Module 3

Prof. Mohammed Juned

Compare ALOHA & Slotted ALOHA

Pure Aloha	**Slotted Aloha**
In this Aloha, any station can transmit the data at any time.	In this, any station can transmit the data at the beginning of any time slot.
In this, The time is continuous and not globally synchronized.	In this, The time is discrete and globally synchronized.
Vulnerable time for Pure Aloha = 2 x Tt	Vulnerable time for Slotted Aloha = Tt

In Pure Aloha, Probability of successful transmission of the data packet = $G \times e^{-2G}$	In Slotted Aloha, Probability of successful transmission of the data packet = $G \times e^{-G}$
In Pure Aloha, Maximum efficiency = 18.4%	In Slotted Aloha, Maximum efficiency = 36.8%
Pure Aloha doesn't reduce the number of collisions to half.	Slotted Aloha reduces the number of collisions to half and doubles the efficiency of Pure Aloha.

CSMA

- CSMA (Carrier Sense Multiple Access)
- It is a **carrier sense multiple access** based on media access protocol to sense the traffic on a channel (idle or busy) before transmitting the data.
- It means that if the channel is idle, the station can send data to the channel. Otherwise, it must wait until the channel becomes idle. Hence, it reduces the chances of a collision on a transmission medium.

CSMA

- **CSMA Access Modes**
- **1-Persistent:**
- In the 1-Persistent mode of CSMA that defines each node, first sense the shared channel and if the channel is idle, it immediately sends the data.
- Else it must wait and keep track of the status of the channel to be idle and broadcast the frame unconditionally as soon as the channel is idle.

CSMA

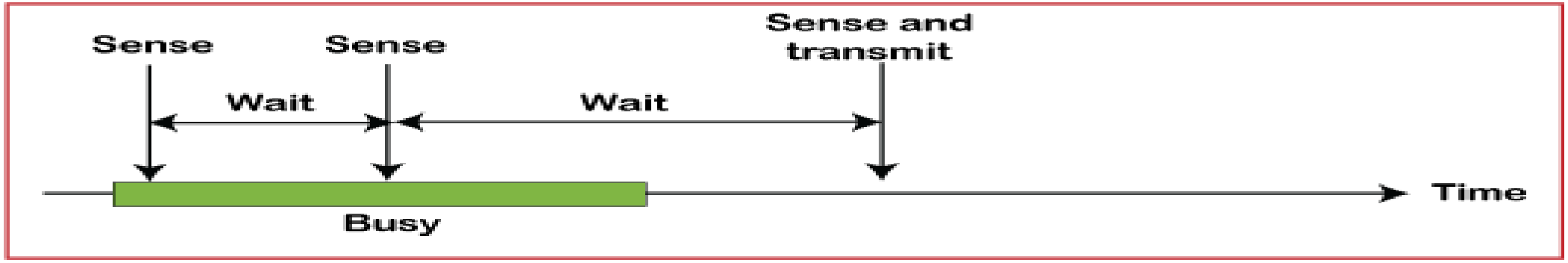
- **Non-Persistent:** It is the access mode of CSMA that defines before transmitting the data, each node must sense the channel.
- If the channel is inactive, it immediately sends the data. Otherwise, the station must wait for a random time (not continuously), and when the channel is found to be idle, it transmits the frames.

CSMA

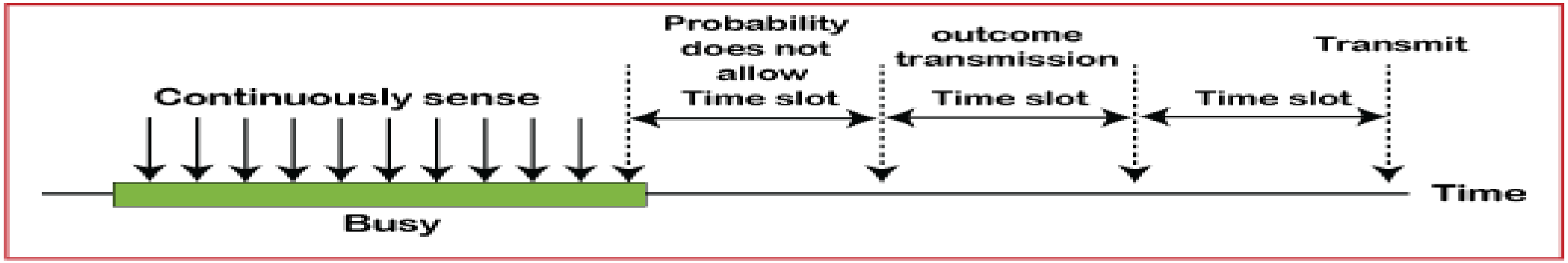
- **P-Persistent:** It is the combination of 1-Persistent and Non-persistent modes.
- The P-Persistent mode defines that each node senses the channel, and if the channel is inactive, it sends a frame with a **P** probability.
- If the data is not transmitted, it waits for a (**q = 1-p probability**) random time and resumes the frame with the next time slot.

CSMA

a. 1-persistent



b. Nonpersistent

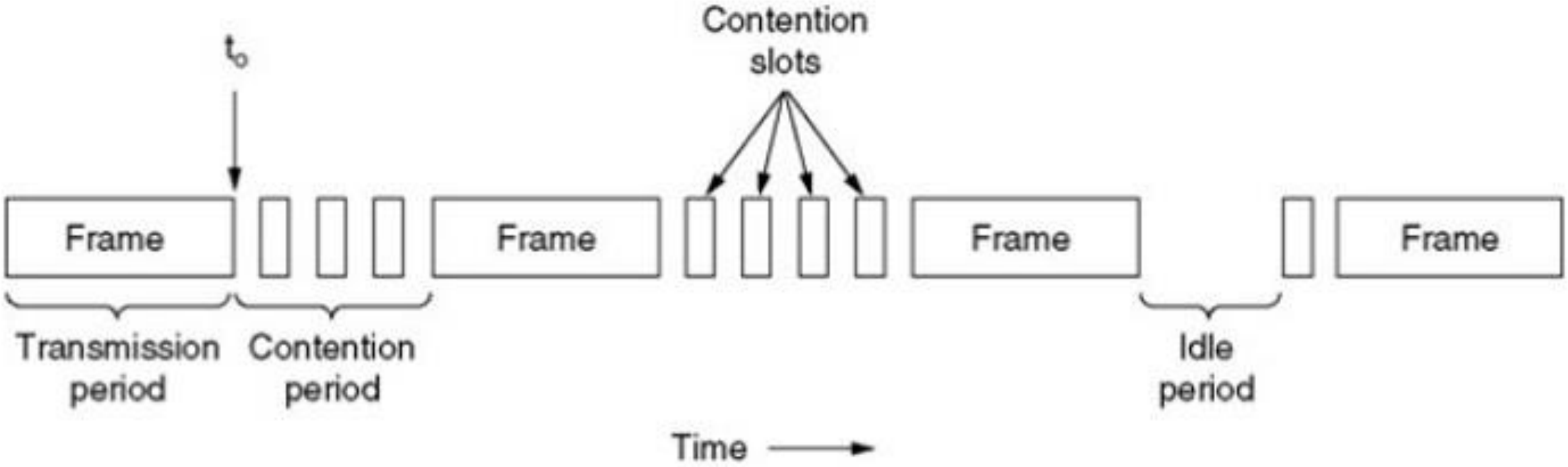


c. p-persistent

CSMA/CD

- Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is a network protocol for carrier transmission that operates in the Medium Access Control (MAC) layer.
- CSMA/CD-CSMA Collision Detection
- The concept of Collision Detection (CSMA/CD) is explained below in stepwise manner:
- **Step 1** – If two stations sense the channel to be ideal and they begin transmitting simultaneously and collision occurs, rather than finish transmitting the frames should just stop transmitting the frames as soon as collision dictated.
- **Step 2** – By terminating frames it would save time and bandwidth. This is called CSMA/CD.
- **Step 3** – It is mainly used LAN in the MAC sub player (part of data link layer network) and Ethernet.

CSMA/CD


Contention
slots
t_0
Frame
Frame
Frame
Frame
Transmission
period
Contention
period
Idle
period
Time

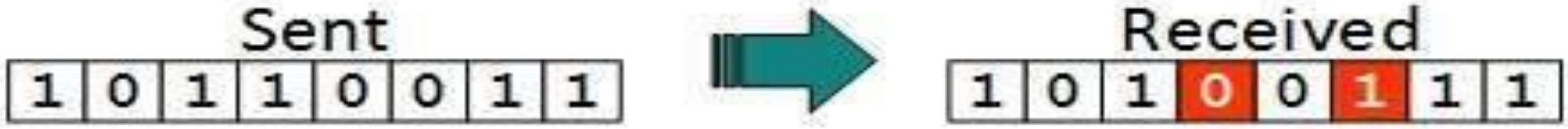
- At the time t0 transition period ends and at the next frame sent between there is the contention period.
- This period is the minimum time a host must transmit such that it can be sure that no other host packet it has been transmitting.
- It will be a minimum period. By this way we can avoid collisions.
- Collision can be detected by looking at the power or pulse width of the received signal and compared with the transmitted signal.
- Power signal is better than the transmitted signal. The ideal period is when all stations are quiet.
- So, at consecutive transmission and cotentition periods the frame can check if a collision is found.

- **Algorithm :**
- The algorithm of CSMA/CD is given below –
- **Step 1** – Check whether the station that wants to transmit the data senses is busy or idle. If a carrier is idle, then the transmission is carried out.
- **Step 2** – The transmission station detects a collision, by using the condition: Tt >= 2 * Tp where Tt is the transmission delay and Tp is the propagation delay.
- **Step 3** – Whenever it detects the collision the station releases the jam signal.
- **Step 4** – After collision has occurred, the transmitting station stops transmitting and waits for some random amount of time called the 'back-off time'. After this time, the station retransmits again.

- **Error Detection & Correction :**
- There are many reasons such as noise, cross-talk etc., which may help data to get corrupted during transmission.
- Data-link layer uses some error control mechanism to ensure that frames (data bit streams) are transmitted with certain level of accuracy.
- But to understand how errors is controlled, it is essential to know what types of errors may occur.

- Types of Errors
- There may be three types of errors:
- **Single bit error**

- In a frame, there is only one bit, anywhere though, which is corrupt.
- **Multiple bits error**

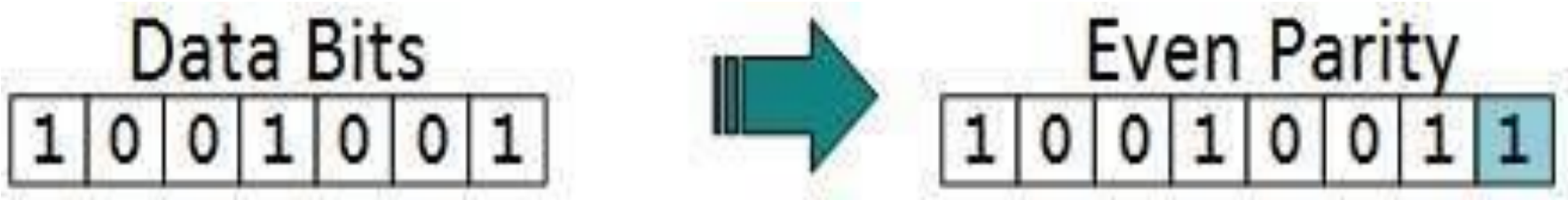


- **Burst error**
- Frame contains more than1 consecutive bits corrupted.

- **Error control mechanism may involve two possible ways:**
- **Error detection**
- **Error correction**

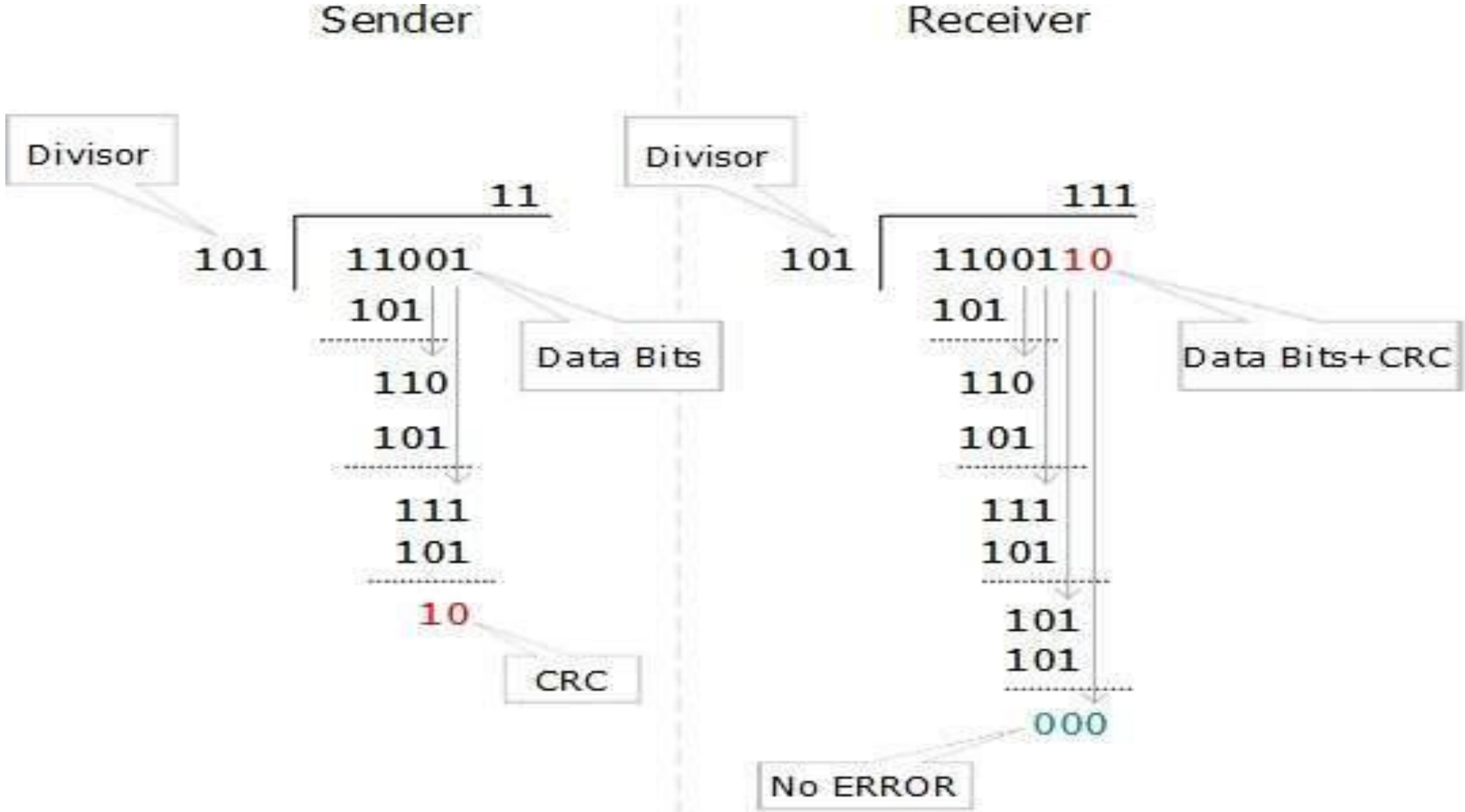
- **Error Detection**
- Errors in the received frames are detected by means of Parity Check and Cyclic Redundancy Check (CRC).
- In both cases, few extra bits are sent along with actual data to confirm that bits received at other end are same as they were sent.
- If the counter-check at receiver' end fails, the bits are considered corrupted.

- **Parity Check**
- One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.
- The sender while creating a frame counts the number of 1s in it. For example, if even parity is used and number of 1s is even then one bit with value 0 is added. This way number of 1s remains even. If the number of 1s is odd, to make it even a bit with value 1 is added.



- The receiver simply counts the number of 1s in a frame.
- If the count of 1s is even and even parity is used, the frame is not-corrupted and is accepted. If the count of 1s is odd and odd parity is used, the frame is still not corrupted.
- If a single bit flips in transit, the receiver can detect it by counting the number of 1s.
- But when more than one bits are erroneous, then it is very hard for the receiver to detect the error.

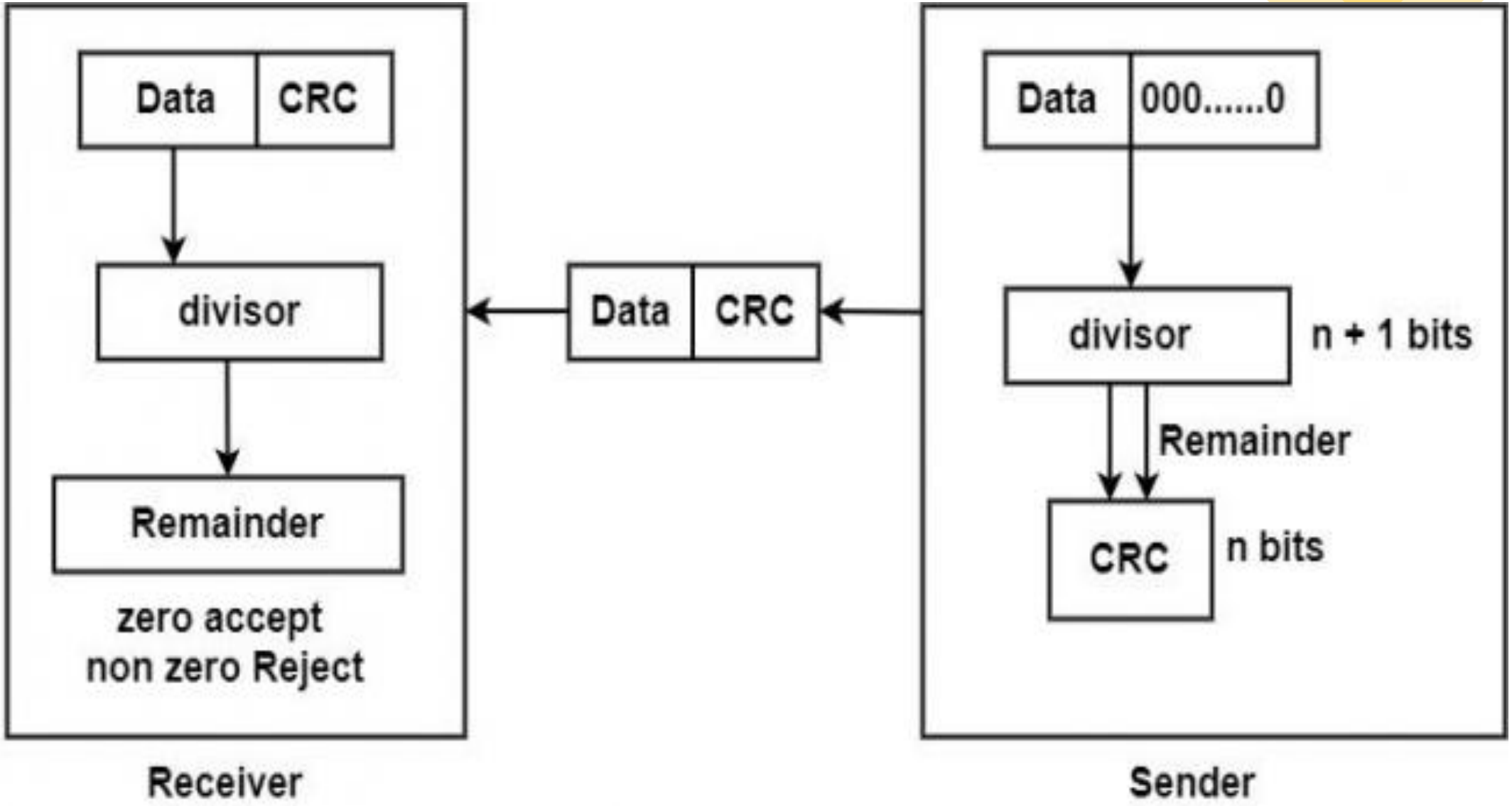
- **Cyclic Redundancy Check (CRC)**
- CRC is a different approach to detect if the received frame contains valid data.
- This technique involves binary division of the data bits being sent.
- The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder.
- Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as codewords.


Sender
Receiver
Divisor
Divisor
11
111
101
11001
101
1100110
101
Data Bits
Data Bits+CRC
110
110
101
101
111
111
101
101
101
101
10
000
CRC
No ERROR

- At the other end, the receiver performs division operation on codewords using the same CRC divisor.
- If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

- **Cyclic Redundancy Check**
- The Cyclic Redundancy Checks (CRC) is the most powerful method for Error-Detection and Correction.
- It is given as a kbit message, and the transmitter creates an (n – k) bit sequence called frame check sequence.
- The out coming frame, including n bits, is precisely divisible by some fixed number.
- Modulo 2 Arithmetic is used in this binary addition with no carries, just like the XOR operation.

- Redundancy means **duplicacy.**
- The redundancy bits used by CRC are changed by splitting the data unit by a fixed divisor. The remainder is CRC.
- **Qualities of CRC**
- It should have accurately one less bit than the divisor.
- Joining it to the end of the data unit should create the resulting bit sequence precisely divisible by the divisor.


Data
CRC
divisor
Remainder
zero accept
non zero Reject
Receiver
Data
CRC
Data
000.....0
divisor
n + 1 bits
Remainder
CRC
n bits
Sender