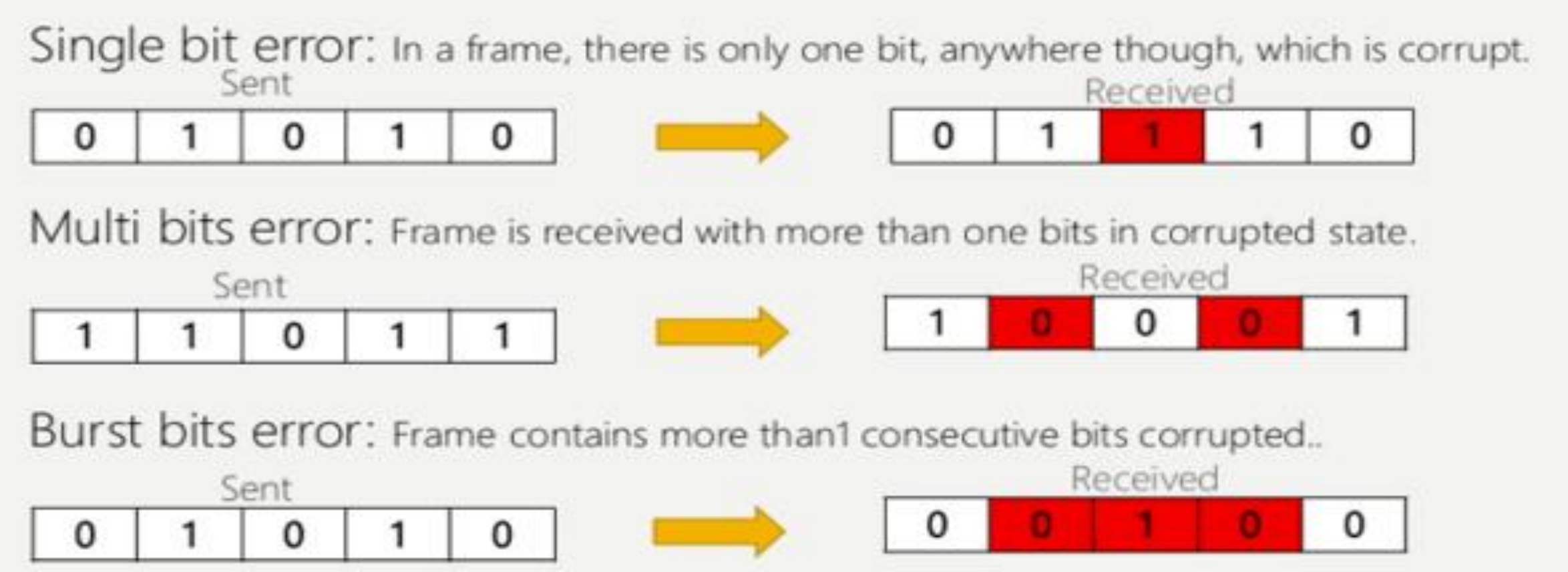
Error Detection & Correction

There are many reasons such as noise, cross-talk etc., which may help data to get corrupted during transmission. The upper layers work on some generalized view of network architecture and are not aware of actual hardware data processing. Hence, the upper layers expect error-free transmission between the systems. Applications such as voice and video may not be that affected and with some errors they may still function well.

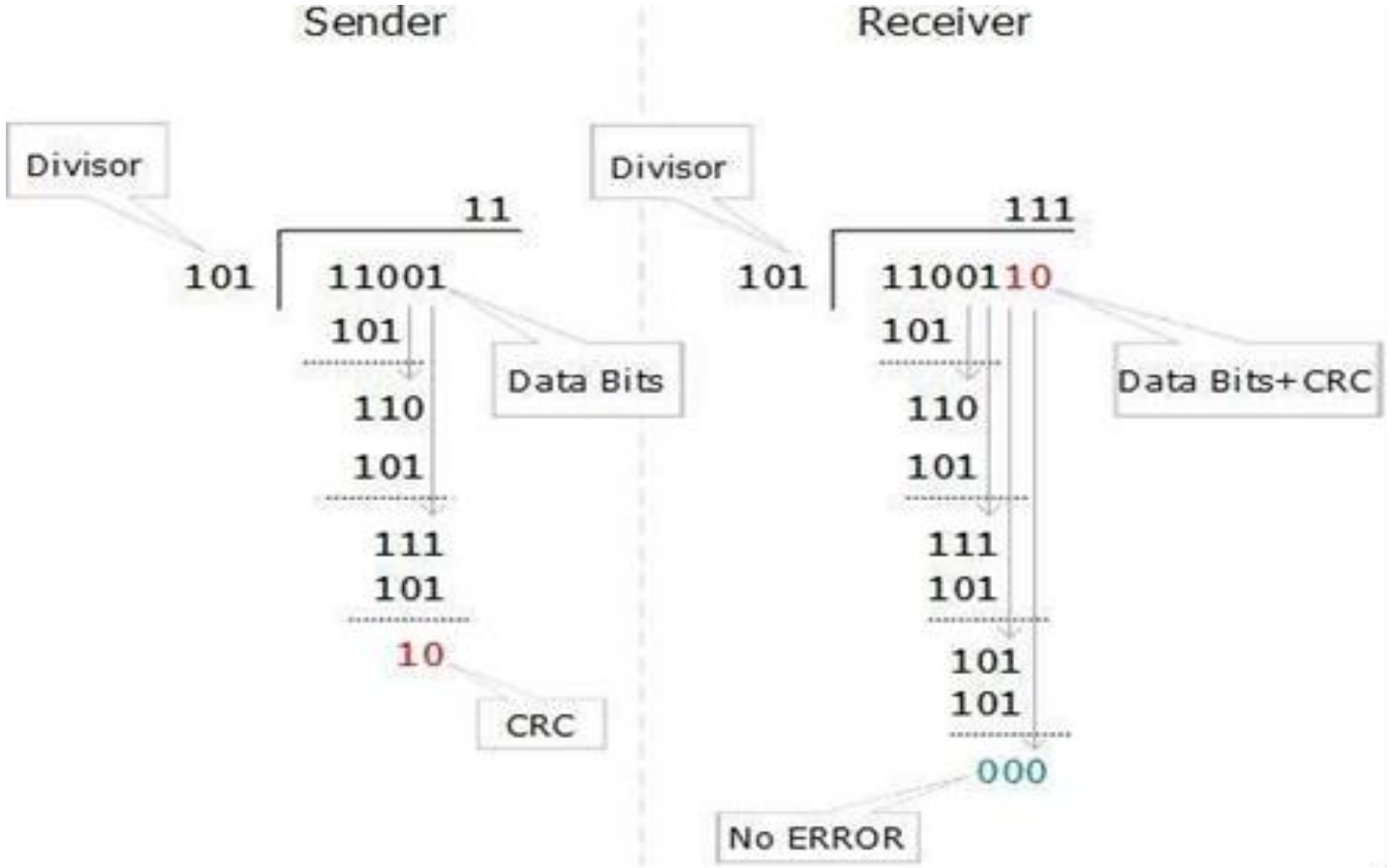
Types of Errors

There may be three types of errors:



Cyclic Redundancy Check (CRC)

This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder.



Hamming Code

Hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors. It was developed by R.W. Hamming for error correction. In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction. When the destination receives this message, it performs recalculations to detect errors and find the bit position that has error.

Encoding a message by Hamming Code

The procedure used by the sender to encode the message encompasses the following steps −

Step 1 − Calculation of the number of redundant bits.

Step 2 − Positioning the redundant bits.

Step 3 − Calculating the values of each redundant bit.

The redundant bits are parity bits. A parity bit is an extra bit that makes the number of 1s either even or odd. The two types of parity are −

- **Even Parity** − Here the total number of bits in the message is made even.
- **Odd Parity** − Here the total number of bits in the message is made odd.

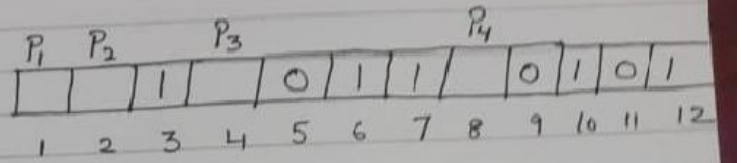
- r_1 is the parity bit for all data bits in positions whose binary representation includes a 1 in the least significant position excluding 1 (3, 5, 7, 9, 11 and so on)
- r_2 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 2 from right except 2 (3, 6, 7, 10, 11 and so on)

- r_3 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 3 from right except 4 (5-7, 12-15, 20-23 and so on)

HAMMING CODE

Data - 1 0 1 1 0 1 0 1 [8 bit]

FRAME-



Calculation of Parity bit

Bit Position

1st bit — $P_1 = \{3, 5, 7, 9, 11\} = \{1, 0, 1, 0, 0\} = 0$

2nd bit — $P_2 = \{3, 6, 7, 10, 11\} = \{1, 1, 1, 1, 0\} = 0$

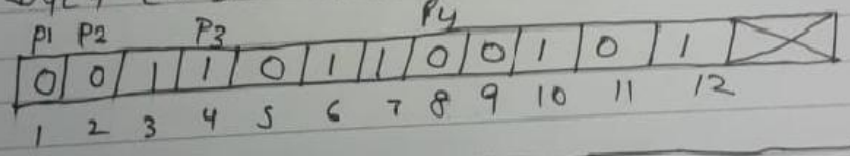
4th bit — $P_3 = \{5, 6, 7, 12\} = \{0, 1, 1, 1\} = 1$

8th bit — $P_4 = \{9, 10, 11, 12\} = \{0, 1, 0, 1\} = 0$

SO

$P_1 = 0$ $P_2 = 0$ $P_3 = 1$ $P_4 = 0$

Data to be Sent :-



4 bit Parity , 8 bit data