**LAB - 3**

**ERROR DETECTION**

**Objective:**

To implement the Error detection mechanism in Client Server Application with the parity, 2D parity and Cyclic Redundancy Check methods.

**Introduction:**

An Error is **a** condition when the receiver’s information does not matches 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.

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

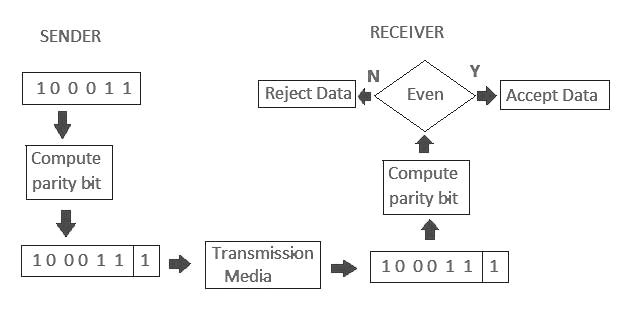
4. Cyclic redundancy check

**1.Simple Parity check:**

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

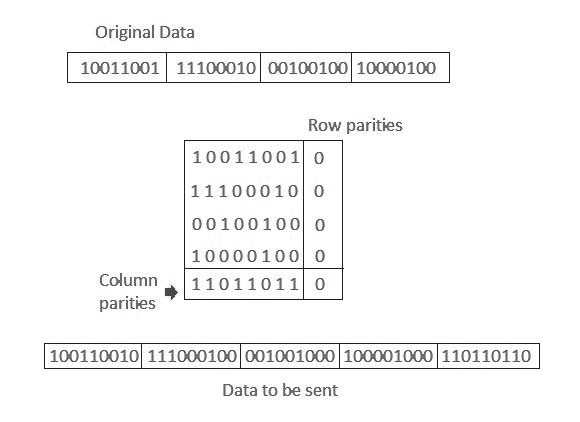
* 1 is added to the block if it contains odd number of 1’s, and
* 0 is added if it contains even number of 1’s

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

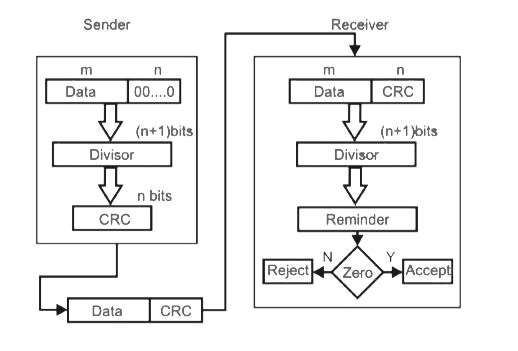


**2. Two-dimensional Parity check:**

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.



**3. 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.

Signature