



Royal University of Phnom Penh

Data Communication II

Chapter 4_0: Error Detection and Correction



Lecturer: CHHORN SYLUN

Email: chhorn.sylun@rupp.edu.kh

Room: A311B, RUPP

Objective

- ❑ Introduction
- ❑ Type of errors
- ❑ Redundancy
- ❑ Technique of Detection Errors

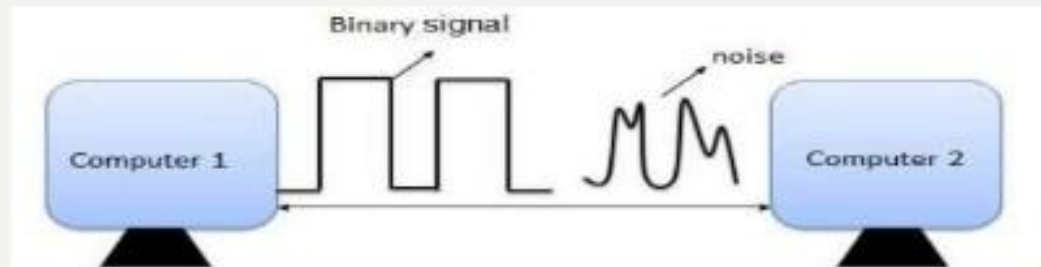
Introduction

- Data can be corrupted during transmission.
- For reliable communication, errors must be detected and corrected.
- Error detection and correction are implemented either at **data link layer** or the **transport layer** of the OSI model.

So what is Error?

Introduction

- Error is a condition when the output information does not match with the input information.
- During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from one system to the other.
- That means a 0 bit may change to 1 or 1 bit may change to 0.

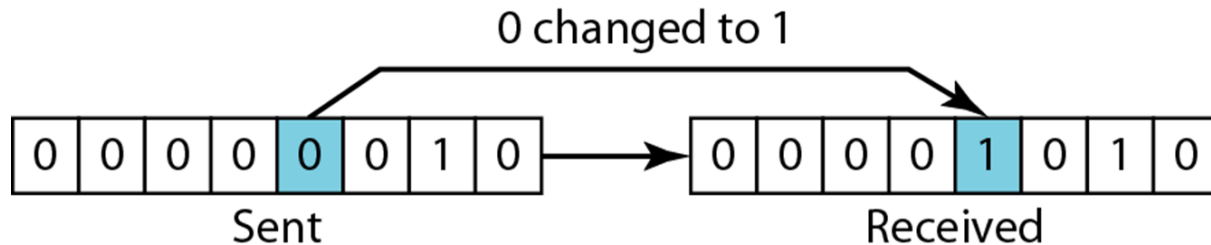


Introduction

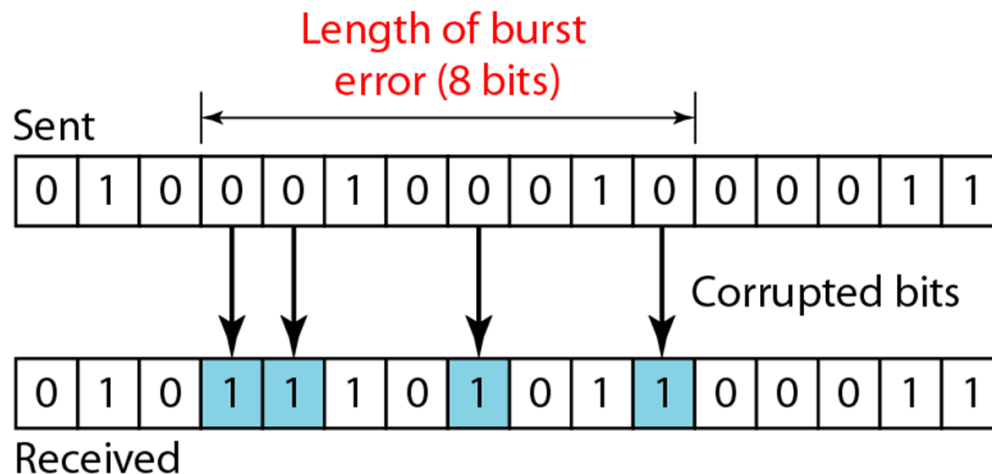
- **Error Detection:** Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver.
- **Error Correction:** We need to know the exact number of bits that are corrupted and their location in the message. The number of the errors and size of message are important.

Type of Errors

- **Single Bit Error:** only 1 bit in the data unit or frame has changed.



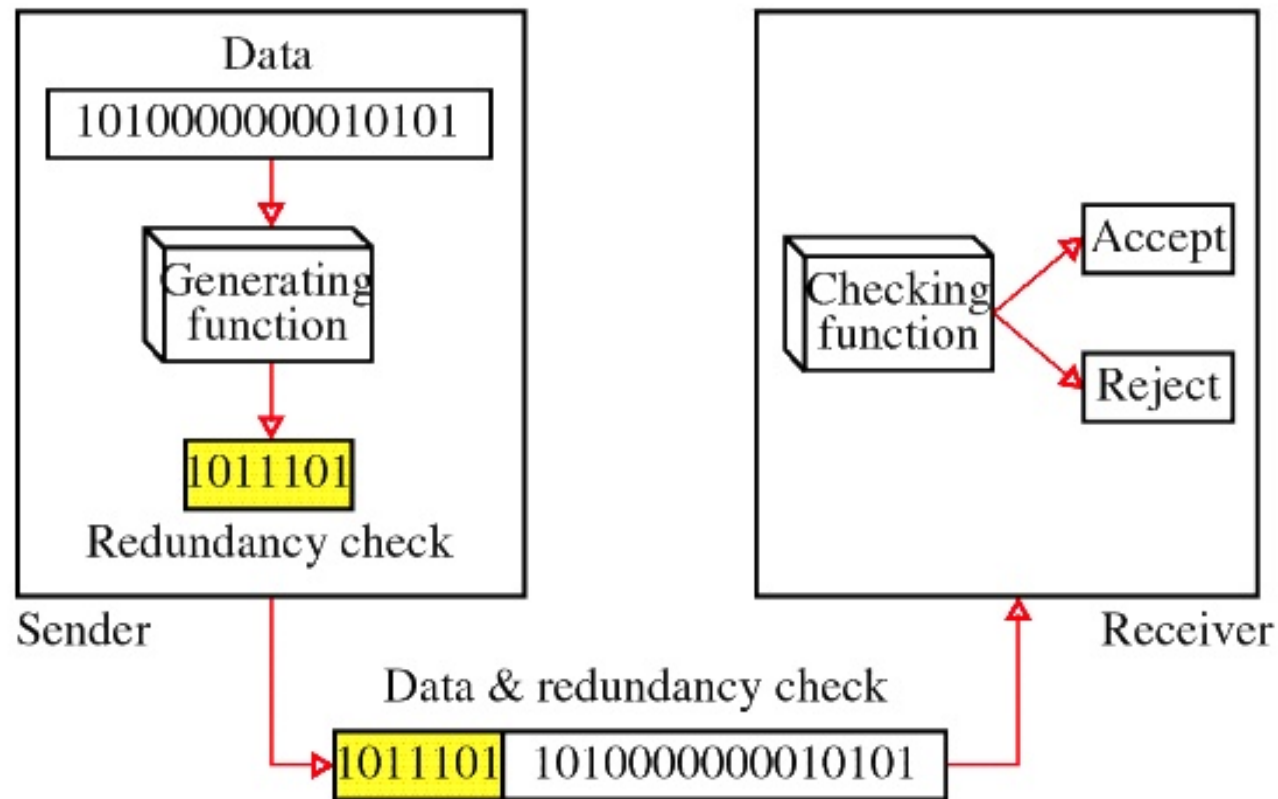
- **Burst Error:** 2 or more bits in the data unit or frame have changed.



Redundancy

- To detect or correct errors, we need to send extra (redundant) bits with data.
- Instead of repeating the entire data stream, a shorter group of bits may be appended to the end of each unit. This technique is called **Redundancy** because the extra bit are redundant to the information.
- They are discarded as soon as the accuracy of the transmission has been determined.

Redundancy



XOR logic

$$0 \oplus 0 = 0$$

$$1 \oplus 1 = 0$$

a. Two bits are the same, the result is 0.

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

b. Two bits are different, the result is 1.

	1	0	1	1	0
\oplus	1	1	1	0	0
<hr/>					
	0	1	0	1	0

c. Result of XORing two patterns

Block Coding

- In block coding, we divide our message into blocks, each of k bits, called **datawords**.
- We add r redundant bits to each block to make the length $n = k + r$.
- The result n -bit blocks are called **codewords**.

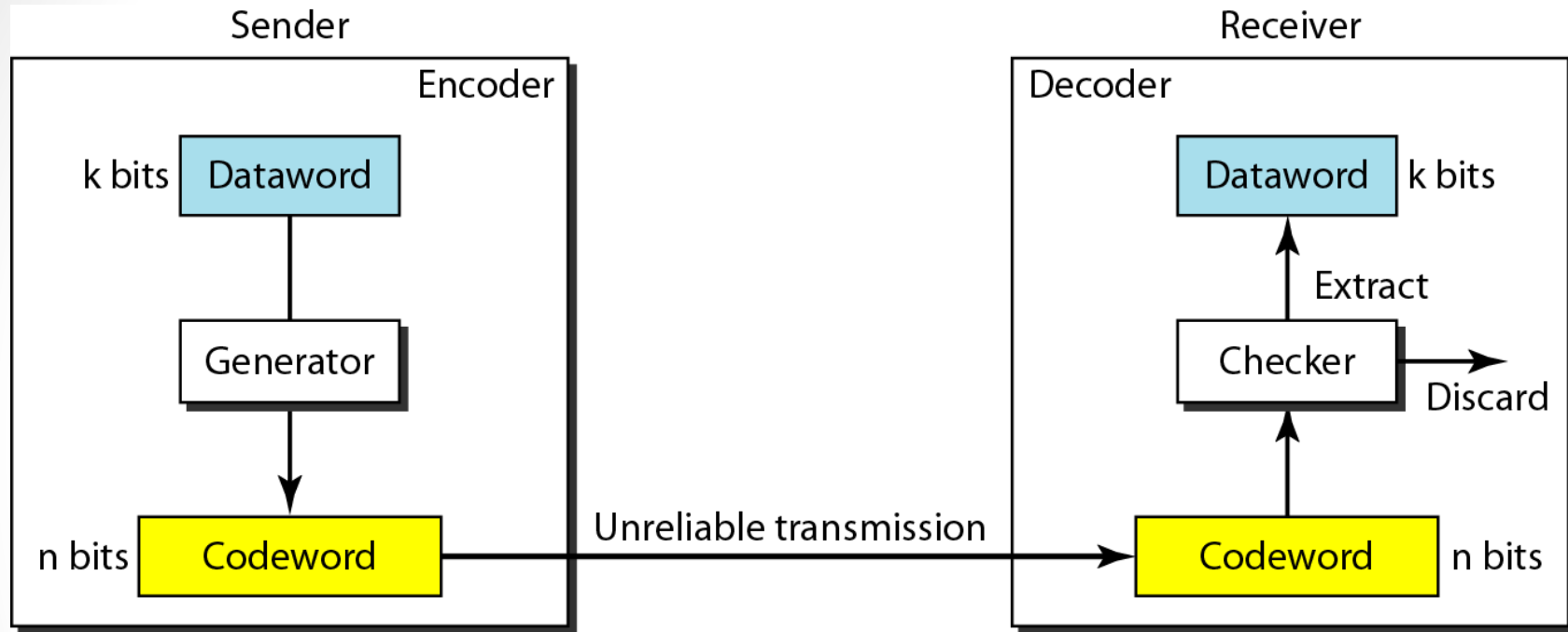


2^k Datawords, each of k bits



2^n Codewords, each of n bits (only 2^k of them are valid)

Block Coding



Process of error detection in block coding

Techniques of Error Detection

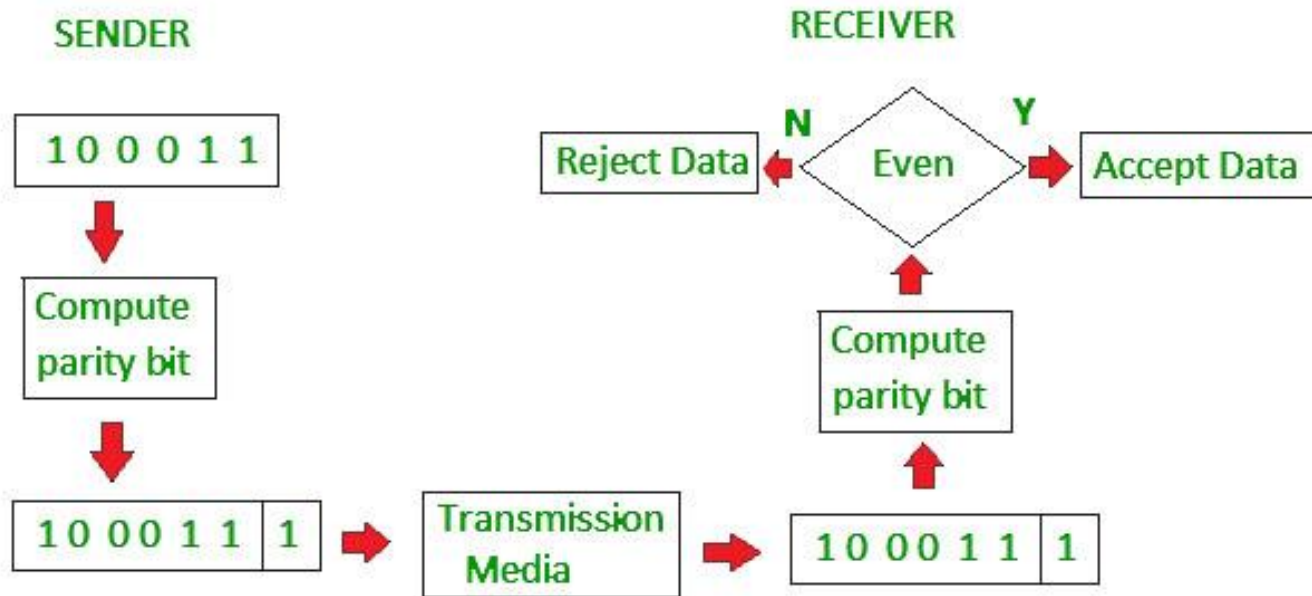
Some techniques for error detection are:

1. Simple Parity Check
2. Two Dimensional Check
3. Checksum
4. Cyclic Redundancy Check (CRC)

Simple Parity Check

- The most common and least expensive mechanism for error detection is the simple parity check.
- Block of data from the source are subjected to a check bit or parity bit generator form, where a parity of:
 - a) 1 is added to the block if it contains odd number of 1's
 - b) 0 is added to the block if it contains even number of 1's

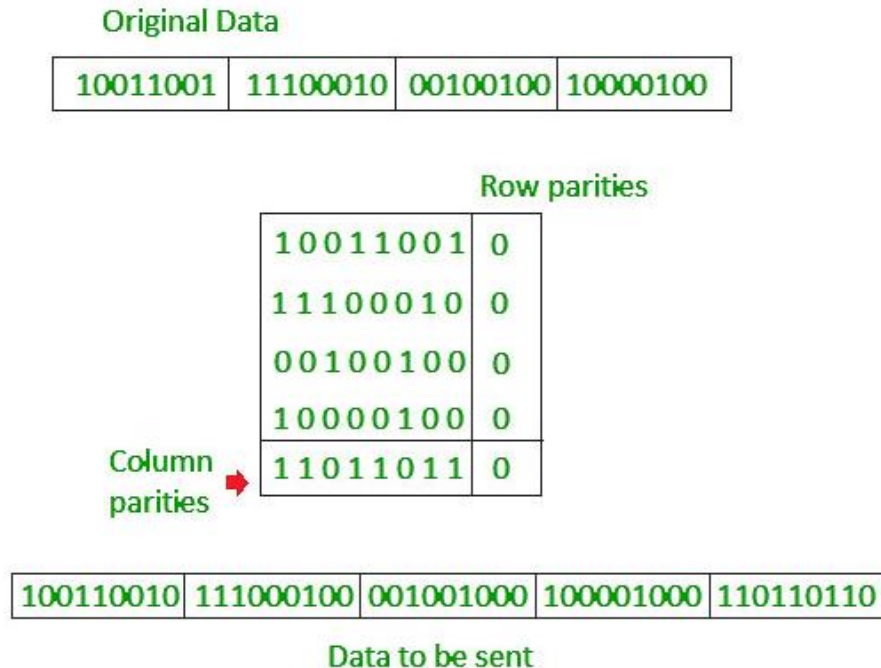
Simple Parity Check



Two Dimensional Parity Check

- Parity check bits are calculated for each row, which is equivalent to a simple parity check bit.
- Parity check bits are also calculated for all columns, then both are sent along with the data.
- At receiving end these are compared with the parity bits calculated on the received data.

Example



Two Dimensional Parity Check

Exercise

Original data

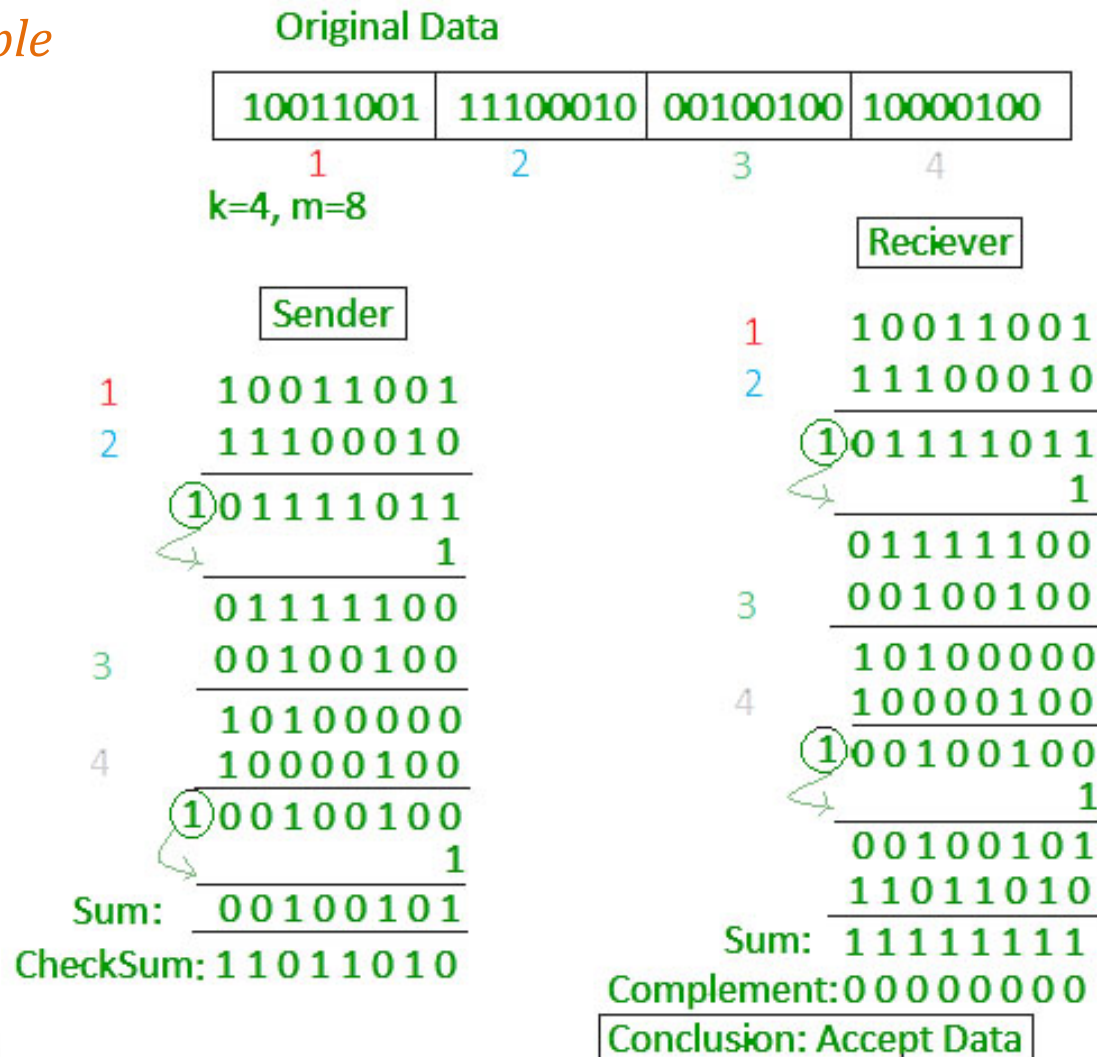
10110011	⋮	10101011	⋮	01011010	⋮	11010101
----------	---	----------	---	----------	---	----------

Checksum

- In Checksum error detection scheme, the **data** is divided into **k** segments each of **m** bits.
 - In the sender's end the segments are added using 1's complement arithmetic to get the sum. **The sum is complemented to get the checksum.**
 - The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
 - If the result is zero, the received data is accepted; otherwise discarded.

Checksum

Example



Checksum

Exercise

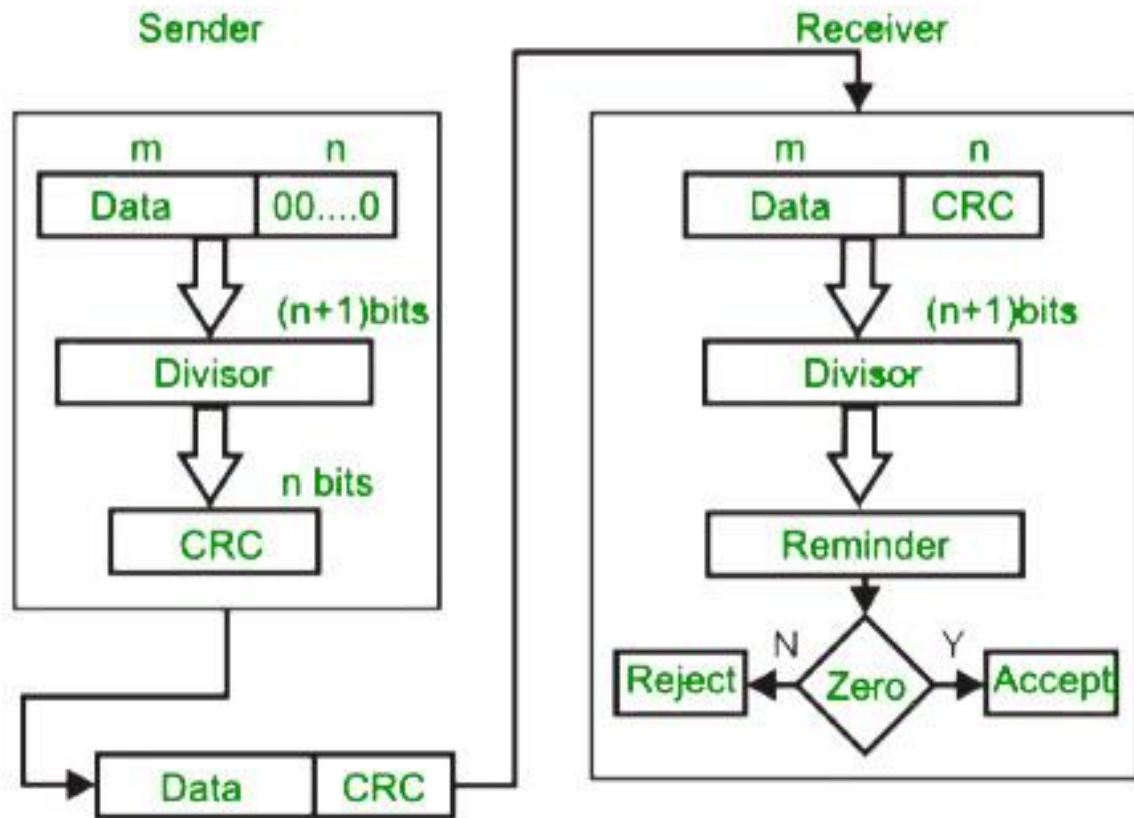
Original data

10110011	⋮	10101011	⋮	01011010	⋮	11010101
----------	---	----------	---	----------	---	----------

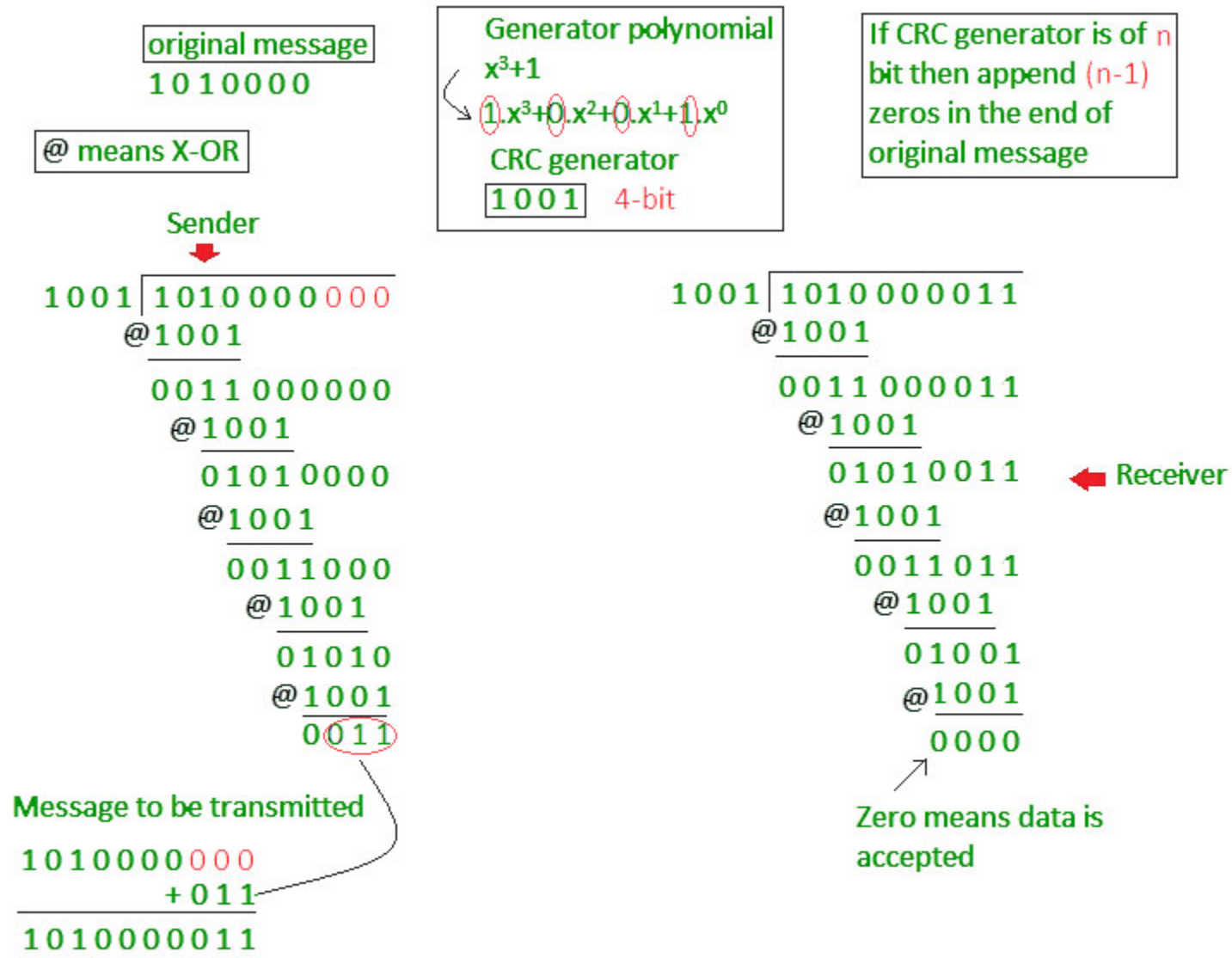
Cyclic Redundancy Check(CRC)

- Unlike Checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

Cyclic Redundancy Check(CRC)



Cyclic Redundancy Check(CRC)



Cyclic Redundancy Check(CRC)

Exercise Data word to be sent : **100100**
Generator Polynomial : **1101**

Cyclic Redundancy Check(CRC)

CRC Performance

- CRC is a very effective error detection technique.
- CRC can detect all single-bit errors
- CRC can detect all double-bit errors (three 1's)
- CRC can detect any odd number of errors ($X+1$)
- CRC detects most of the larger burst errors with a high probability.

Q&A

