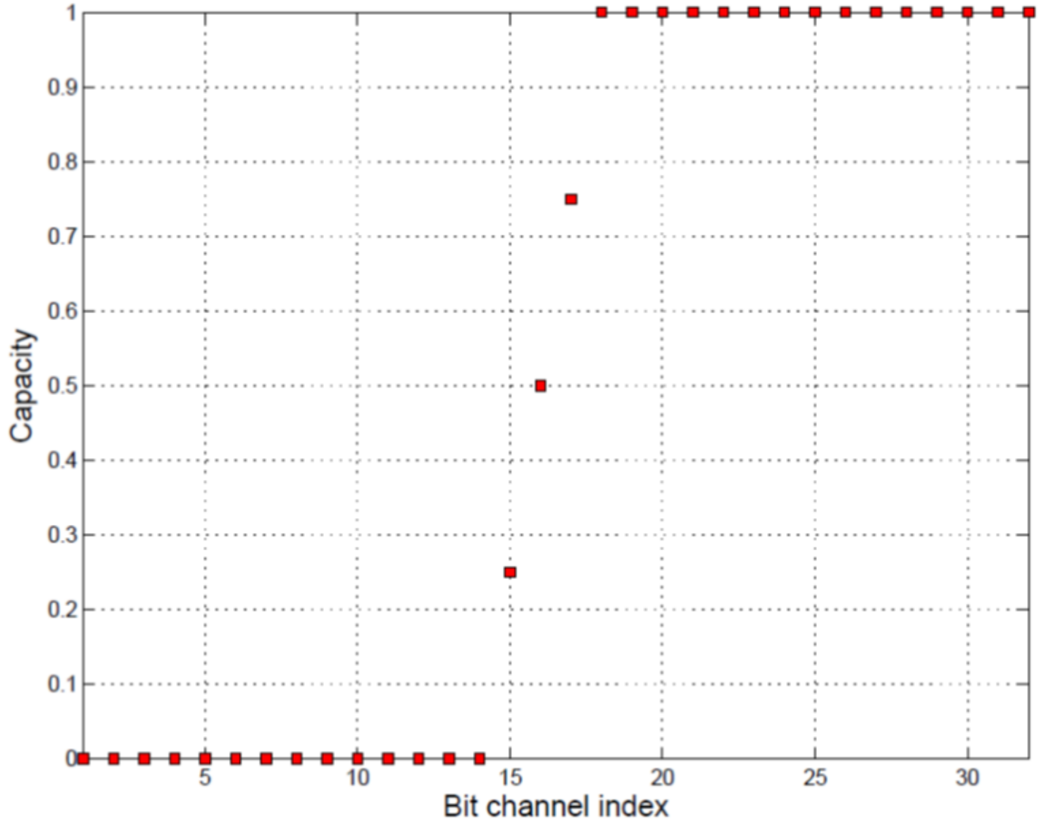
$$\mathcal{C}_0 = (1 - p_{CE})^2 < 1 - p_{CE} < 1 - p_{CE}^2 = \mathcal{C}_1$$

Aggregated Vector Channel



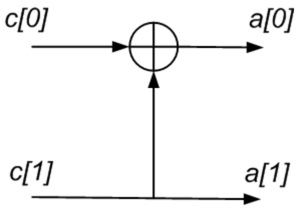
Vector Channel #i: from c[i] to [c[0],…, c[i-1];d[0],d[1],…, d[N-1]]

Random Polarizer (precoder) & Polarization Theorem

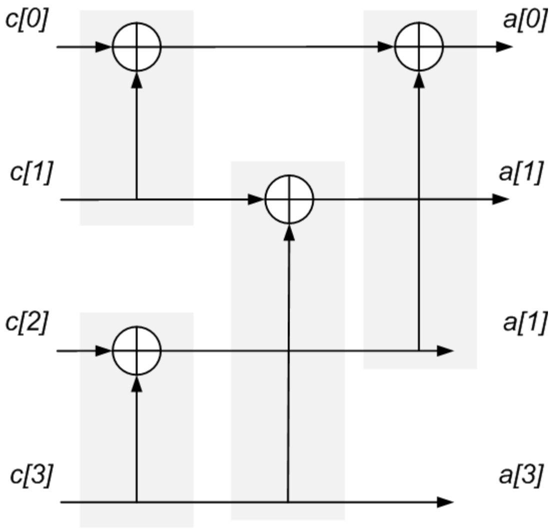


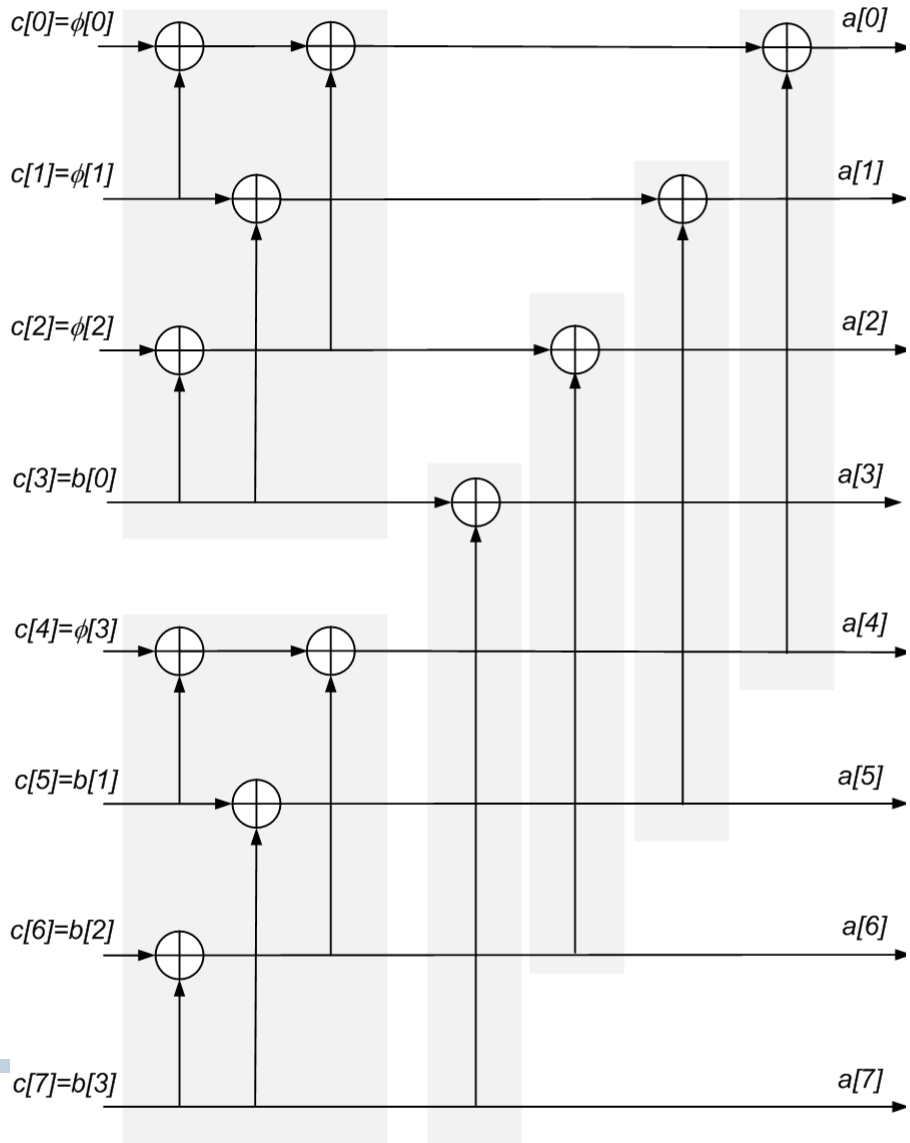
Dati N canali scalari indipendenti con capacità C ciascuno, se N è sufficientemente grande, per ogni ε (arbitrariamente piccolo), il numero di canali polarizzati ottenuti dagli N canali originali e la cui capacità è maggiore di 1-ε è pari a NC, e il numero di canali la cui capacità è più piccola di ε è $N - NC$

Order-Recursive Polar Precoder



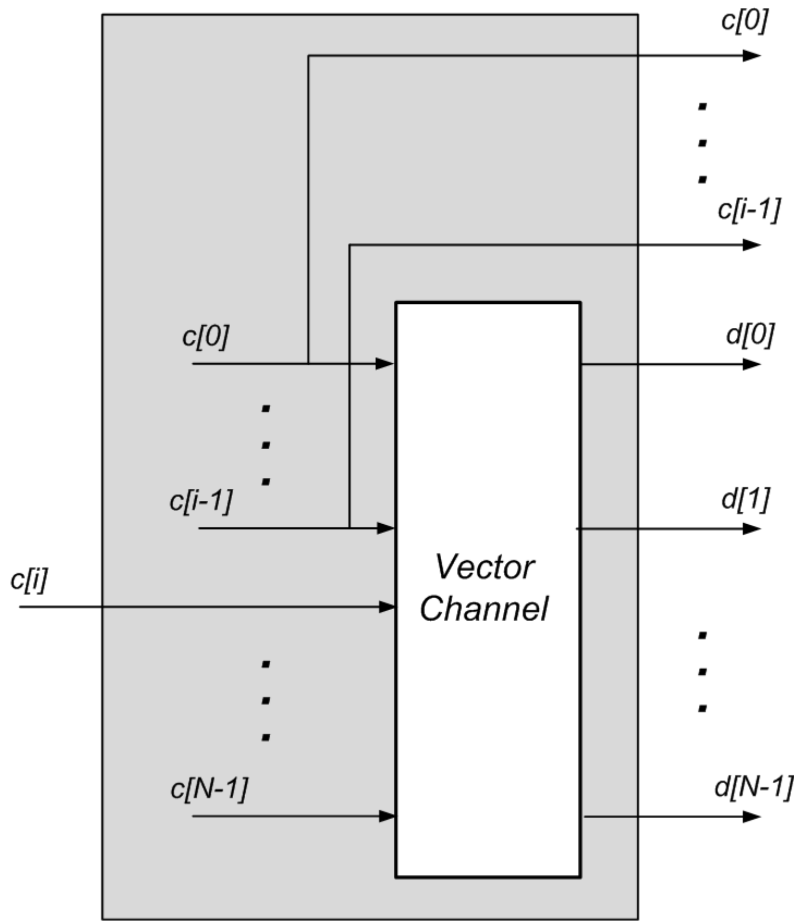
$$\mathbf{G}^{(1)} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$





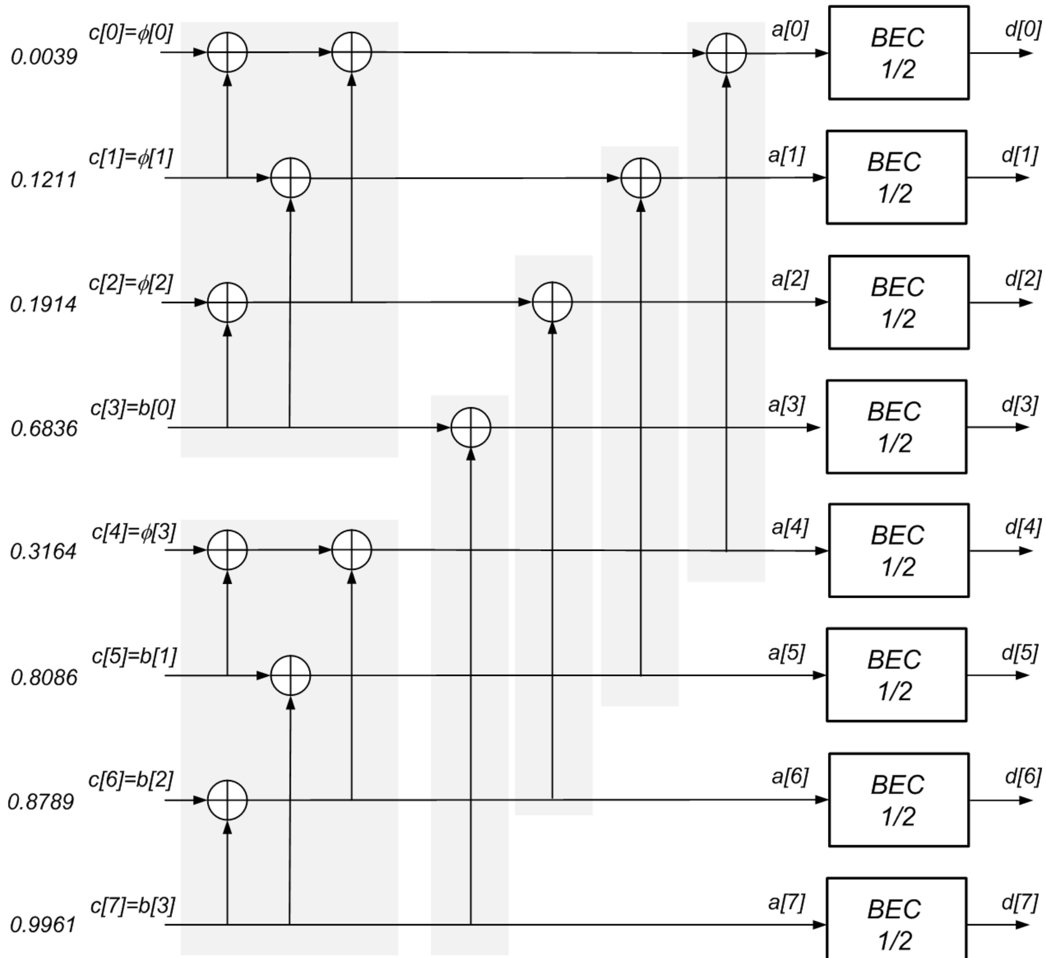
$$\mathbf{G}^{(m+1)} = \mathbf{G}^{(m)} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Aggregated Vector Channel



Vector Channel #i: from c[i] to [c[0],…, c[i-1];d[0],d[1],…, d[N-1]]

Polarization & Frozen Bits



$$\mathbf{a} = [\phi[0], \phi[1], \phi[2], b[0], \phi[3], b[1], b[2], b[3]]\,\mathbf{G}^{(3)}$$

Successive-Cancelation Decoding

1

$$L(c[3]) = \log \frac{\Pr\{c[3]=1 \mid \mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2]\}}{\Pr\{c[3]=0 \mid \mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2]\}}$$

Equivalently (LAPPR=LLR)

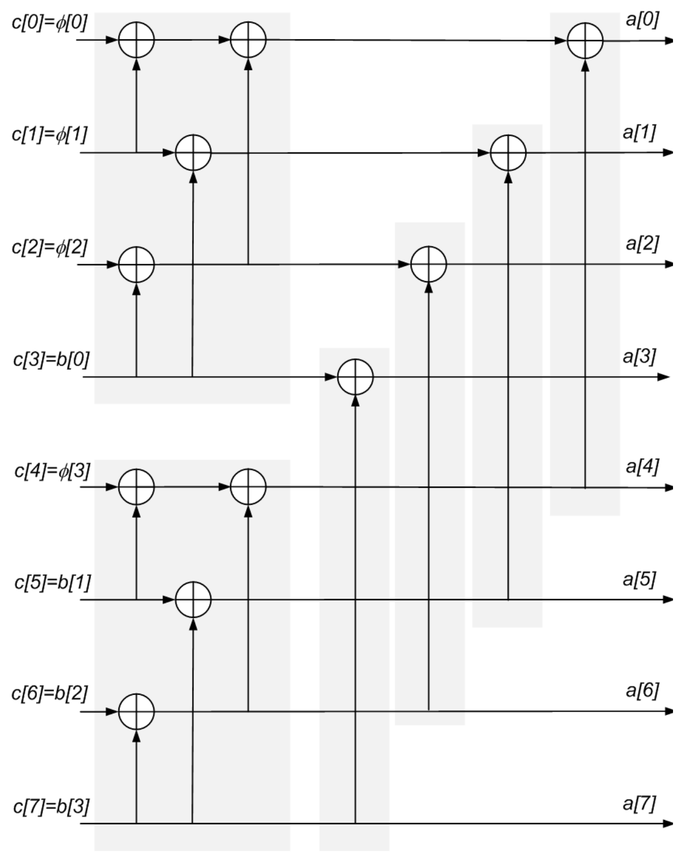
$$L(c[3]) = \log \frac{\Pr\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2] \mid c[3]=1\}}{\Pr\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2] \mid c[3]=0\}}$$

2

$$L(c[5]) = \log \frac{\Pr\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2], \hat{c}[3], \hat{c}[4] \mid c[5]=1\}}{\Pr\{\mathbf{d}, \hat{c}[0], \hat{c}[1], \hat{c}[2], \hat{c}[3], \hat{c}[4] \mid c[5]=0\}}$$

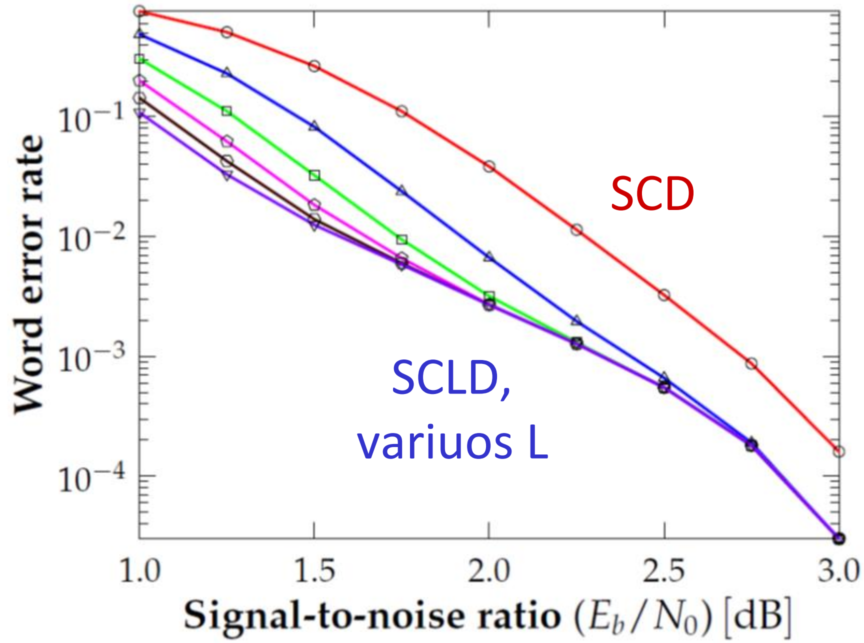
⋮

$$L(c[n]) = \log \frac{\Pr\{\mathbf{d}, \hat{c}[0], \ldots, \hat{c}[n-1] \mid c[n]=1\}}{\Pr\{\mathbf{d}, \hat{c}[0], \ldots, \hat{c}[n-1] \mid c[n]=0\}}$$



Coding/Decoding

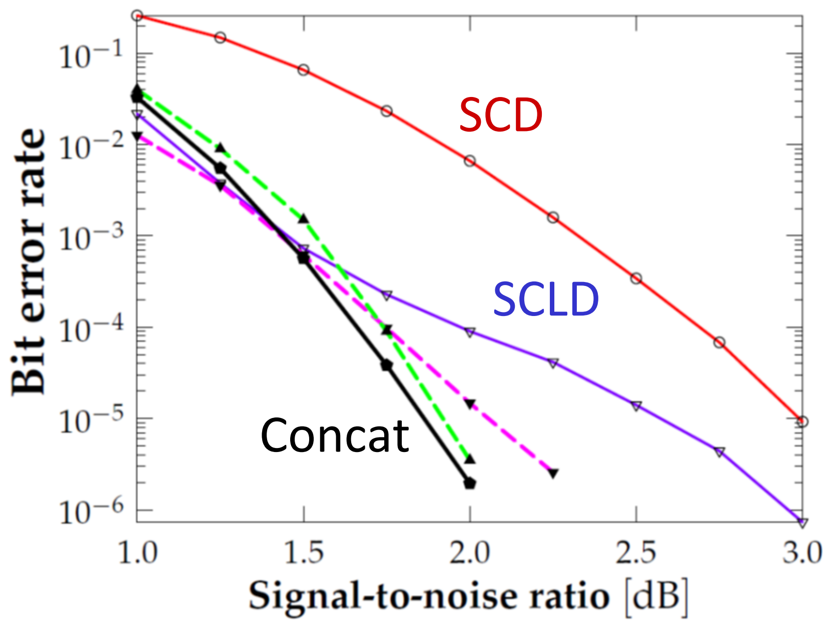
- Il decoder SCD non è in realtà iterativo, una volta decisi i bit non si modificano più
- E' necessario allora «rinviare» la decisione sui bit: si mantengono le due ipotesi «0» e «1» e si procede con due ulteriori decodifiche parallele
- Si evita la crescita esponenziale delle ipotesi mantenendo solo le *L* ipotesi a maggiore verosimiglianza, ordinale in una LIST: *Successive cancelation List Decoder*
- Complessità *L* volte maggiore



N=2048, *r*=1/2

Improved Coding/Decoding

- Final Touch 5G: *concatenazione* con codice esterno CRC e mantenimento in lista delle sole ipotesi che soddisfano il parity check (expurgated SCLD)



N=2048, *r*=1/2

Codes Comparison

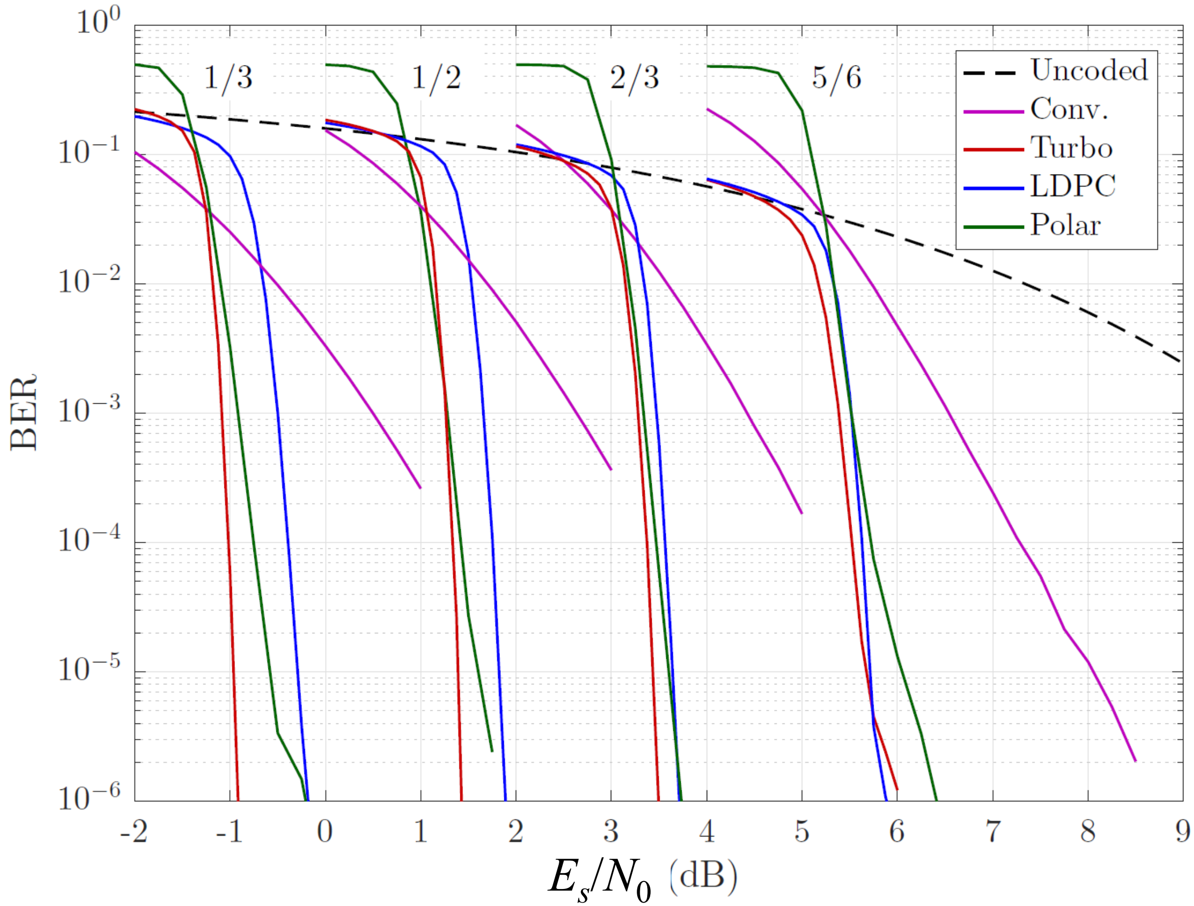
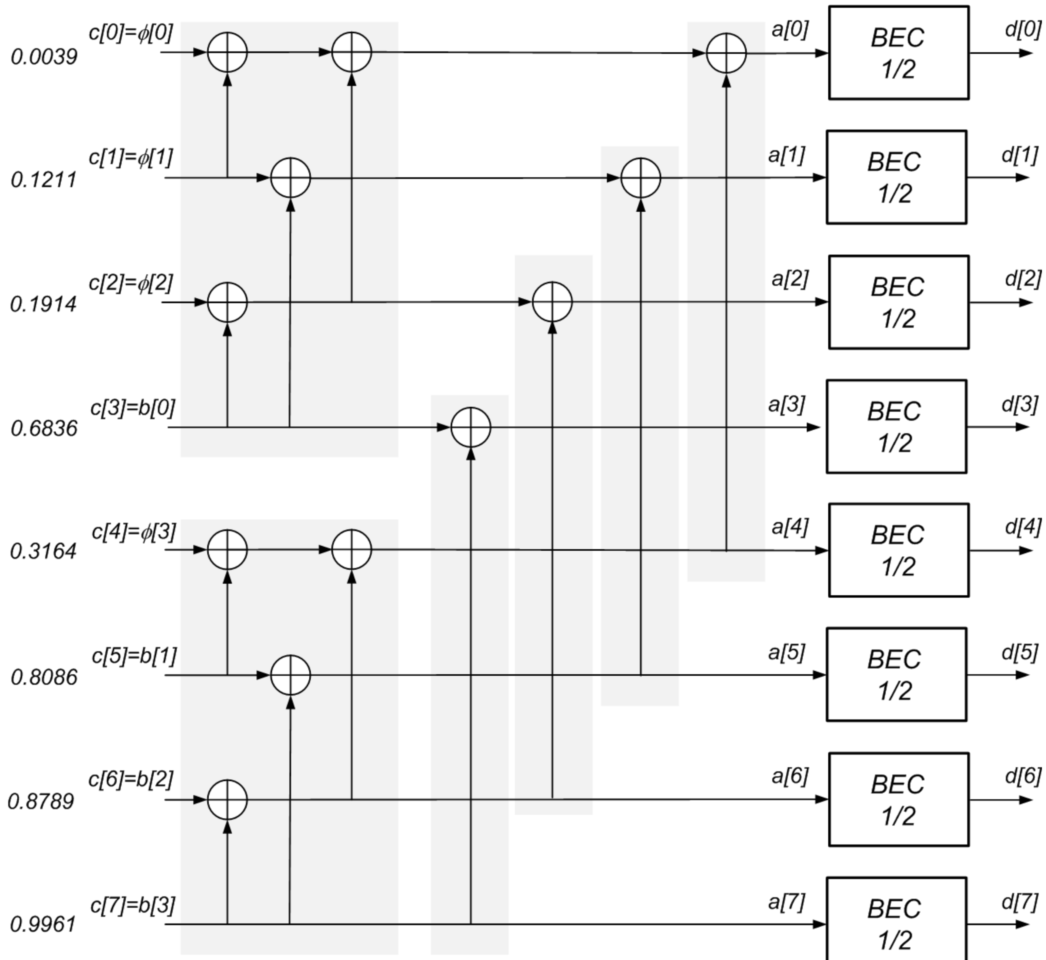


Fig. 14. BER comparison for different code rates, $K = 8192$ (For LDPC, $K = 8196$ for $R = 1/2$, and $1/3$, and $K = 8200$ for $R = 5/6$.)

Ingegneria delle Telecomunicazioni

Digital Communications

Polarization & Frozen Bits



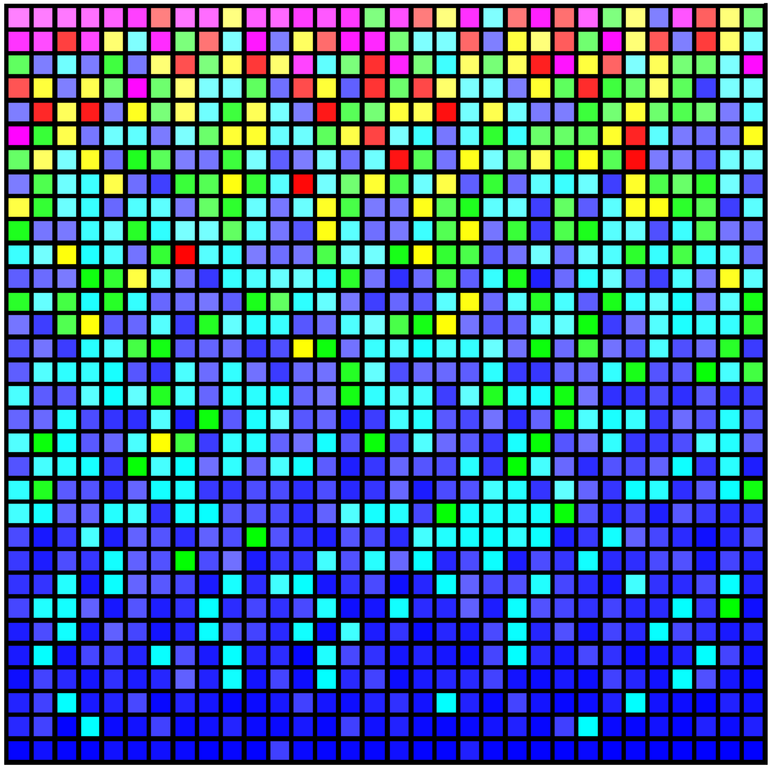
$$\mathbf{a} = [\phi[0], \phi[1], \phi[2], b[0], \phi[3], b[1], b[2], b[3]]\,\mathbf{G}^{(3)}$$

5G Polar Codes for PUCCH

- K_p parity check are added to he K' information bits, $K=K'+K_p$ via suited interleaving (most of them are at the end of the CRC code block)

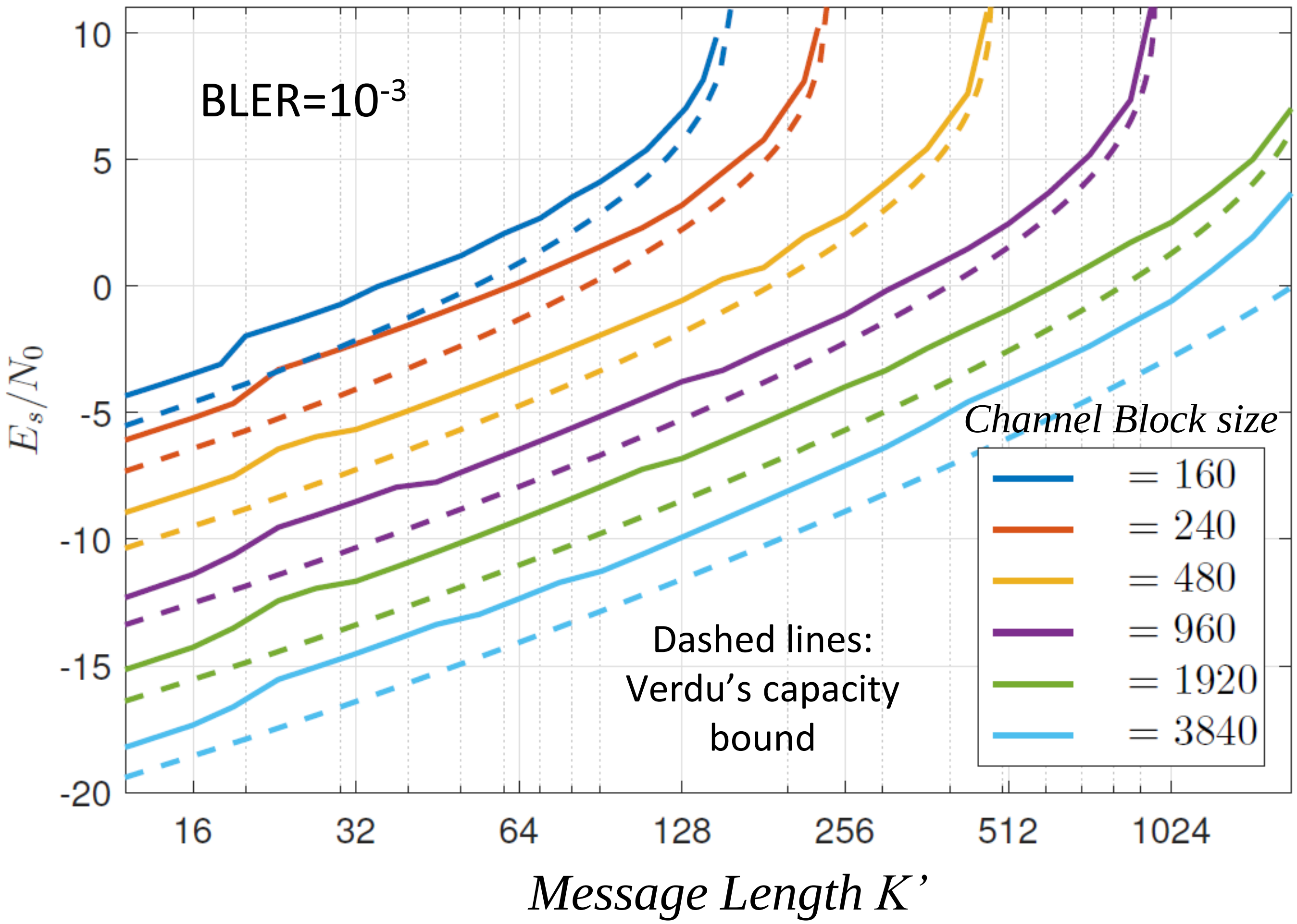
- Encoding is always power-of-2 but with different lengths, according to the message size

- To set the coding rate, the frozen bits are in a variable number and are selected in a «universal» sequence of channel capacity (reliability). The rate can be made also very small (down to r=1/8)



Dip. Ingegneria dell'Informazione
University of Pisa, Pisa, Italy

5G Codes for the PUCCH



xDSL per l'Ultimo Chilometro