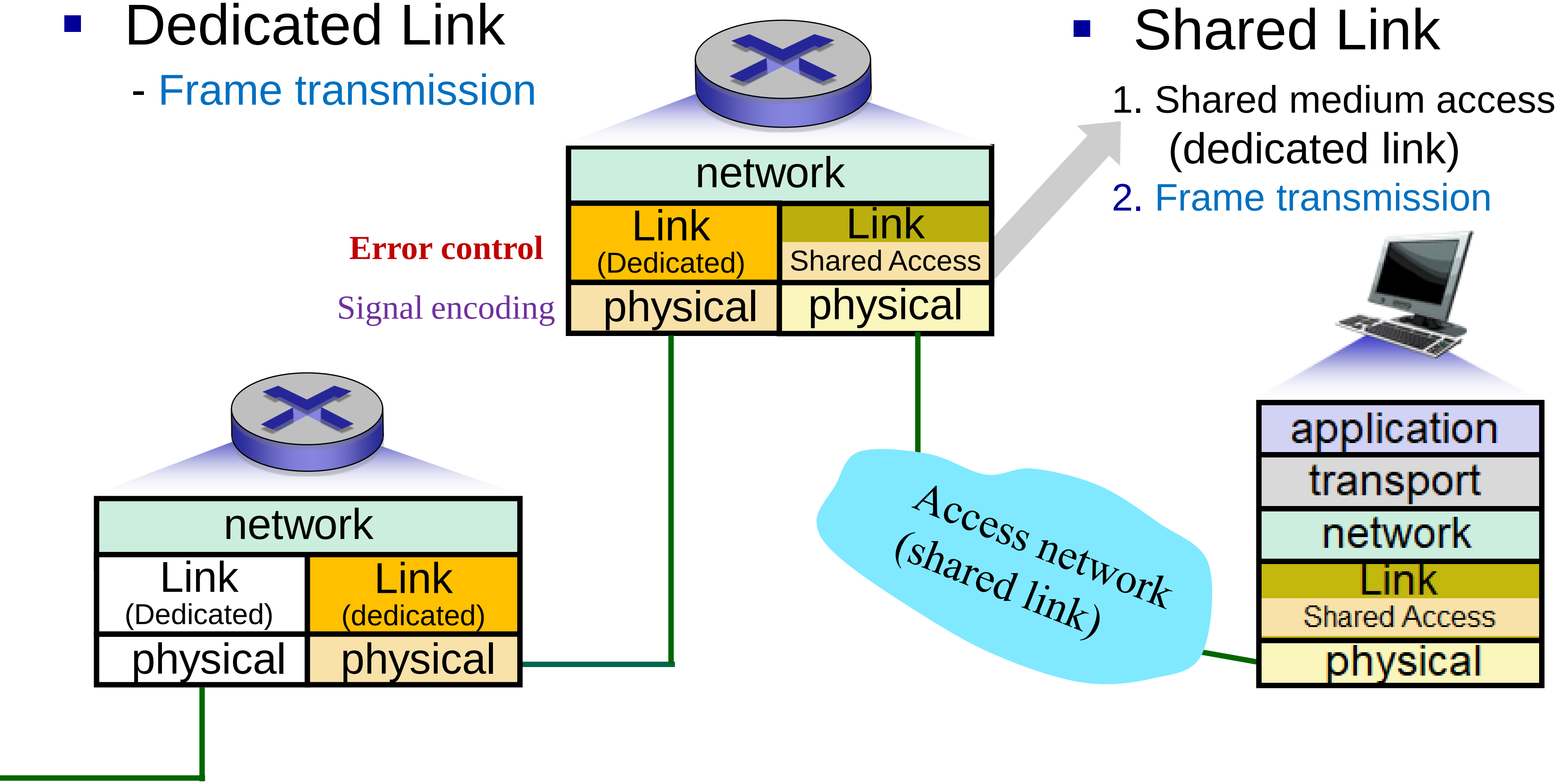
Preview

- Dedicated Link

  - Frame transmission

- Shared Link

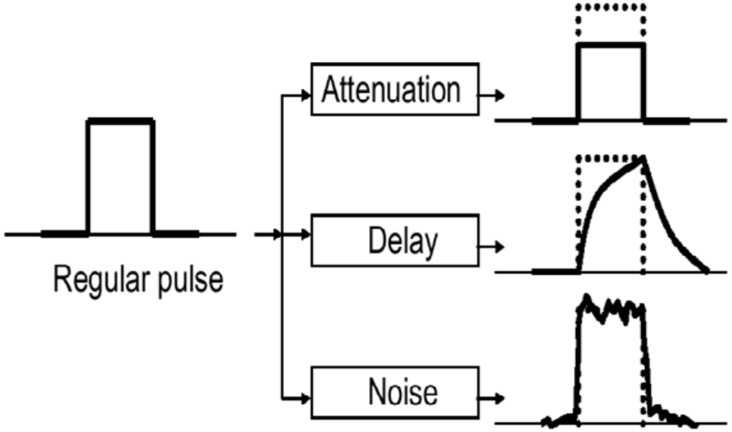
  1. Shared medium access right (dedicated link)
  2. Frame transmission



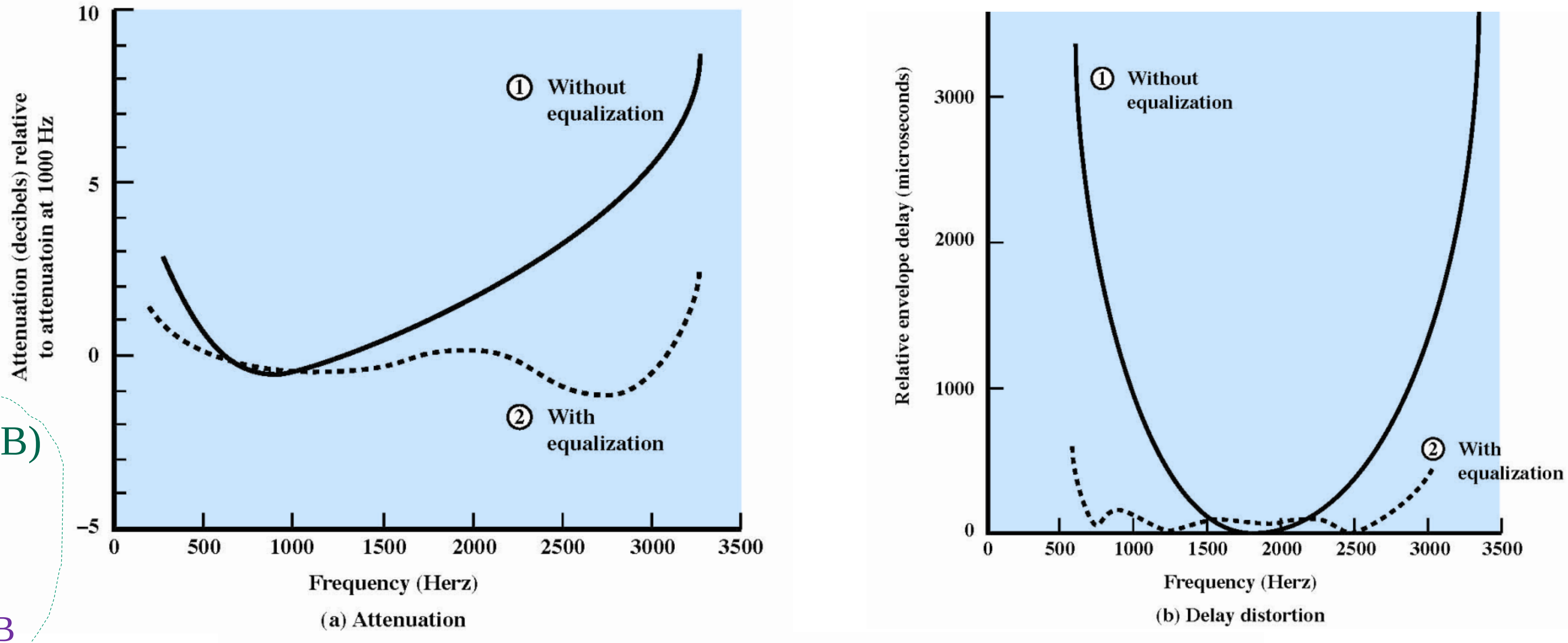
Chapter 6

Error Detection
Error Correction

Signal Impairment



⇒ Transmission Error



(a) Attenuation

(b) Delay distortion

Figure 3.15 Attenuation and Delay Distortion Curves for a Voice Channel

x :$10\log_{10}x$ (dB)

x=1: 0 dB

x=2: 3 dB

x=10: 10 dB

x=100: 20 dB

Types of Transmission Errors (1)

- An error occurs when a bit is altered between transmission and reception
- Single bit error
  - Isolated error that alters one bit but does not affect nearby bits
  - Can occur in the presence of white noise
- Burst error
  - Contiguous sequence of *B* bits in which the first and last bits and any number of intermediate bits are received in error
  - Can be caused by impulse noise or by fading in a mobile wireless environment
  - Effects of burst errors are greater at higher data rates

Types of Transmission Errors (2)

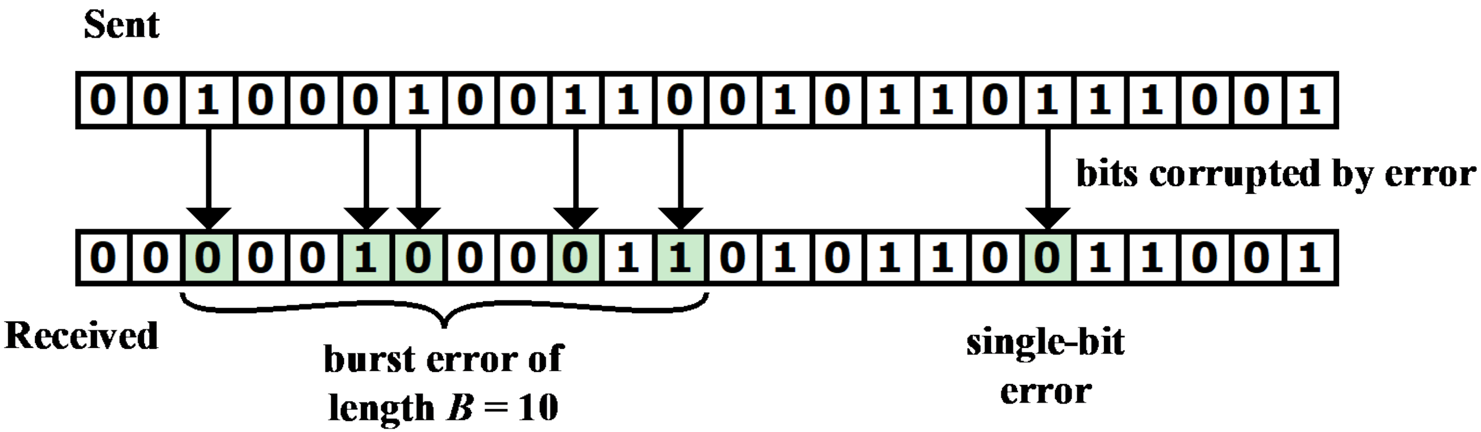
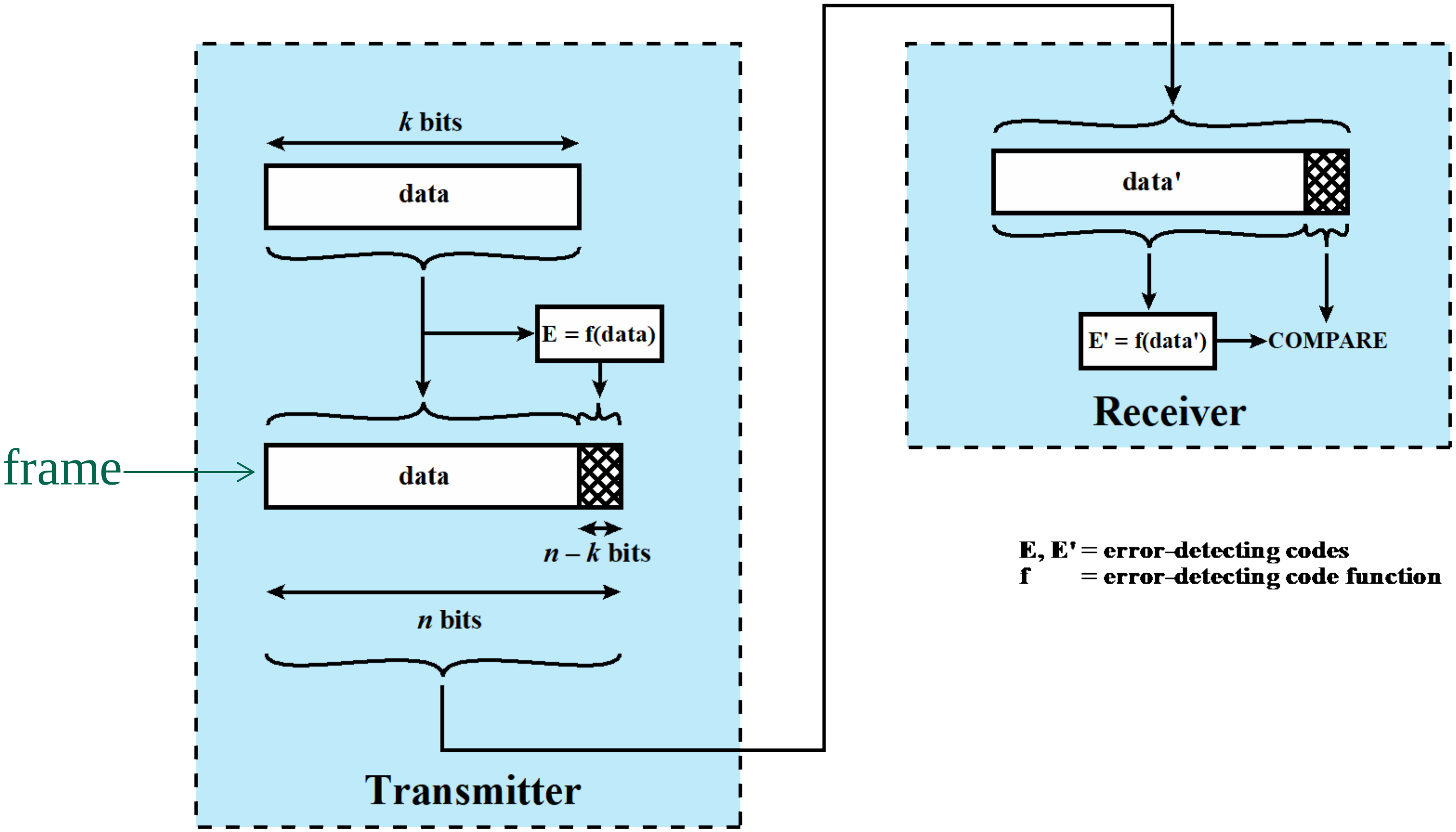


Figure 6.1 Burst and Single-Bit Errors

Coping with Data Transmission Errors

- Error detection and Retransmission
  - detect the presence of an error
  - Automatic repeat request (ARQ) protocols
    - Receiver discards a block of data with error Transmitter retransmits that block of data
- Error correction codes, or forward error correction (FEC)
  - Designed to detect and correct errors

Error Detection (1)



E, E' = error-detecting codes
f = error-detecting code function

Figure 6.2 Error Detection Process

Error Detection (2)

- Two common techniques
  - Parity checks
  - Cyclic redundancy checks (CRC)
- Parity check
  - One extra “parity” bit is added to each word
  - Simplest error detection technique
  - If any even number of bits are inverted due to error, an undetected error occurs
  - Single parity is very effective with white noise, but not very robust with noise bursts

Two-Dimensional Parity Check

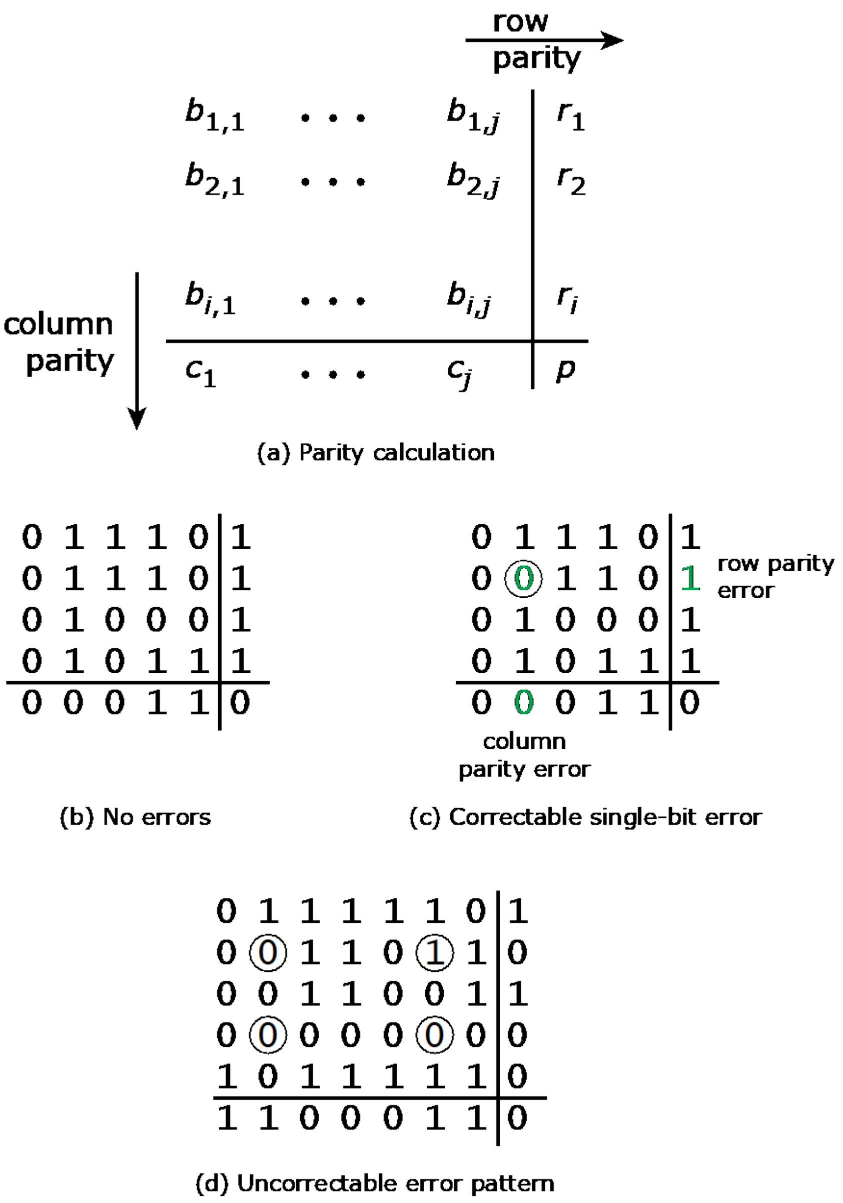


Figure 6.3 A Two-Dimensional Even Parity Scheme

Cyclic Redundancy Checks (1)

- Powerful error detection method
- Easily implemented
- Message (D) to be transmitted is appended with extra frame checksum bits (F), so that bit pattern transmitted (T) is perfectly divisible by a special “generator” pattern (P)
- At destination, divide received message by the same P.
- If remainder is nonzero ⇒ error
- Use modulo-2 arithmetic
  - no carries/borrows
  - add ≡ subtract ≡ xor

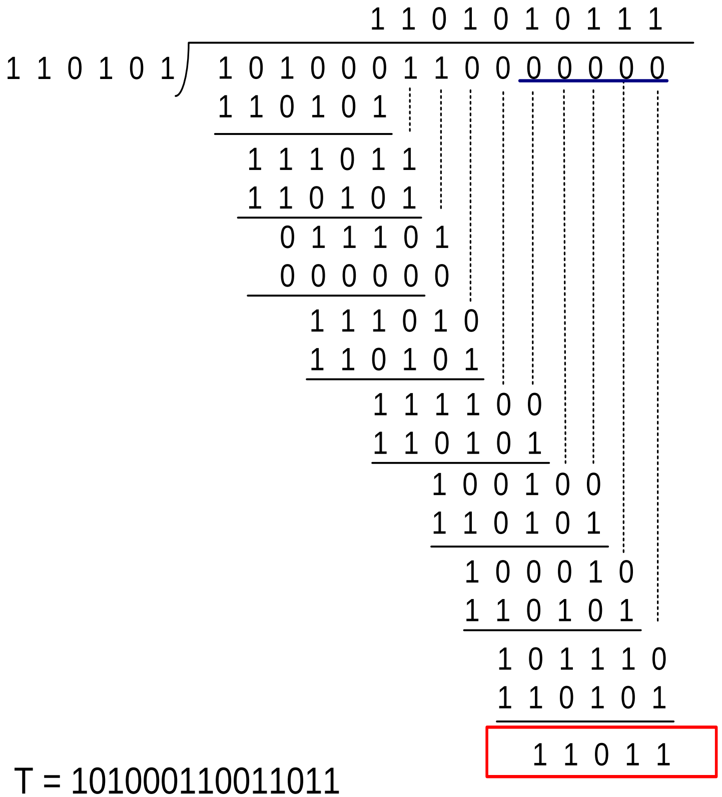
Cyclic Redundancy Checks (2)

- Let
  - T = n-bit frame to be transmitted,
  - D = k-bit message, the first k bits of T
  - F = (n-k)-bit FCS, the last n-k bits of T
  - P = n-k+1 bits, generator pattern (predetermined divisor)
- Method
  - Extend D with (n-k) ‘0’s to the right ($\equiv$ 2^{n-k}D)
  - Divide extended message by P to get R (2^{n-k}D/P=Q+R/P)
  - Add R to extended message to form T (T= 2^{n-k}D+R)
  - Transmit T
  - At receiver, divide T by P. Nonzero rem.$\Rightarrow$ error

$$\frac{T}{P}=\frac{2^{n-k}D+R}{P}=\frac{2^{n-k}D}{P}+\frac{R}{P}=Q+\frac{R+R}{P}=Q$$

Cyclic Redundancy Check (3)

Example 6.6: Message D = 1010001100, Pattern P = 110101



```
                          1 1 0 1 0 1 0 1 1 1
              _____________________________
1 1 0 1 0 1 ) 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0
              1 1 0 1 0 1
              -----------
                1 1 1 0 1 1
                1 1 0 1 0 1
                -----------
                  0 1 1 1 0 1
                  0 0 0 0 0 0
                  -----------
                    1 1 1 0 1 0
                    1 1 0 1 0 1
                    -----------
                      1 1 1 1 0 0
                      1 1 0 1 0 1
                      -----------
                        1 0 0 1 0 0
                        1 1 0 1 0 1
                        -----------
                          1 0 0 0 1 0
                          1 1 0 1 0 1
                          -----------
                            1 0 1 1 1 0
                            1 1 0 1 0 1
                            -----------
                                1 1 0 1 1
```

T = 101000110011011

Exercise:

Compute the frame to be transmitted for message 1101011011 using P=10011

- Answer: 11010110111110

Cyclic Redundancy Check (4)

- Can view CRC generation in terms of polynomial arithmetic
- Any bit pattern : polynomial in dummy variable X

Ex) $\mathrm{D} = 110011$

$$D(X) = 1 \cdot X^5 + 1 \cdot X^4 + 0 \cdot X^3 + 0 \cdot X^2 + 1 \cdot X^1 + 1 \cdot X^0$$
$$= X^5 + X^4 + X + 1$$

- CRC generation in terms of polynomial
  - Append (n-k) ‘0’s : $X^{n-k} D(X)$
  - Modulo 2 division: $\dfrac{X^{n-k} D(X)}{P(X)} = Q(X) + \dfrac{R(X)}{P(X)}$
  - Transmit $T(X) = X^{n-k} D(X) + R(X)$
  - At Receiver

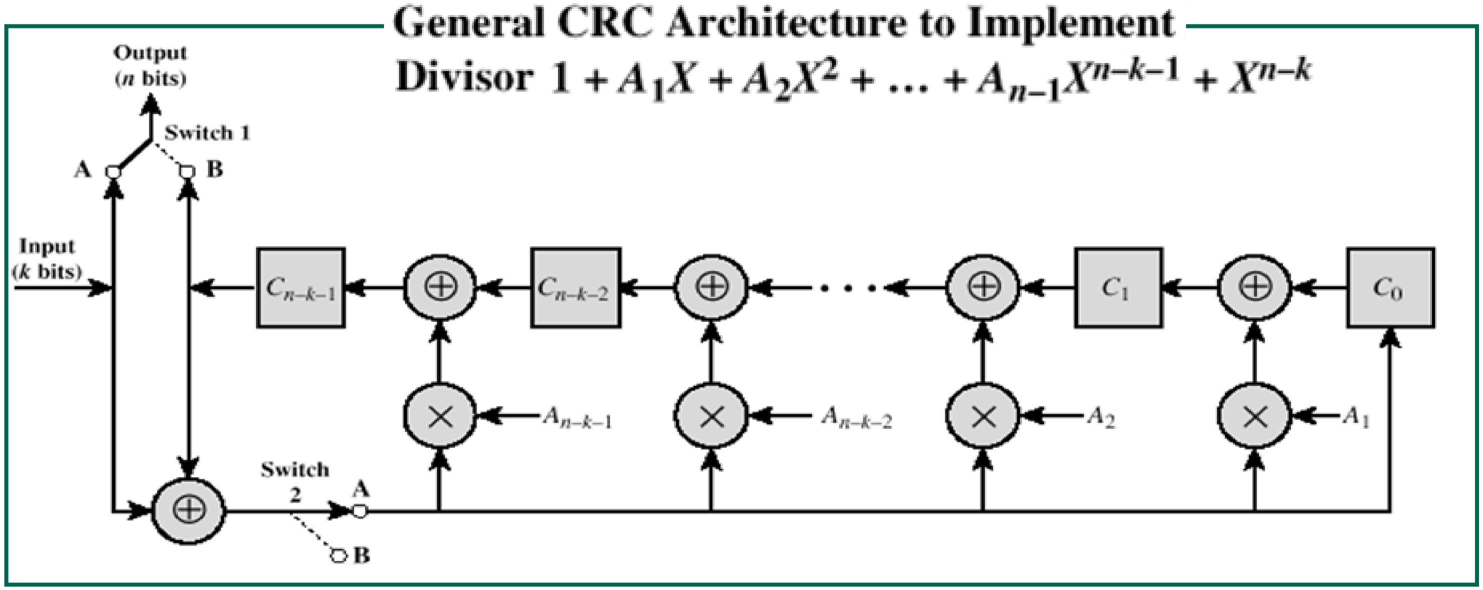
$$\frac{T(X)}{P(X)} = \frac{X^{n-k} D(X) + R(X)}{P(X)} = Q(X) + \frac{R(X) + R(X)}{P(X)}$$

Cyclic Redundancy Check (5)

- Commonly used polynomials, P(X)
  - CRC-12 = $X^{12}+X^{11}+X^{3}+X^{2}+X+1=(X+1)(X^{11}+X^{2}+1)$
  - CRC-ANSI = $X^{16}+X^{15}+X^{2}+1=(X+1)(X^{15}+X+1) \leftarrow X^{16}+X^{15}+X^{2}+X+X+1$
  - CRC-CCITT = $X^{16}+X^{12}+X^{5}+1$
    $= (X+1)(X^{15}+X^{14}+X^{13}+X^{12}+X^{4}+X^{3}+X^{2}+X+1)$
  - IEEE-802 = $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}$
    $+X^{7}+X^{5}+X^{4}+X^{2}+X+1$
- Can detect
  - All single-bit errors if P(X) has more than one nonzero term
  - All double-bit errors and any odd number of errors, as long as P(X) contains a factor (X+1).
  - Any burst error for which the length of the burst is less than or equal to $\underbrace{\text{the length of the FCS}}_{n-k}$.
  - A fraction of error burst of length n-k+1: $1-2^{-(n-k-1)}$
  - A fraction of error burst of length greater than n-k+1: $1-2^{-(n-k)}$

Cyclic Redundancy Check (6)

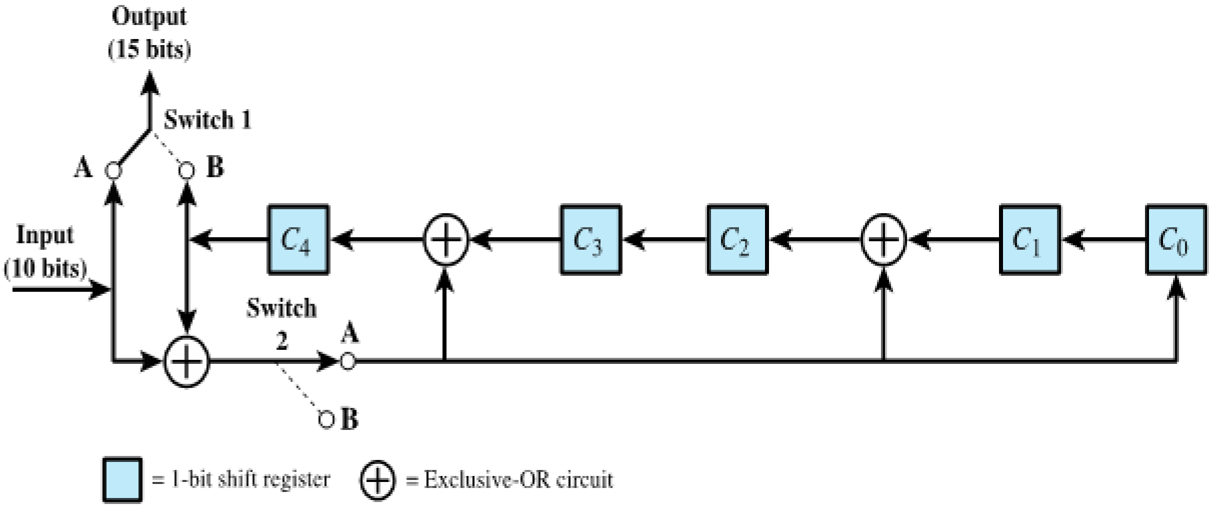
- Implementation
  - Implemented by a circuit consisting of exclusive-or gates and a shift register
    - The shift register contains (n-k) bits (length of FCS)
    - There are up to (n-k) exclusive-or gates
    - The presence or absence of a gate corresponds to the presence or absence of a term in P(X)



General CRC Architecture to Implement Divisor $1 + A_1X + A_2X^2 + \ldots + A_{n-1}X^{n-k-1} + X^{n-k}$

CRC Implementation Example

Circuit with Shift Registers for Dividing by the Polynomial $X^5 + X^4 + X^2 + 1$

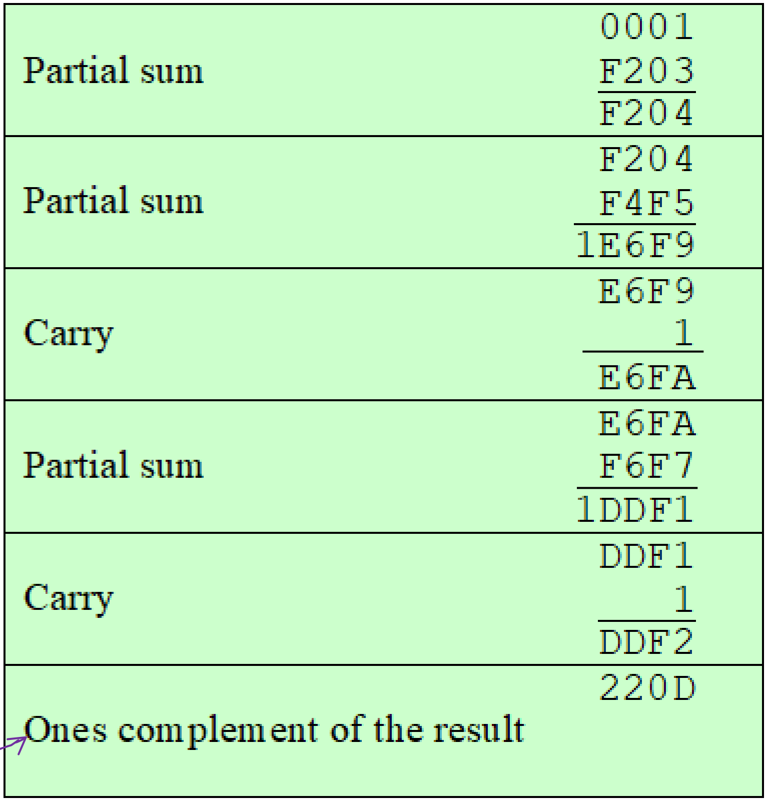


Internet Checksum

- Error detecting code used in many Internet standard protocols, including IP (IP header), TCP, and UDP (optional)
- Ones-complement addition
  - The two numbers are treated as unsigned binary integers and added
  - If there is a carry out of the leftmost bit, add 1 to the sum (end-around carry)
- Less effective than CRC
- Little overhead (implemented in software)
- It is assumed that at the lower link level, a strong code such as CRC is used
- An additional end-to-end checksum

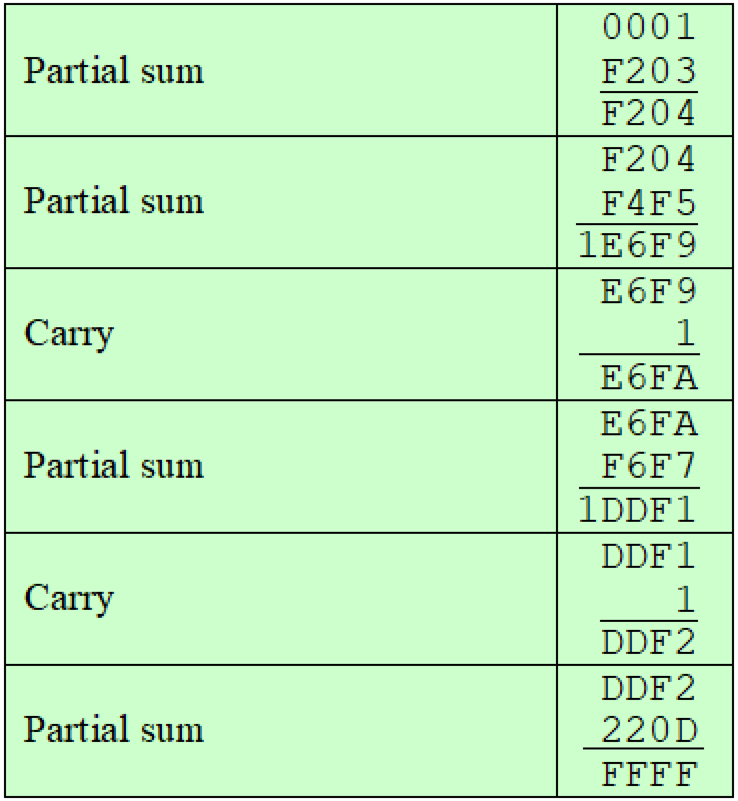
Internet Checksum Example

0001 F203 F4F5 F6F7 220D



Partial sum	0001 F203 F204
Partial sum	F204 F4F5 1E6F9
Carry	E6F9 1 E6FA
Partial sum	E6FA F6F7 1DDF1
Carry	DDF1 1 DDF2
Ones complement of the result	220D

(a) Checksum calculation by sender



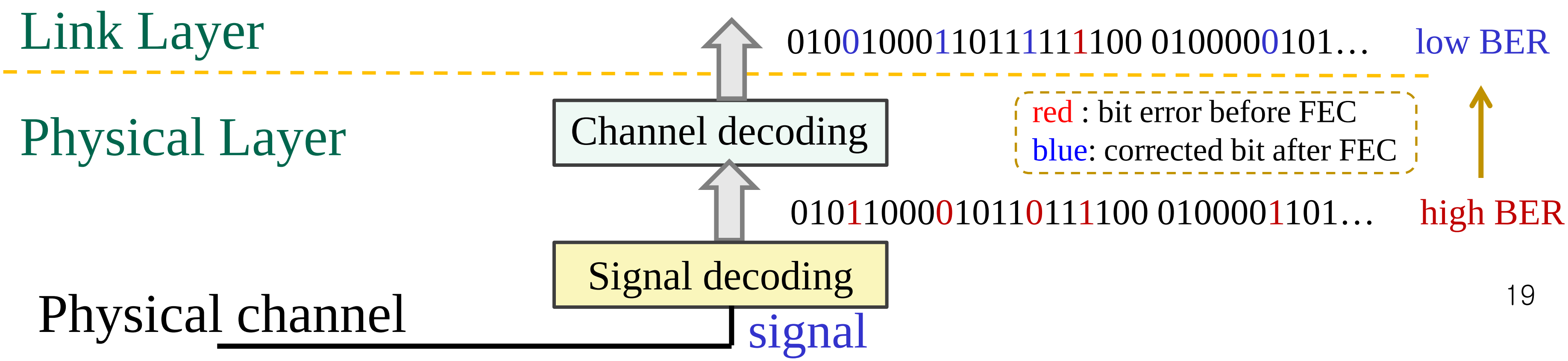
Partial sum	0001 F203 F204
Partial sum	F204 F4F5 1E6F9
Carry	E6F9 1 E6FA
Partial sum	E6FA F6F7 1DDF1
Carry	DDF1 1 DDF2
Partial sum	DDF2 220D FFFF

(b) Checksum verification by receiver

1101 1101 1111 0010 (DDF2)
0010 0010 0000 1101 (220D)

Error Correction (1)

- Forward error correction (channel coding)
  - enough redundancy is transmitted in the code that errors can be corrected by the receiver without retransmission
  - Block code
    - Mapping a data block to the corresponding codeword
    - Hamming code, BCH code, Reed-Solomon code
  - Convolutional code, turbo code
  - BER on physical channel to the BER requirements of the upper layer (layer 2)



Link Layer

Physical Layer

Physical channel

Channel decoding

Signal decoding

signal

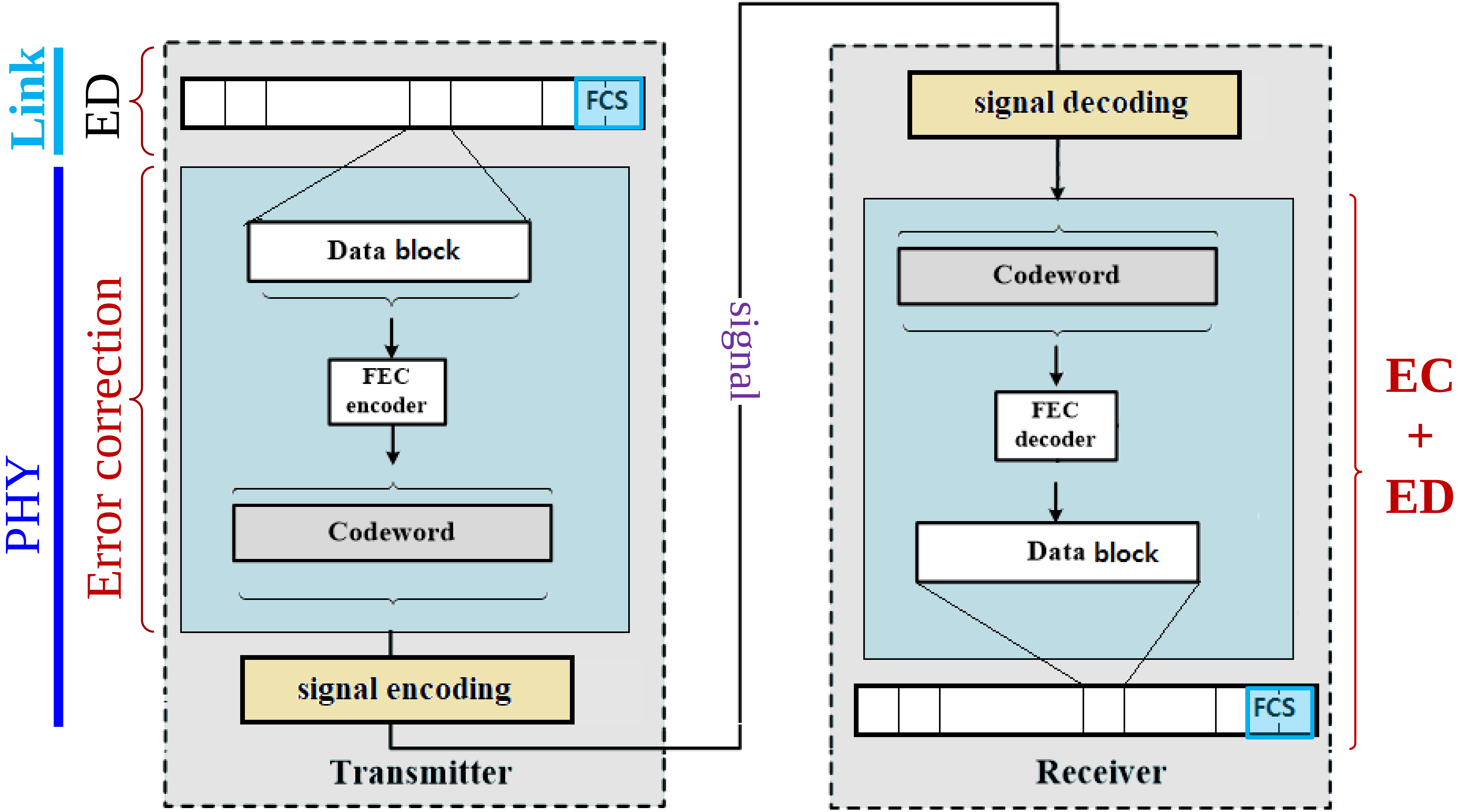
01001000110111111100 0100000101… low BER

01011000010110111100 0100001101… high BER

red : bit error before FEC

blue: corrected bit after FEC

Error Correction (2)



Link

ED

PHY

Error correction

FCS

Data block

FEC encoder

Codeword

signal encoding

Transmitter

signal

signal decoding

Codeword

FEC decoder

Data block

FCS

Receiver

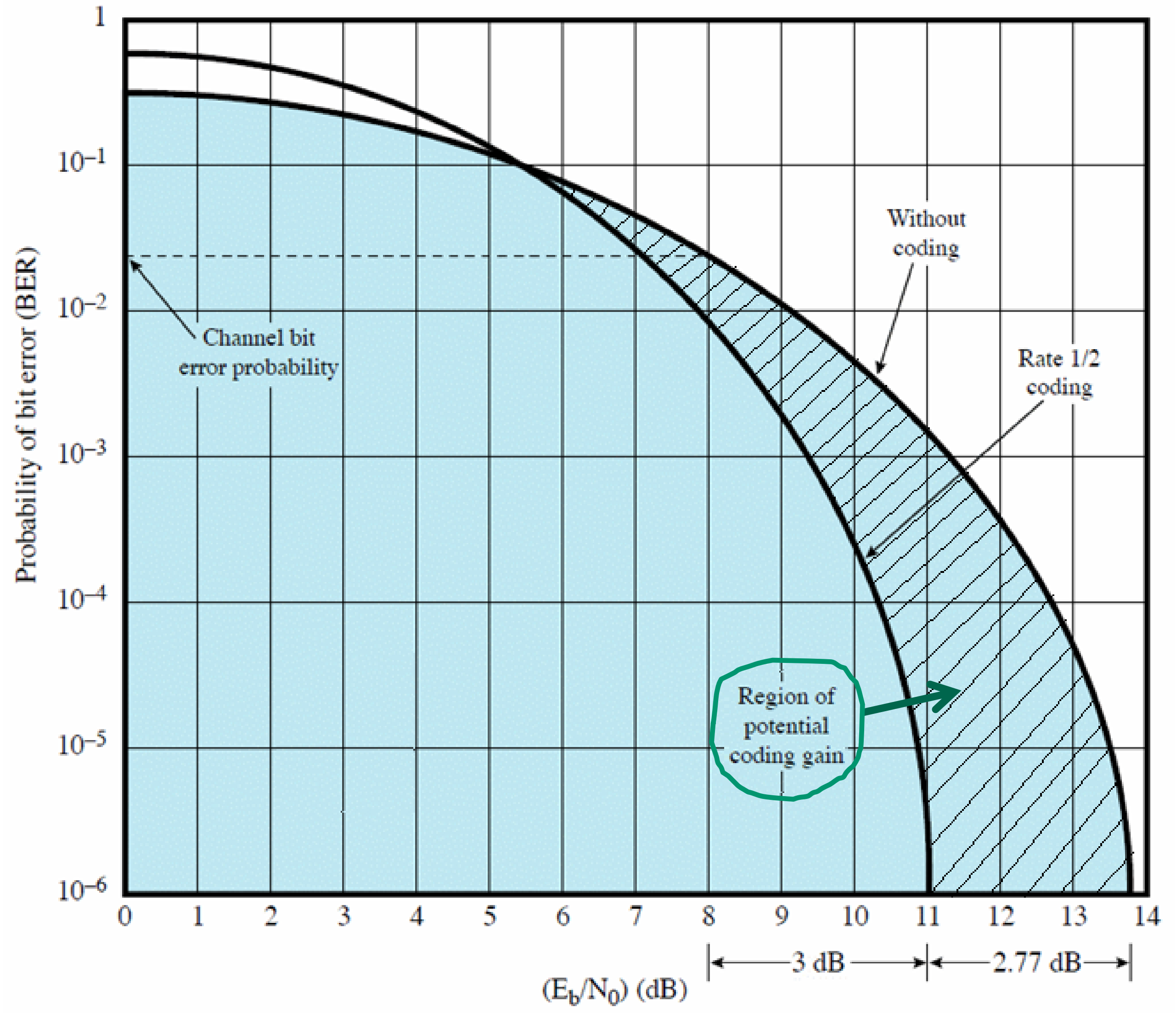
EC + ED

Error correction (Layer 1), Error Detection (Layer 2)

Basics

- Hamming distance
  - for 2 *n*-bit binary sequences, the number of different bits
  - E.g., v_1=011011; v_2=110001; d(v1, v_2)=3
- Coding rate
  - ratio of data bits to total bits
- Coding gain
  - the reduction in the required E_b/N_0 to achieve a specified BER of an error-correcting coded system

Coding Gain



1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
Probability of bit error (BER)
Channel bit error probability
Without coding
Rate 1/2 coding
Region of potential coding gain
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
3 dB
2.77 dB
(E_b/N_0) (dB)

Block Code Example

Example 6.9

For k=2, n=5

Data block	Codeword
00	00000
01	00111
10	11001
11	11110

- Minimum distance 1: 00001 ⇒ 00000 , 00011 ⇒ 00111 : correction
- Minimum distance 2: 01010 ⇒ 00000 or 11110 : detection
- Singe bit error correction and double bit error detection

BCH code (Chap 16.)

- A kind of block code
- Cyclic code
  - Valid code: $(c_0,c_1, \ldots, c_{n-1}) \overset{shift}{\Longrightarrow} (c_{n-1},c_0,\ldots, c_{n-2})$
- BCH (n,k)
  - Data block length: k
  - Error check bit: n-k
  - Codeword length: n
- Example: BCH(7,4)
  - a single bit error correction
  - $P(X)=X^3+X+1$

	A single bit error	syndrome
1	0000001	001
X	0000010	010
X^2	0000100	100
X^3	0001000	011
X^4	0010000	110
X^5	0100000	111
X^6	1000000	101

$$
\begin{array}{r|l}
 & X^3+ \quad X + 1 \\
\hline
X^3+X+1 & X^6 \\
 & X^6+ \quad X^4+X^3 \\
\cline{2-2}
 & \quad X^4+X^3 \\
 & \quad X^4+ \quad X^2+X \\
\cline{2-2}
 & \quad\quad X^3+X^2+X \\
 & \quad\quad X^3+ \quad X+1 \\
\cline{2-2}
 & \quad\quad\quad X^2+ \quad 1 \Rightarrow 101
\end{array}
$$

BCH(7,4)

Data Block	Valid Codeword
0000	0000000
0001	0001011
0010	0010110
0011	0011101
0100	0100111
0101	0101100
0110	0110001
0111	0111010
1000	1000101
1001	1001110
1010	1010011
1011	1011000
1100	1100010
1101	1101001
1110	1110100
1111	1111111

❖ BCH(15,5): can correct three or fewer bit-errors

BCH code (7,4)

Data Block	Valid Codeword
0000	0000000
0001	0001011
0010	0010110
0011	0011101
0100	0100111
0101	0101100
0110	0110001
0111	0111010
1000	1000101
1001	1001110
1010	1010011
1011	1011000
1100	1100010
1101	1101001
1110	1110100
1111	1111111

Example:

Transmitter

Data Block: 1010
Codeword: 1010011 ⇒ $T(X)=X^6+X^4+X+1$

A single bit error	syndrome
0000001	001
0000010	010
0000100	100
0001000	011
0010000	110
0100000	111
1000000	101

Receiver

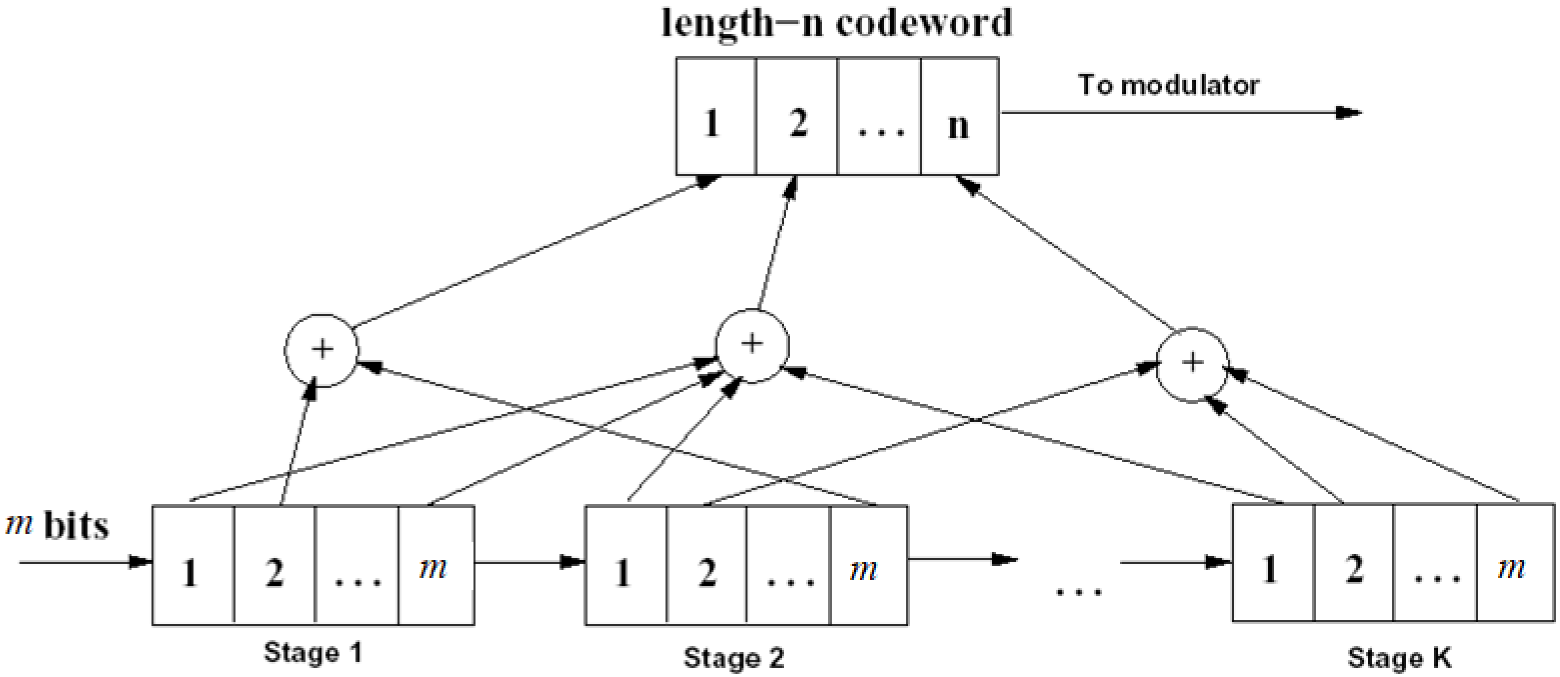
$R(X)=X^6+X+1$ (1000011)

$$
\begin{array}{r|l}
 & X^3+\qquad X+1 \\
\hline
X^3+X+1 & X^6+\qquad\qquad\quad X+1 \\
 & X^6+\quad X^4+X^3 \\
\hline
 & \qquad X^4+X^3+\quad X \\
 & \qquad X^4+\quad X^2+X \\
\hline
 & \qquad\quad X^3+X^2+\quad 1 \\
 & \qquad\quad X^3+\qquad X+1 \\
\hline
 & \qquad\qquad X^2+X \quad \Rightarrow 110
\end{array}
$$

$C(X)=R(X)+E(X)$
$=X^6+X+1+X^4$ (1010011)

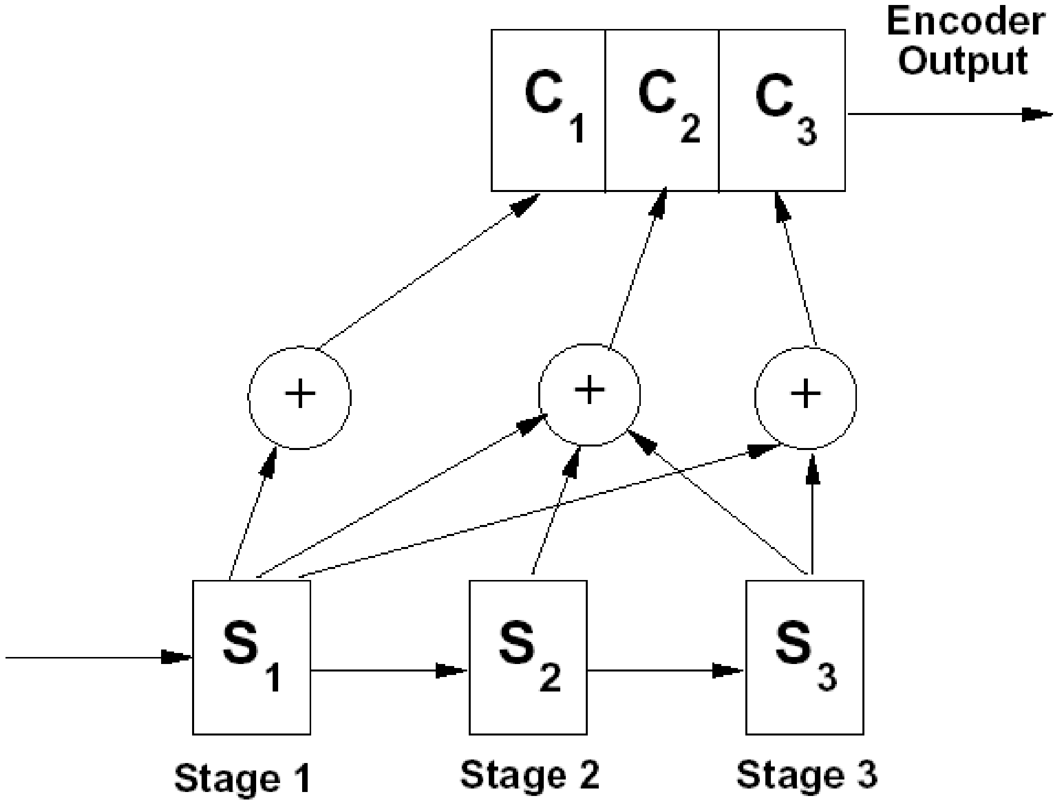
Convolutional Encoder (1)

- The encoder generates a codeword of length *n* for *m*-bit input sequence
  - a shift register: *K* stages with *m* bits per stage (*m*-bits shift at a time)
  - *n* binary addition operator
  - Constraint length: *mK* bits



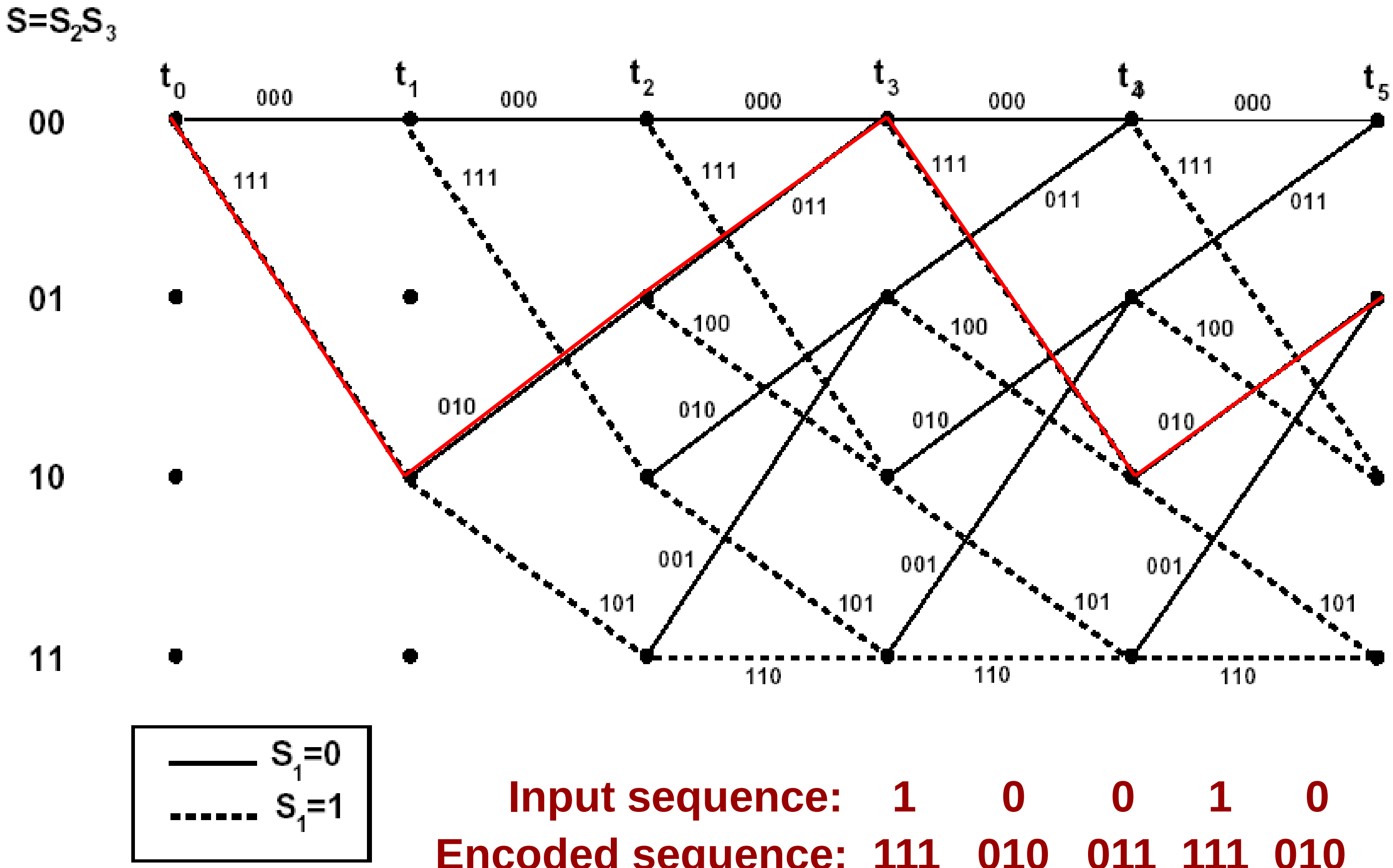
Convolutional Encoder (2)

- Example (*n*=3, *m*=1, *K*=3)



Convolutional Encoder (3)

- Trellis Diagram

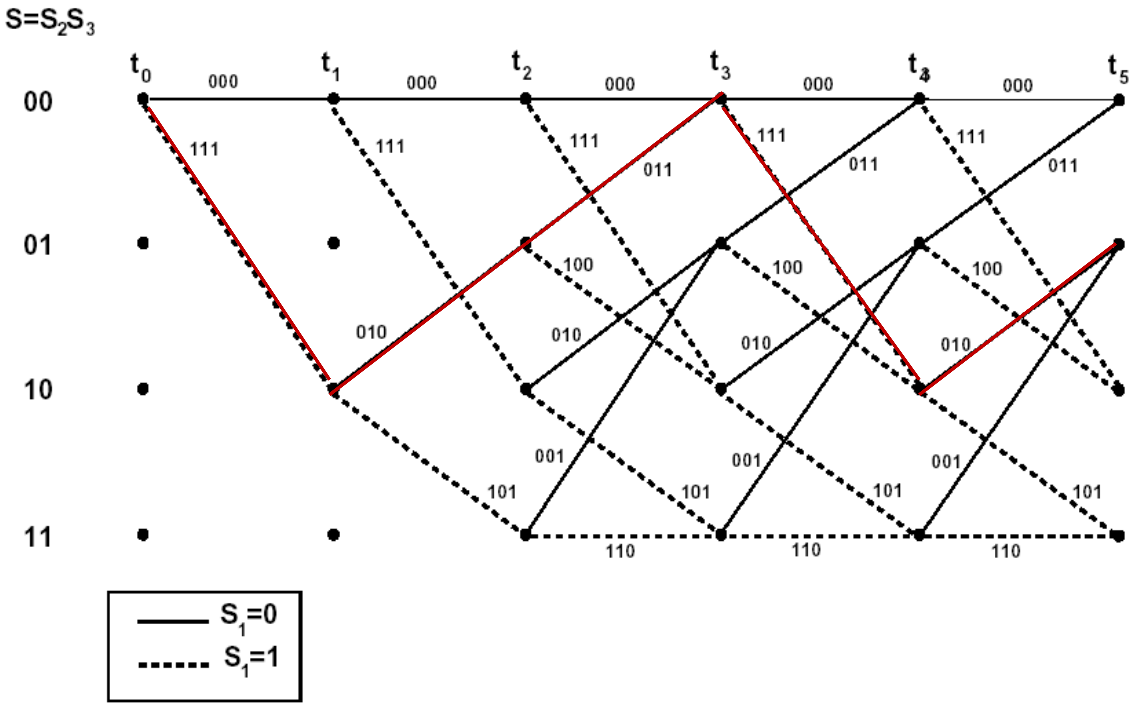


Input sequence: 1 0 0 1 0

Encoded sequence: 111 010 011 111 010

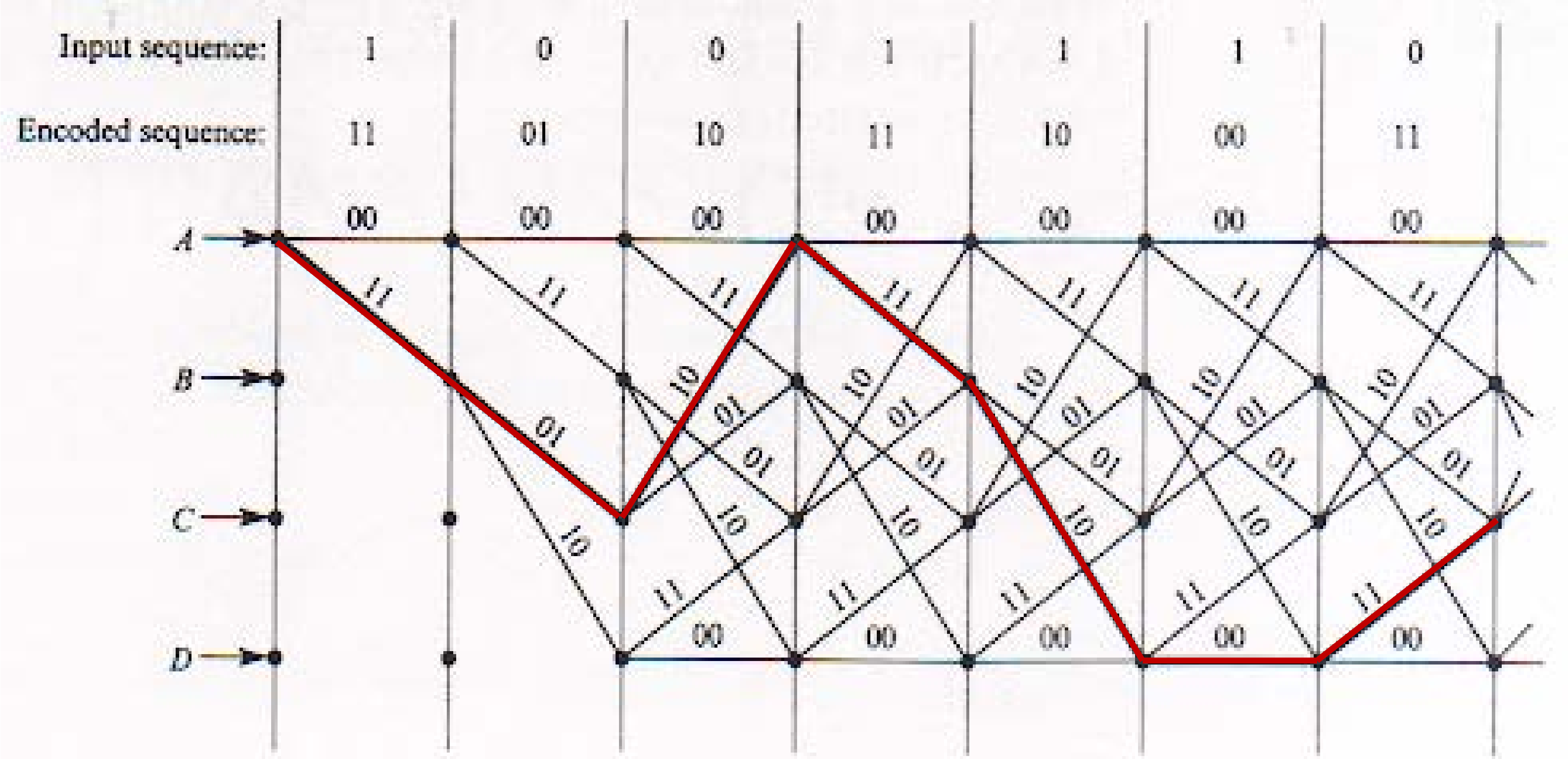
Convolutional Code Decoding (1)

- Example 1
  - Input Sequence : 1 0 0 1 0 …
  - Encoded (Output) sequence : 111 010 011 111 010 …
  - Corrupted encoded sequence : 111 010 111 111 011 …
  - Decoded sequence: 1 0 0 1 0 … (error correction)



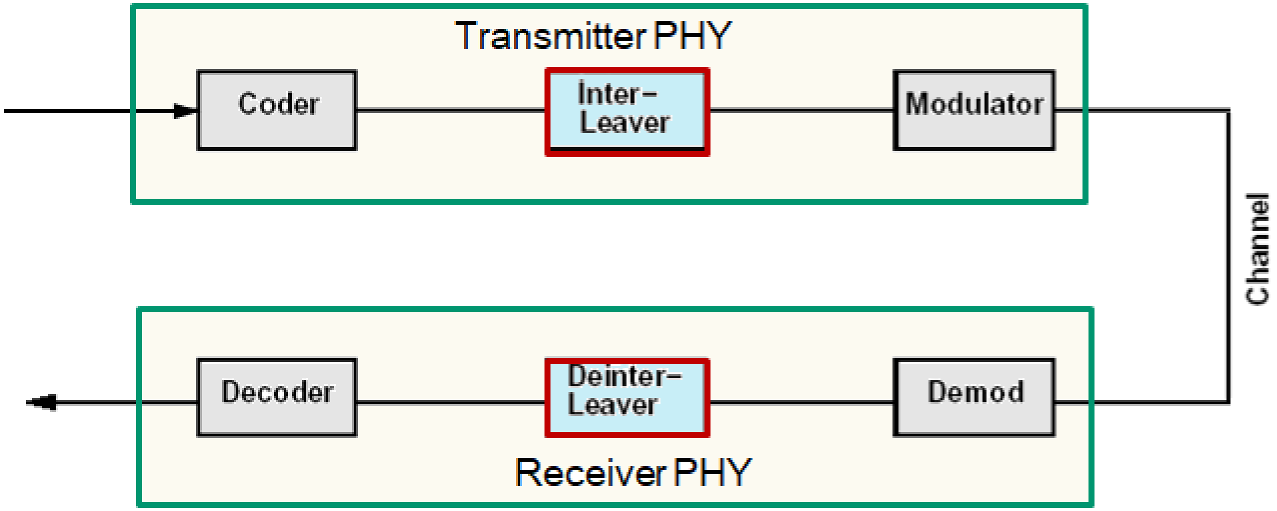
Convolutional Code Decoding (2)

- Example 2
  - Input Sequence : 1 0 0 1 1 1 0 …
  - Encoded (Output) sequence : 11 01 10 11 10 00 11 …
  - Corrupted encoded sequence : 11 01 00 11 11 00 11 …
  - Decoded sequence: 1 0 0 1 1 1 0 … (error correction)

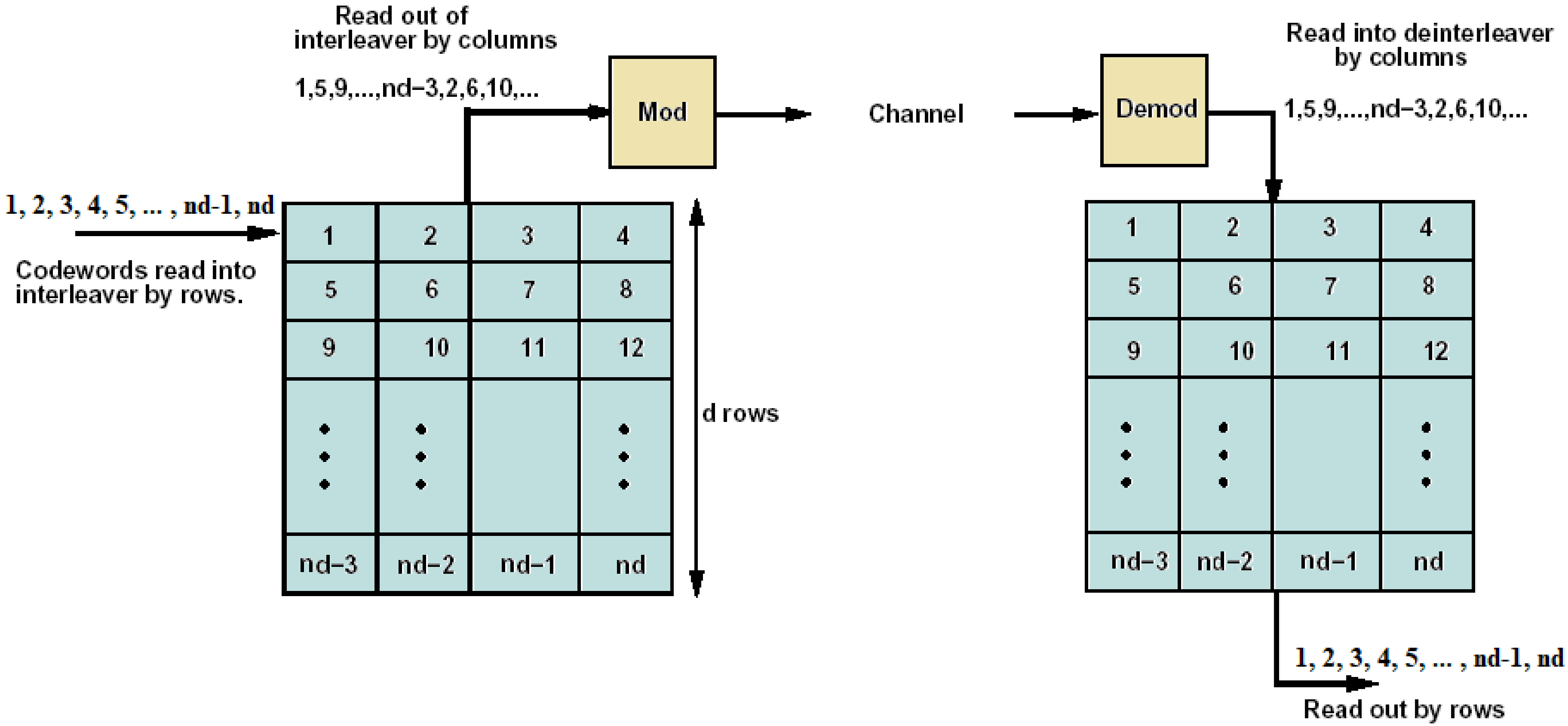


Interleaving(1)

- To mitigate the effects of error bursts, coding is typically combined with interleaving.
  - Deinterleaver: spreading out error bursts
  - Channel decoder: error correction over the spread error



Interleaving (2)



1, 2, 3, 4, 5, ... , nd-1, nd

Codewords read into interleaver by rows.

Read out of interleaver by columns

1,5,9,...,nd−3,2,6,10,...

Mod

Channel

Demod

Read into deinterleaver by columns

1,5,9,...,nd−3,2,6,10,...

d rows

1	2	3	4
5	6	7	8
9	10	11	12
• • •	• • •		• • •
nd−3	nd−2	nd−1	nd

1	2	3	4
5	6	7	8
9	10	11	12
• • •	• • •		• • •
nd−3	nd−2	nd−1	nd

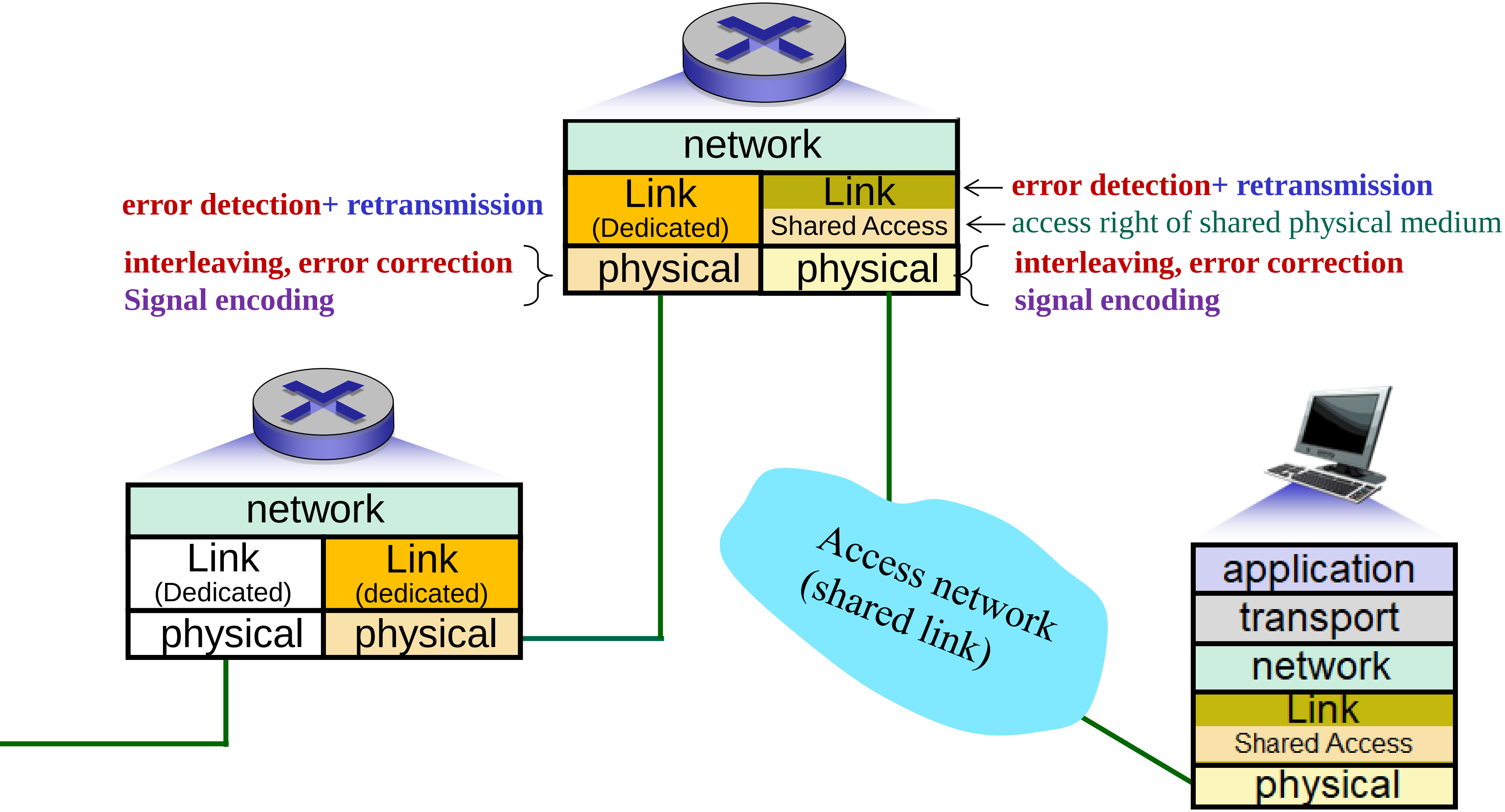
1, 2, 3, 4, 5, ... , nd-1, nd

Read out by rows

INTERLEAVER
(Transmitter)

DE−INTERLEAVER
(Receiver)

Rreview



error detection+ retransmission

interleaving, error correction
Signal encoding

error detection+ retransmission
access right of shared physical medium

interleaving, error correction
signal encoding