[2CEIT503 COMPUTER NETWORKS]

Practical: 5

AIM- Write a program to implement various Error Detection Mechanisms.

a. find minimum hamming distance
b. Checksum
c. CRC

Department of Computer Engineering/Information Technology

Hamming Distance

- The Hamming distance between two words is the number of differences between corresponding bits.
- Hamming distance between two words x and y as d(x,y)
- The Hamming distance d(000, 011) is 2 because

$$000 \oplus 011 \text{ is } 011 \text{ (two 1s)}$$

- The **minimum Hamming distance** is the smallest Hamming distance between all possible pairs in a set of words.

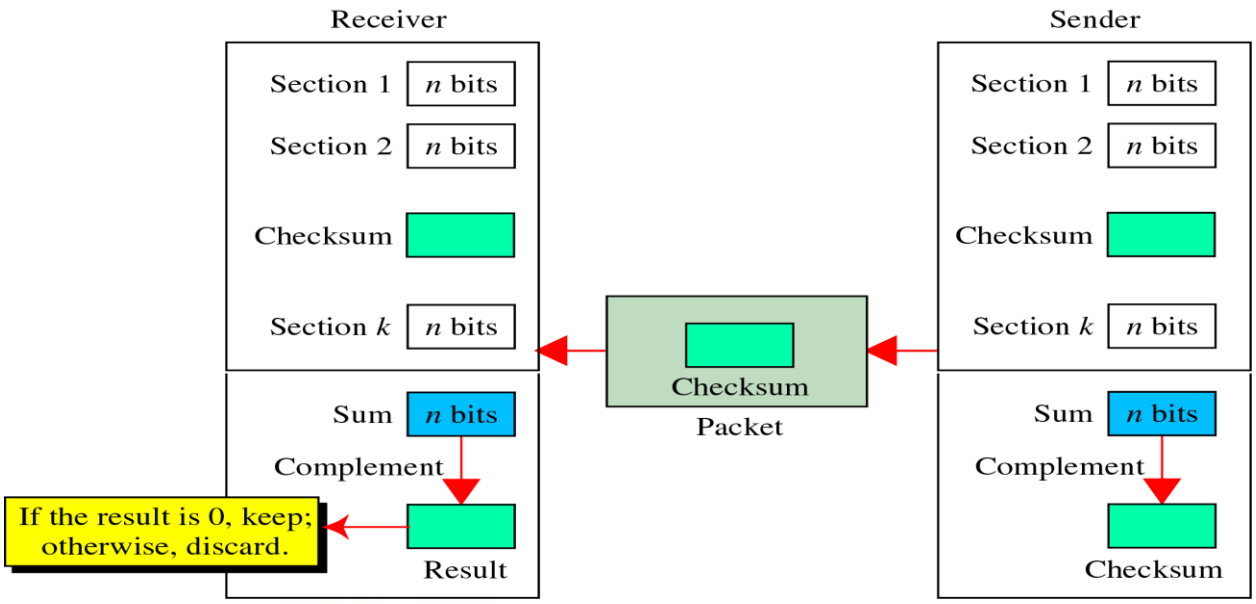
Datawords	*Codewords*
00	000
01	011
10	101
11	110

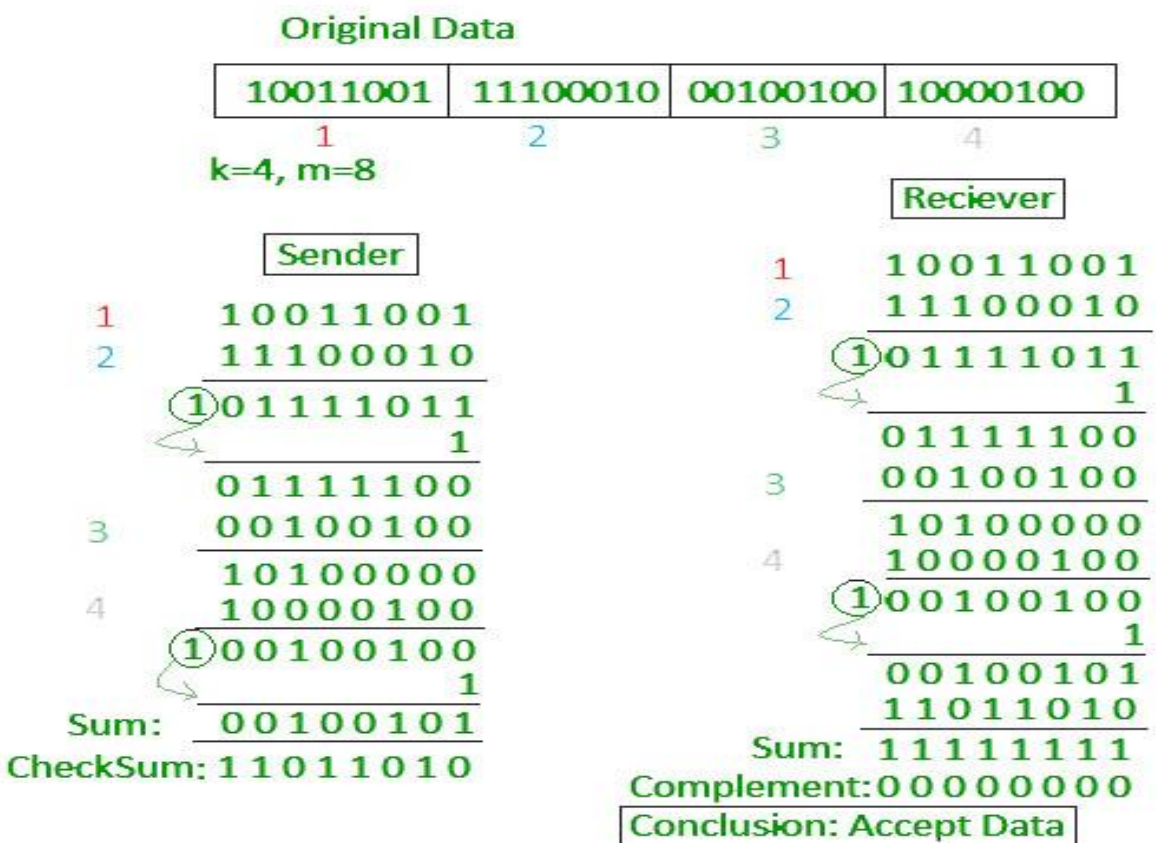
$$d(000, 011) = 2 \quad d(000, 101) = 2 \quad d(000, 110) = 2 \quad d(011, 101) = 2$$
$$d(011, 110) = 2 \quad d(101, 110) = 2$$

The d_{min} in this case is 2.

Checksum

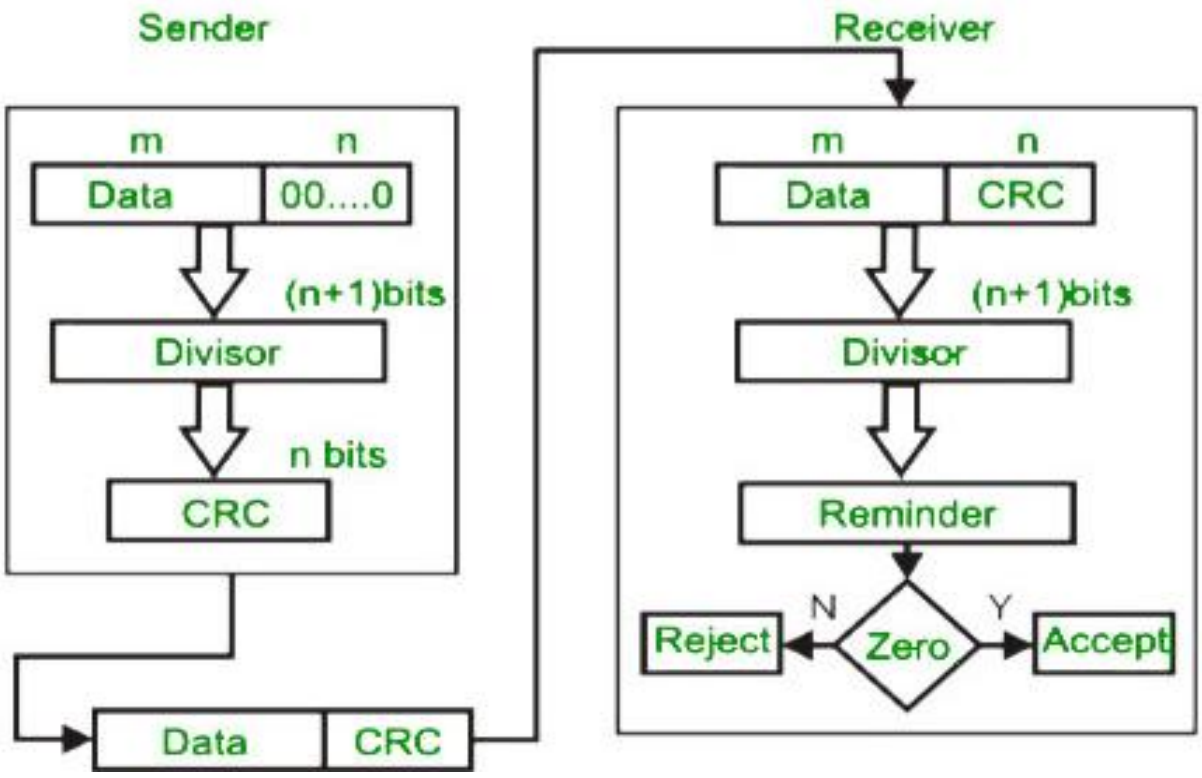
- In checksum error detection scheme, the data is divided into *k* segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's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

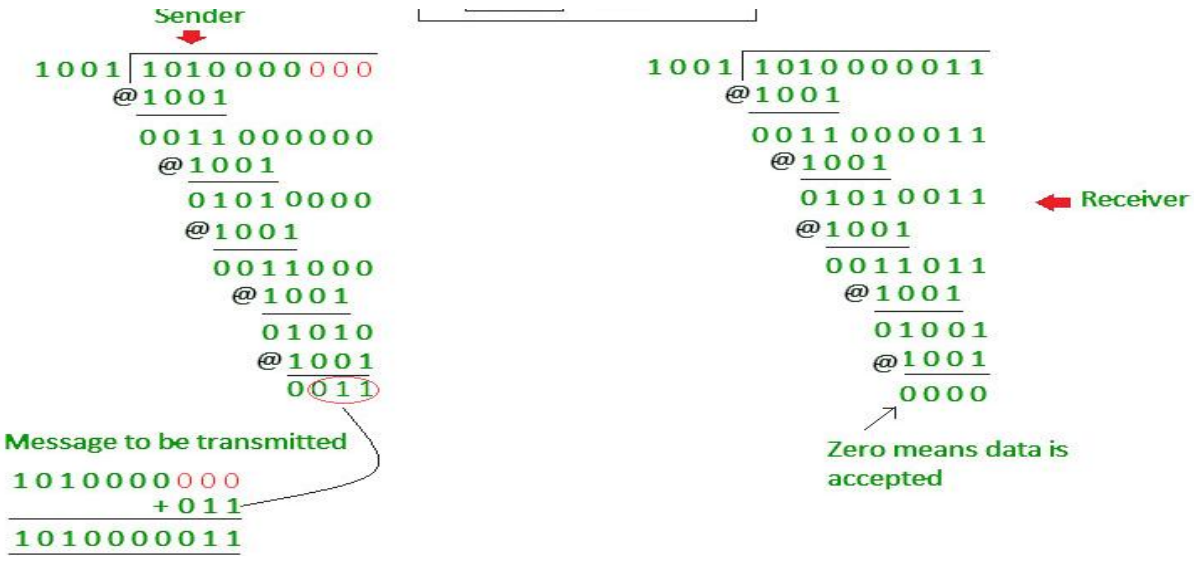

Receiver
Section 1
n bits
Section 2
n bits
Checksum
Section k
n bits
Sum
n bits
Complement
Result
If the result is 0, keep; otherwise, discard.
Checksum
Packet
Sender
Section 1
n bits
Section 2
n bits
Checksum
Section k
n bits
Sum
n bits
Complement
Checksum


Original Data
10011001
11100010
00100100
10000100
1
2
3
4
k=4, m=8
Sender
1 10011001
2 11100010
1 01111011
1
01111100
3 00100100
10100000
4 10000100
1 00100100
1
Sum: 00100101
CheckSum: 11011010
Reciever
1 10011001
2 11100010
1 01111011
1
01111100
3 00100100
10100000
4 10000100
1 00100100
1
00100101
11011010
Sum: 11111111
Complement: 00000000
Conclusion: Accept Data

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
1001 | 10100000000
@1001
00110000000
@1001
01010000
@1001
0011000
@1001
01010
@1001
0011
Message to be transmitted
10100000000
+011
10100000011
1001 | 10100000011
@1001
00110000011
@1001
01010011
@1001
0011011
@1001
01001
@1001
0000
Receiver
Zero means data is accepted