VISHNU
UNIVERSAL LEARNING

Topic-2
Error Detection

Prepared by K.Lakshmipathi raju, Assistant professor
Dept of IT

A condition when the receiver's information does not match with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model)

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.

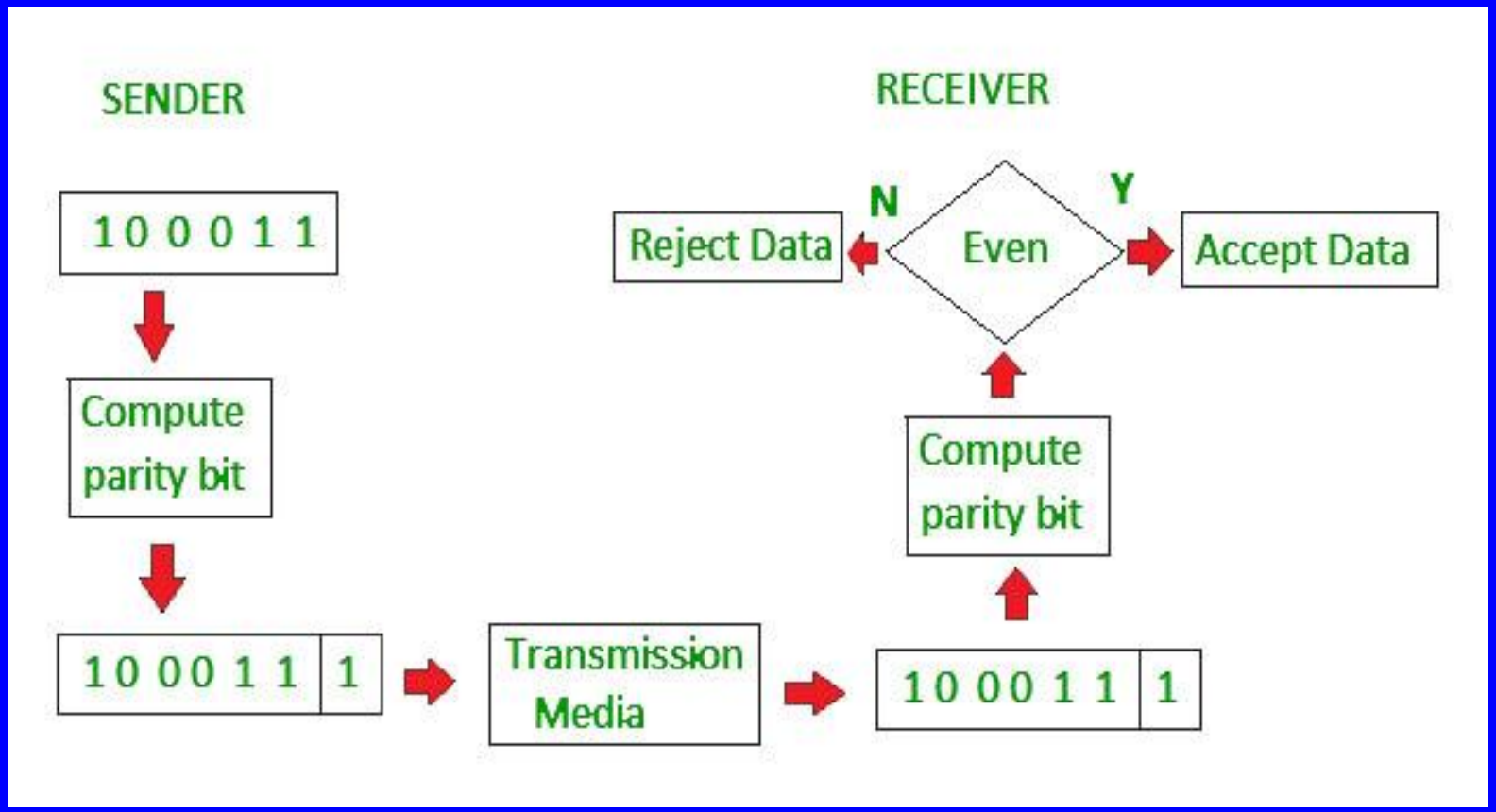
Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.

Some popular techniques for error detection are:

1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check

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.

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

Two-dimensional Parity check

Original Data

10011001	11100010	00100100	10000100

	Row parities
10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column parities

100110010	111000100	001001000	100001000	110110110

Data to be sent

Checksum

In checksum error detection scheme, the data is divided into k segments each of m bits.

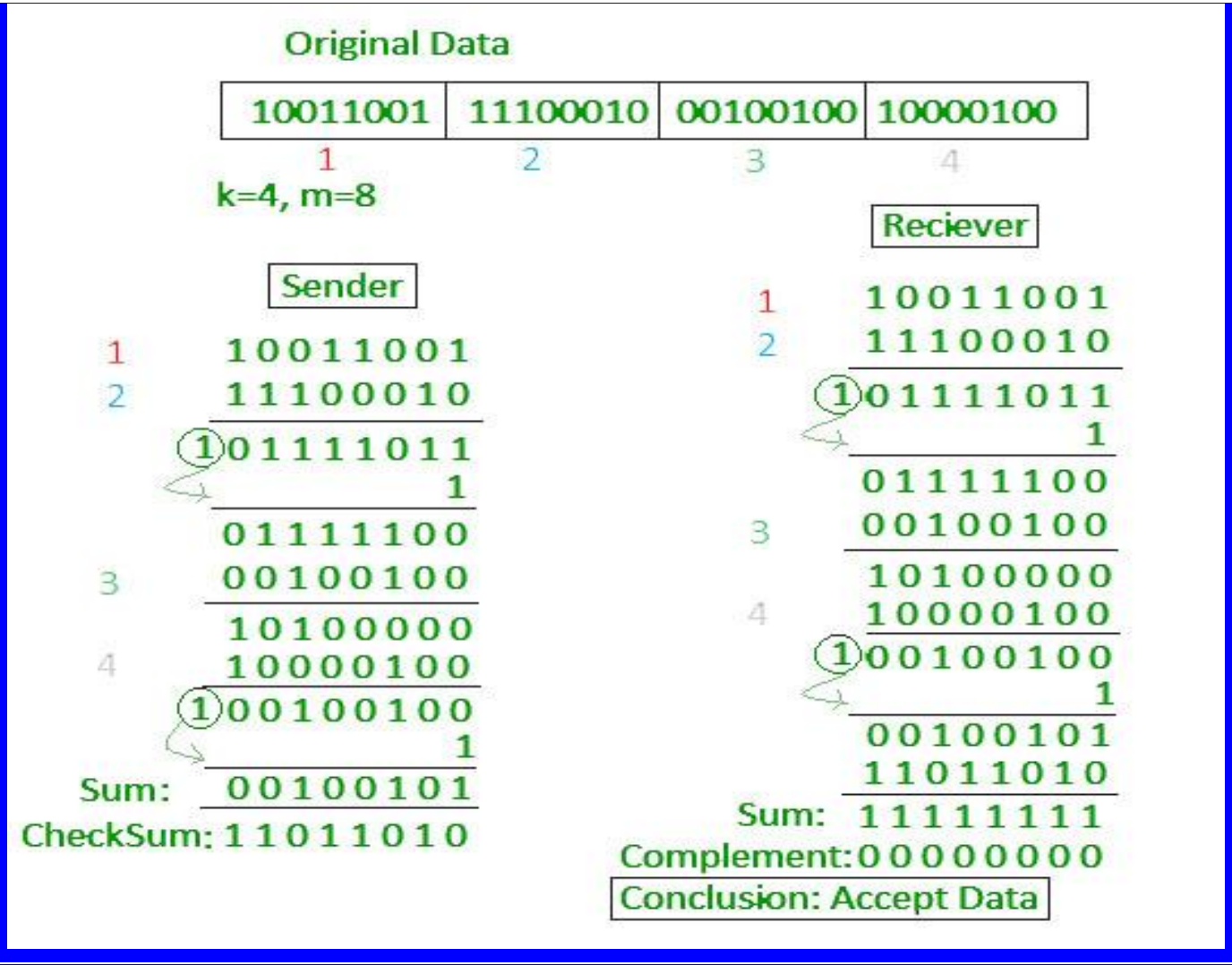
In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.

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


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

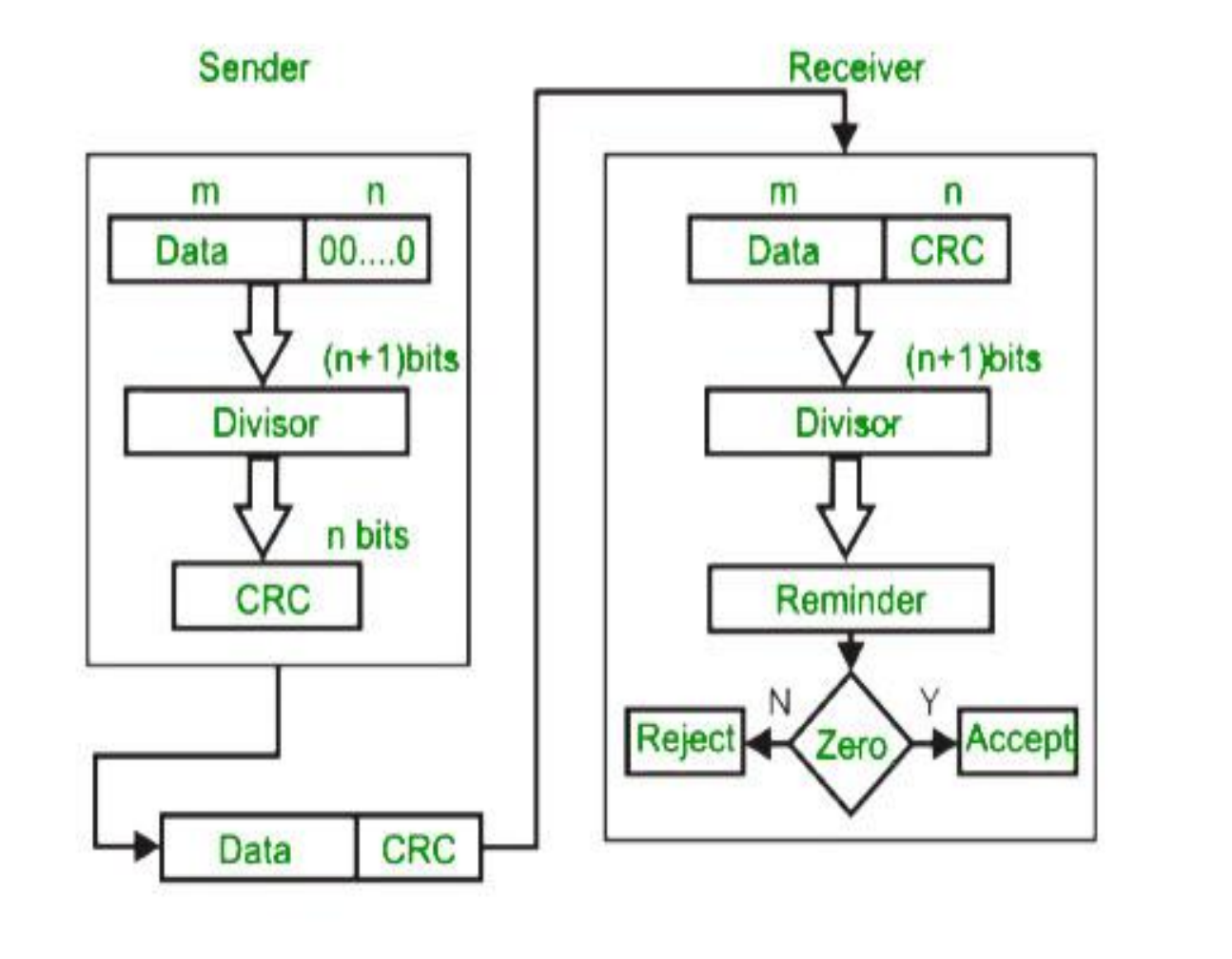
Unlike checksum scheme, which is based on addition, CRC is based on binary division.

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.

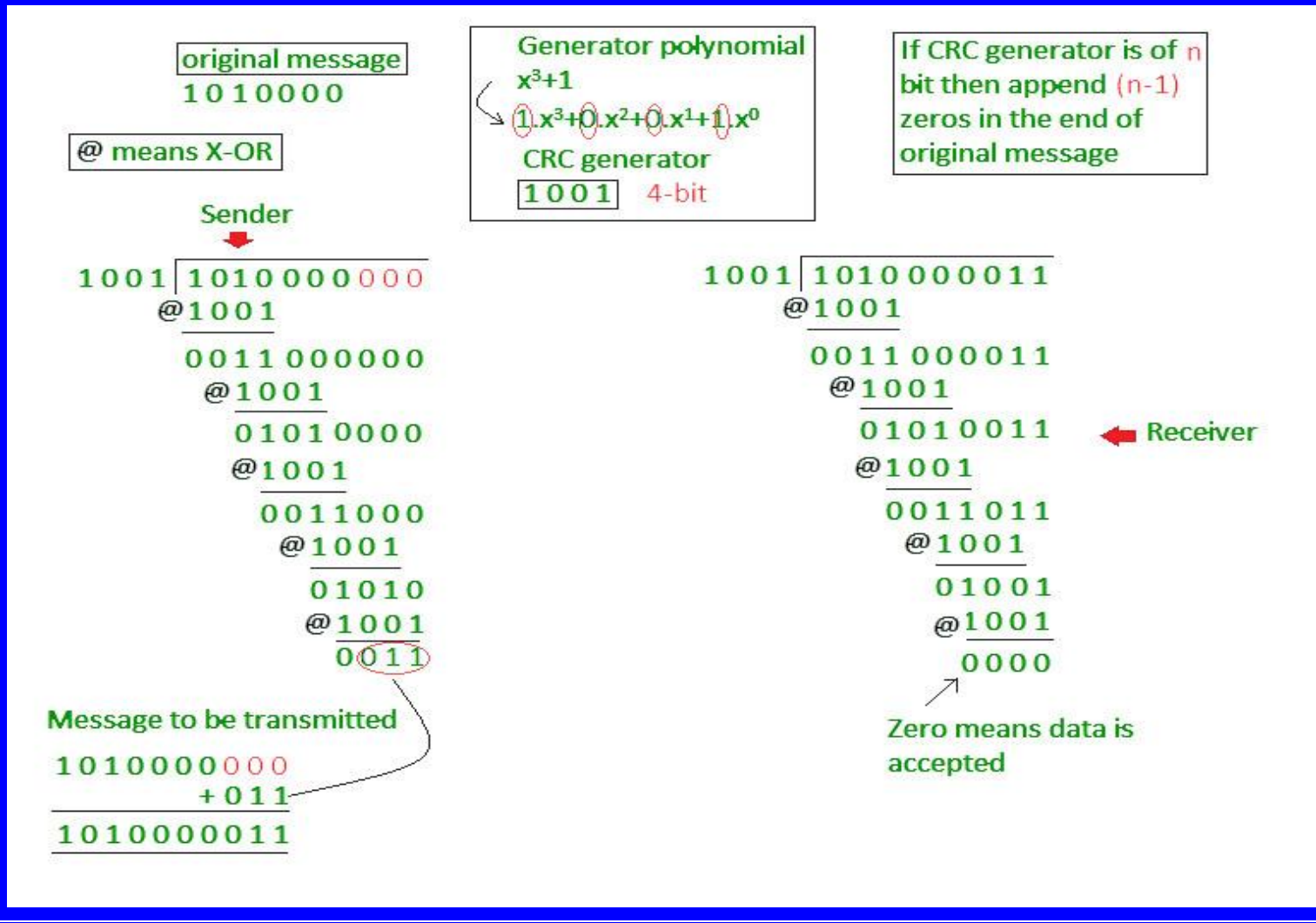
At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.

A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

Cyclic redundancy check (CRC)



Example


original message
1010000
@ means X-OR
Generator polynomial
x^3+1
$1.x^3+0.x^2+0.x^1+1.x^0$
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

VISHNU
UNIVERSAL LEARNING

Thank You