Computer Networks

Error Detection and Correction

Amitangshu Pal
Computer Science and Engineering
IIT Kanpur

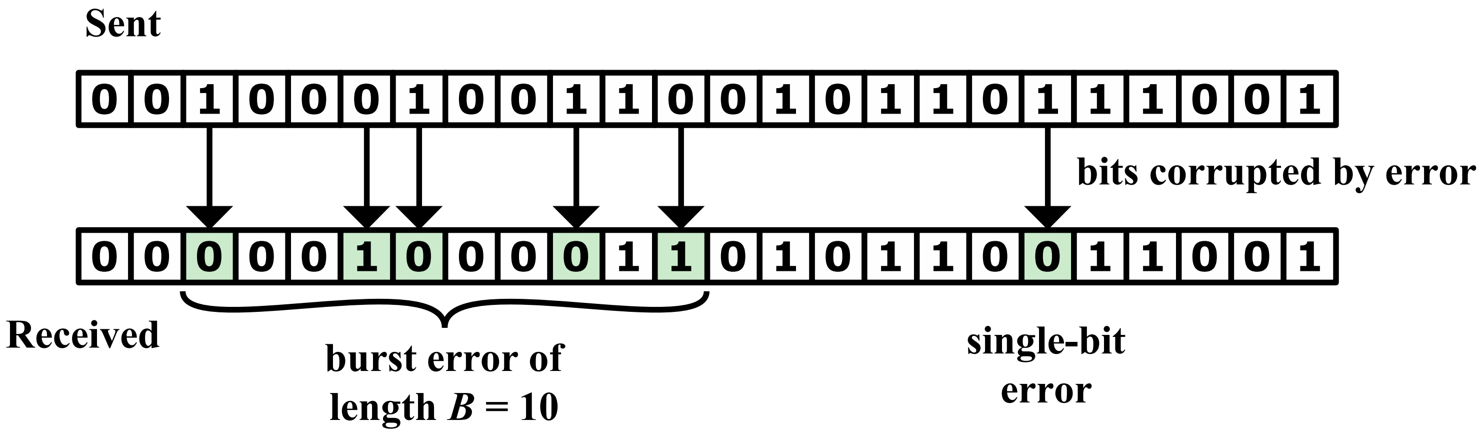
Types of Errors

- An error occurs when a bit is altered between transmission and reception

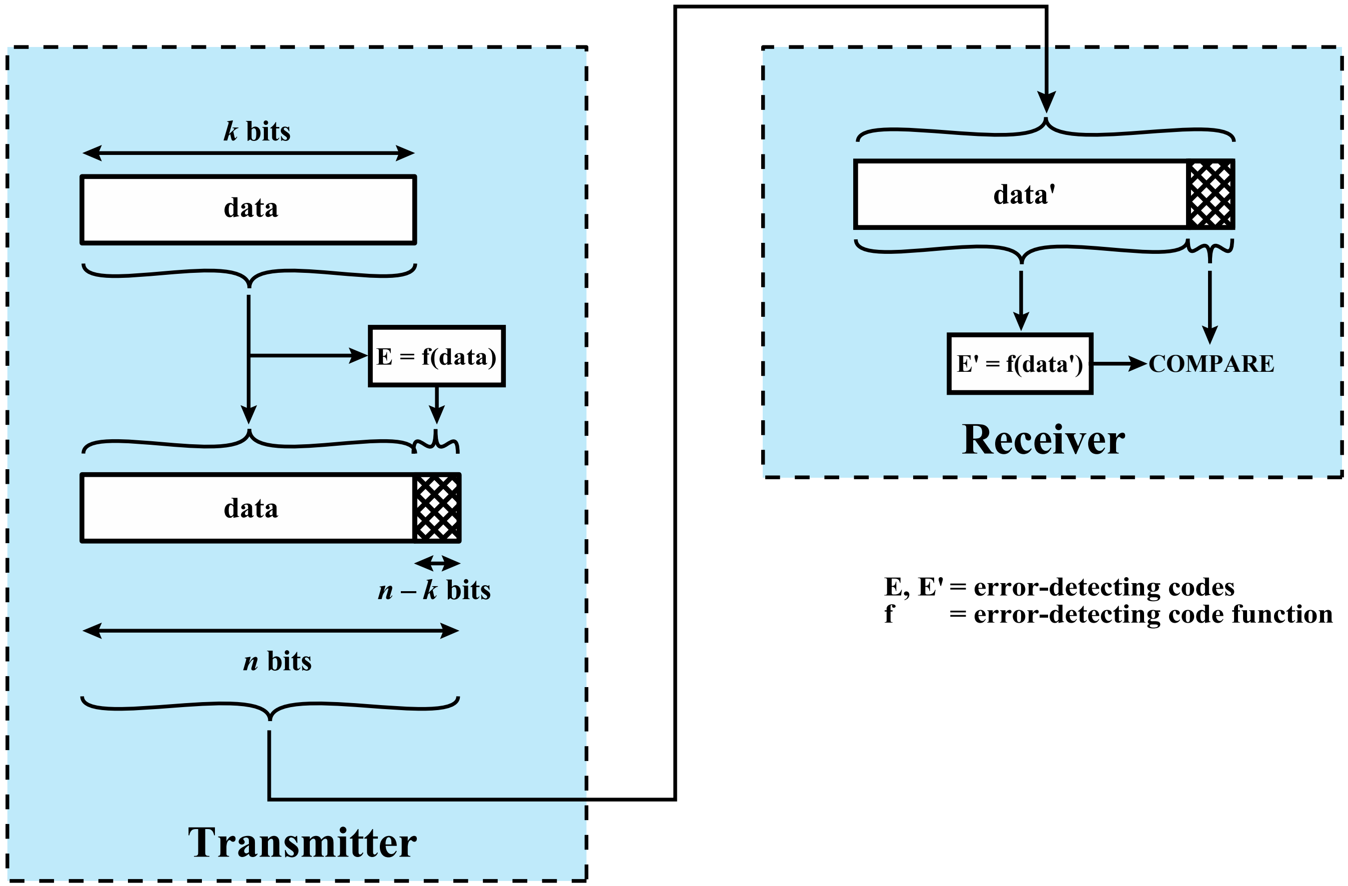
- Single bit error:
  - Isolated error that alters one bit but does not affect nearby bits
  - Can occur in presence of white noise

- 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 occur due to impulse noise or fading in wireless environment
  - Effects are greater at higher data rates

Types of Errors



Error Detection

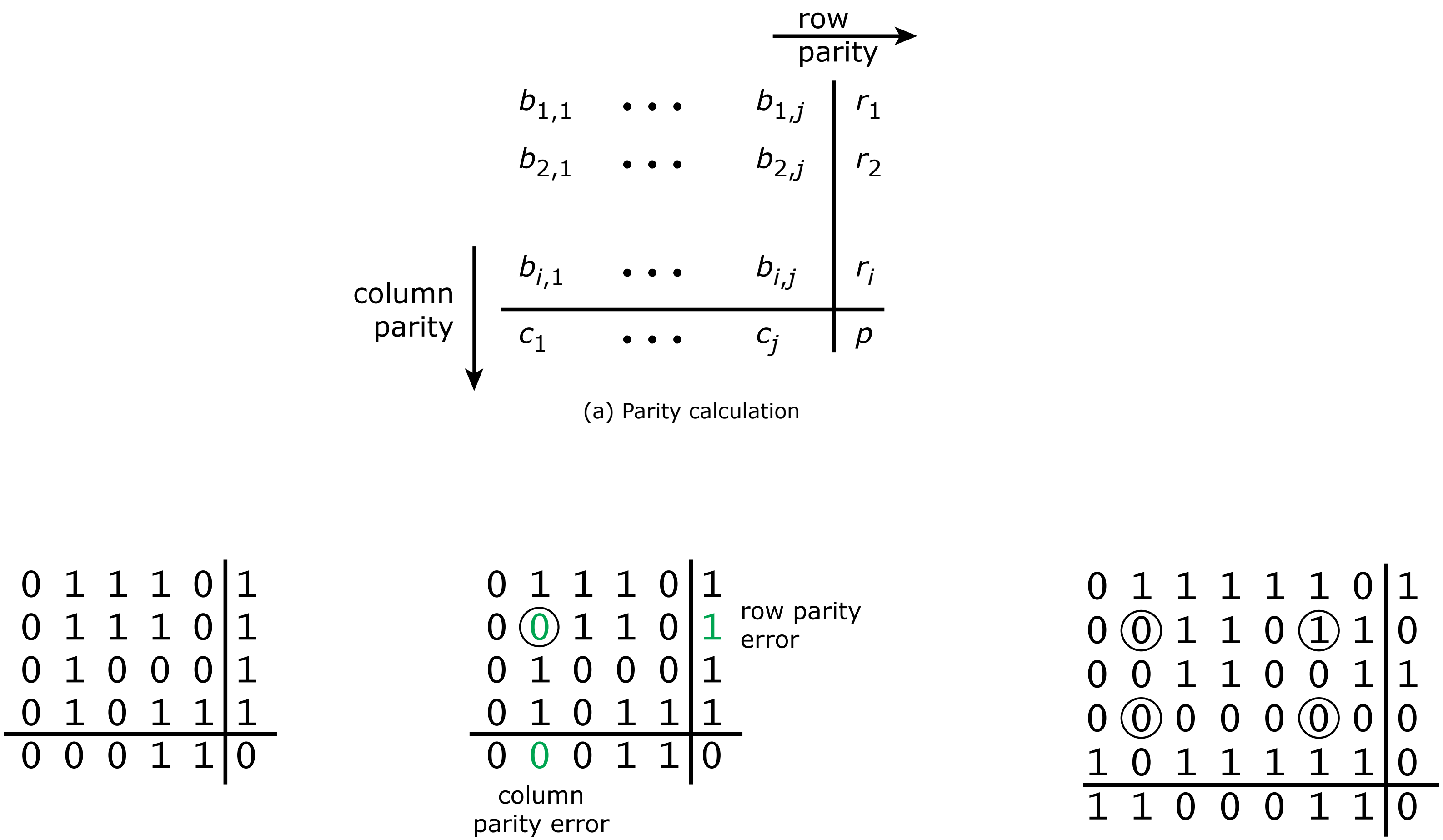


Parity Check

- The simplest error detecting scheme is to append a parity bit to the end of a block of data
  - Even parity: Total number of ones will be odd
  - Odd parity: Total number of ones will be even

- If any even number of bits are inverted due to error, an undetected error occurs

Two-dimensional Parity Check



(a) Parity calculation

(b) No errors

(c) Correctable single-bit error

(d) Uncorrectable error pattern

The Internet Checksum

- Error detecting code used in many Internet standard protocols, including IP, TCP, and UDP
- Ones-complement operation
  - Replace 0 digits with 1 digits and 1 digits with 0 digits
- 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)

The Internet Checksum

- 00 01 F2 03 F4 F5 F6 F7 00 00

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

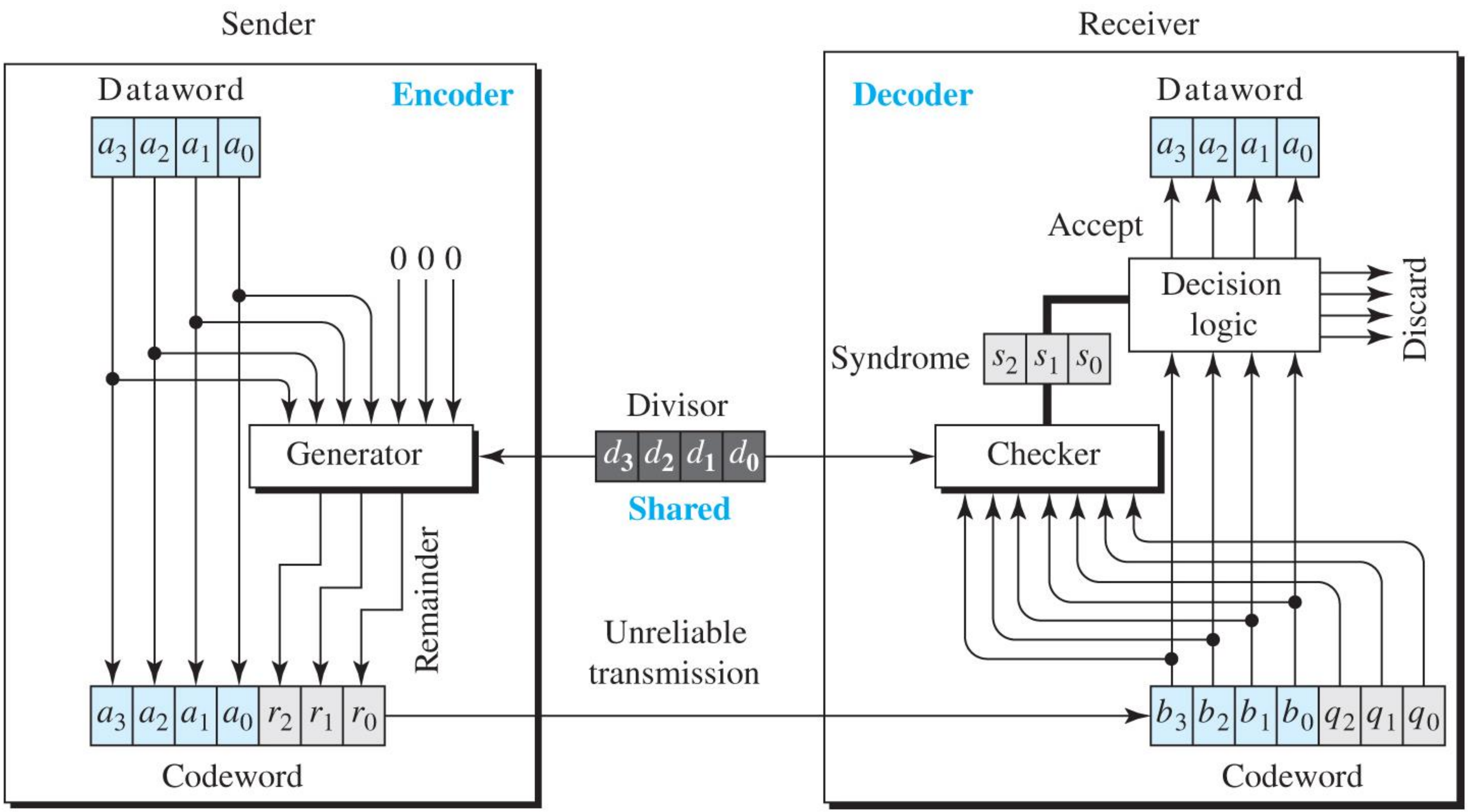
Partial sum	0001 F203 F204
Partial sum	F204 F4F5 1E6F9
Carry	E6F9 1 E6FA
Partial sum	E6FA F6F7 1DDF1
Carry	DDF1 1 DDF2
Partial sum	DDF2 220D FFFF

(b) Checksum verification by receiver

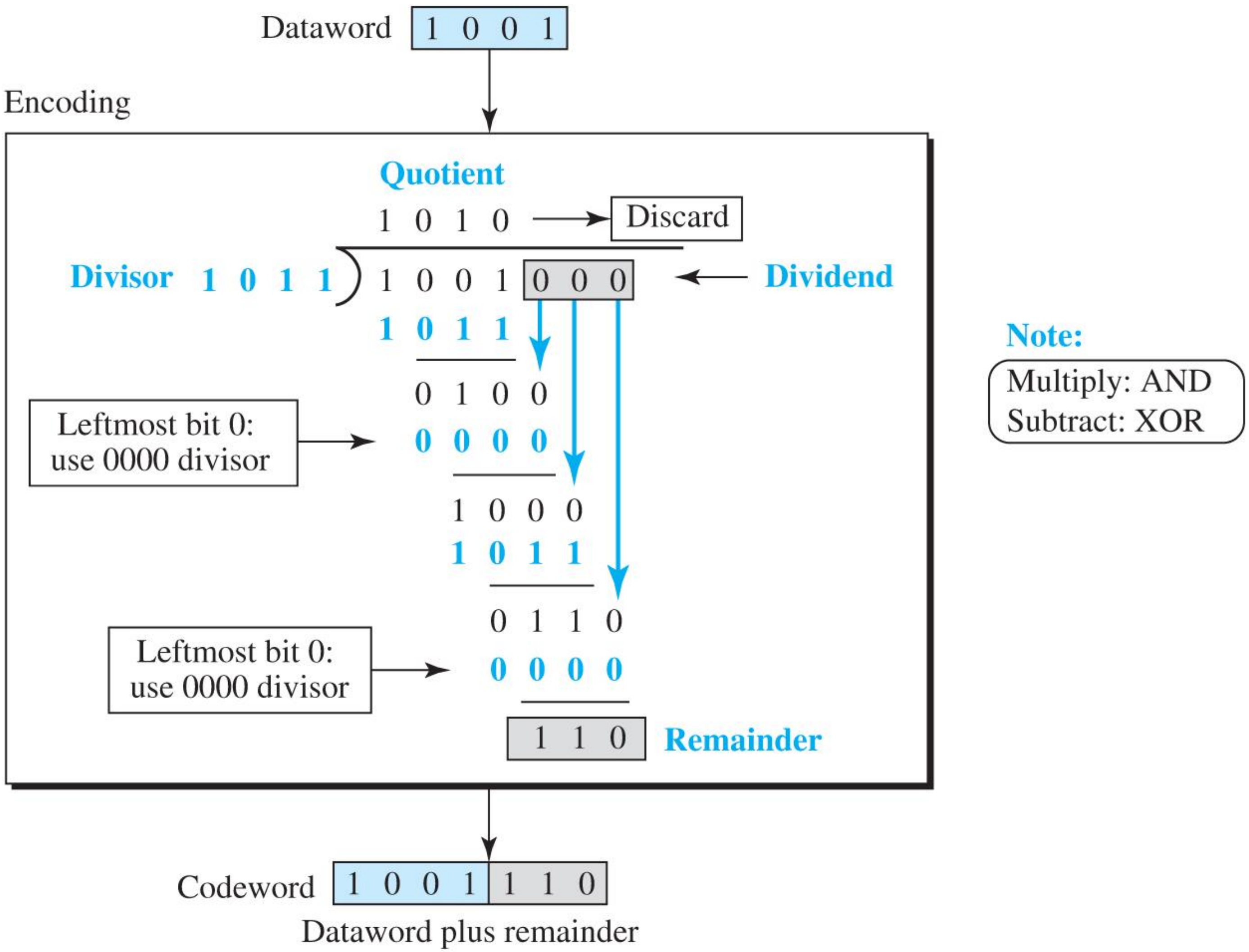
Cyclic Redundancy Check (CRC)

- One of the most common and powerful error-detecting codes
- Given a block of bits, the transmitter generates an bit frame check sequence (FCS) which is exactly divisible by some predetermined number
- Receiver divides the incoming frame by that number
  - If there is no remainder, assume there is no error

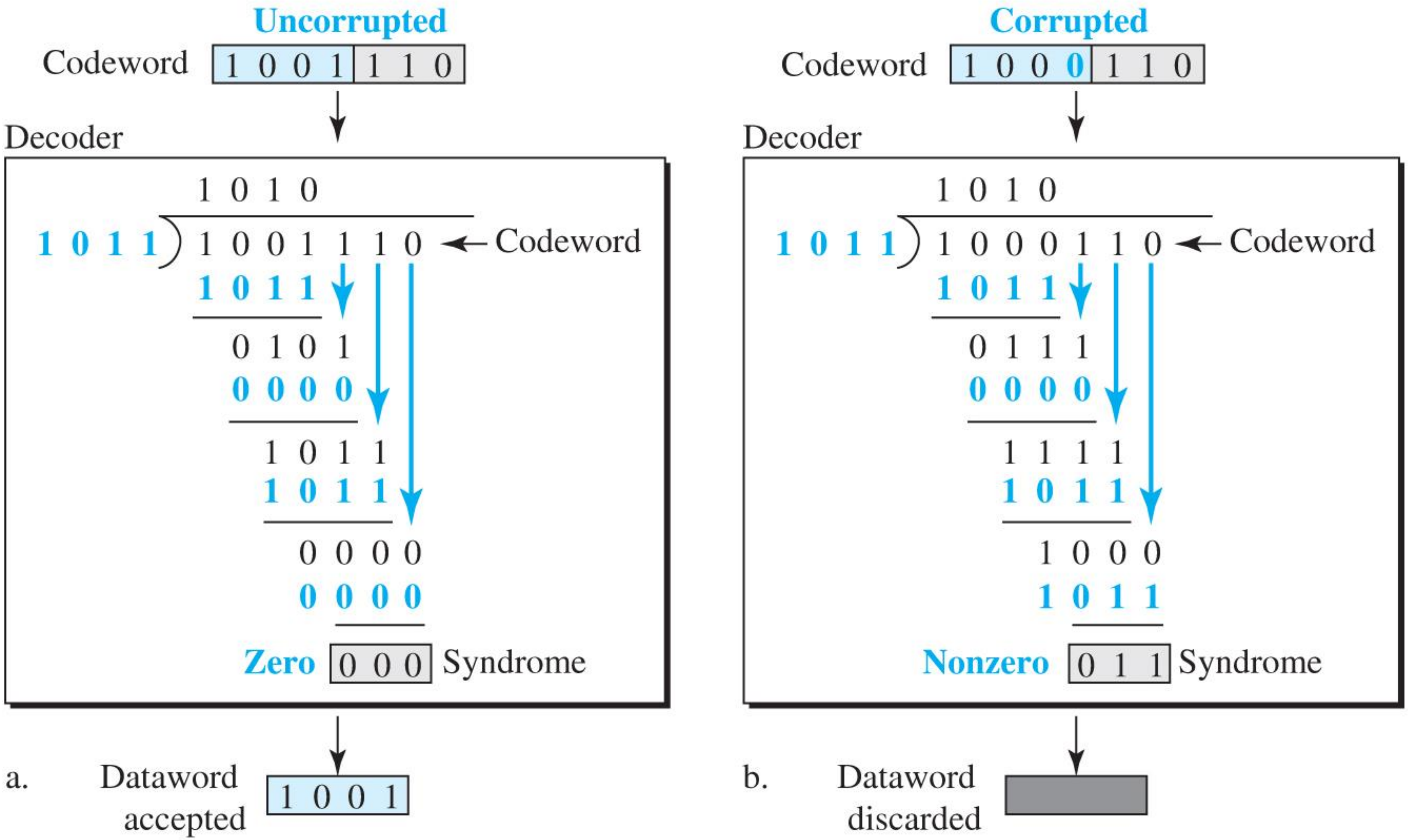
Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)

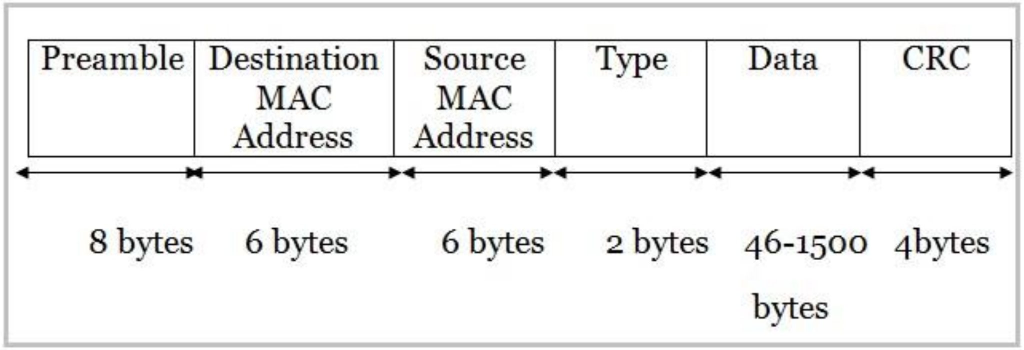


Cyclic Redundancy Check (CRC)

$$
\begin{array}{rl}
 & \quad X^9 + X^8 + X^6 + X^4 + X^2 + X \qquad\qquad \leftarrow Q(X) \\
P(X) \rightarrow X^5 + X^4 + X^2 + 1 & \overline{\big)\, X^{14} \quad X^{12} \qquad\qquad X^8 + X^7 + \quad X^5} \quad \leftarrow X^5 D(X) \\
 & \quad \underline{X^{14} + X^{13} + \quad X^{11} + \quad X^9} \\
 & \qquad X^{13} + X^{12} + X^{11} + \quad X^9 + X^8 \\
 & \qquad \underline{X^{13} + X^{12} + \quad X^{10} + \quad X^8} \\
 & \qquad\qquad X^{11} + X^{10} + X^9 + \quad X^7 \\
 & \qquad\qquad \underline{X^{11} + X^{10} + \quad X^8 + \quad X^6} \\
 & \qquad\qquad\qquad X^9 + X^8 + X^7 + X^6 + X^5 \\
 & \qquad\qquad\qquad \underline{X^9 + X^8 + \quad X^6 + \quad X^4} \\
 & \qquad\qquad\qquad\qquad X^7 + \quad X^5 + X^4 \\
 & \qquad\qquad\qquad\qquad \underline{X^7 + X^6 + \quad X^4 + \quad X^2} \\
 & \qquad\qquad\qquad\qquad\quad X^6 + X^5 + \quad X^2 \\
 & \qquad\qquad\qquad\qquad\quad \underline{X^6 + X^5 + \quad X^3 + \quad X} \\
 & \qquad\qquad\qquad\qquad\qquad X^3 + X^2 + X \quad \leftarrow R(X)
\end{array}
$$

Cyclic Redundancy Check (CRC)

Name	Binary	Application
CRC-8	100000111	ATM header
CRC-10	11000110101	ATM AAL
CRC-16	10001000000100001	HDLC
CRC-32	100000100110000010001110110110111	LANs



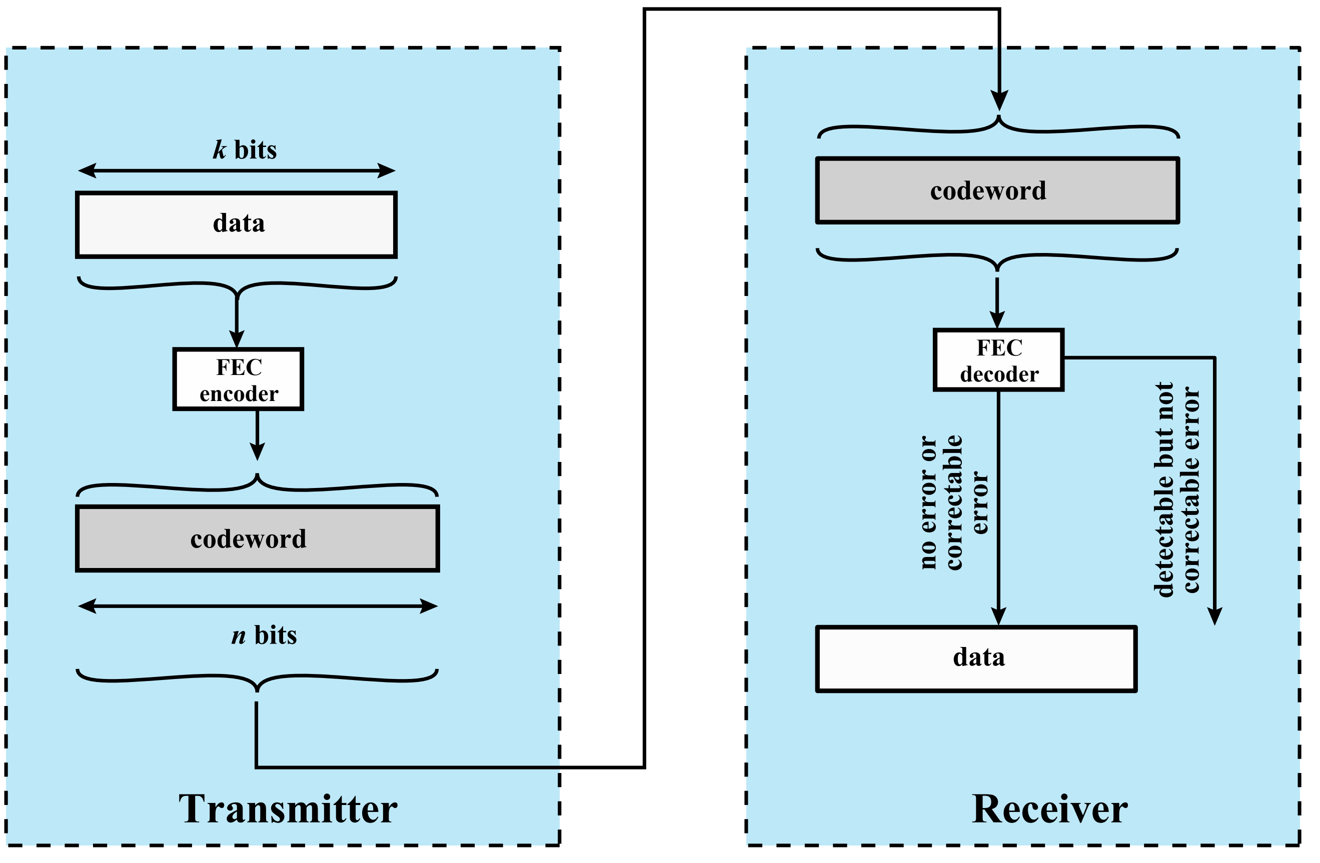
Src: https://www.minitool.com/lib/ethernet-frame.html

Error Correction

- Backword error correction: Correction of detected errors usually requires data blocks to be retransmitted
- Not appropriate for wireless applications:
  - The bit error rate (BER) on a wireless link can be quite high, which would result in a large number of retransmissions
  - Propagation delay is very long compared to the transmission time of a single frame

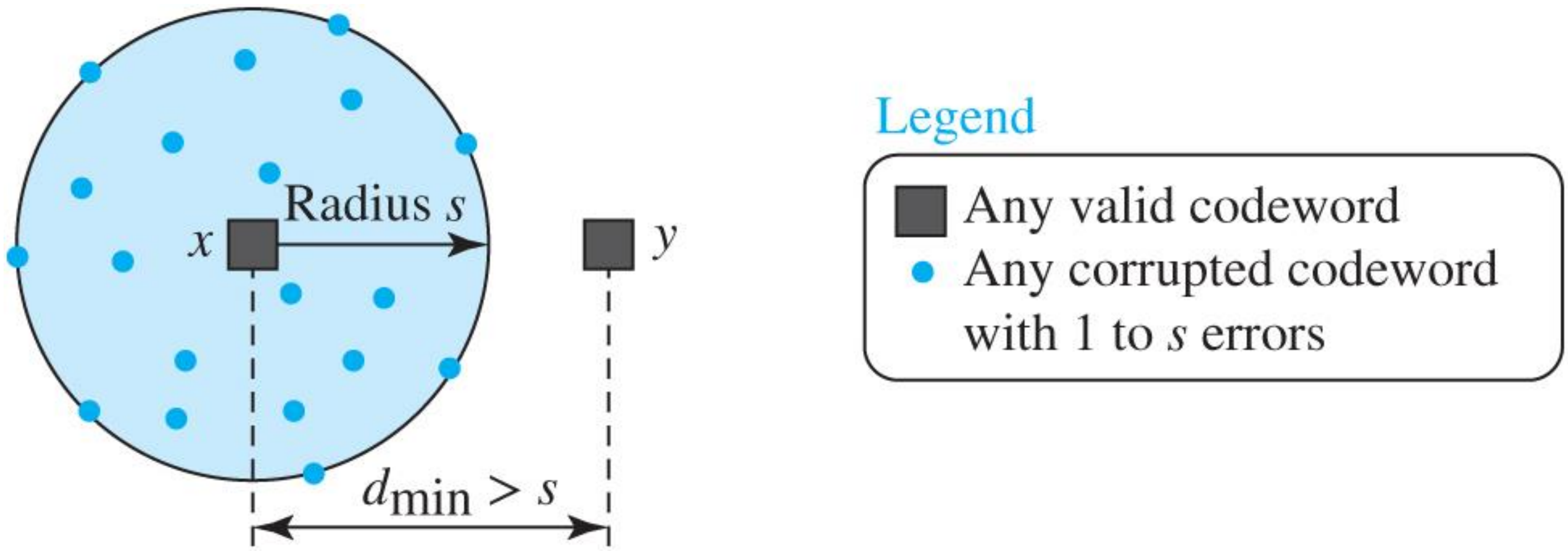
Error Correction

- Forward error correction: Need to correct errors on basis of bits received



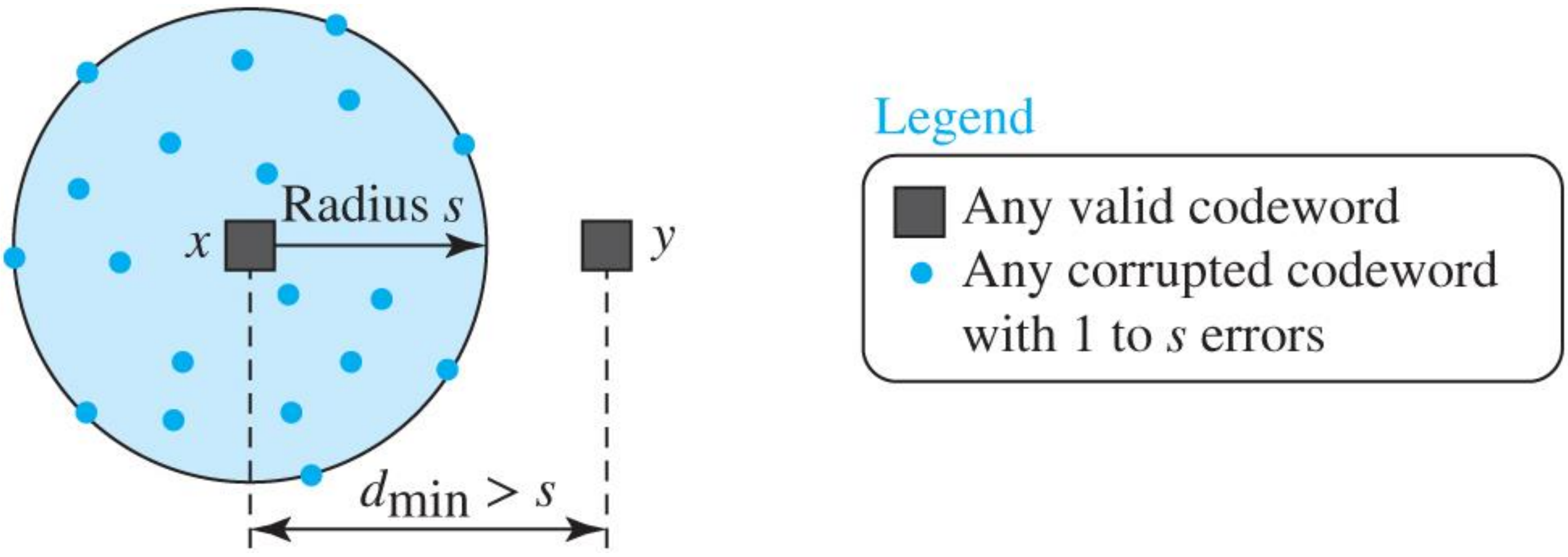
Error Correction

- Hamming distance
  - $d(v_1, v_2)$–bit binary sequences v_1 and v_2 is the number of bits in which v_1 and v_2 disagree



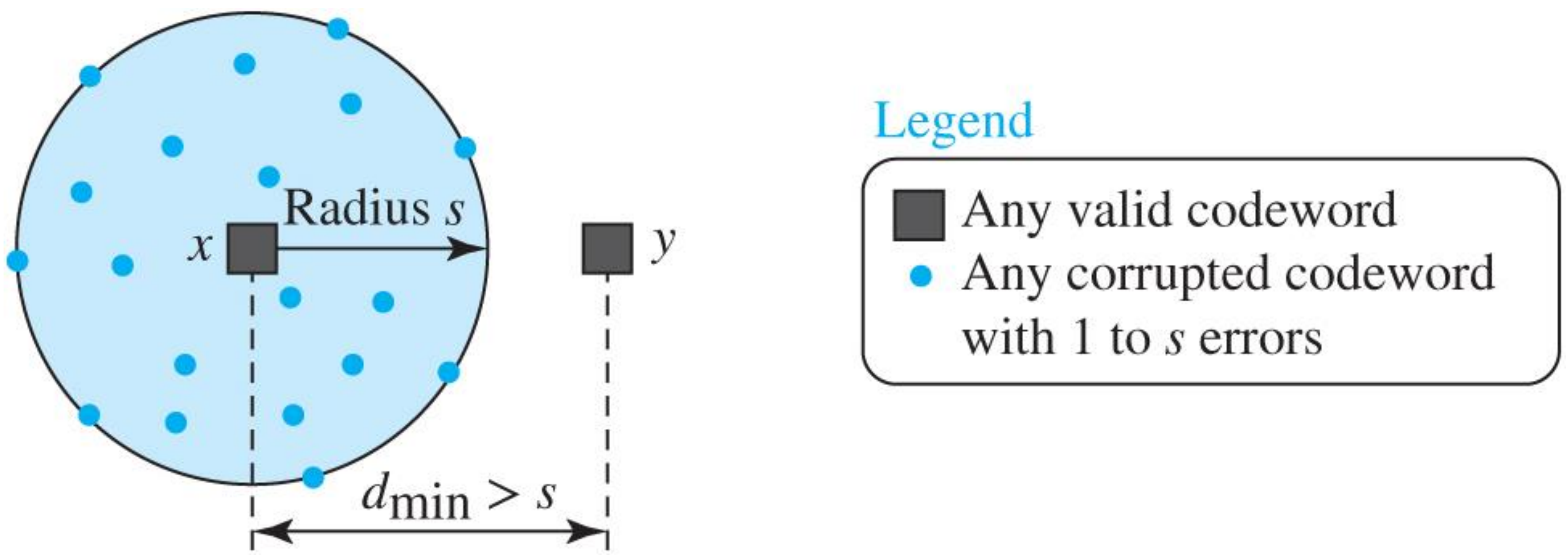
Error Correction

- Valid/Legal and invalid/illegal codeword
- Hamming distance of the codewords: Smallest Hamming distance in between any two valid codewords



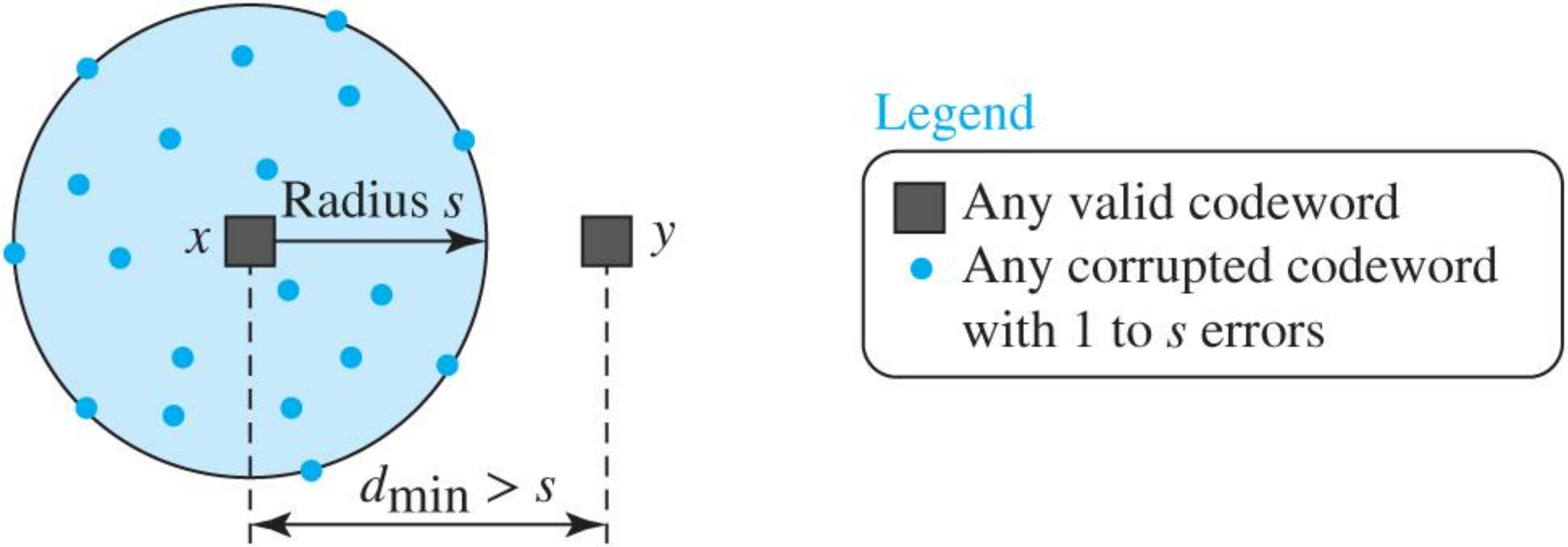
Error Correction

- To detect d bit errors, we need a distance of $d + 1$ code
  - With such a code, there is no way that a d single bit error can change a valid codeword to another valid codeword



Error Correction

- To correct d bit errors, we need a distance of $2d + 1$ code
  - With such a code, the legal codewords are so far apart that even d changes, the original codeword is still closer, than any other codeword
  - Original codeword can be uniquely detected based on the assumption that a larger number of errors are less likely → time consuming search



Error Correction

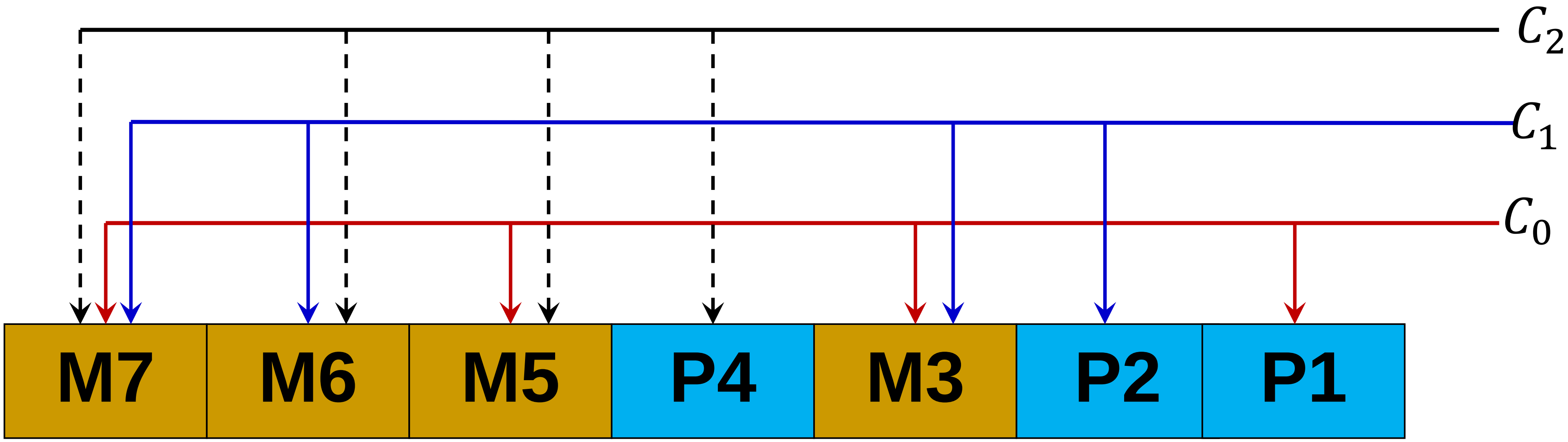
- Codeblock of n bits = m message bit + r check bit

- We want to design a code that allows all single bit error to be corrected
  - Each of the 2^m legal messages → there is n illegal codewords at a distance 1 from it
  - As there are 2^n total number of bit patterns

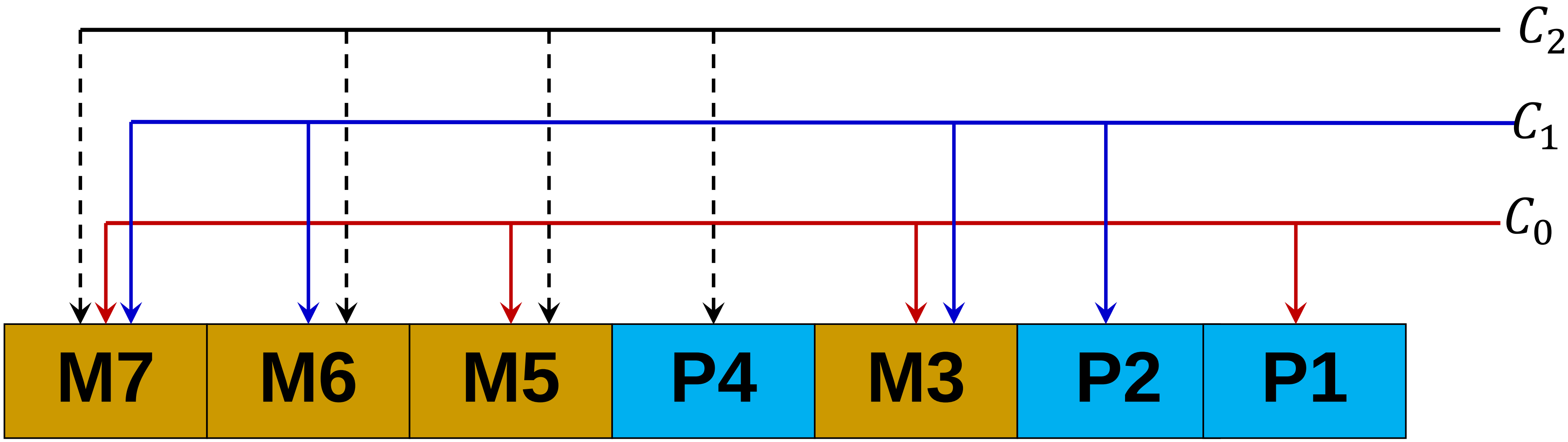
$$2^m(n+1) \leq 2^n \qquad \therefore n+1 \leq 2^{n-m} = 2^r \qquad \therefore m+r+1 \leq 2^r$$

  - So, given m, this put a lower bound on the check bits that are needed to correct single bit errors
    - So for m = 4, r = 3
    - For m = 7, r = 4

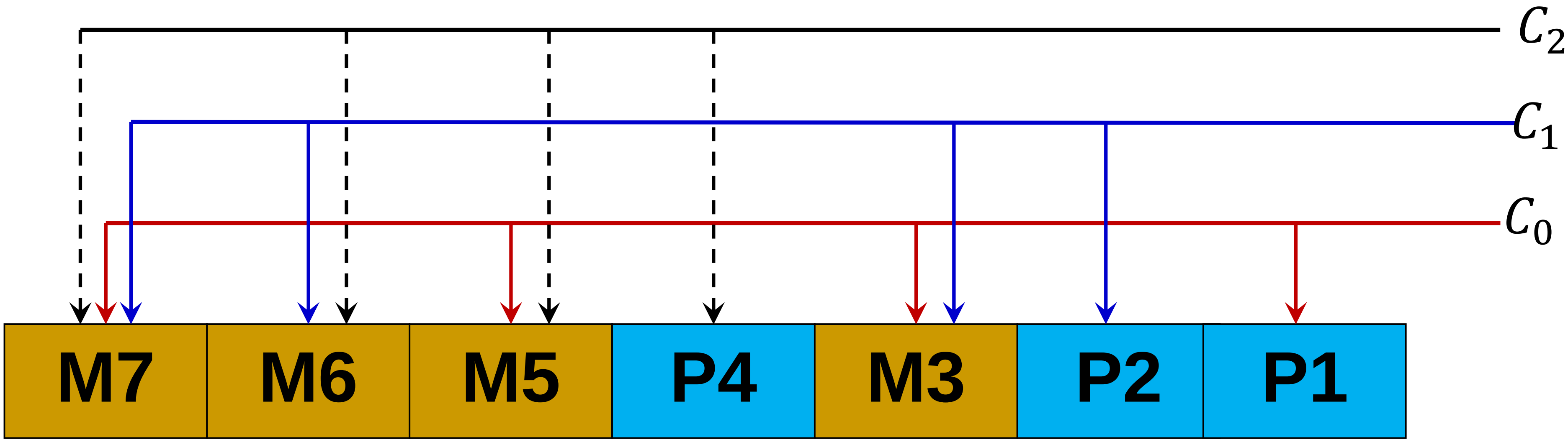
Hamming Code



Hamming Code



Hamming Code



Error Correction Codes

- Convolution codes: GSM mobile phone system, satellite communications, 802.11
- Reed-Solomon code: DSL, data over cable, satellite communications, CDs
- Low-density parity check: Digital video broadcasting, Ethernet, 802.11

THANK YOU

QUESTIONS???