UNIT II

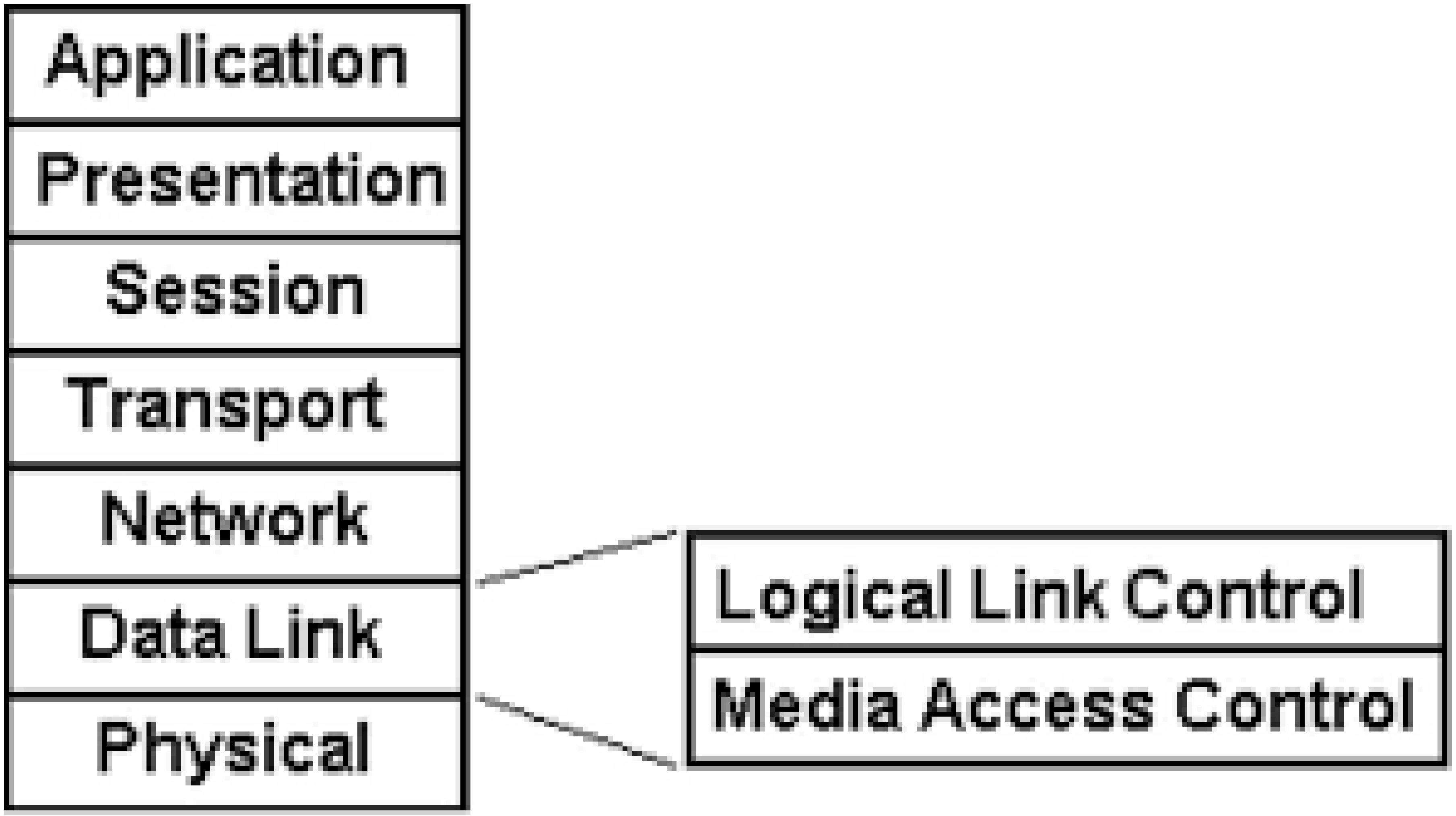
DATA LINK LAYER

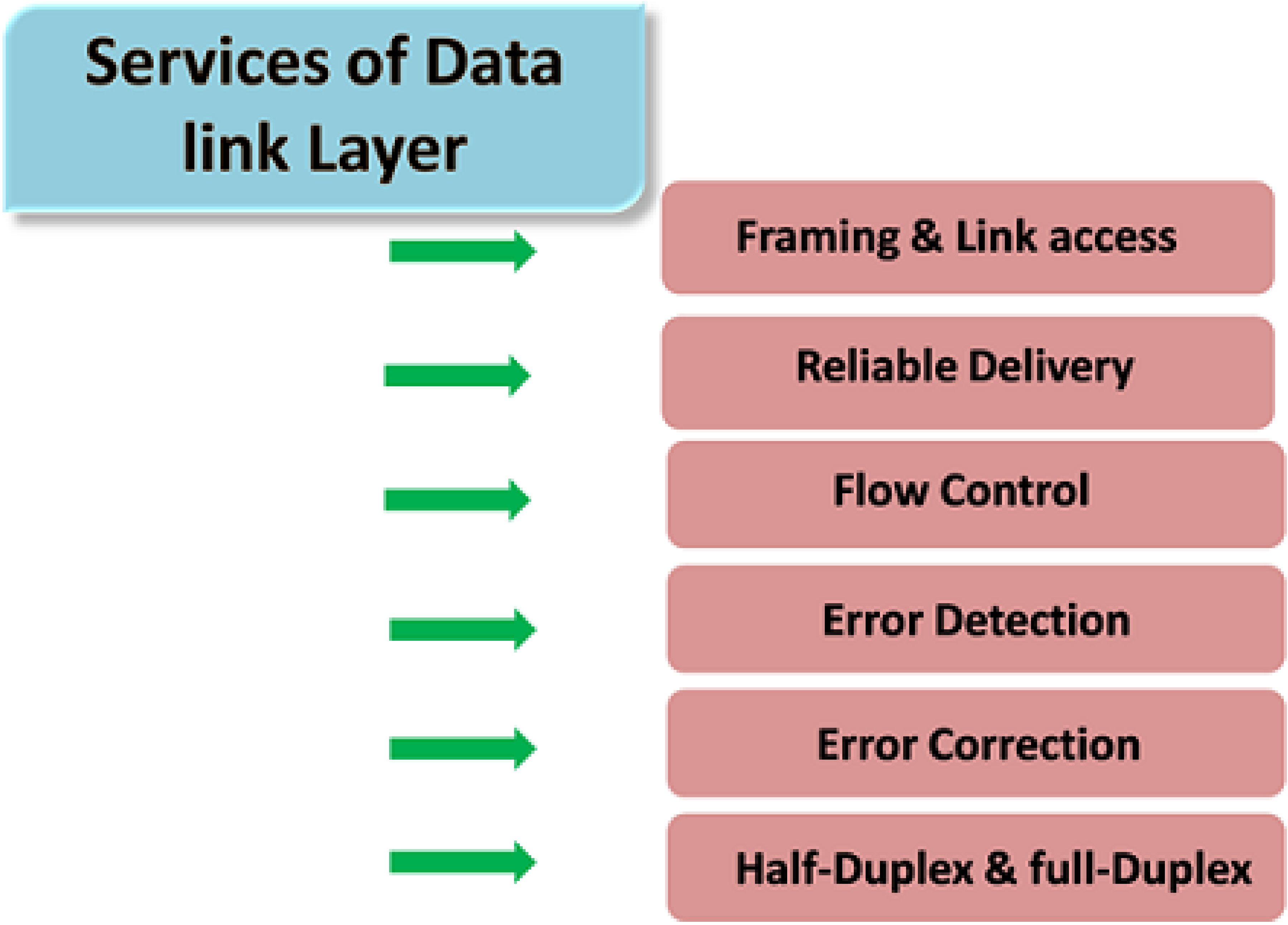
INTRODUCTION

- Data Link Layer is second layer of OSI Layered Model. This layer is one of the most complicated layers and has complex functionalities and liabilities.
- Data link layer hides the details of underlying hardware and represents itself to upper layer as the medium to communicate.
- Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware.
- At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer.

Data link layer has two sub-layers:

- **Logical Link Control:** It deals with protocols, flow-control, and error control
- **Media Access Control:** It deals with actual control of media


Application
Presentation
Session
Transport
Network
Data Link
Physical
Logical Link Control
Media Access Control


Services of Data link Layer
Framing & Link access
Reliable Delivery
Flow Control
Error Detection
Error Correction
Half-Duplex & full-Duplex

Functionality of Data-link Layer

- **Framing**

  Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, it sends each frame bit-by-bit on the hardware. At receiver’ end, data link layer picks up signals from hardware and assembles them into frames.

- **Addressing**

  Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.

- **Synchronization**

  When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.

- **Error Control**

  Sometimes signals may have encountered problem in transition and the bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.

- **Flow Control**

  Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machine to exchange data on same speed.

- **Multi-Access**

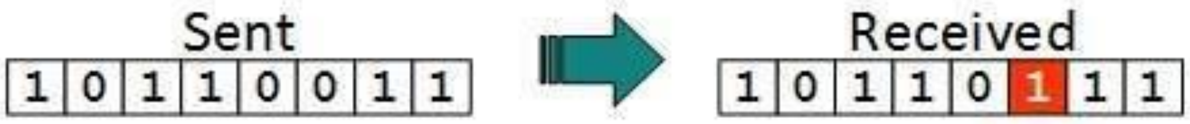
  When host on the shared link tries to transfer the data, it has a high probability of collision. Data-link layer provides mechanism such as CSMA/CD to equip capability of accessing a shared media among multiple Systems.

Error Detection & Correction

- Data-link layer uses some error control mechanism to ensure that frames (data bit streams) are transmitted with certain level of accuracy. But to understand how errors is controlled, it is essential to know what types of errors may occur.

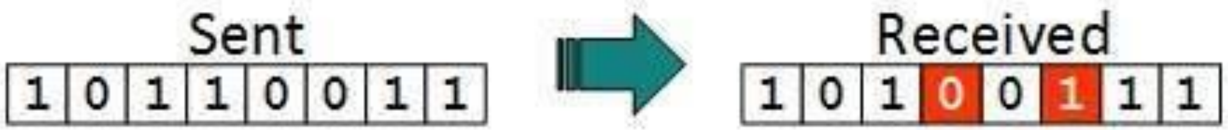
Types of Errors

There may be three types of errors:



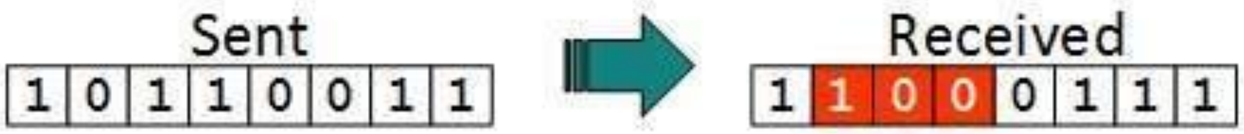
- **Single bit error**

  In a frame, there is only one bit, anywhere though, which is corrupt.



- **Multiple bits error**

  Frame is received with more than one bits in corrupted state.



- **Burst error**

  Frame contains more than1 consecutive bits corrupted.

Error Detecting Codes

- Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted.

- To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

- Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.

- Some popular techniques for error detection are:
  1. Simple Parity check
  2. Two-dimensional Parity check
  3. Checksum
  4. Cyclic redundancy check

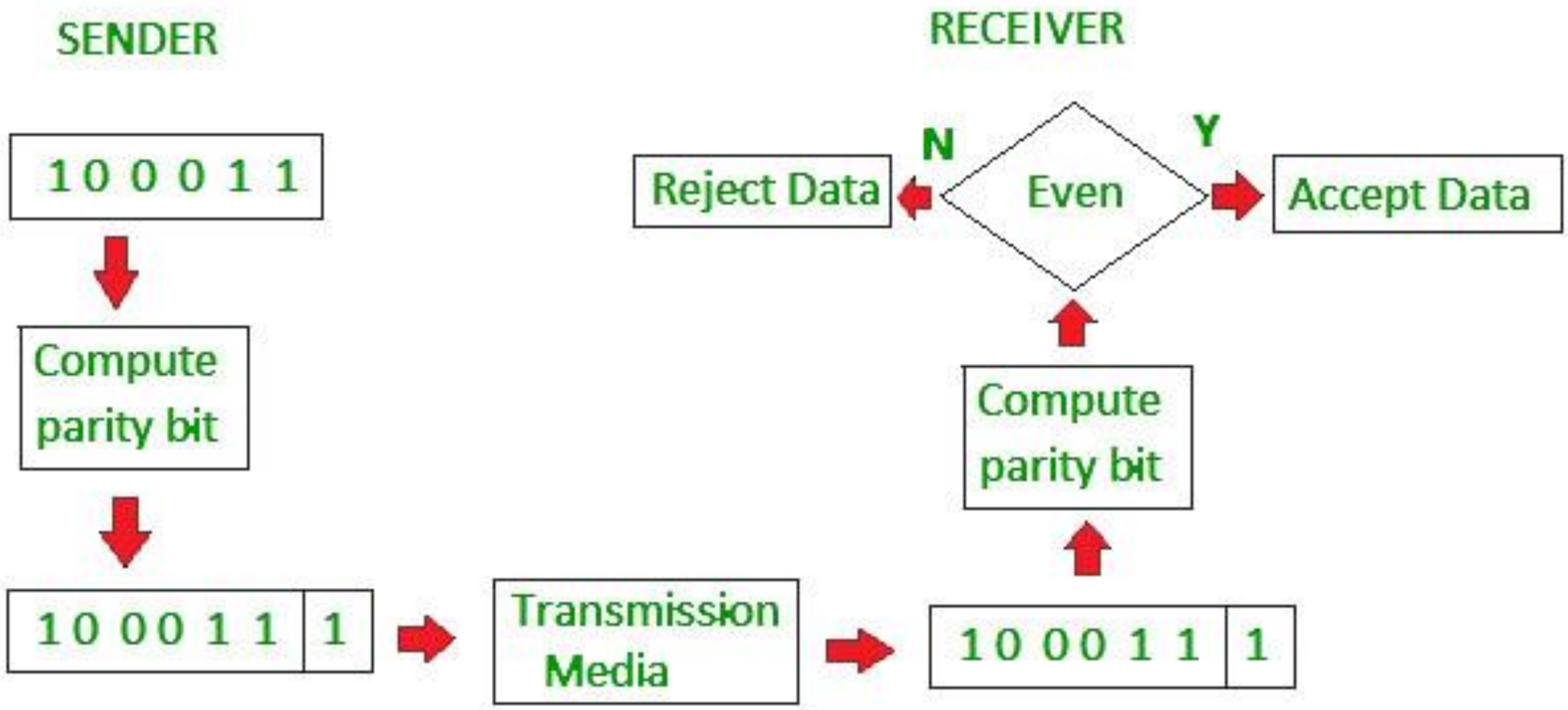
➢ **Simple Parity check**

- Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

➢ 1 is added to the block if it contains odd number of 1’s, and

➢ 0 is added if it contains even number of 1’s

- This scheme makes the total number of 1’s even, that is why it is called even parity checking.

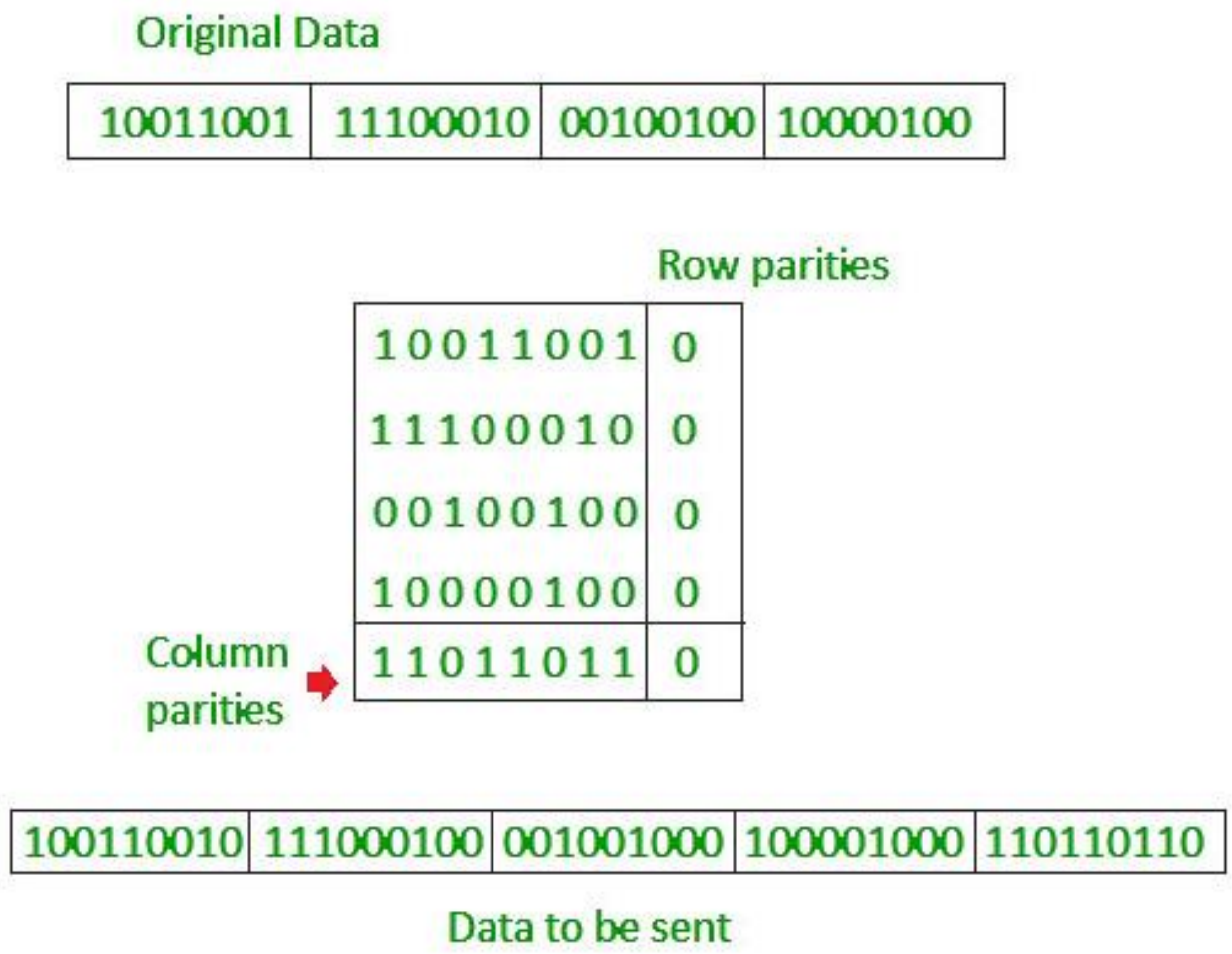

SENDER
RECEIVER
1 0 0 0 1 1
Compute
parity bit
1 0 0 0 1 1
1
Transmission
Media
1 0 0 0 1 1
1
Compute
parity bit
Even
N
Y
Reject Data
Accept Data

➢ 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 the receiving end these are compared with the parity bits calculated on the received data.



➢ 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.

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

k=4, m=8

Sender

```
1          1 0 0 1 1 0 0 1
2          1 1 1 0 0 0 1 0
          ----------------
          1 0 1 1 1 1 0 1 1
                          1
          ----------------
           0 1 1 1 1 1 0 0
3          0 0 1 0 0 1 0 0
          ----------------
           1 0 1 0 0 0 0 0
4          1 0 0 0 0 1 0 0
          ----------------
          1 0 0 1 0 0 1 0 0
                          1
          ----------------
Sum:       0 0 1 0 0 1 0 1
          ----------------
CheckSum:  1 1 0 1 1 0 1 0
```

Reciever

```
1             1 0 0 1 1 0 0 1
2             1 1 1 0 0 0 1 0
             ----------------
             1 0 1 1 1 1 0 1 1
                             1
             ----------------
              0 1 1 1 1 1 0 0
3             0 0 1 0 0 1 0 0
             ----------------
              1 0 1 0 0 0 0 0
4             1 0 0 0 0 1 0 0
             ----------------
             1 0 0 1 0 0 1 0 0
                             1
             ----------------
              0 0 1 0 0 1 0 1
              1 1 0 1 1 0 1 0
             ----------------
Sum:          1 1 1 1 1 1 1 1
Complement:   0 0 0 0 0 0 0 0
```

Conclusion: Accept Data

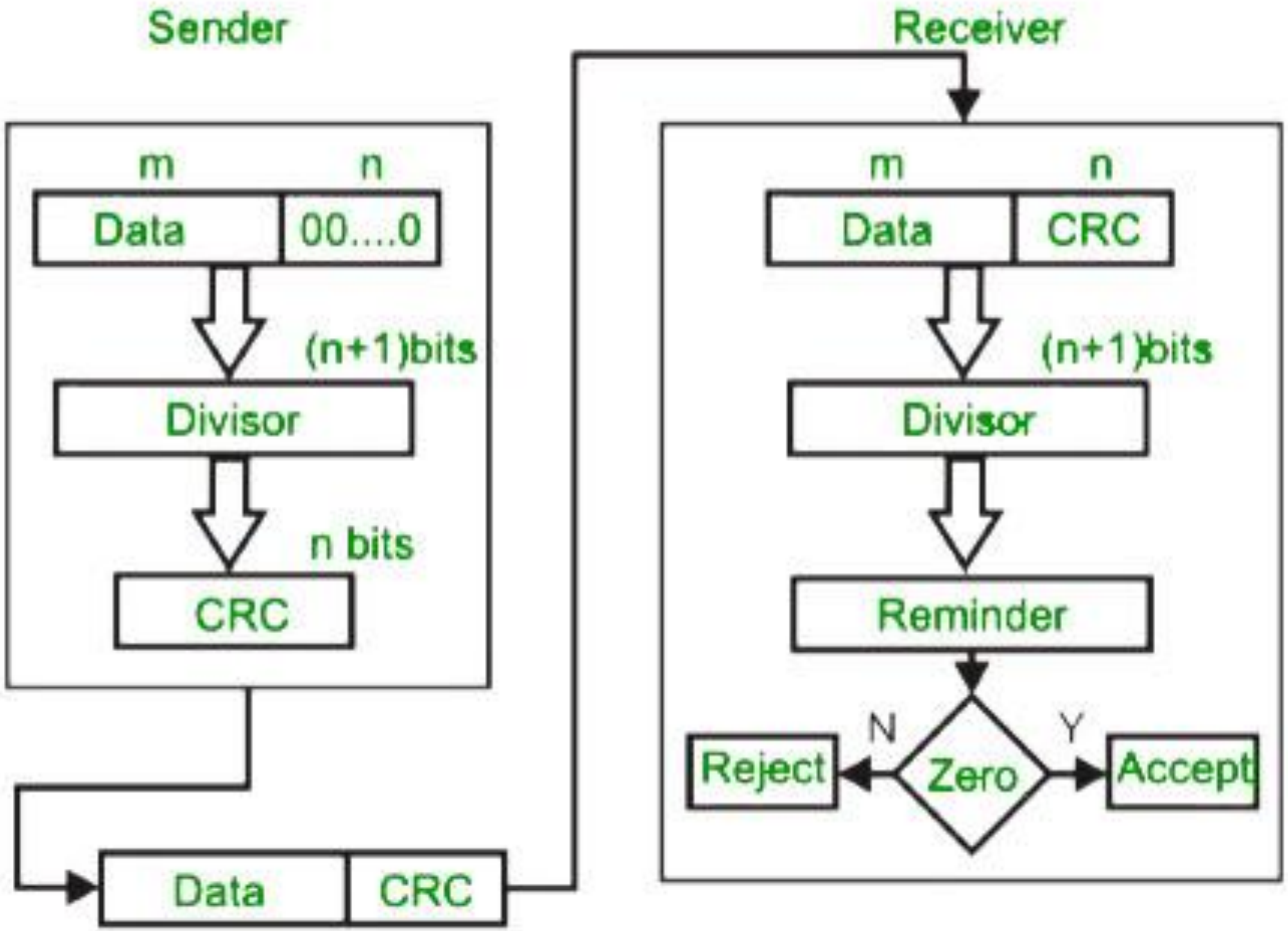
➢ Cyclic redundancy check (CRC)

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.

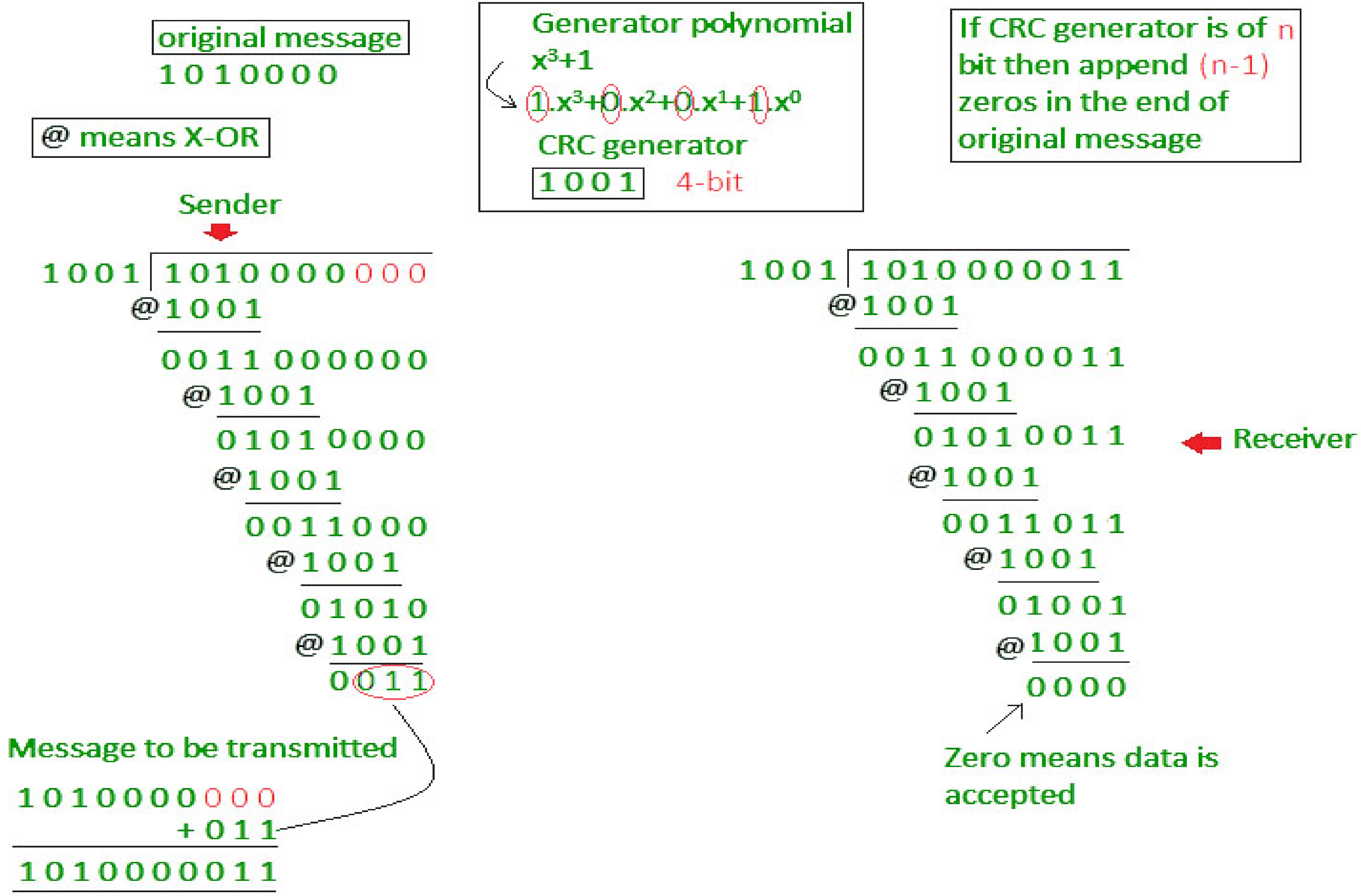
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.

- 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.



Sender
Receiver
m
n
Data
00....0
(n+1)bits
Divisor
n bits
CRC
m
n
Data
CRC
(n+1)bits
Divisor
Reminder
N
Y
Reject
Zero
Accept
Data
CRC


original message
1010000
@ means X-OR
Generator polynomial
x3+1
1.x3+0.x2+0.x1+1.x0
CRC generator
1001
4-bit
If CRC generator is of n bit then append (n-1) zeros in the end of original message
Sender
1001
1010000000
@1001
0011000000
@1001
01010000
@1001
0011000
@1001
01010
@1001
0011
Message to be transmitted
1010000000
+011
1010000011
1001
1010000011
@1001
0011000011
@1001
01010011
@1001
0011011
@1001
01001
@1001
0000
Receiver
Zero means data is accepted

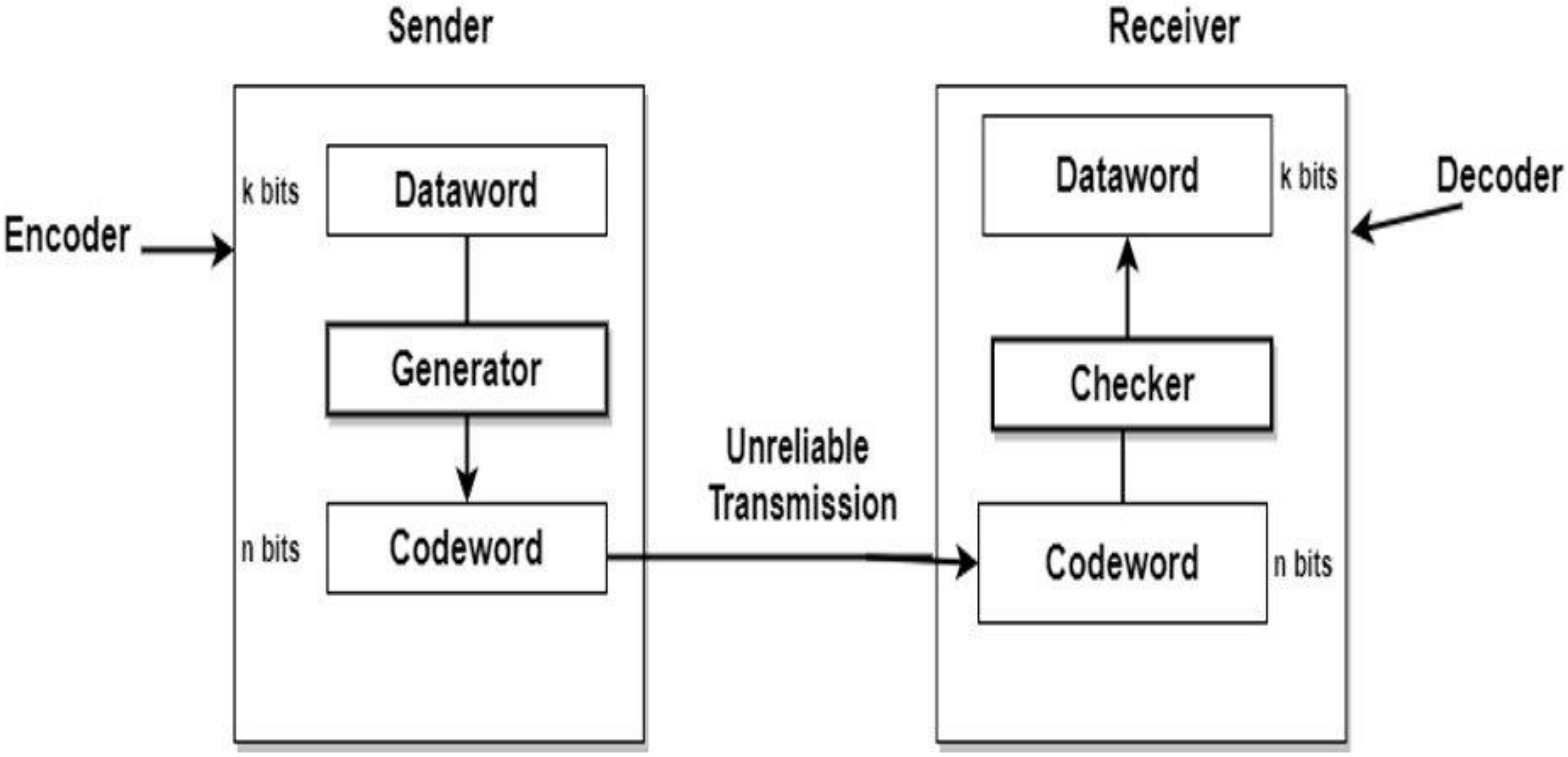
Error Correction

- In Error Detection, the receiver only needs to know that the received codeword is invalid; But in Error Correction the receiver needs to guess the Original codeword that is sent. In this way, error Correction is much more difficult than Error Detection.

- The need for redundant bits is more during error correction rather than for error detection.

- In order to detect or correct the errors, there is a need to send some extra bits along with the data. These extra bits are commonly known as Redundant bits.

- Original data is divided into segments of k bits; it is referred to as **dataword**. When we add **r** redundant bits to each block in order to make the length;**n=k+r** then it is referred to as **Codeword**.


Sender
Receiver
Encoder
k bits
Dataword
Generator
n bits
Codeword
Unreliable Transmission
Dataword
k bits
Decoder
Checker
Codeword
n bits

There are two ways to handle the error correction:

- **Backward Error correction technique**
- **Forward Error Correction technique**

- Whenever an error discovered, the receiver can have the sender in order to retransmit the entire data unit. This technique is known as the **Backward Error correction technique**.

- This technique is simple and inexpensive in the case of wired transmission like fiber optics; there is no expense in retransmitting the data. In the case of wireless transmission, retransmission costs too much thus forward error correction technique is used then.

- The receiver can use an error-correcting code that automatically contains certain errors. This technique is known as the **Forward Error Correction technique**.