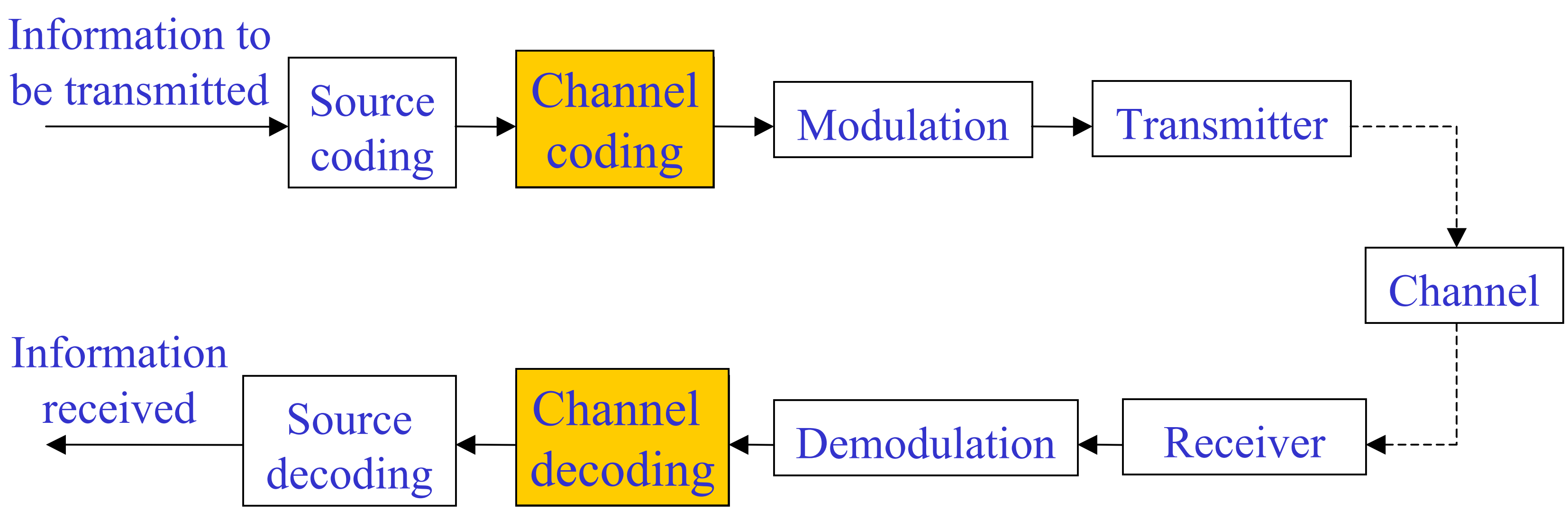
Chapter 4

Channel Coding

Outline

- FEC (Forward Error Correction)
- Block Codes
- Convolutional Codes
- Interleaving
- Information Capacity Theorem
- Turbo Codes
- CRC (Cyclic Redundancy Check)
- ARQ (Automatic Repeat Request)
  - Stop-and-wait ARQ
  - Go-back-N ARQ
  - Selective-repeat ARQ

Channel Coding in Digital Communication Systems



Forward Error Correction (FEC)

- The key idea of FEC is to transmit enough redundant data to allow receiver to recover from errors all by itself. No sender retransmission required.
- The major categories of FEC codes are
  - Block codes,
  - Cyclic codes,
  - Reed-Solomon codes (Not covered here),
  - Convolutional codes, and
  - Turbo codes, etc.

Block Codes

- Information is divided into blocks of length k
- r parity bits or check bits are added to each block (total length $n = k + r$),.
- Code rate $R = k/n$
- Decoder looks for codeword closest to received vector (code vector + error vector)
- Tradeoffs between
  - Efficiency
  - Reliability
  - Encoding/Decoding complexity

Block Codes: Linear Block Codes

- Linear Block Code

The block length $c(x)$ or $\boldsymbol{C}$ of the Linear Block Code is

$$c(x) = m(x)\, g(x) \qquad \text{or} \qquad \boldsymbol{C} = m\, \boldsymbol{G}$$

where $m(x)$ or m is the information codeword block length, $g(x)$ is the generator polynomial, $\boldsymbol{G}$ is the generator matrix.

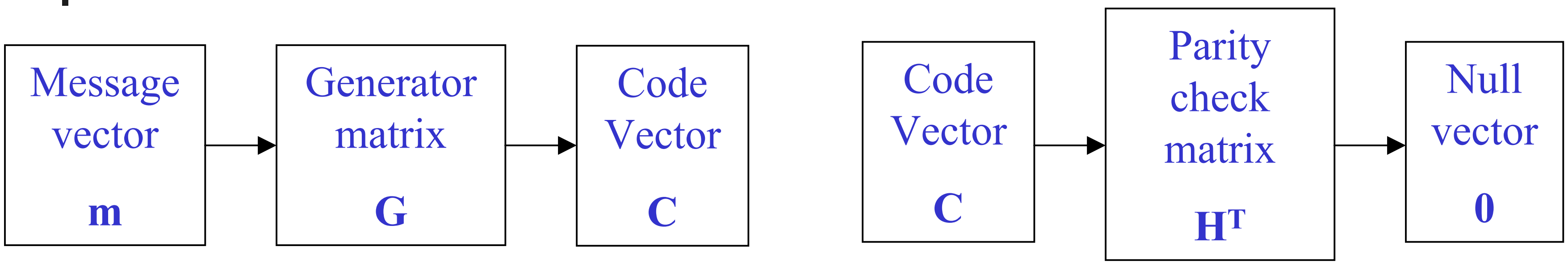
$$\boldsymbol{G} = [\boldsymbol{p} \mid \boldsymbol{I}],$$

where p_i = Remainder of $[x^{n-k+i-1}/g(x)]$ for $i=1, 2, .., k$, and $\boldsymbol{I}$ is unit matrix.

- The parity check matrix

$\boldsymbol{H} = [\boldsymbol{p}^T \mid \boldsymbol{I}\,]$, where $\boldsymbol{p}^T$ is the transpose of the matrix $\boldsymbol{p}$.

Block Codes: Linear Block Codes



Operations of the generator matrix and the parity check matrix

The parity check matrix H is used to detect errors in the received code by using the fact that $c * H^T = \mathbf{0}$ (null vector)

Let $x = c \oplus e$ be the received message where c is the correct code and e is the error

Compute $S = x * H^T = (c \oplus e) * H^T = c\,H^T \oplus e\,H^T = e\,H^T$

If S is 0 then message is correct else there are errors in it, from common known error patterns the correct message can be decoded.

Block Codes: Example

Example : Find linear block code encoder **G** if code generator polynomial $g(x)=1+x+x^3$ for a (7, 4) code.

We have n = Total number of bits = 7, k = Number of information bits = 4, r = Number of parity bits = n - k = 3.

$\because$

$$G = [P \mid I] = \begin{bmatrix} p_1 \; 1\,0 \cdots 0 \\ p_2 \; 0\,1 \cdots 0 \\ \cdots\cdots\cdots\cdots \\ p_k \; 0\,0 \cdots 1 \end{bmatrix},$$

where

$$p_i = \mathrm{Re}\,mainder\ of \left[\frac{x^{n-k+i-1}}{g(x)} \right], \quad i = 1, 2, \cdots, k$$

Block Codes: Example (Continued)

$$p_1 = \mathrm{Re}\left[\frac{x^3}{1+x+x^3}\right] = 1 + x \rightarrow [110]$$

$$p_2 = \mathrm{Re}\left[\frac{x^4}{1+x+x^3}\right] = x + x^2 \rightarrow [011]$$

$$p_3 = \mathrm{Re}\left[\frac{x^5}{1+x+x^3}\right] = 1 + x + x^2 \rightarrow [111]$$

$$p_4 = \mathrm{Re}\left[\frac{x^6}{1+x+x^3}\right] = 1 + x^2 \rightarrow [101]$$

$$G = \begin{bmatrix} 1101000 \\ 0110100 \\ 1110010 \\ 1010001 \end{bmatrix}$$

Cyclic Codes

It is a block code which uses a shift register to perform encoding and decoding

The code word with n bits is expressed as

$$c(x)=c_1x^{n-1}+c_2x^{n-2}\ldots\ldots+c_n$$

where each c_i is either a 1 or 0.

$$c(x) = m(x)\,x^{n-k} + c_p(x)$$

where $c_p(x)$ = remainder from dividing $m(x)\,x^{n-k}$ by generator $g(x)$

if the received signal is $c(x) + e(x)$ where $e(x)$ is the error.

To check if received signal is error free, the remainder from dividing $c(x) + e(x)$ by $g(x)$ is obtained(syndrome). If this is 0 then the received signal is considered error free else error pattern is detected from known error syndromes.

Cyclic Redundancy Check (CRC)

- Using parity, some errors are masked - careful choice of bit combinations can lead to better detection.
- Binary (*n, k*) CRC codes can detect the following error patterns
  1. All error bursts of length n-k or less.
  2. All combinations of minimum Hamming distance d_{min}- 1 or fewer errors.
  3. All error patters with an odd number of errors if the generator polynomial $g(x)$ has an even number of nonzero coefficients.

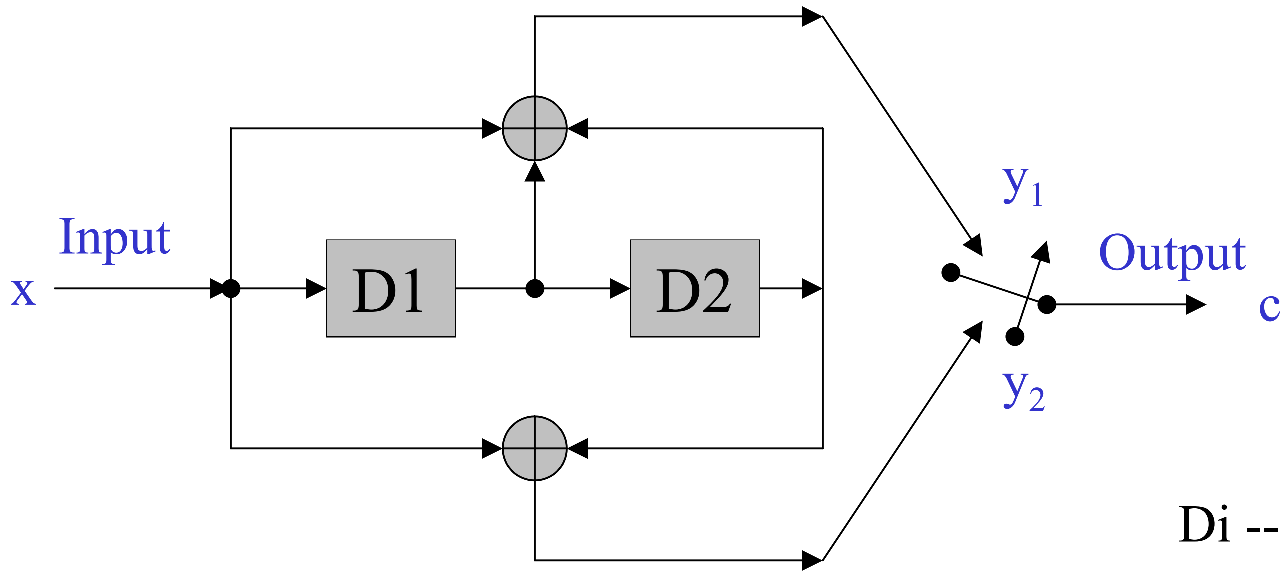
Common CRC Codes

Code	Generator polynomial g(x)	Parity check bits
CRC-12	$1+x+x^2+x^3+x^{11}+x^{12}$	12
CRC-16	$1+x^2+x^{15}+x^{16}$	16
CRC-CCITT	$1+x^5+x^{15}+x^{16}$	16

Convolutional Codes

- Encoding of information stream rather than information blocks
- Value of certain information symbol also affects the encoding of next *M* information symbols, i.e., memory *M*
- Easy implementation using shift register
  → Assuming *k* inputs and *n* outputs
- Decoding is mostly performed by the Viterbi Algorithm (not covered here)

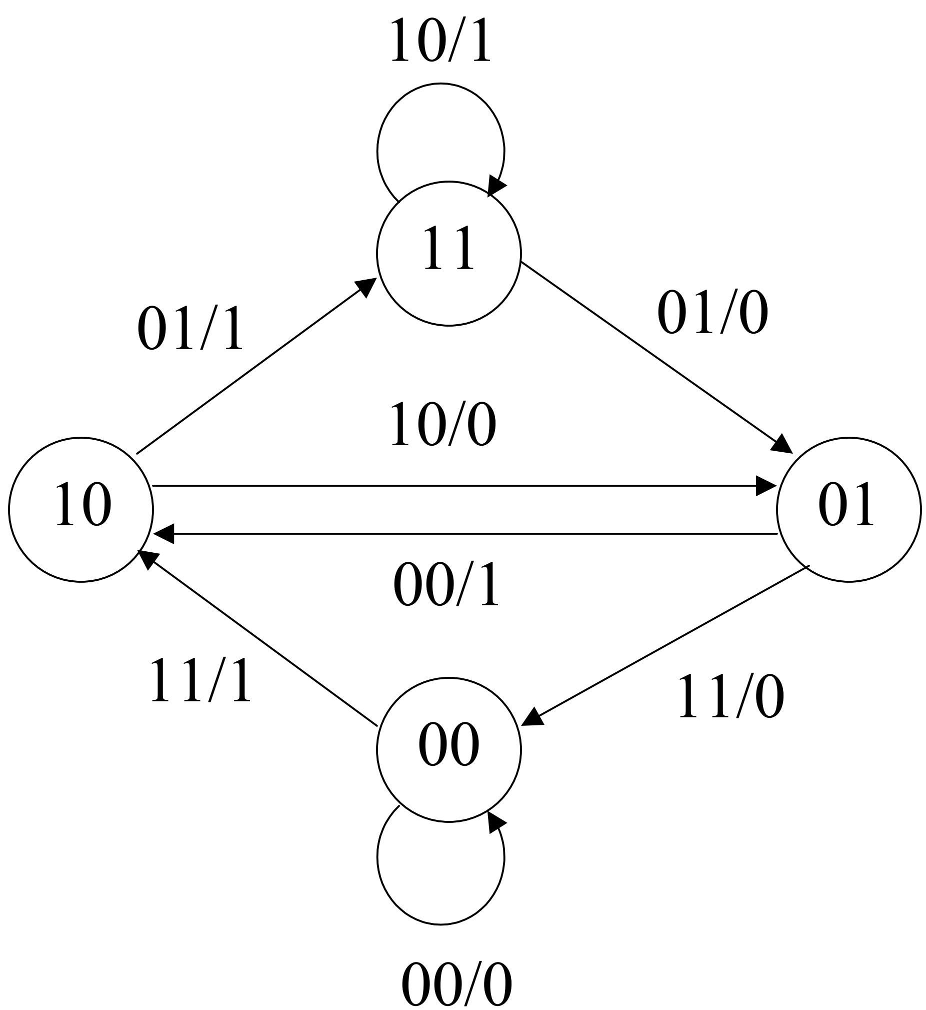
Convolutional Codes: (n=2, k=1, M=2) Encoder



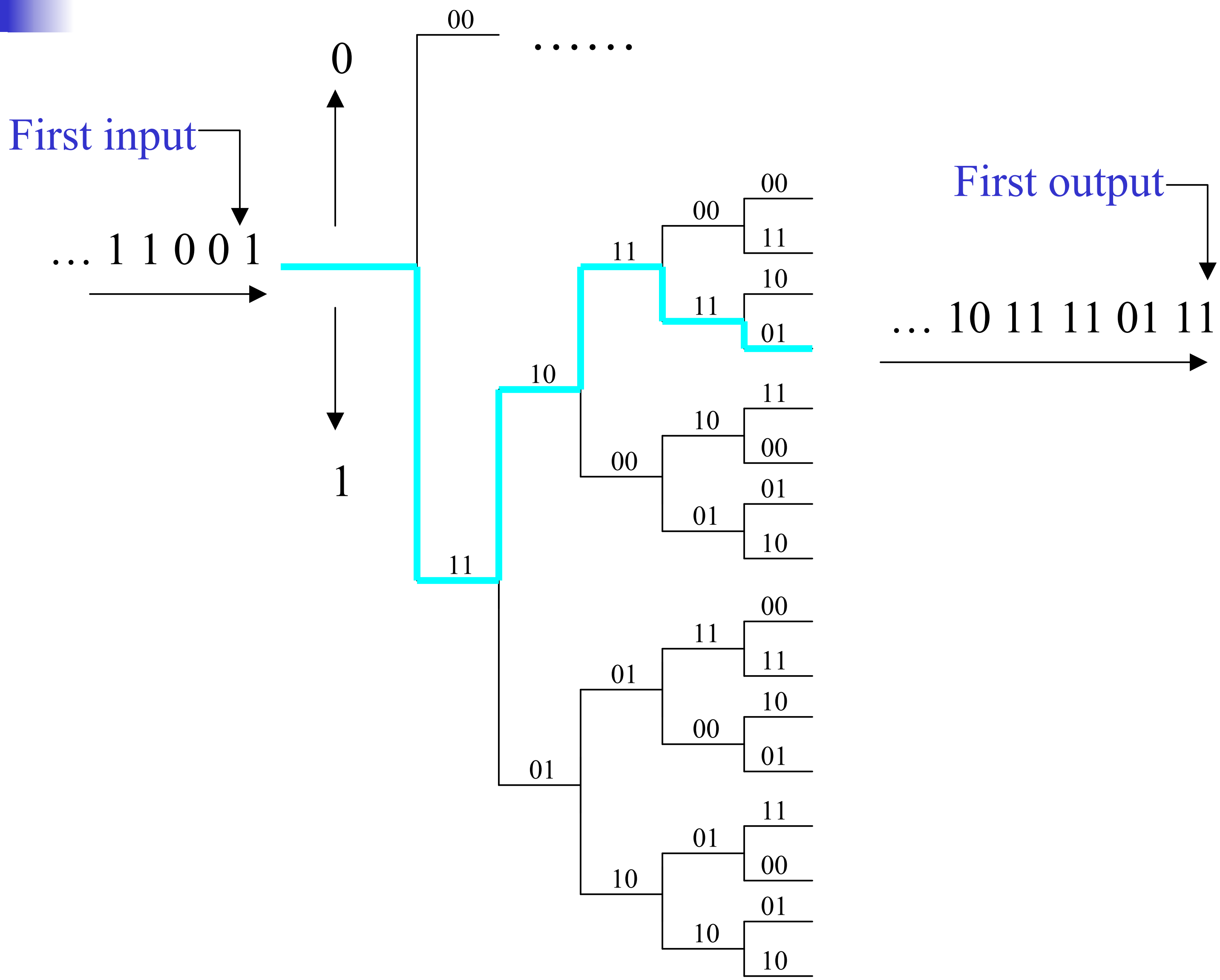
Di -- Register

Input:	1	1	1	0	0	0	…
Output:	11	01	10	01	11	00	…
Input:	1	0	1	0	0	0	…
Output:	11	10	00	10	11	00	…

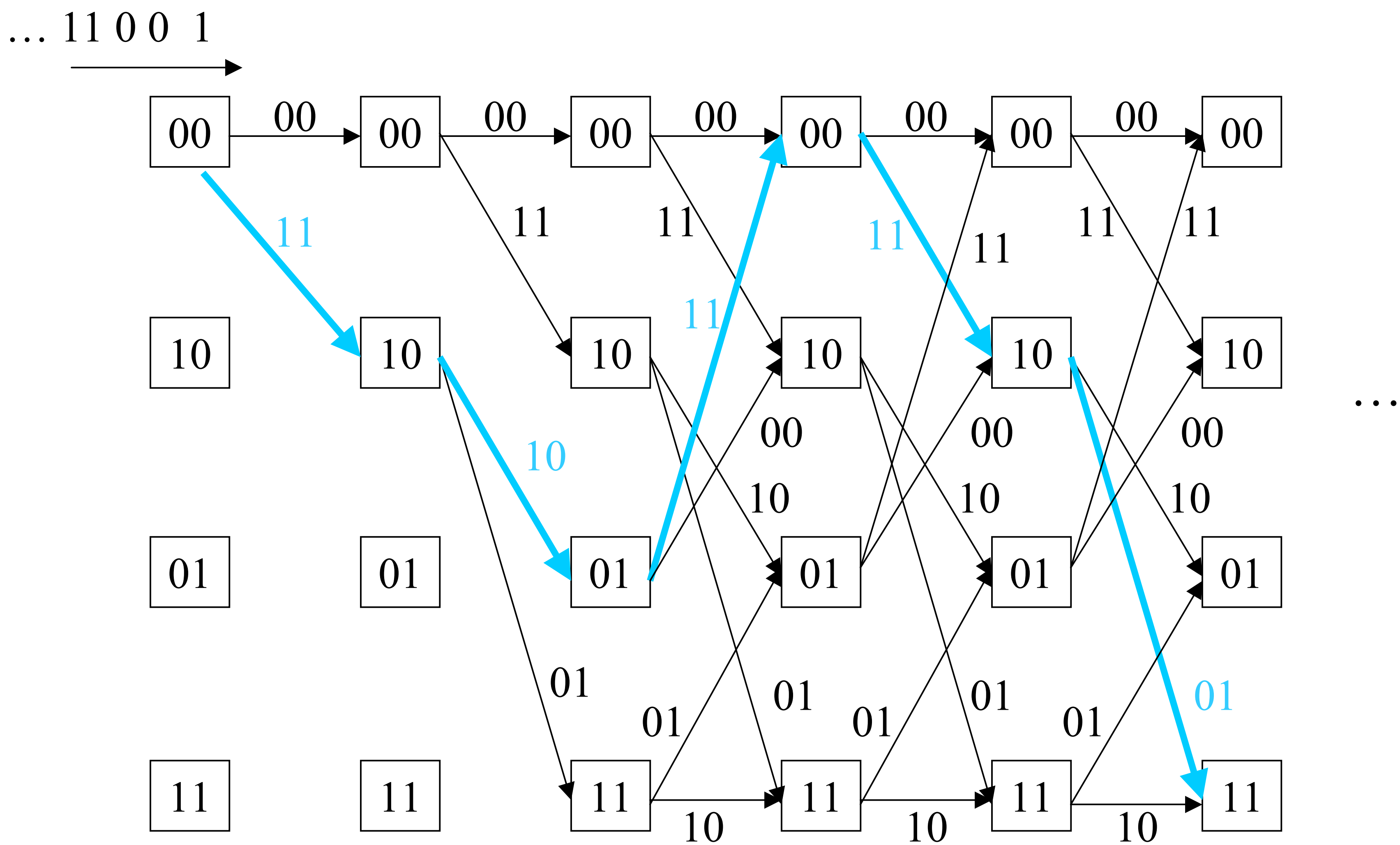
State Diagram



Tree Diagram



Trellis



… 11 0 0 1
00 00 00 00 00 00
10 10 10 10 10 10
01 01 01 01 01 01
11 11 11 11 11 11
00 00 00 00 00
11 11 11 11 11 11 11 11
11 10 11 01
00 00 00
10 10 10
01 01 01 01 01 01 01
10 10 10
…

Interleaving

Input Data: a1, a2, a3, a4, a5, a6, a7, a8, a9, …

Interleaving (Write →, Read ↓):

a1,	a2,	a3,	a4
a5,	a6,	a7,	a8
a9,	a10,	a11,	a12
a13,	a14,	a15,	a16

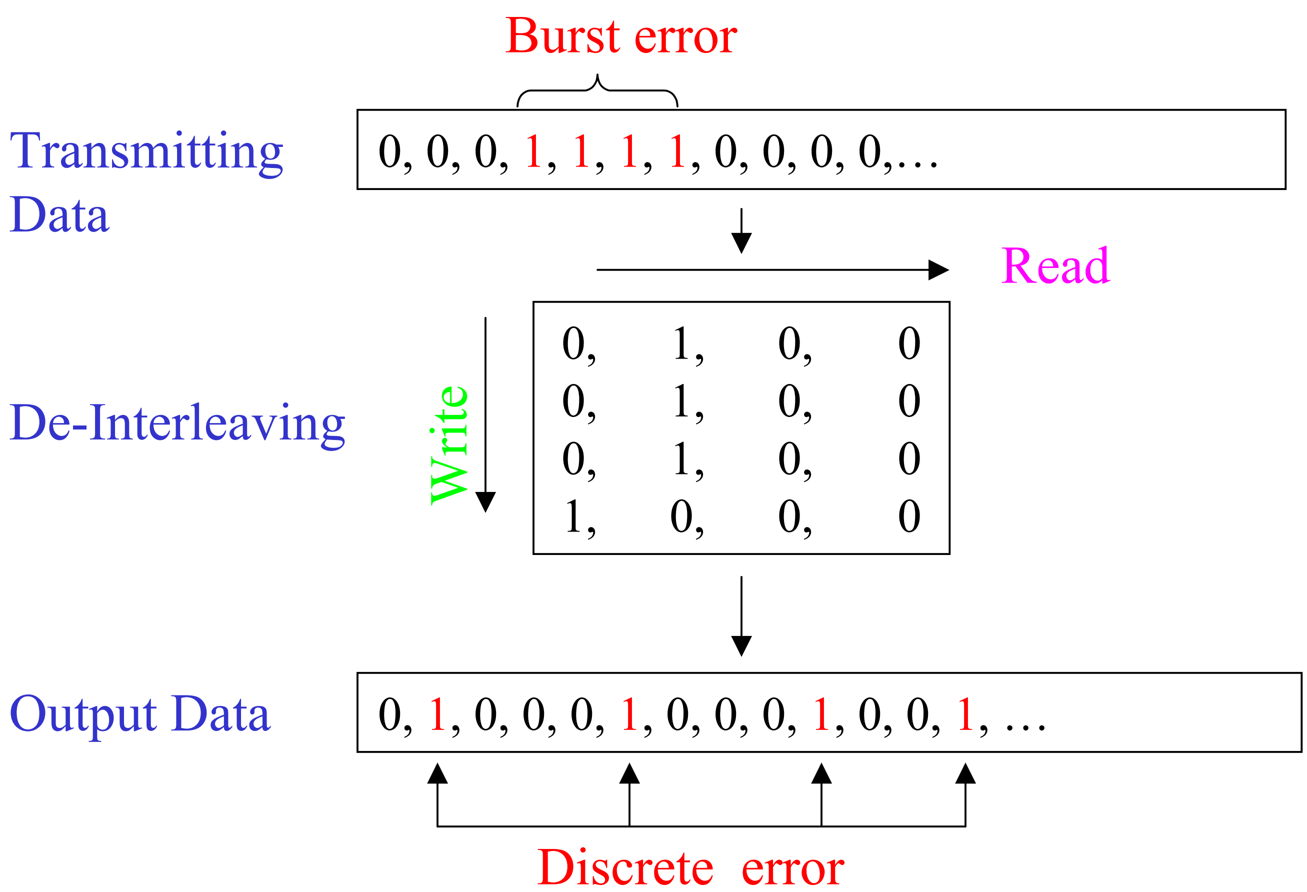
Transmitting Data: a1, a5, a9, a13, a2, a6, a10, a14, a3, …

De-Interleaving (Write ↓, Read →):

a1,	a2,	a3,	a4
a5,	a6,	a7,	a8
a9,	a10,	a11,	a12
a13,	a14,	a15,	a16

Output Data: a1, a2, a3, a4, a5, a6, a7, a8, a9, …

Interleaving (Example)



Information Capacity Theorem (Shannon Limit)

- The information capacity (or channel capacity) C of a continuous channel with bandwidth B Hertz can be perturbed by additive Gaussian white noise of power spectral density $N_0/2$, provided bandwidth B satisfies

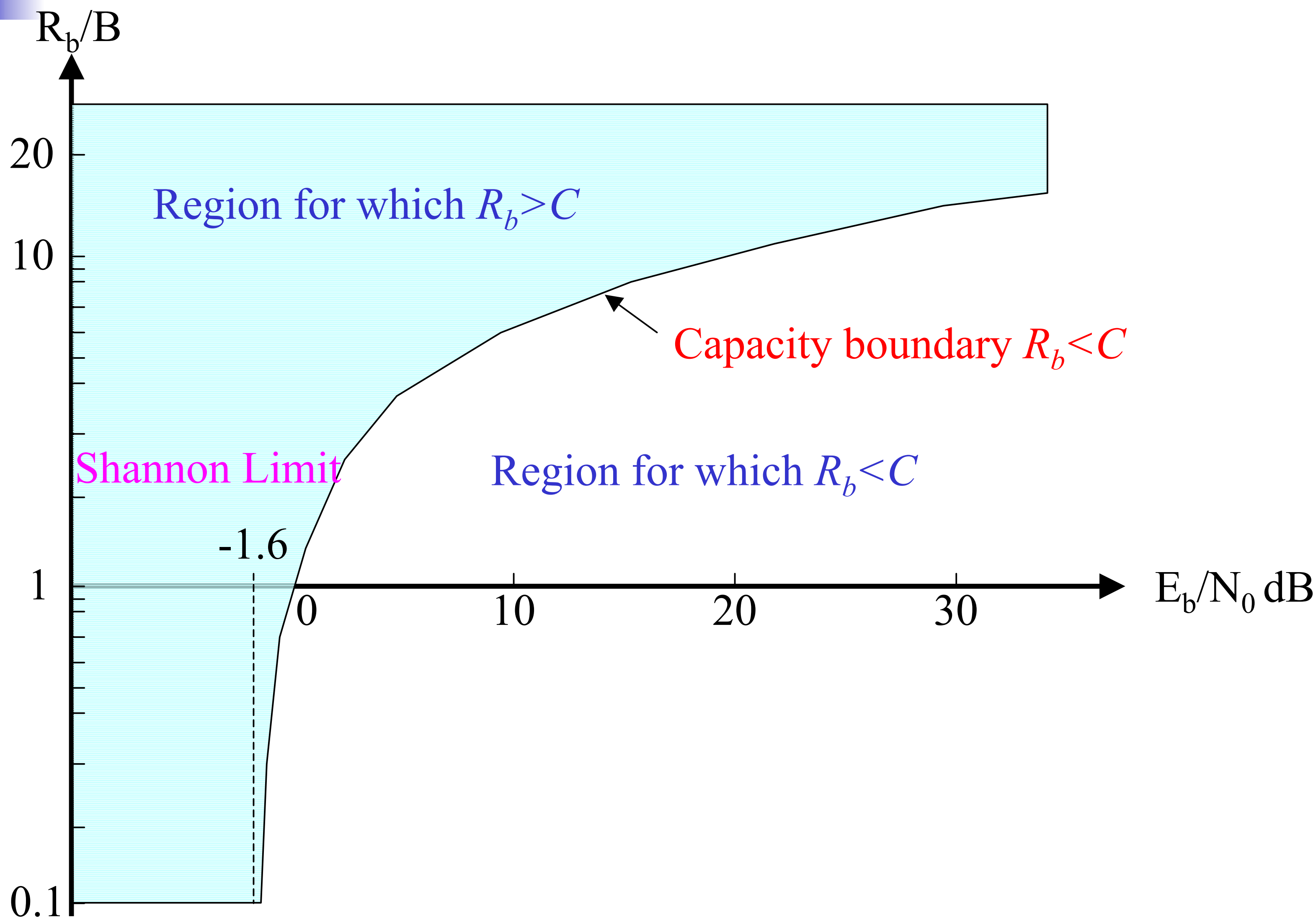
$$C = B\log_2\left(1+\frac{P}{N_0 B}\right) \quad bits/\sec ond$$

where P is the average transmitted power $P = E_b R_b$ (for an ideal system, $R_b = C$).

E_b is the transmitted energy per bit,

R_b is transmission rate.

Shannon Limit

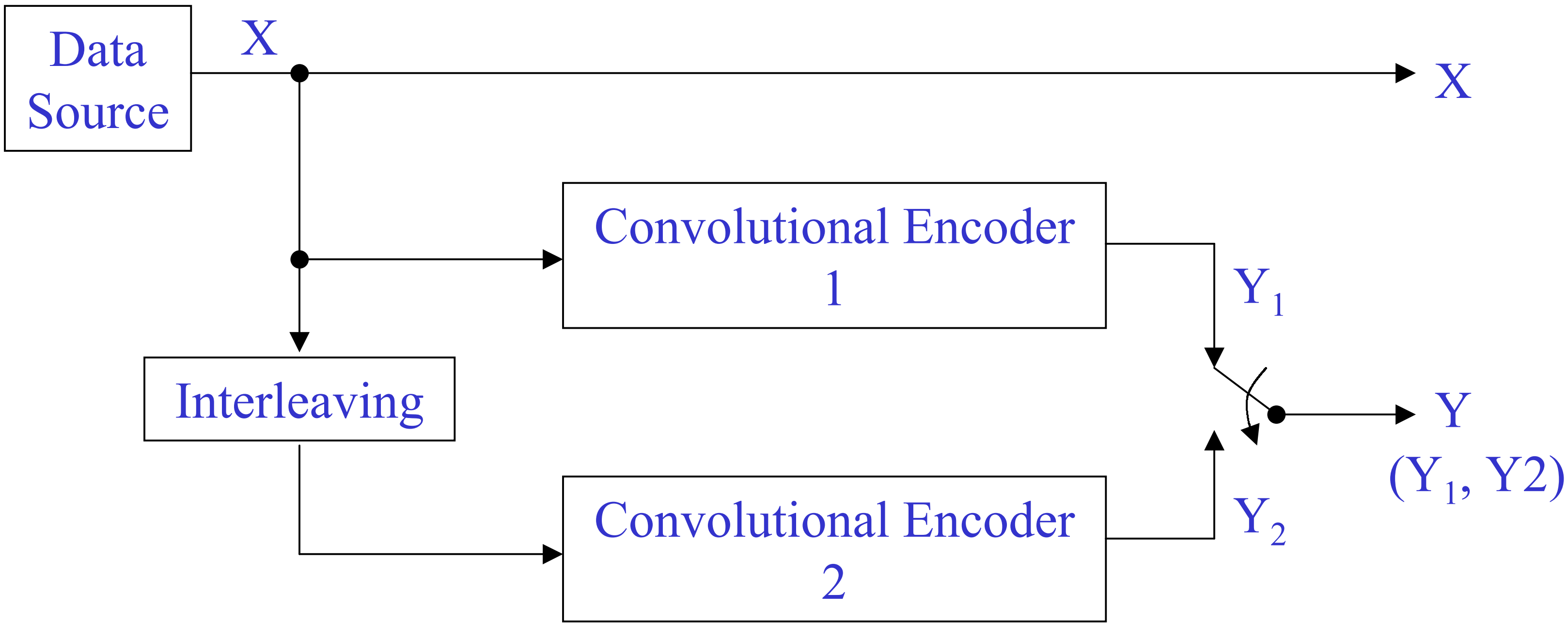


Turbo Codes

- A brief historic of turbo codes :

  The turbo code concept was first introduced by C. Berrou in 1993. Today, Turbo Codes are considered as the most efficient coding schemes for FEC.
- Scheme with known components (simple convolutional or block codes, interleaver, soft-decision decoder, etc.)
- Performance close to the Shannon Limit (E_b/N_0 = -1.6 db if $R_b \rightarrow 0$) at modest complexity!
- Turbo codes have been proposed for low-power applications such as deep-space and satellite communications, as well as for interference limited applications such as third generation cellular, personal communication services, ad hoc and sensor networks.

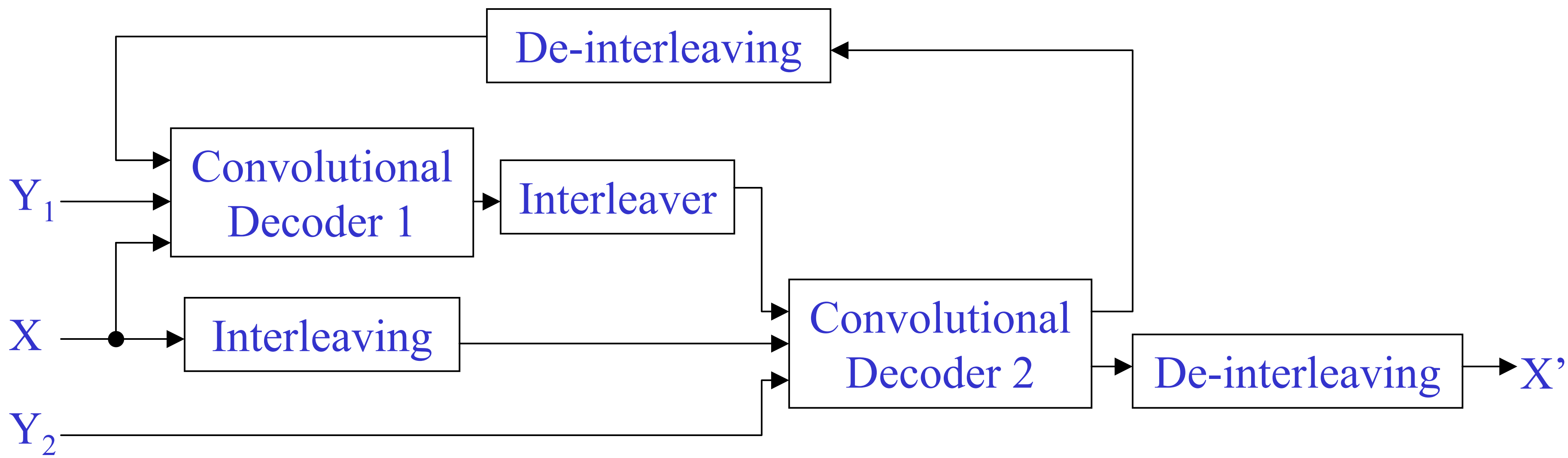
Turbo Codes: Encoder



X: Information

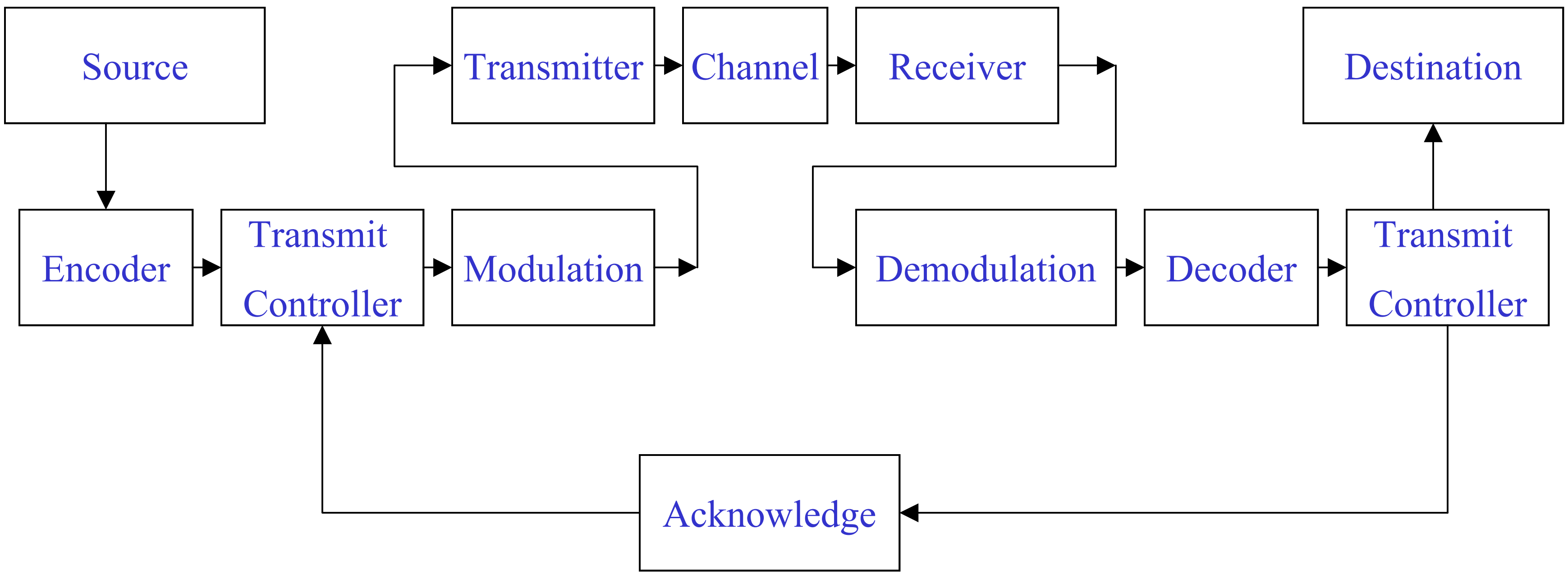
Y_i: Redundancy Information

Turbo Codes: Decoder

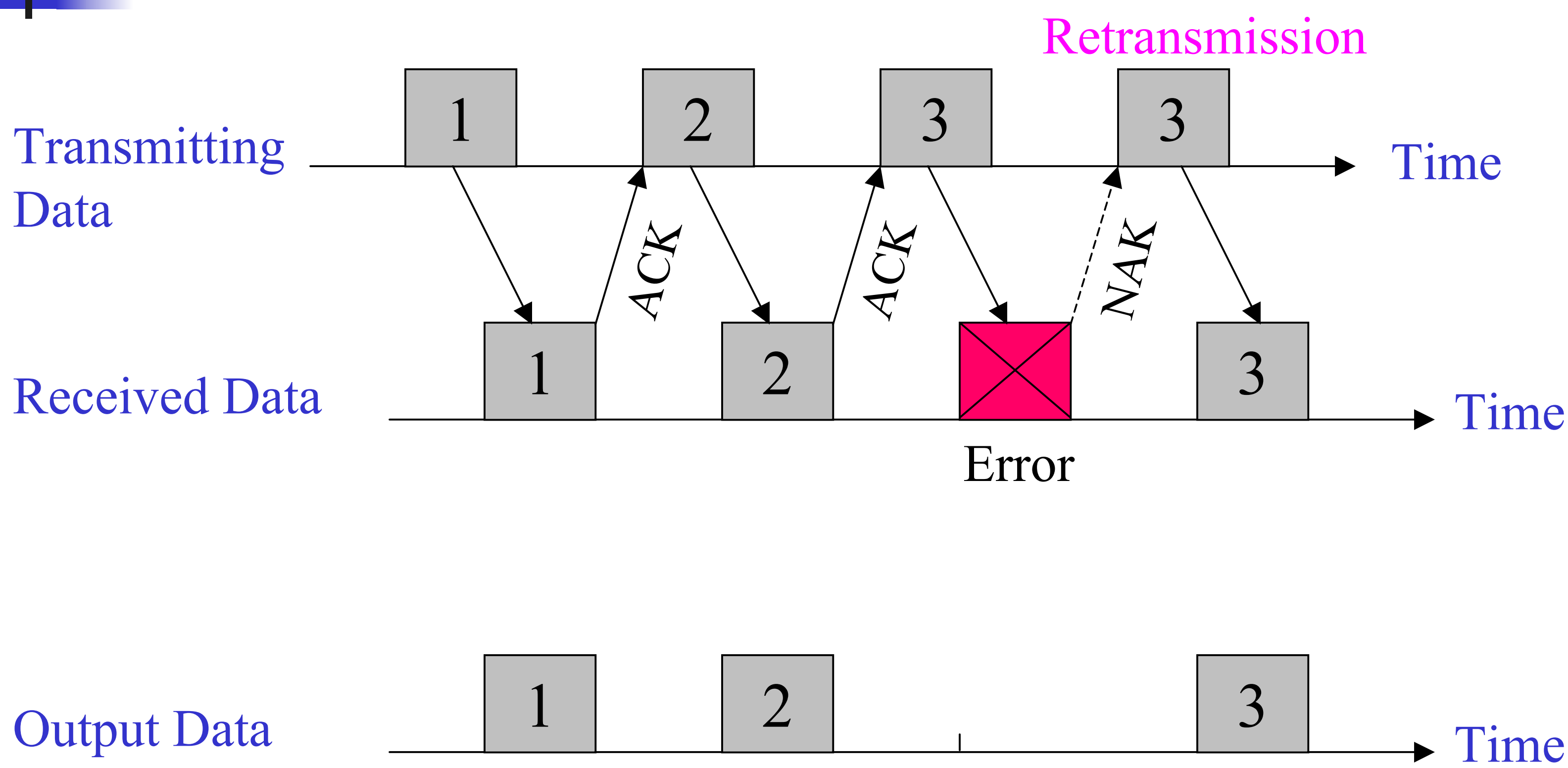


X': Decoded Information

Automatic Repeat Request (ARQ)



Stop-And-Wait ARQ (SAW ARQ)



ACK: Acknowledge

NAK: Negative ACK

Stop-And-Wait ARQ (SAW ARQ)

Throughput:

$$S = (1/T) * (k/n) = [(1- P_{\mathbf{b}})^n \;/\; (1 + D * R_b/ n)] * (k/n)$$

where T is the average transmission time in terms of a block duration

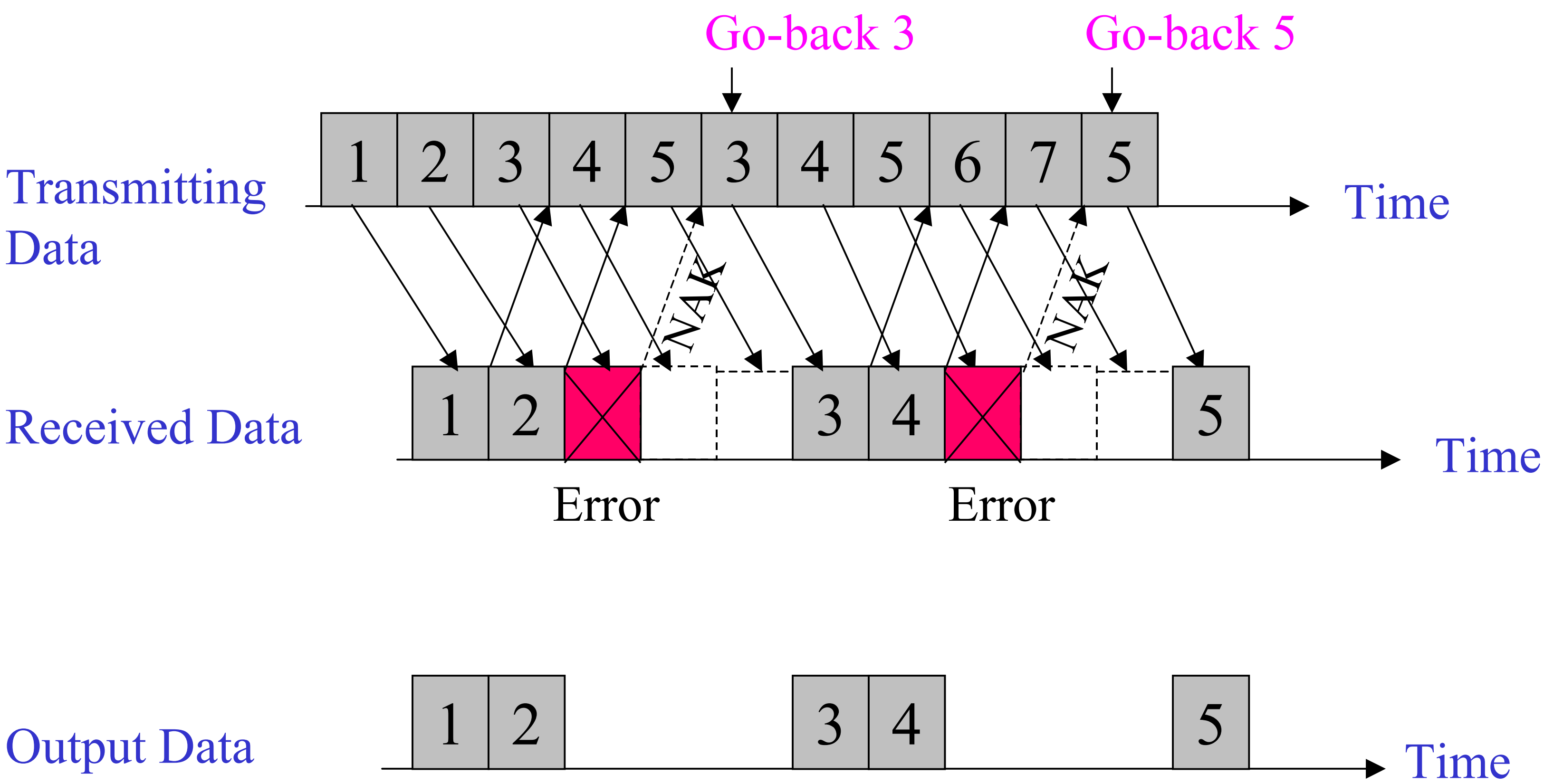
$$T = (1 + D * R_b/ n) * P_{ACK} + 2 * (1 + D * R_b/ n) * P_{ACK} * (1- P_{ACK})$$

$$+ 3 * (1 + D * R_b/ n) * P_{ACK} * (1- P_{ACK})^2 + \ldots..$$

$$= (1 + D * R_b/ n) / (1- P_{\mathbf{b}})^n$$

where n = number of bits in a block, k = number of information bits in a block, D = round trip delay, R_b= bit rate, $P_{\mathbf{b}}$ = BER of the channel, and $P_{ACK} = (1- P_{\mathbf{b}})^n$

Go-Back-N ARQ (GBN ARQ)



Go-Back-N ARQ (GBN ARQ)

Throughput

$$S = (1/T) * (k/n)$$

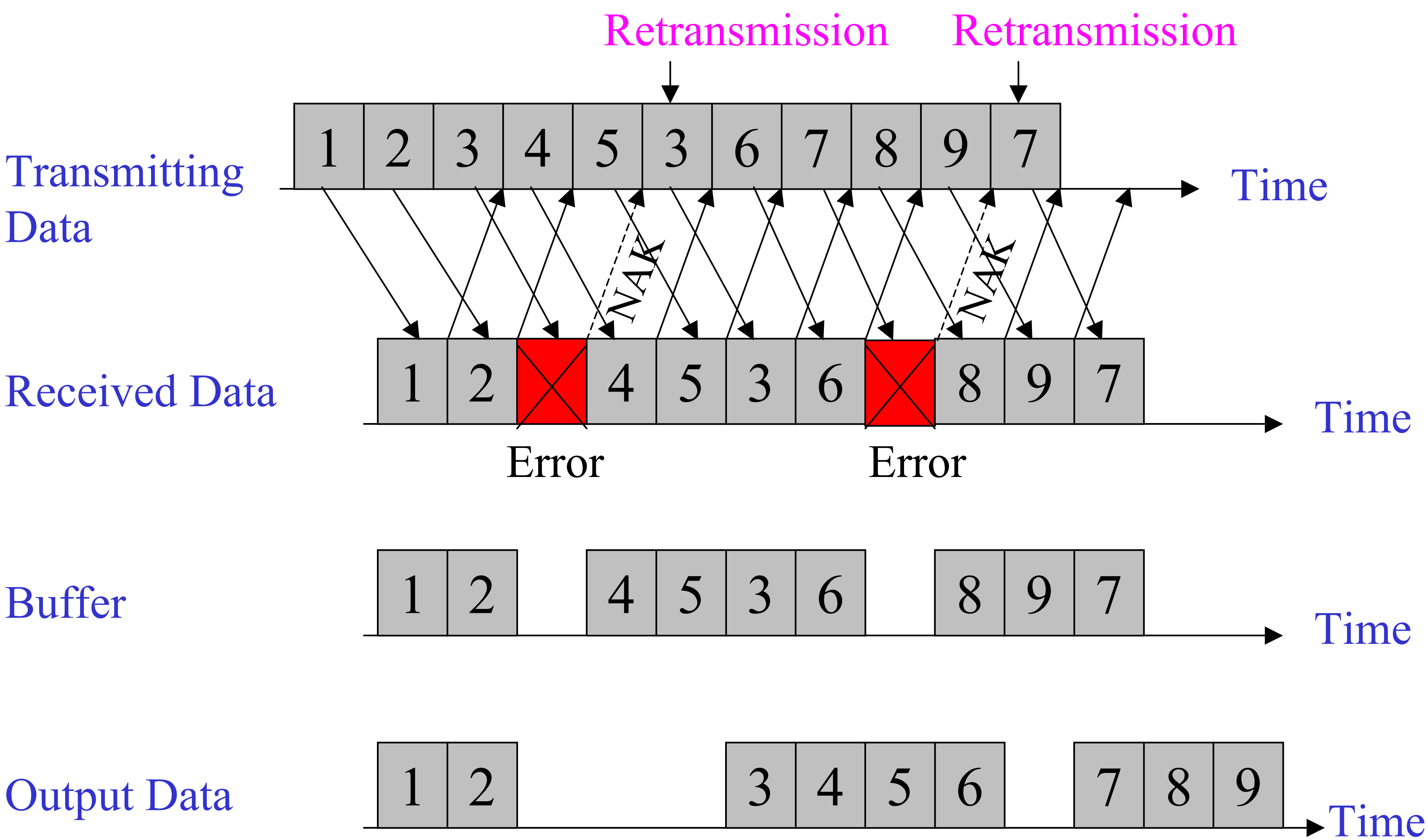
$$= [(1- P_b)^n \; / \; ((1- P_b)^n + N * (1-(1- P_b)^n))] * (k/n)$$

where

$$T = 1 * P_{ACK} + (N+1) * P_{ACK} * (1- P_{ACK}) +2 * (N+1) * P_{ACK} * (1- P_{ACK})^2 + \ldots.$$

$$= 1 + (N * [1 - (1- P_b)^n])/ (1- P_b)^n$$

Selective-Repeat ARQ (SR ARQ)



Selective-Repeat ARQ (SR ARQ)

Throughput

$$S = (1/T) * (k/n)$$

$$= (1- P_{\mathbf{b}})^n * (k/n)$$

where

$$T = 1 * P_{ACK} + 2 * P_{ACK} * (1- P_{ACK}) + 3 * P_{ACK} * (1- P_{ACK})^2$$

$$+ \ldots.$$

$$= 1/(1- P_{\mathbf{b}})^n$$