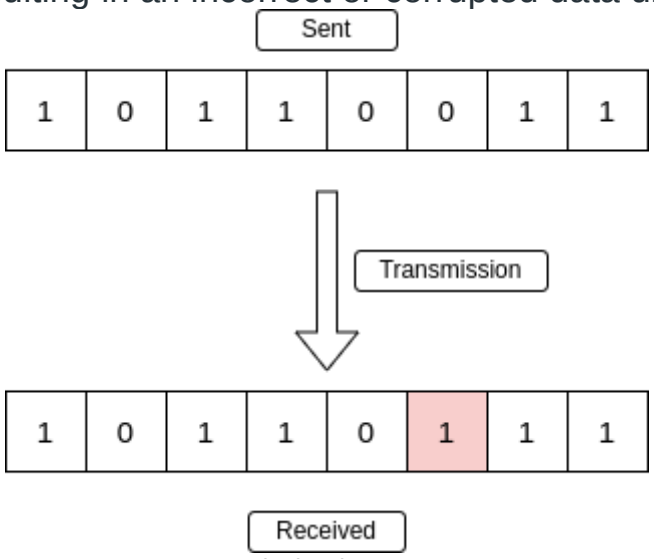
Error is a condition when the receiver's information does not match the sender's. Digital signals suffer from noise during transmission that can introduce errors in the binary bits traveling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.
Data (Implemented either at the Data link layer or Transport Layer of the OSI Model) may get scrambled by noise or get corrupted whenever a message is transmitted. To prevent such errors, error-detection codes are added as extra data to digital messages. This helps in detecting any errors that may have occurred during message transmission.

Types of Errors

Single-Bit Error

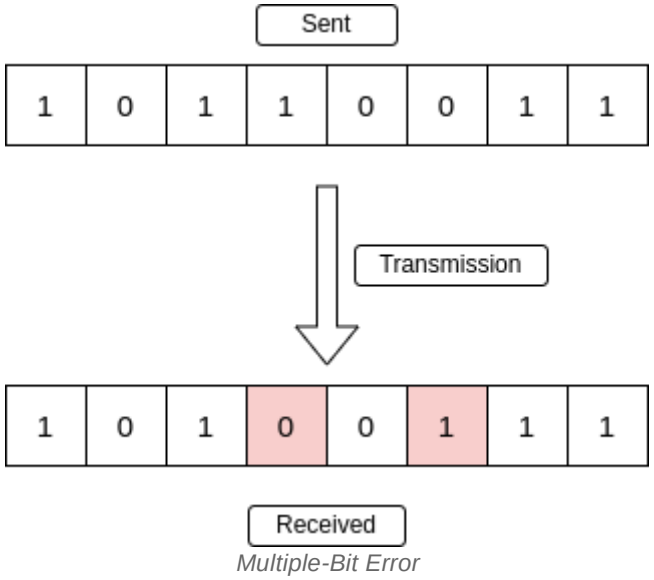
A single-bit error refers to a type of data transmission error that occurs when one bit (i.e., a single binary digit) of a transmitted data unit is altered during transmission, resulting in an incorrect or corrupted data unit.



Single-Bit Error

Multiple-Bit Error

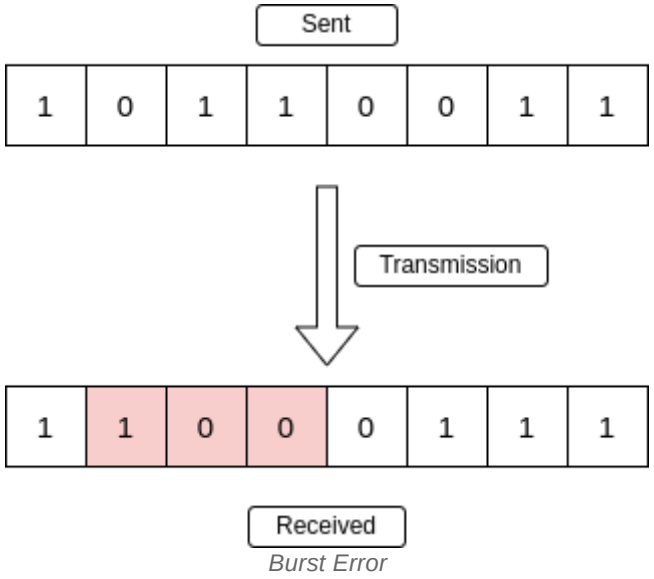
A multiple-bit error is an error type that arises when more than one bit in a data transmission is affected. Although multiple-bit errors are relatively rare when compared to single-bit errors, they can still occur, particularly in high-noise or high-interference digital environments.



Multiple-Bit Error

Burst Error

When several consecutive bits are flipped mistakenly in digital transmission, it creates a burst error. This error causes a sequence of consecutive incorrect values.



Burst Error

Error Detection Methods

To detect errors, a common technique is to introduce redundancy bits that provide additional information. Various techniques for error detection include:

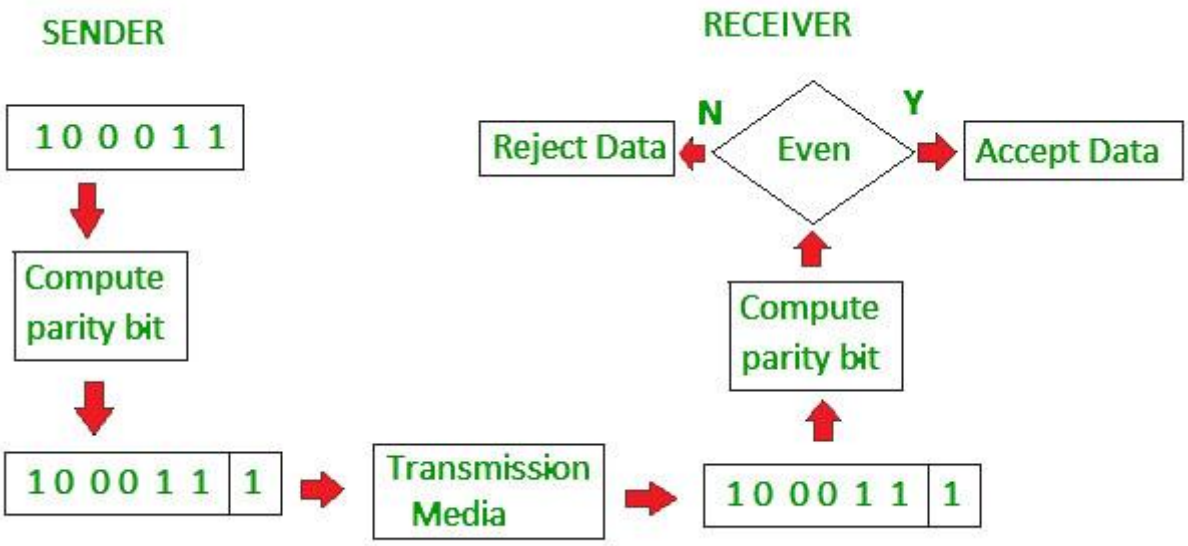
- Simple Parity Check
- Two-Dimensional Parity Check
- Checksum
- Cyclic Redundancy Check (CRC)

Simple Parity Check

Simple-bit parity is a simple error detection method that involves adding an extra bit to a data transmission. It works as:

- 1 is added to the block if it contains an odd number of 1's, and
- 0 is added if it contains an even number of 1's

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



Advantages of Simple Parity Check

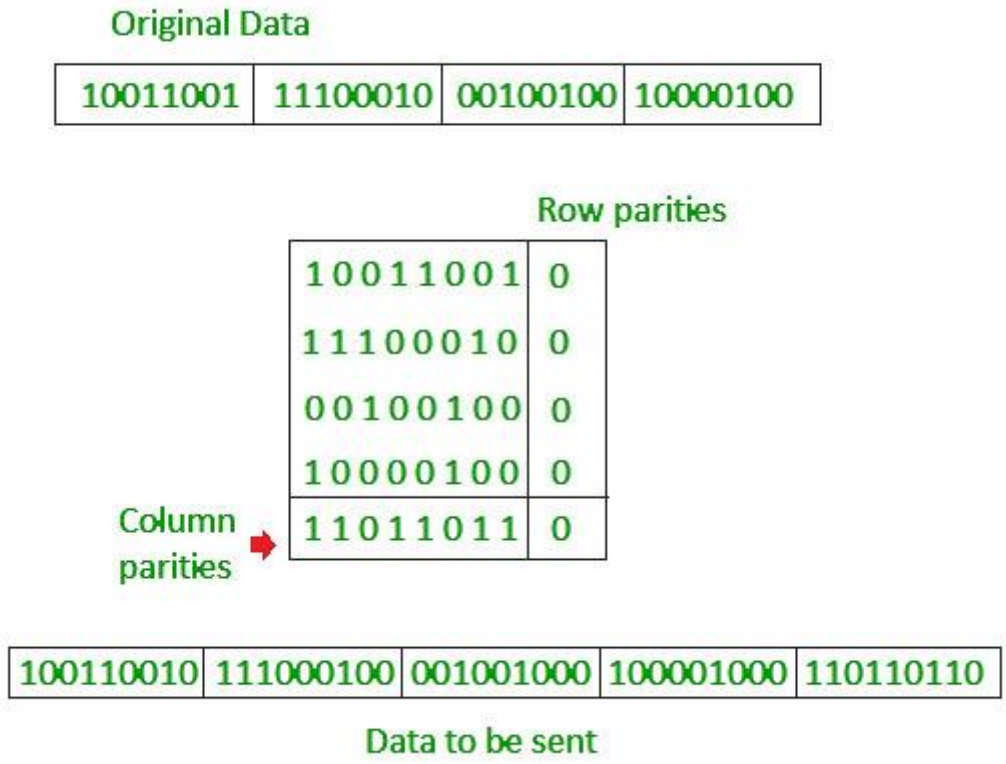
- Simple parity check can detect all single bit error.
- Simple parity check can detect an odd number of errors.
- **Implementation**: Simple Parity Check is easy to implement in both hardware and software.
- **Minimal Extra Data**: Only one additional bit (the parity bit) is added per data unit (e.g., per byte).
- **Fast Error Detection**: The process of calculating and checking the parity bit is quick, which allows for rapid error detection without significant delay in data processing or communication.
- **Single-Bit Error Detection**: It can effectively detect single-bit errors within a data unit, providing a basic level of error detection for relatively low-error environments.

Disadvantages of Simple Parity Check

- Single Parity check is not able to detect even no. of bit error.
- **For example,** the Data to be transmitted is **101010**. Codeword transmitted to the receiver is 1010101 (we have used even parity). Let's assume that during transmission, two of the bits of code word flipped to 1111101.
  On receiving the code word, the receiver finds the no. of ones to be even and hence **no error,** *which is a wrong assumption.*

Two-Dimensional Parity Check

Two-dimensional 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.



Advantages of Two-Dimensional Parity Check

- Two-Dimensional Parity Check can detect and correct all single bit error.
- Two-Dimensional Parity Check can detect two or three bit error that occur any where in the matrix.

Disadvantages of Two-Dimensional Parity Check

- Two-Dimensional Parity Check can not correct two or three bit error. It can only detect two or three bit error.
- If we have a error in the parity bit then this scheme will not work.

Checksum

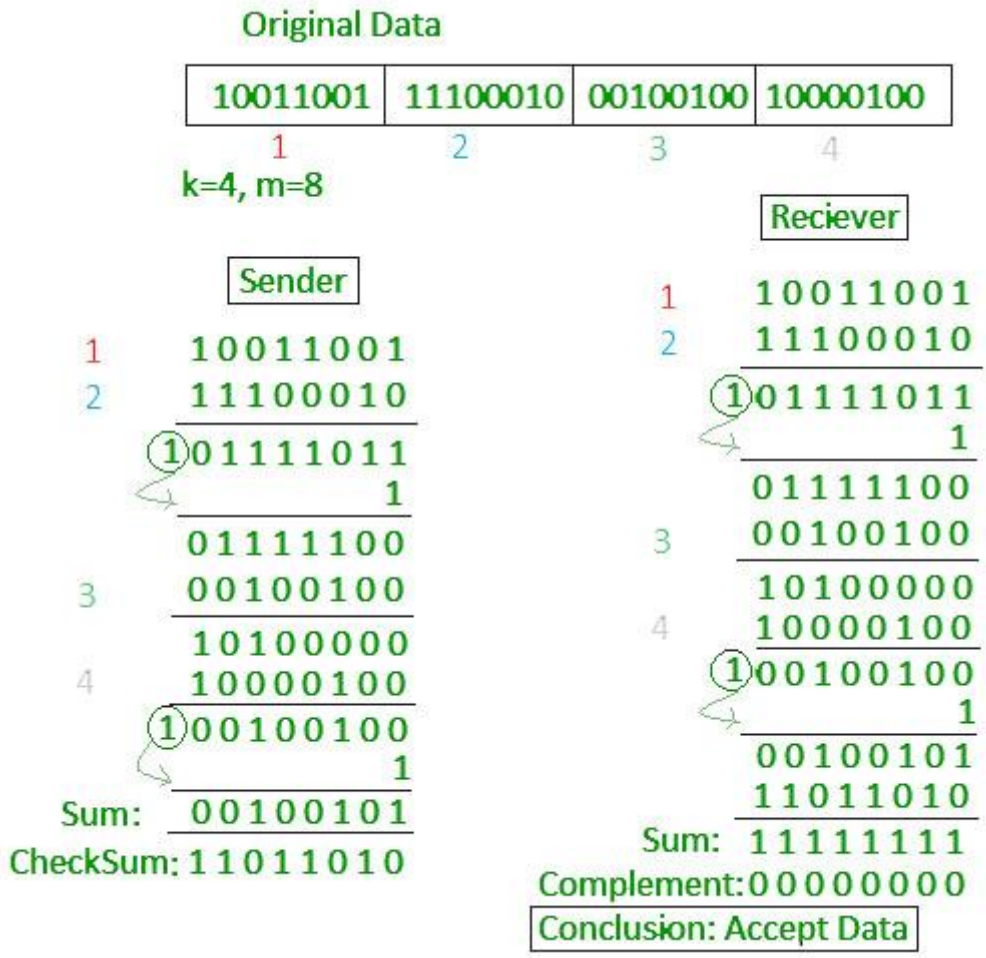
Checksum error detection is a method used to identify errors in transmitted data. The process involves dividing the data into equally sized segments and using a 1's complement to calculate the sum of these segments. The calculated sum is then sent along with the data to the receiver. At the receiver's end, the same process is repeated and if all zeroes are obtained in the sum, it means that the data is correct.

Checksum – Operation at Sender's Side

- Firstly, the data is divided into k segments each of m bits.
- On 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.

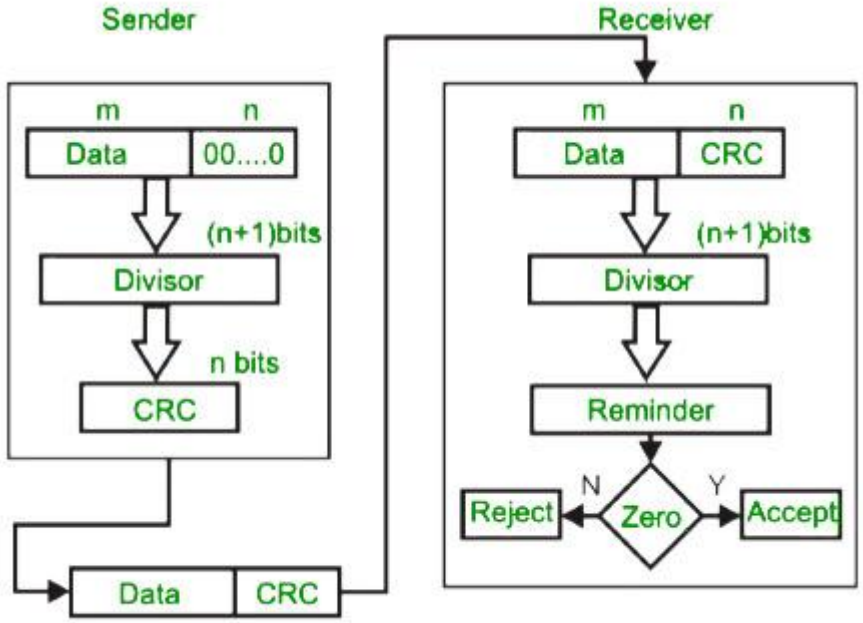
Checksum – Operation at Receiver's Side

- 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.



Cyclic Redundancy Check (CRC)

- Unlike the 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 the 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.



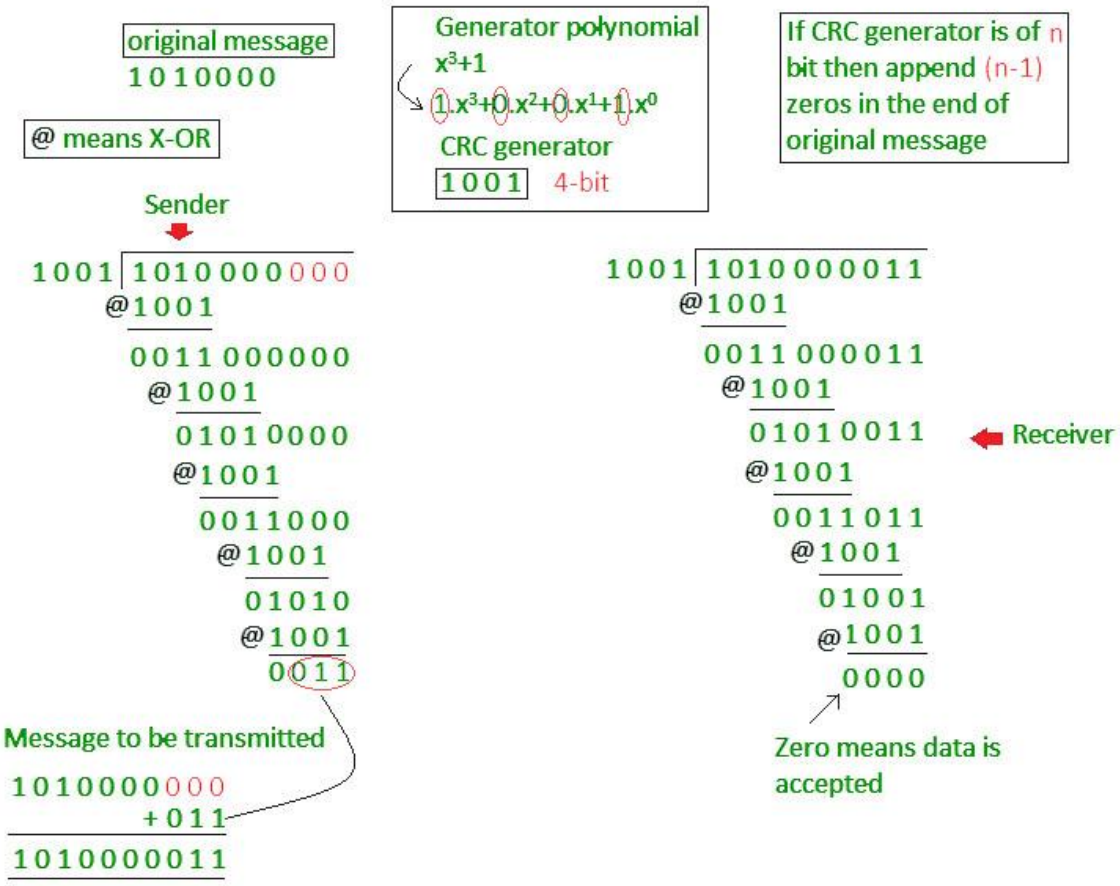
CRC

Working

We have given dataword of length n and divisor of length k.
Step 1: Append (k-1) zero's to the original message
Step 2: Perform modulo 2 division
Step 3: Remainder of division = CRC
Step 4: Code word = Data with append k-1 zero's + CRC
Note:

- CRC must be k-1 bits
- Length of Code word = n+k-1 bits

Example: Let's data to be send is 1010000 and divisor in the form of polynomial is x^3+1. CRC method discussed below.



Advantages of Error Detection

- **Increased Data Reliability:** Error detection ensures that the data transmitted over the network is reliable, accurate, and free from errors. This ensures that the recipient receives the same data that was transmitted by the sender.
- **Improved Network Performance:** Error detection mechanisms can help to identify and isolate network issues that are causing errors. This can help to improve the overall performance of the network and reduce downtime.
- **Enhanced Data Security:** Error detection can also help to ensure that the data transmitted over the network is secure and has not been tampered with.

Disadvantages of Error Detection

- **Overhead**: Error detection requires additional resources and processing power, which can lead to increased overhead on the network. This can result in slower network performance and increased latency.

- **False Positives:** Error detection mechanisms can sometimes generate false positives, which can result in unnecessary retransmission of data. This can further increase the overhead on the network.
- **Limited Error Correction:** Error detection can only identify errors but cannot correct them. This means that the recipient must rely on the sender to retransmit the data, which can lead to further delays and increased network overhead.