UNIT-2
Data Link Layer

B SAI BABA,M.Tech(Ph.D),VIT,Bhimavaram

Syllabus

The Data Link Layer:

- Design issues
- Error detection and correction
- Elementary data link protocols
- Sliding window protocols
- HDLC
- The data link layer in the internet.

Syllabus

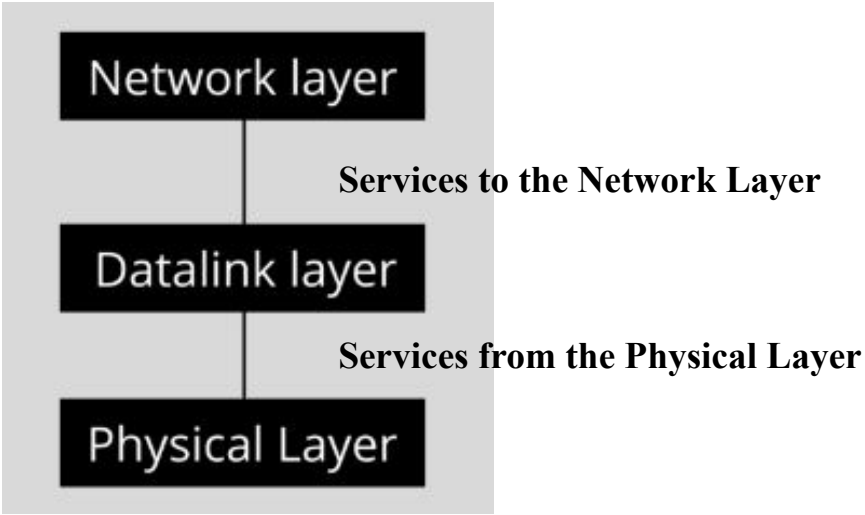
- **The Media Access Sub Layer**
  - Channel allocation problem
  - Multiple access protocols.

Data Link Layer Design Issues

- In the OSI model, the data link layer is a 6th layer from the top and 2nd layer from the bottom.
- The data link layer is responsible for maintaining the data link between two hosts or nodes.
- **The main functions and the design issues of this layer are**
  - **Providing Services to The Network Layer**
  - **Framing**
  - **Addressing**
  - **Error Control**
  - **Flow Control**
  - **Access Control**

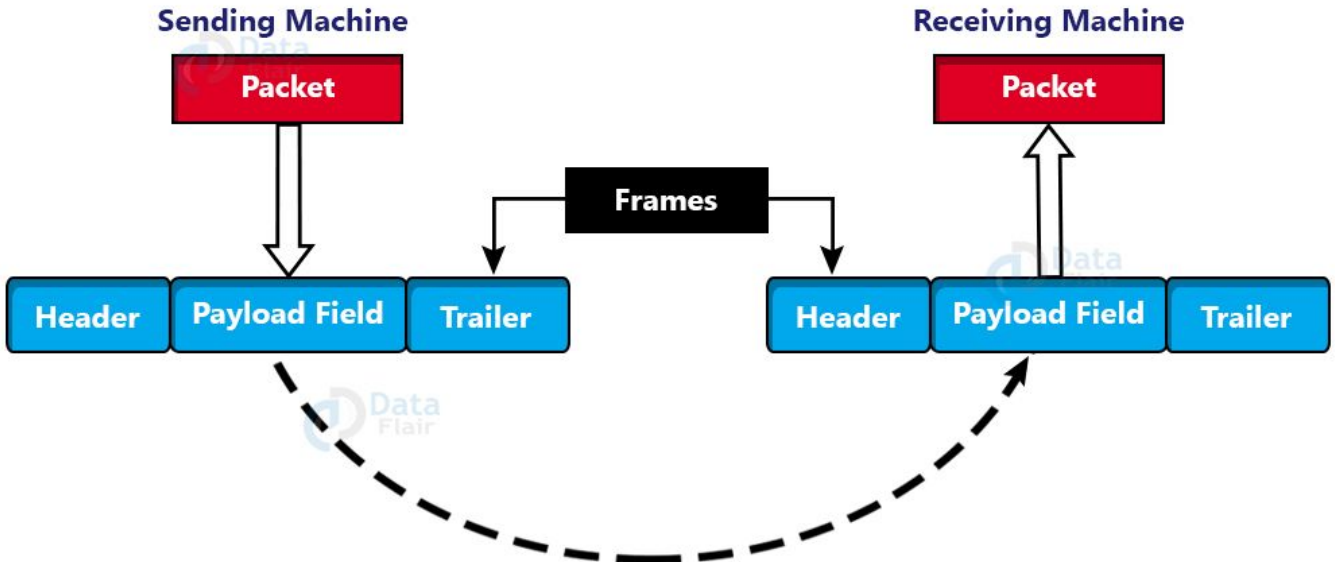
Providing Services to The Network Layer

- In the OSI Model, each layer uses the services of the layer below it and provides services to the layer above it.
- The data link layer uses the services offered by the physical layer.
- The primary function of this layer is to provide a well defined service interface to network layer above it.



Framing

- To provide service to the network layer, the data link layer must use the service provided to it by the physical layer.
- What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination.
- Physical layers only just accept and transfer stream of bits without any regard to meaning or structure. Therefore it is up to data link layer to simply develop and recognize frame boundaries.
- Framing purpose is to divide the stream of data from the network layer into manageable frames that can be transmitted over the physical medium.
- The data link layer encapsulates Network Layer **Packets** into **Frames** by adding header and trailer information.



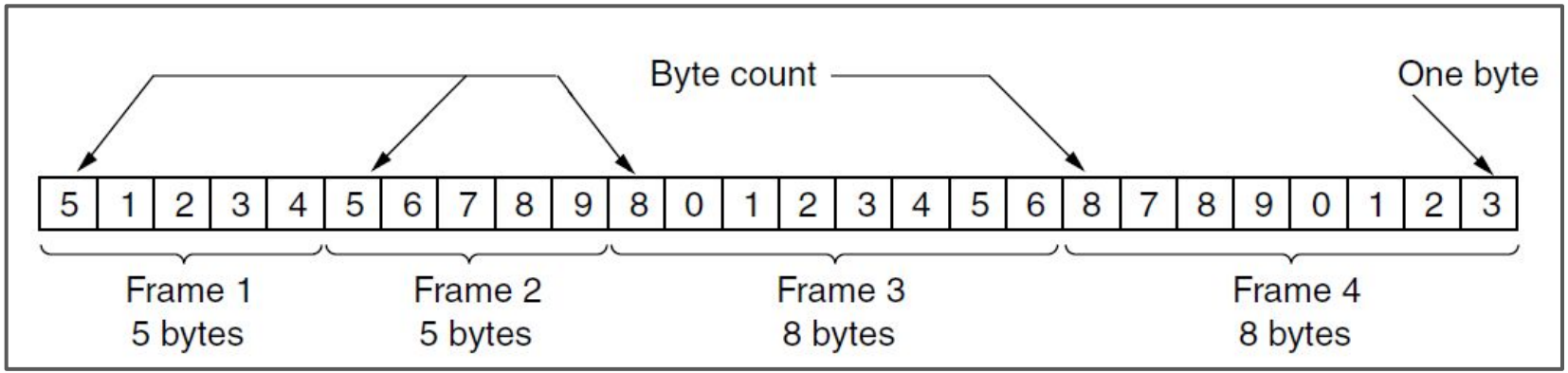
1. **Frame Header:** It contains the source and the destination addresses of the frame and the control bytes.
2. **Payload field :** It contains the data packet from network layer.
3. **Trailer:** It contains the error detection and error correction bits. It is also called a Frame Check Sequence (FCS)

Framing Methods

1. Byte count
2. Flag bytes with byte stuffing
3. Flag bits with bit stuffing

1. Byte count

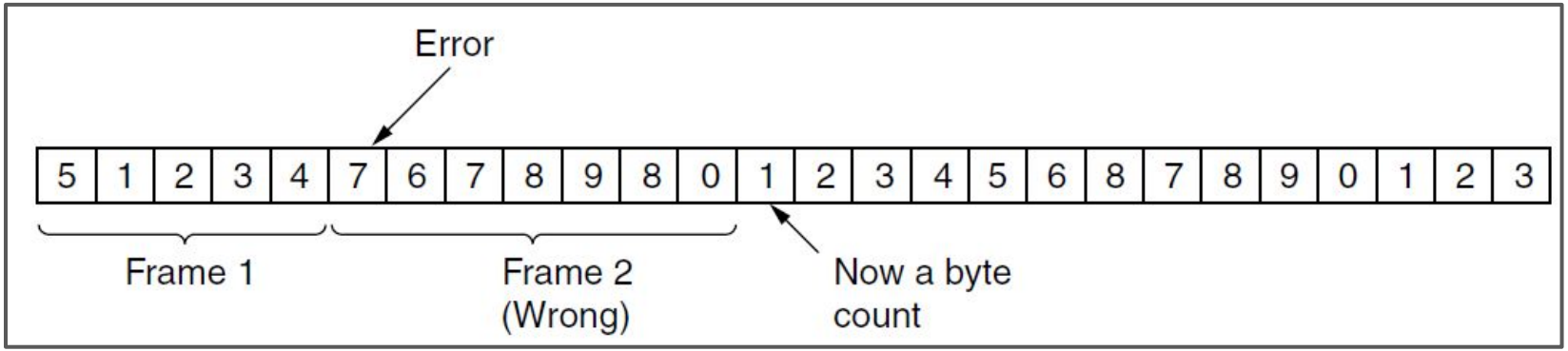
- This method is **rarely used** and is generally required to count total number of Bytes that are present in frame. This is be done by using field in header.
- Byte count method ensures data link layer at the receiver about total number of Bytes that follow, and about where the frame ends.



: Without errors:

1. Byte count:

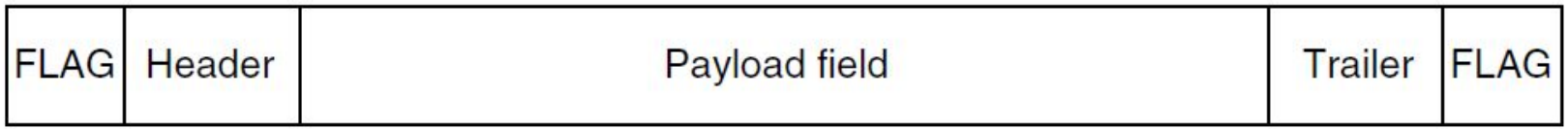
: With errors:



- There is disadvantage also of using this method i.e., if anyhow Byte count is disturbed or distorted by an error occurring during transmission, then destination or receiver might lose synchronization.
- The destination or receiver might also be not able to locate or identify beginning of next frame.

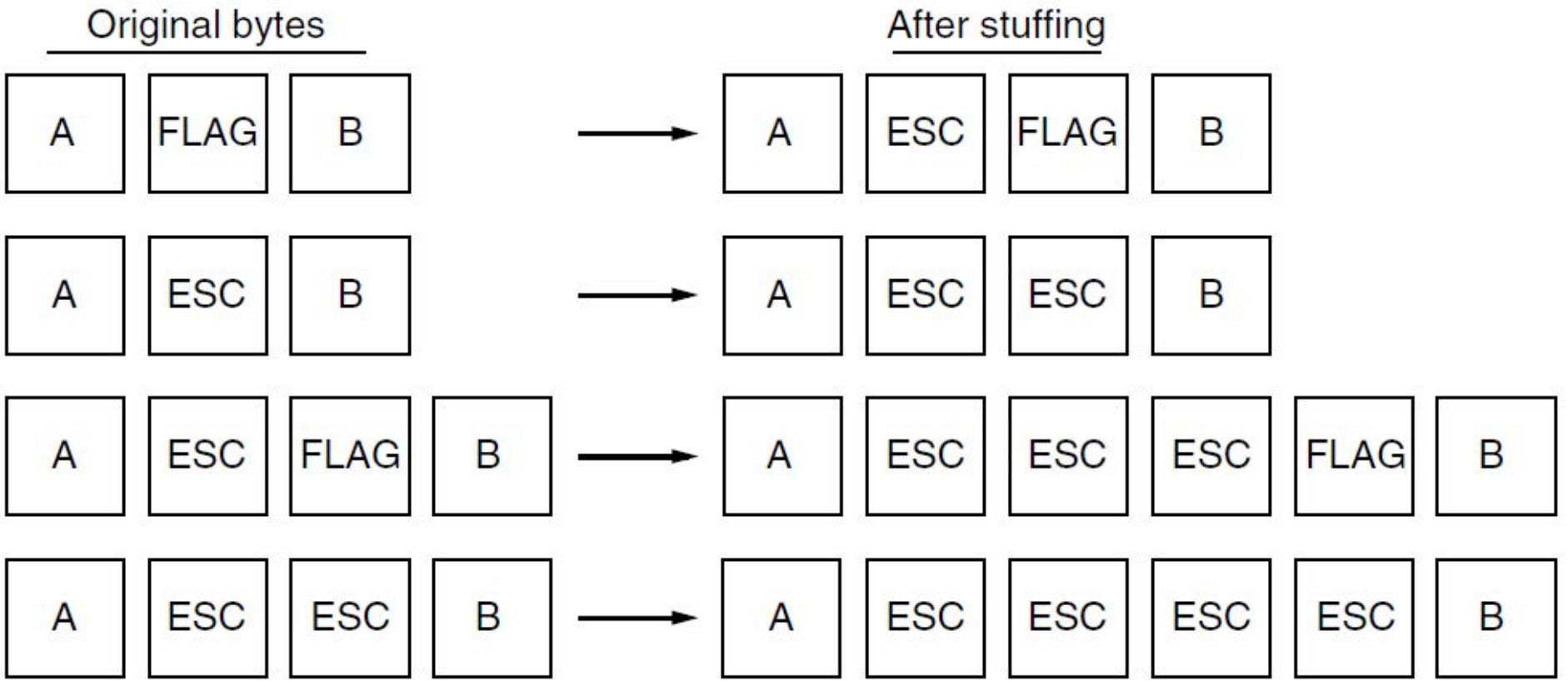
2. Flag bytes with byte stuffing

- The second framing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes.
- Often the same byte, called a flag byte, is used as both the starting and ending delimiter.
- This byte is shown in Fig. 3-4(a) as FLAG.
- Two consecutive flag bytes indicate the end of one frame and the start of the next. Thus, if the receiver ever loses synchronization it can just search for two flag bytes to find the end of the current frame and the start of the next frame.



A Frame delimited by flag bytes

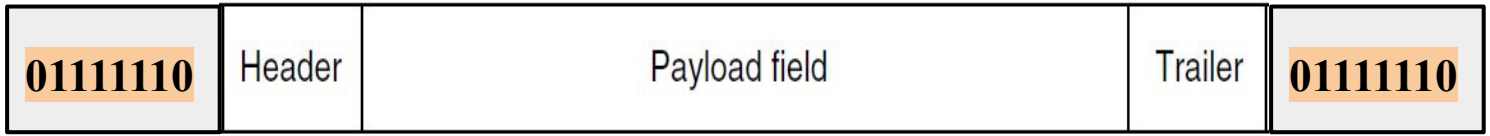
Four examples of byte sequences before and after byte stuffing



- It may happen that t**he flag byte occurs in the data** .One way to solve this problem is to have the sender's data link layer insert a special **escape byte (ESC)** just before each ‘‘accidental’’ flag byte in the data.
- Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.
- The data link layer on the receiving end removes the escape bytes before giving the data to the network layer. This technique is called **byte stuffing.**

3. Flag bits with bit stuffing

- Framing can be also be done at **the bit level**, so frames can contain an arbitrary number of bits made up of units of any size.
- It was developed for the once very popular HDLC (Highlevel Data Link Control) protocol.
- **Each frame begins and ends with a special bit pattern, 01111110**



A Frame delimited by a flag byte 01111110

- Whenever **the sender's data link layer encounters five consecutive 1s in the data**, it automatically **stuffs a 0 bit into the outgoing bit stream.**

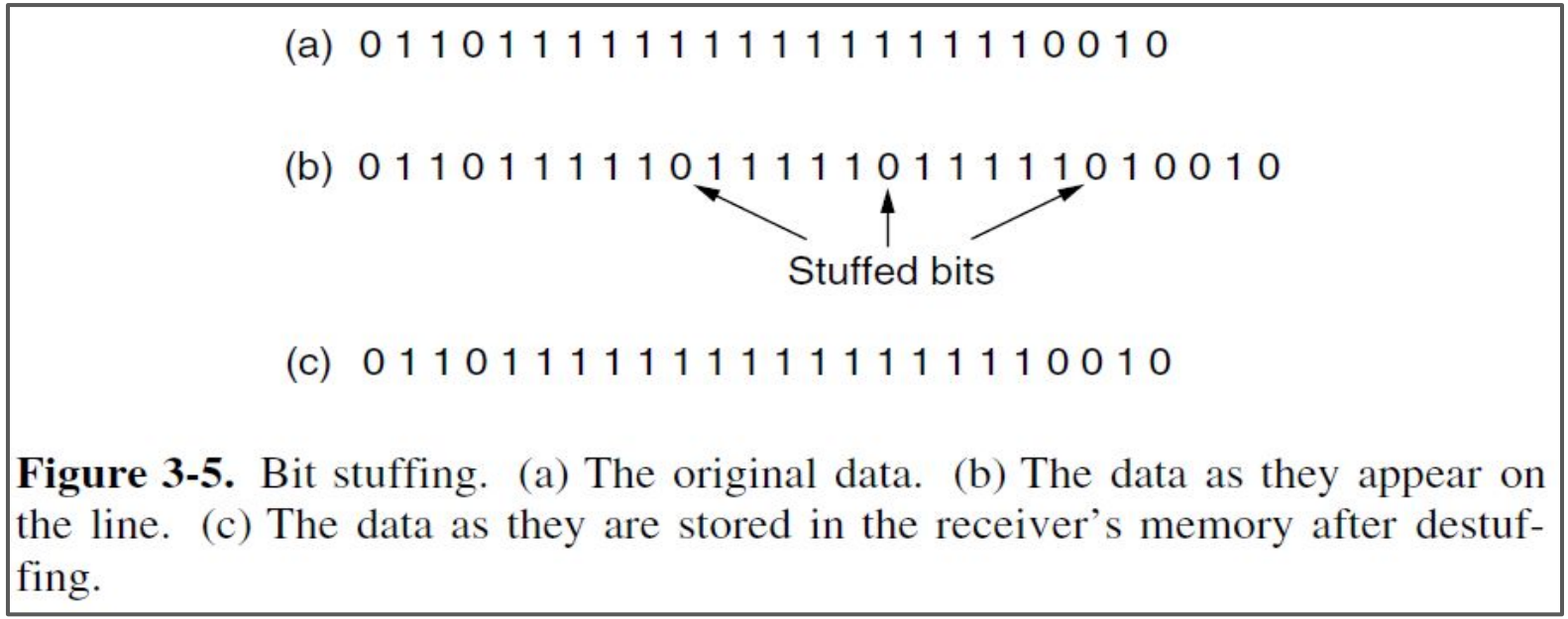
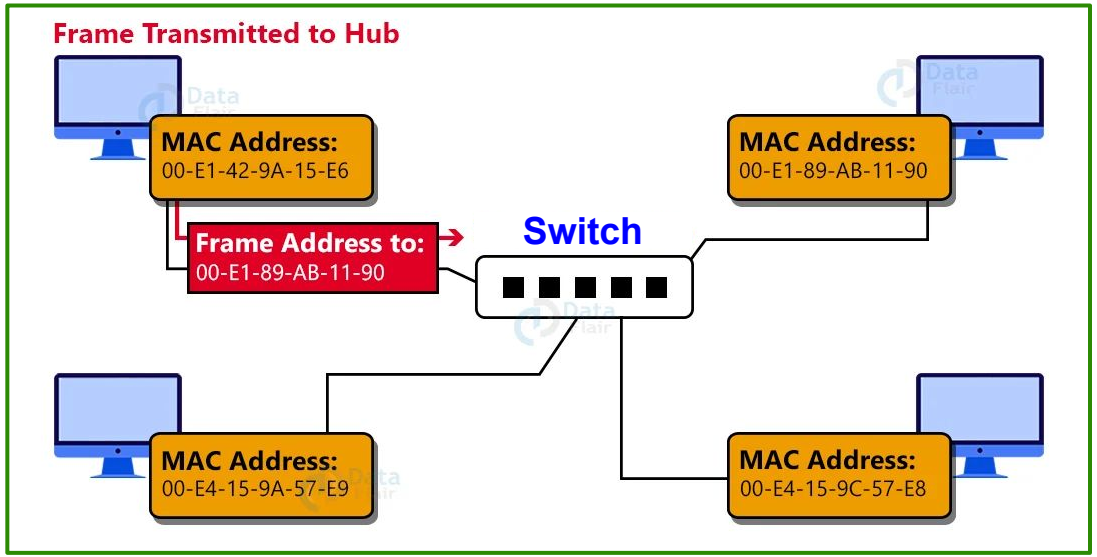


Figure 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

- When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically destuffs (i.e., deletes) the 0 bit.
- Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing.
- If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110. Figure 3-5 gives an example of bit stuffing.

Data Link Layer Addressing

- In the data link layer of the network protocol stack, addressing is used to identify network nodes (devices) within a local network or a LAN (Local Area Network).
- There are **two main types of a** the data link layer:
  - **Media Access Control (MAC) Layer**
  - **Logical Link Control (LLC) Layer**



1. MAC Address (Media Access Control Address)

- MAC addresses are **unique identifiers** assigned to **Network Interface Cards (NICs)** at the factory.
- They consist of **6 bytes (48 bits)** and are usually represented in 12-digit hexadecimal notation (e.g., 00:1A:2B:3C:4D:5E).
- MAC addresses are globally unique, allowing each network device to have a unique identifier.
- The source and destination MAC addresses are included in the **frame header** of data frames at the data link layer.
- This allows devices to identify the intended recipient of the frame.

2. Logical Link Control Layer

- The LLC layer is responsible for **managing communication** between devices on the same network segment.
- It provides **a standardized interface** to the Network Layer (Layer 3) above it and the MAC layer below it.
- The LLC layer helps to ensure that upper-layer protocols can communicate efficiently over various types of data link technologies.

Key functions of the LLC layer include:

Error Control: Detecting and handling errors in the data link layer by using mechanisms like acknowledgments and retransmissions.

Flow Control: The Data Link Layer manages the flow of data between devices to prevent congestion and ensure efficient data transmission.

Link Management: Establishing, maintaining, and terminating logical links between devices.

Error Control

- The data link layer ensures error free link for data transmission. The issues it caters to with respect to error control are −
  - **Dealing with transmission errors**
  - **Sending acknowledgement frames in reliable connections**
  - **Retransmitting lost frames**
  - **Identifying duplicate frames and deleting them**
  - **Controlling access to shared channels in case of broadcasting.**

Flow Control

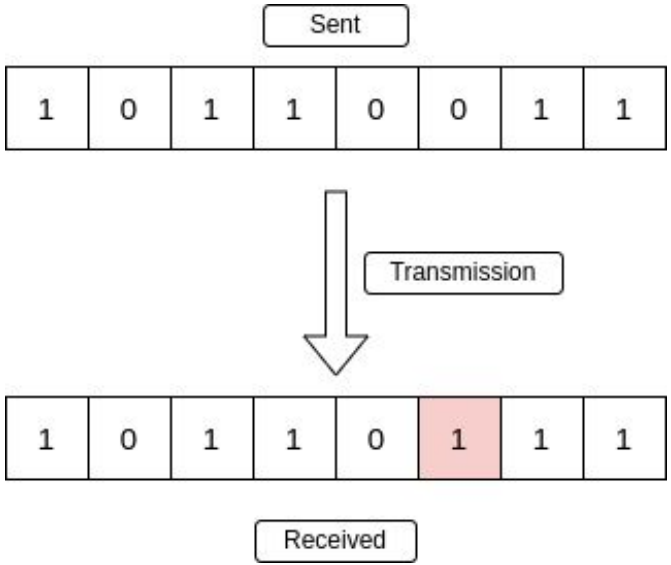
- The data link layer regulates flow control so that a fast sender does not drown a slow receiver.
- When the sender sends frames at very high speeds, a slow receiver may not be able to handle it. There will be frame losses even if the transmission is error-free.
- The two common approaches for flow control are −
  - **Feedback based flow control**
  - **Rate based flow control**

Error detection and correction

- **Error** is a condition when the receiver's information **does not match** the sender's information.
- During transmission, digital signals suffer from **noise** 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**
  - **Multiple Bit Error**
  - **Burst Error**

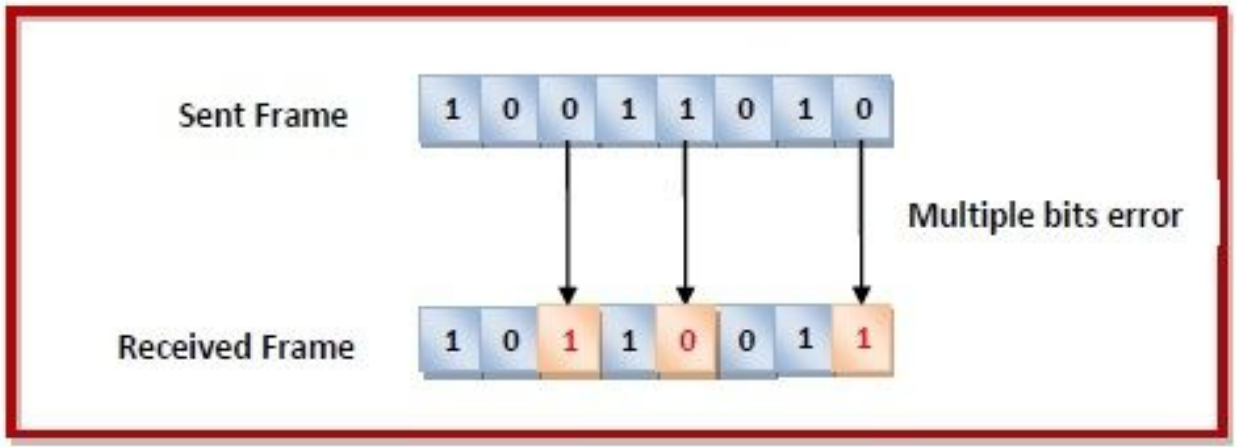
Single Bit Error

- 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.



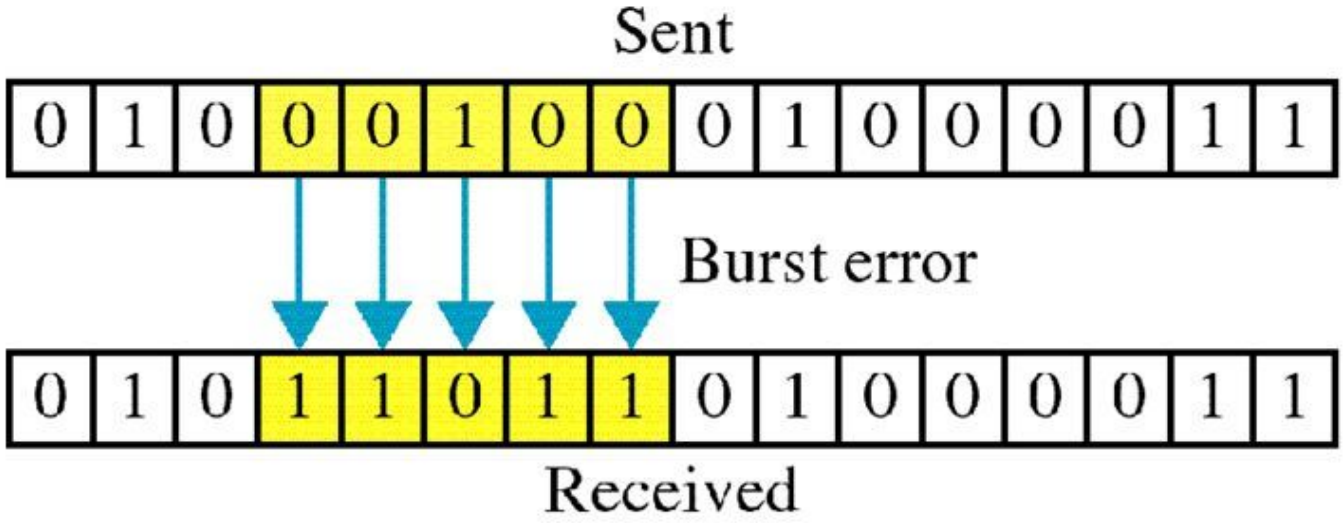
Multiple-Bit Error

- 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.



Burst Error

- **More than one consecutive bit is corrupted in the received frame.**



Error Control

- Error control can be done in two ways
  - **Error detection** − Error detection involves checking whether any error has occurred or not. The number of error bits and the type of error does not matter.
  - **Error correction** − Error correction involves ascertaining the exact number of bits that has been corrupted and the location of the corrupted bits.

Error Detection Methods:

- ★ **Parity**
- ★ **Checksums**
- ★ **Cyclic Redundancy Checks (CRCs)**

Error Correction Methods:

- ★ **Hamming codes**
- ★ **Binary convolutional codes**
- ★ **Reed-Solomon codes**
- ★ **Low-Density Parity Check codes**

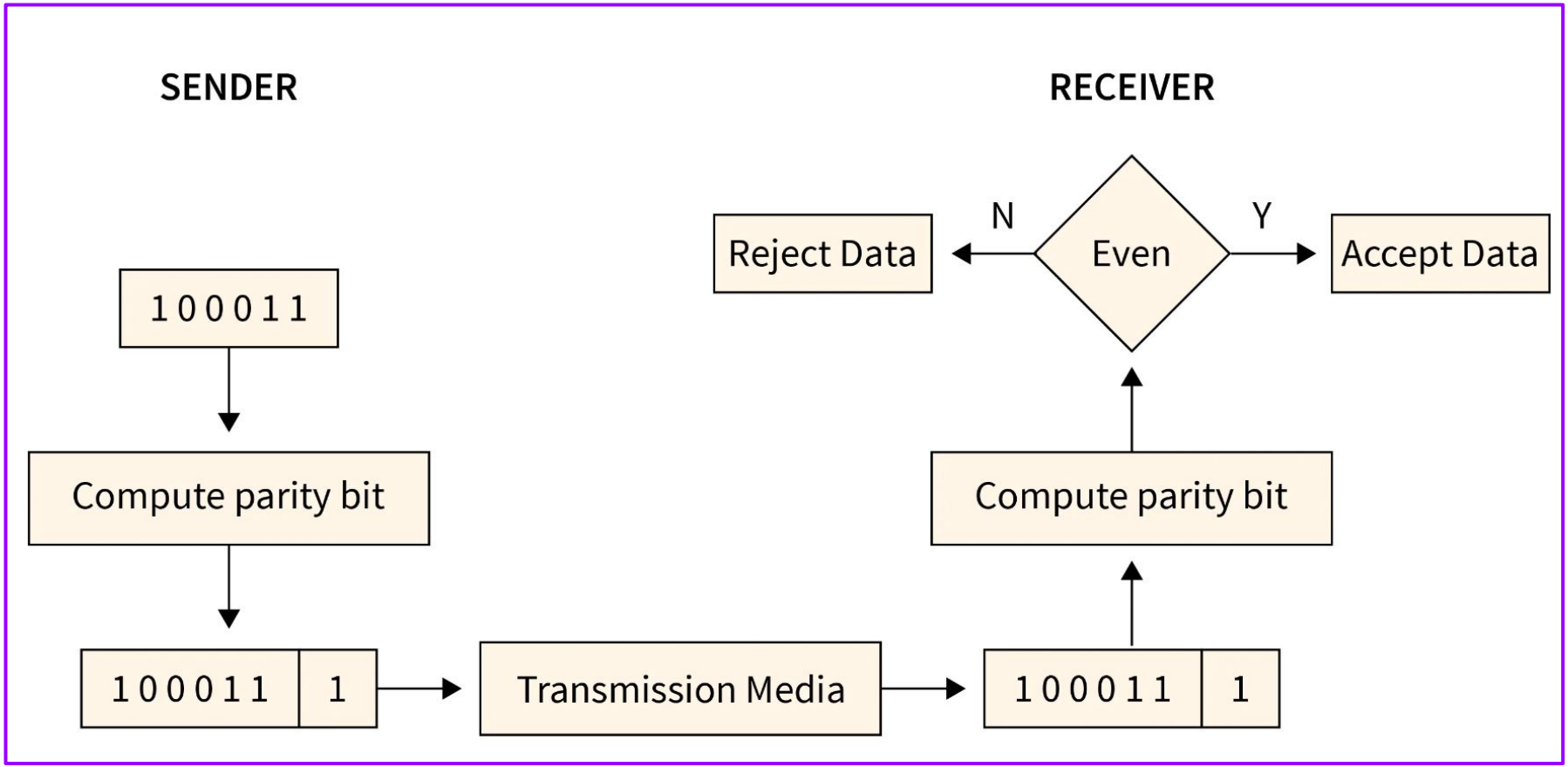
Error Detection Methods:

1. Parity Check:

- **One extra bit** is transmitted in addition to the original bits to make **the number of 1s even in the case of even parity or odd in the case of odd parity.**
- While creating a frame, the sender counts the number of 1s in it and adds the parity bit in the following way
  - **In case of even parity:** If a number of 1s is even then parity bit value is 0. If the number of 1s is odd then parity bit value is 1.
  - **In case of odd parity:** If a number of 1s is odd then parity bit value is 0. If a number of 1s is even then parity bit value is 1.

On receiving a frame, the receiver counts the number of 1s in it. In case of even parity check, if the count of 1s is even, the frame is accepted, otherwise, it is rejected. A similar rule is adopted for odd parity check.

Example of Simple Even Parity Check



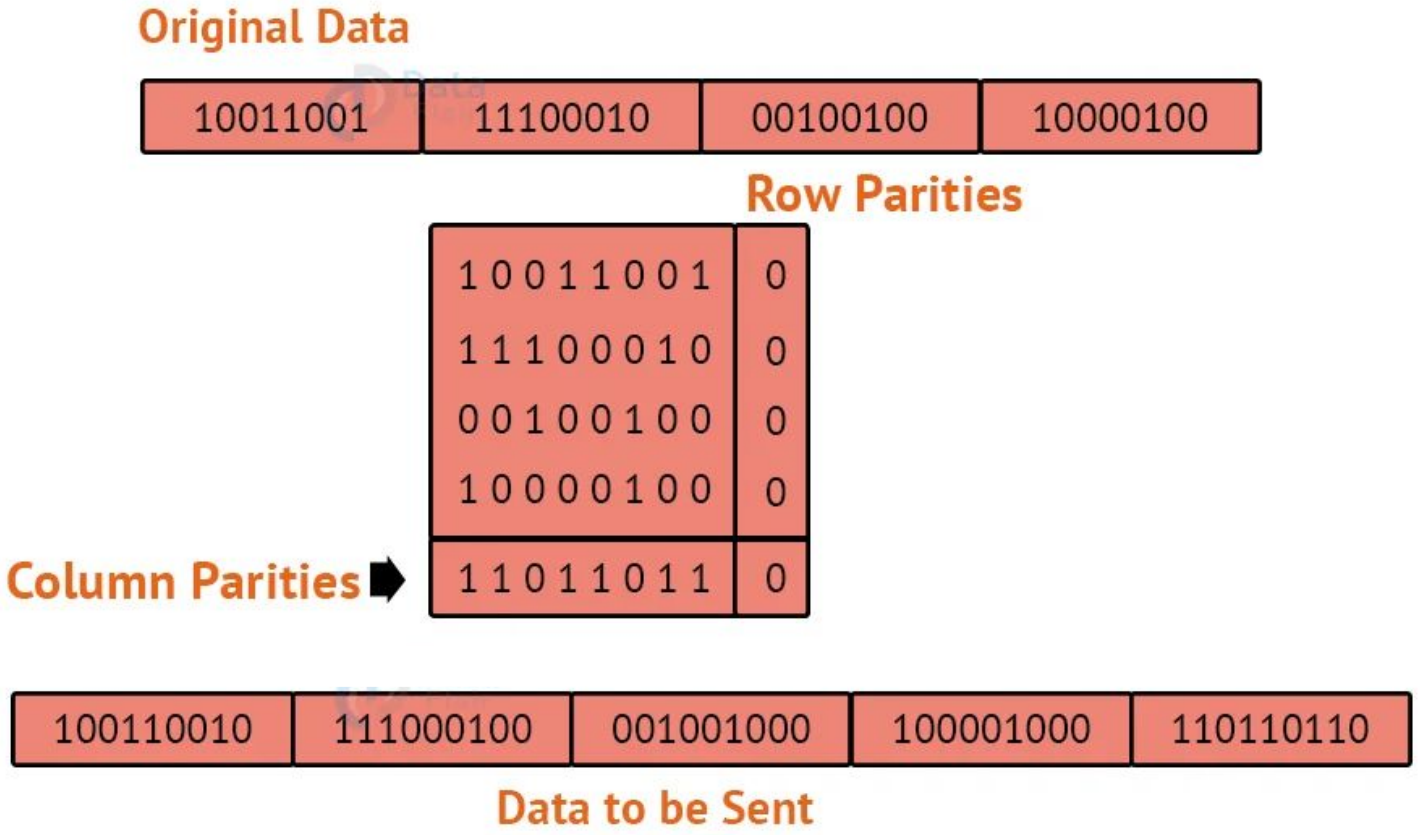
Disadvantage:

- **Only single-bit error** is detected by this method, **it fails in multi-bit error detection** .
- It can not detect an error in case of **an error in two bits**.

* Two-Dimensional Parity Check:

- For each row, parity check bits are calculated, which is identical to a basic parity check bit.
- For each column, parity check bits are computed and transmitted together with the data.
- These are compared with the parity bits calculated on the received data at the receiving end.

Two-Dimensional Parity Check

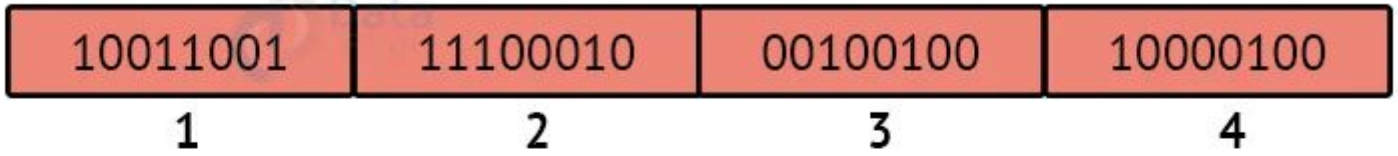


2. Checksum:

Checksum is a error detection which detects the error by dividing the data into the segments of equal size and then use 1's complement to find the sum of the segments and then sum is transmitted with the data to the receiver and same process is done by the receiver and at the receiver side, all zeros in the sum indicates the correctness of the data.

- ➔ First of all **data** is divided into **k segments** in a checksum error detection scheme and **each segment has m bits.**
- ➔ For finding out the sum at the sender's side, all segments are added through **1's complement** arithmetic. And for determining the checksum **we complement the sum**.
- ➔ Along with **data segments, the checksum segments** are also **transferred**.
- ➔ All the segments that are received on the receiver's side are added through 1's complement arithmetic to determine the sum. Then complement the sum also.
- ➔ The received data is **accepted only** on the condition that the **result is found to be 0**. And if the **result is not 0** then it will be **discarded.**

Original Data



k=4 , m=8

Original Data

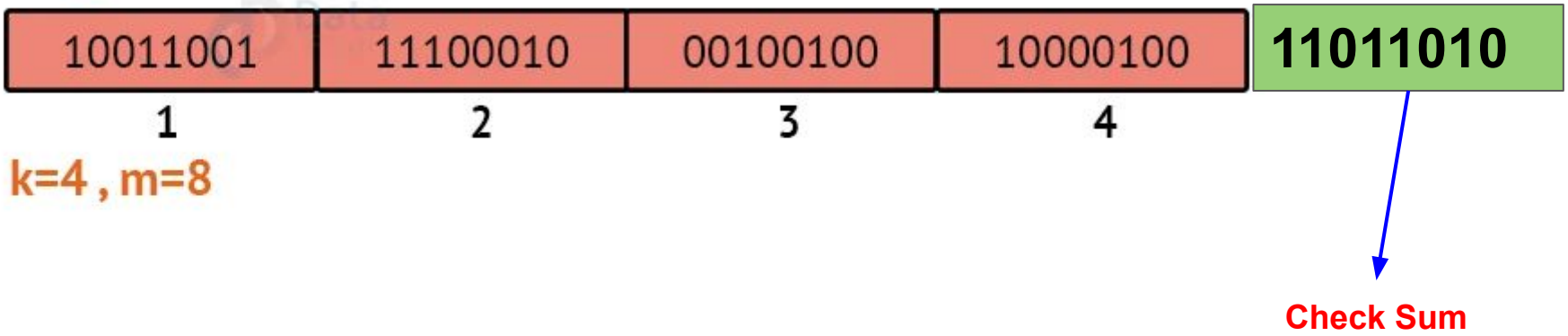
10011001	11100010	00100100	10000100
1	2	3	4

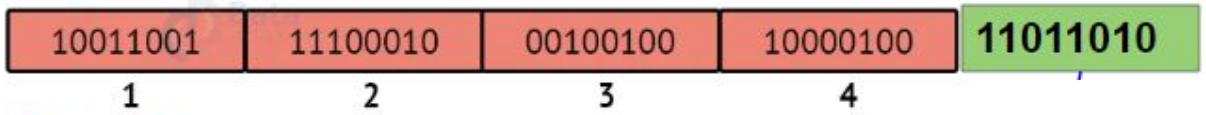
k=4 , m=8

SENDER

```
1        10011001
2        11100010
         ----------
         101111011
                1
         ----------
         01111100
3        00100100
         ----------
         10100000
4        10000100
         ----------
         100100100
               1
         ----------
Sum:     00100101
         ----------
CheckSum: 11011010
```

Original Data





RECIEVER

```
1   10011001
2   11100010
    ----------
    101111011
              1
    ----------
    01111100
3   00100100
    ----------
    10100000
4   10000100
    ----------
    100100100
            1
    ----------
    00100101
    11011010
    ----------
Sum: 11111111
Complement: 00000000
```

Conclusion: Accept Data

Eg: **10110011 10101011 01011010 11010101, K=4 & m=8**

Sender Side:

Sum = ?

CheckSum = ?

Receiver Side:

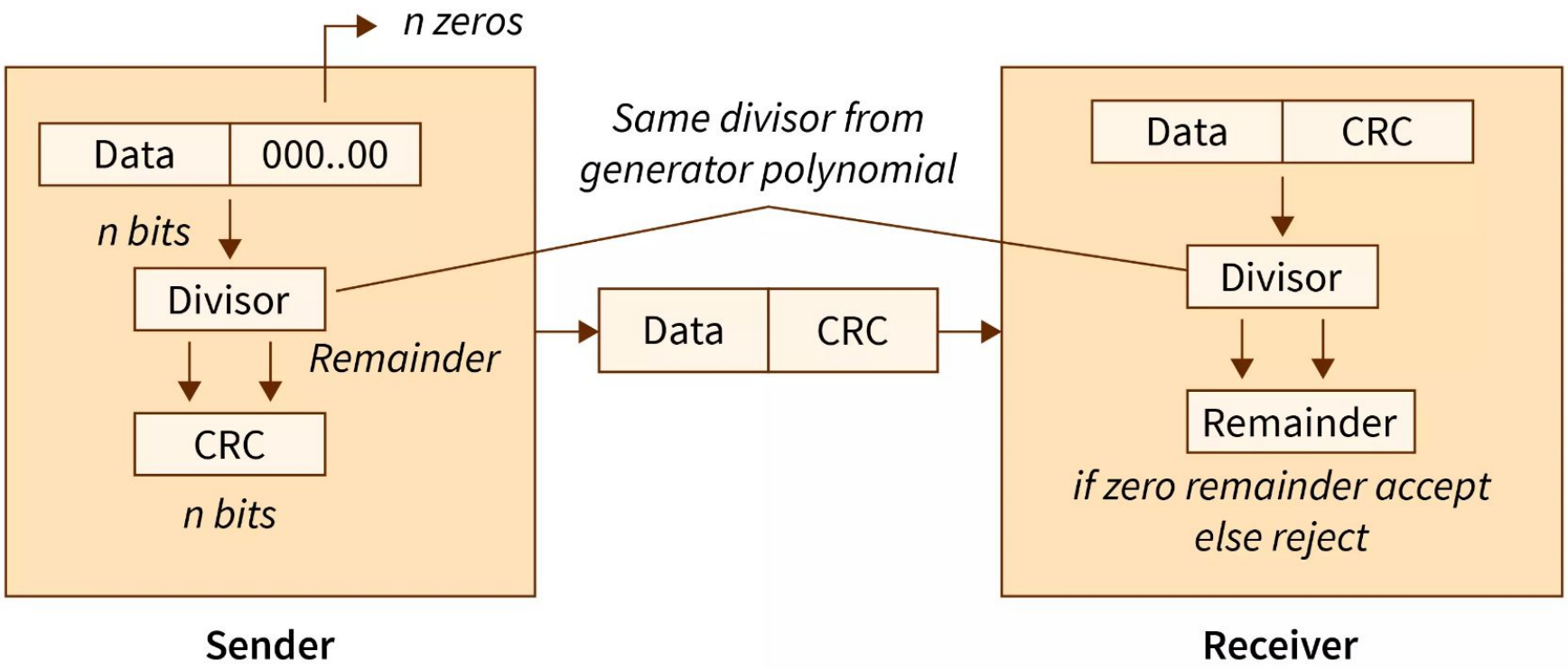
Sum = ?

CheckSum = ?

3. Cyclic Redundancy Checks (CRCs) or polynomial code:

- 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.
- **The sender** divides the bits that are being transferred and calculates the remainder.
- **The sender inserts the remainder at the end of the original bits before sending the actual bits.**
- **A codeword** is made up of the **actual data bits plus the remainder**. The transmitter sends data bits in the form of codewords.
- The receiver, on the other hand, divides the codewords using the same CRC divisor.
- **If the remainder** consists entirely of **zeros**, the data bits are **validated**; otherwise, it is assumed that some data corruption happened during transmission.

Cyclic Redundancy Checks



Eg:

A bit stream 1010000 is transmitted using the standard CRC method. The generator polynomial is x^3+1. What is the actual bit string transmitted? Conclude whether the receiver receive original message or error message.

original message
1 0 1 0 0 0 0

@ means X-OR

Generator polynomial
x^3+1
$1.x^3+0.x^2+0.x^1+1.x^0$
CRC generator
1 0 0 1 4-bit

If CRC generator is of n bit then append (n-1) zeros in the end of original message

Sender

```
1 0 0 1 | 1 0 1 0 0 0 0 0 0 0
        @ 1 0 0 1
        ---------
          0 0 1 1 0 0 0 0 0 0
            @ 1 0 0 1
            ---------
              0 1 0 1 0 0 0 0
                @ 1 0 0 1
                ---------
                  0 0 1 1 0 0 0
                    @ 1 0 0 1
                    ---------
                      0 1 0 1 0
                        @ 1 0 0 1
                        ---------
                          0 0 1 1
```

Message to be transmitted

```
1 0 1 0 0 0 0 0 0 0
        + 0 1 1
-----------------
1 0 1 0 0 0 0 0 1 1
```

```
1 0 0 1 | 1 0 1 0 0 0 0 0 1 1
        @ 1 0 0 1
        ---------
          0 0 1 1 0 0 0 0 1 1
            @ 1 0 0 1
            ---------
              0 1 0 1 0 0 1 1
                @ 1 0 0 1
                ---------
                  0 0 1 1 0 1 1
                    @ 1 0 0 1
                    ---------
                      0 1 0 0 1
                        @ 1 0 0 1
                        ---------
                          0 0 0 0
```

Receiver

Zero means data is accepted

Examples:

1. A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x4+x+1. What is the actual bit string transmitted?

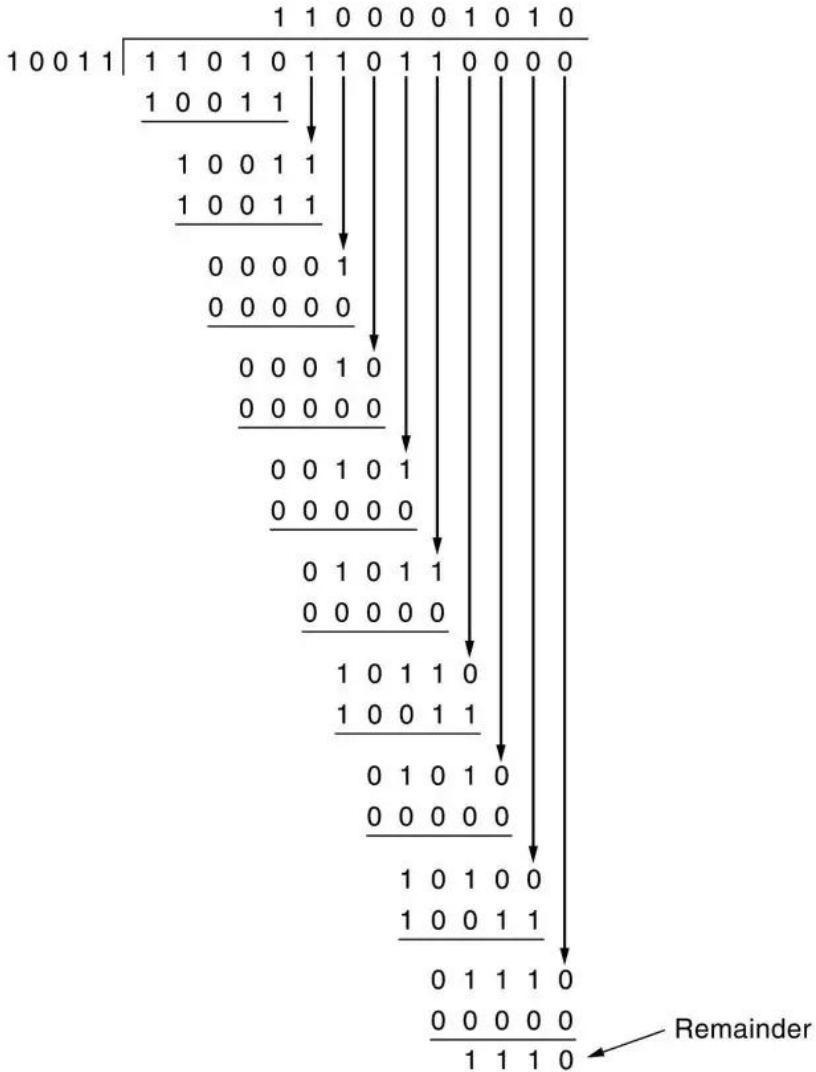
2. What is the remainder obtained by dividing x^ 7 + x^ 5 + 1 by the generator polynomial x ^3 + 1?

Examples:

1. A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is x4+x+1. What is the actual bit string transmitted?

Solution:

- The generator polynomial G(x) = x4 + x + 1 is encoded as **10011**.
- Clearly, the generator polynomial consists of **5 bits**.
- So, a string of **4 zeroes** is appended to the bit stream to be transmitted.
- The resulting bit stream is **11010110110000**.



From here, CRC = **1110**.

Now,

- The code word to be transmitted is obtained by replacing the last 4 zeroes of 11010110110000 with the CRC.
- Thus, the code word transmitted to the receiver = **11010110111110.**

Error Correction Methods

- Hamming codes
- Binary convolutional codes
- Reed-Solomon codes
- Low-Density Parity Check codes

- ★ Error Correction codes are used to detect and correct errors that occur during data transmission from the transmitter to the receiver.
- ★ There are two approaches to error correction:

  1. Backward Error Correction:

  When a backward mistake is detected, the receiver requests that the sender **retransmit** the complete data unit.

  2. Forward Error Correction:

  In this scenario, the error-correcting code is used by the receiver, which automatically corrects the mistakes.
- ★ **A single extra bit can identify but not correct the errors.**
- ★ To correct the mistakes, the specific location of the error must be known.
- ★ If we wish to compute a single-bit mistake, for example, the **error correcting algorithm** will identify which one of seven bits is incorrect. We will need to add some more redundant bits to do this.
- ★ The number of redundant bits is calculated using the following formula: $2^r >= d+r+1$
- ★ The above formula is used to compute the value of r. For example, if the value of d is 4, the least possible number that fulfils the above relation is 3.

Hamming codes

★ It is a block code that is capable of **detecting up to two simultaneous bit errors and correcting single-bit errors.**

★ **Parity bits:** A bit that is added to the original binary data to make sure the total number of 1s is even or odd (in case of even or odd parity respectively).

- **Even parity**: To check for even parity, if the total number of 1s is even, the parity bit value is 0. If the total number of occurrences of 1s is odd, the parity bit value is 1.
- **Odd Parity:** To test for odd parity, if the total number of 1s is even, the parity bit value is 1. If the total number of 1s is odd, the parity bit value is 0.

Hamming codes

Algorithm of Hamming code:

1. An information of 'd' bits are added to the redundant bits 'r' to form d+r.
2. The location of each of the (d+r) digits is assigned a decimal value.
3. The 'r' bits are placed in the positions 1,2,.....2k-1.
4. At the receiving end, the parity bits are recalculated. The decimal value of the parity bits determines the position of an error.

Relationship b/w Error position & binary number:

Error Position	Binary Number
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Let's understand the concept of Hamming code through an example:

- **Suppose the original data is 1010 which is to be sent.**

Total number of data bits 'd' = 4

Number of redundant bits r : $2^r >= d+r+1$

$2^r >= 4+r+1$

Therefore, the value of r is 3 that satisfies the above relation.

Total number of bits = d+r = 4+3 = 7;

- **Determining the position of the redundant bits**
  - The number of redundant bits is 3. The three bits are represented by **r1, r2, r4**.
  - The position of the redundant bits is calculated with corresponds to the raised **power of 2**.

  Therefore, their corresponding positions are 1, 2^1, 2^2.

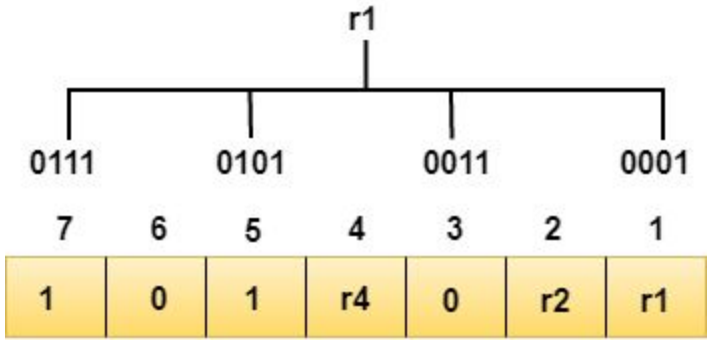
The position of r1 = 1

The position of r2 = 2

The position of r4 = 4

Determining the Parity bits

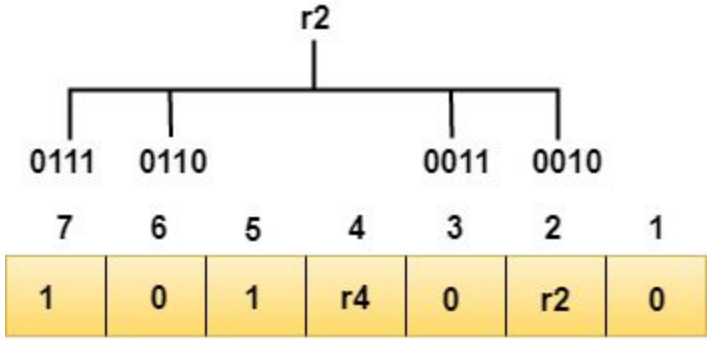
- Determining **the r1 bit**
  - **The r1 bit** is calculated by performing **a parity check on the bit positions** whose binary representation includes 1 in the first position.



- We observe from the above figure that the bit positions that includes 1 in the first position are 1, 3, 5, 7. Now, we perform the even-parity check at these bit positions.
- The total number of 1 at these bit positions corresponding to r1 is even, therefore, **the value of the r1 bit is 0.**

Determining the Parity bits

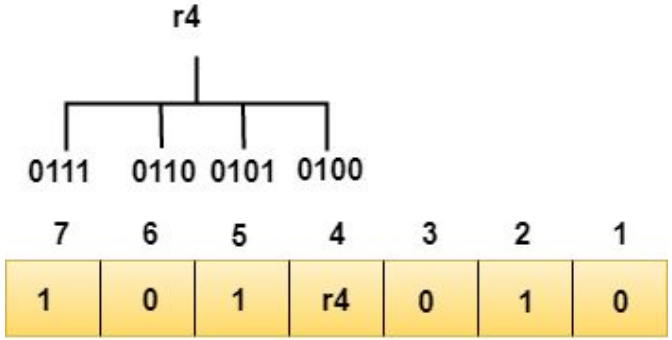
- Determining **the r2 bit**
  - **The r2 bit** is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



- We observe from the above figure that the bit positions that includes 1 in the second position are 2, 3, 6, 7. Now, we perform the even-parity check at these bit positions.
- The total number of 1 at these bit positions corresponding to r2 is odd, therefore, **the value of the r2 bit is 1.**

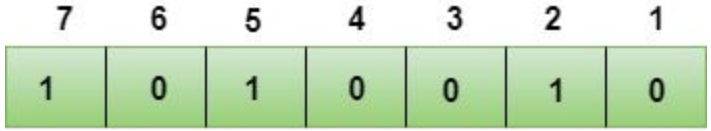
Determining **the r4 bit**

- **The r4 bit is** calculated by performing a parity check on the bit positions whose binary representation includes 1 in the third position.



- We observe from the above figure that the bit positions that includes 1 in the third position are 4, 5, 6, 7. Now, we perform the even-parity check at these bit positions.
- The total number of 1 at these bit positions corresponding to r4 is even, therefore, **the value of the r4 bit is 0.**

*** Data transferred is given below:**



Example: **If the data to be transmitted is 1011001**

Flow Control

Elementary data link protocols

Flow control

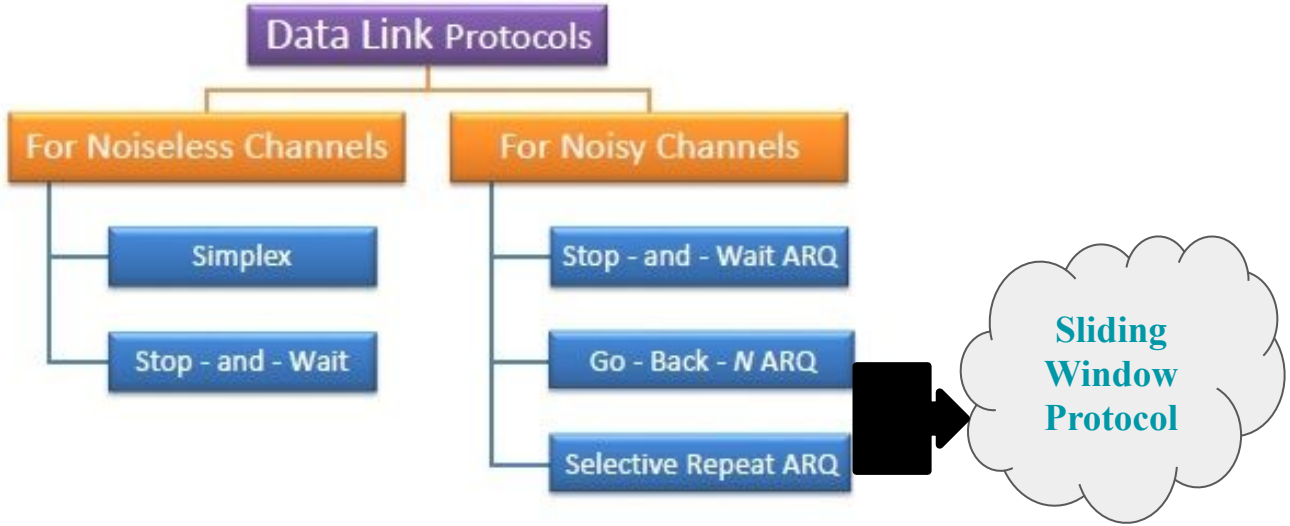
- ★ Flow control is **a Speed Matching Mechanism.**
- ★ Flow control in the data link layer is a mechanism used **to manage the rate of data transmission between the sender and the receiver.**
- ★ It ensures that the sender **does not overwhelm** the receiver with more data than it can handle, **preventing data loss or buffer overflow.**
- ★ Flow control is set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgement from the receiver.
- ★ Receiver must inform the sender before the limits are reached and request that the sender to send fewer frames or stop temporarily.
- ★ There are different **protocols** involved in Flow Control.

What is **Protocol?**

★ In computer networks, a protocol refers to **a set of rules and procedures** that govern the communication and interaction between devices or systems.

★ It defines how data is transmitted, formatted, addressed, routed, and processed in a network.

★ Protocols ensure that devices can understand and interpret the information exchanged, allowing for reliable and standardized communication.



Stop and wait Protocol

★ It is a basic protocol used for reliable data transmission between a sender and a receiver when there is **no noise or errors** in the channel.

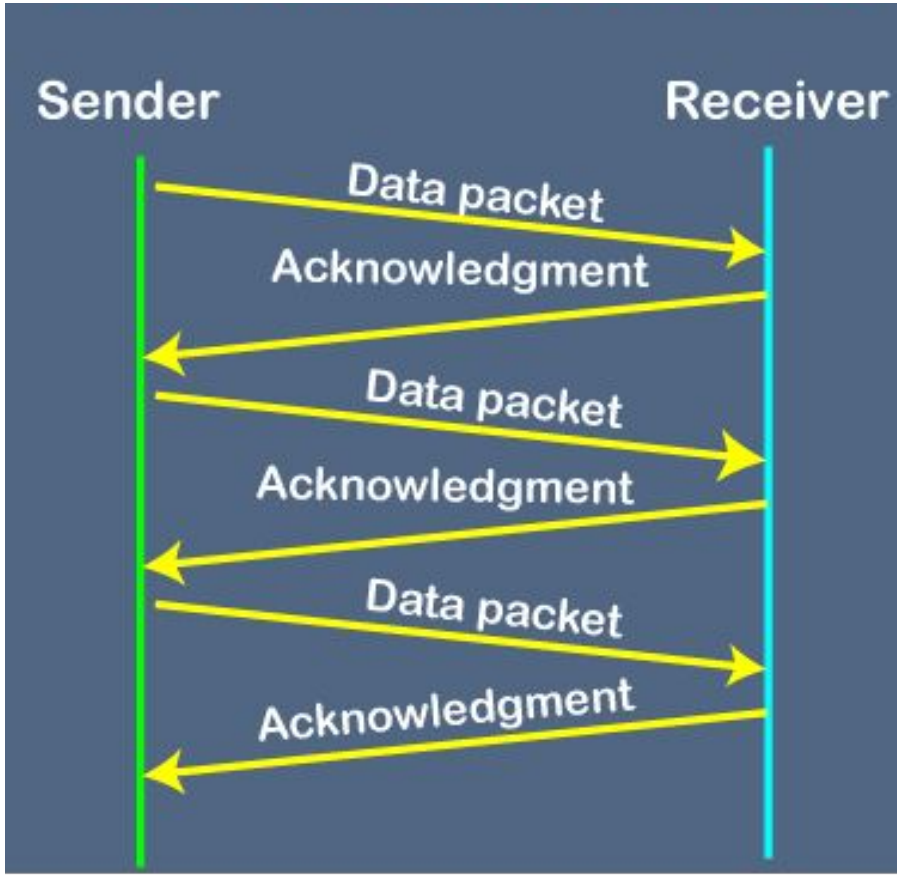
Sender:

- The sender sends a single frame to the receiver.
- After sending the frame, it waits for an acknowledgment (ACK) from the receiver.
- If the sender receives the ACK, it assumes that the frame was successfully received by the receiver and proceeds to send the next frame.
- If the sender does not receive the ACK within a specified time, it assumes that the frame was lost and retransmits the same frame.

Receiver:

- The receiver receives a frame from the sender.
- If the frame is error-free, the receiver sends an ACK back to the sender indicating successful reception.
- If the frame contains errors, the receiver discards the frame and does not send an ACK.
- The receiver waits for the sender to retransmit the frame if it does not receive a valid frame.

Flow diagram for Stop & Wait Protocol



Problems of Stop and Wait Protocol

1. Problems due to Lost Data.

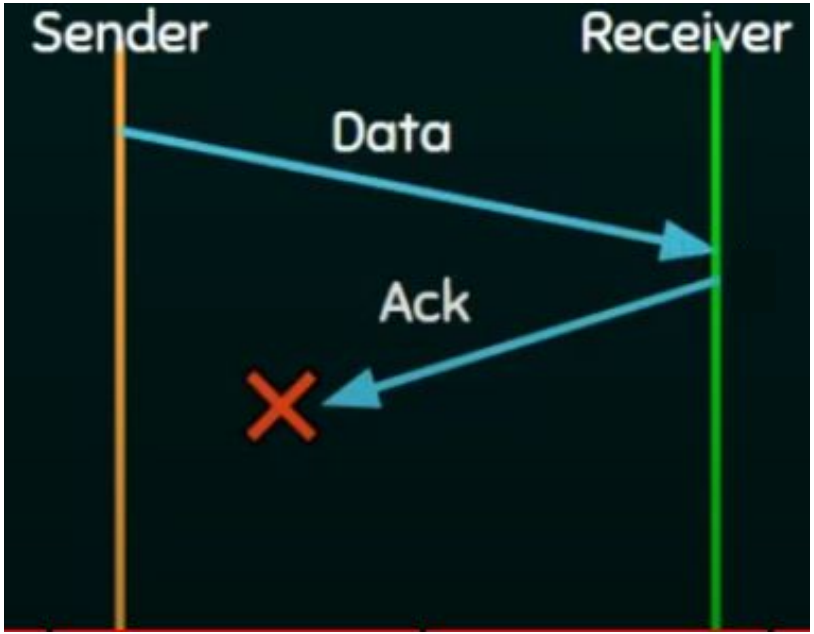
a. **Sender** waits for **Ack** for an **infinite** amount of time.

b. **Receiver** waits for **Data** for an **infinite** amount of time.

Problems of Stop and Wait Protocol

2. Problems due to Lost Ack.

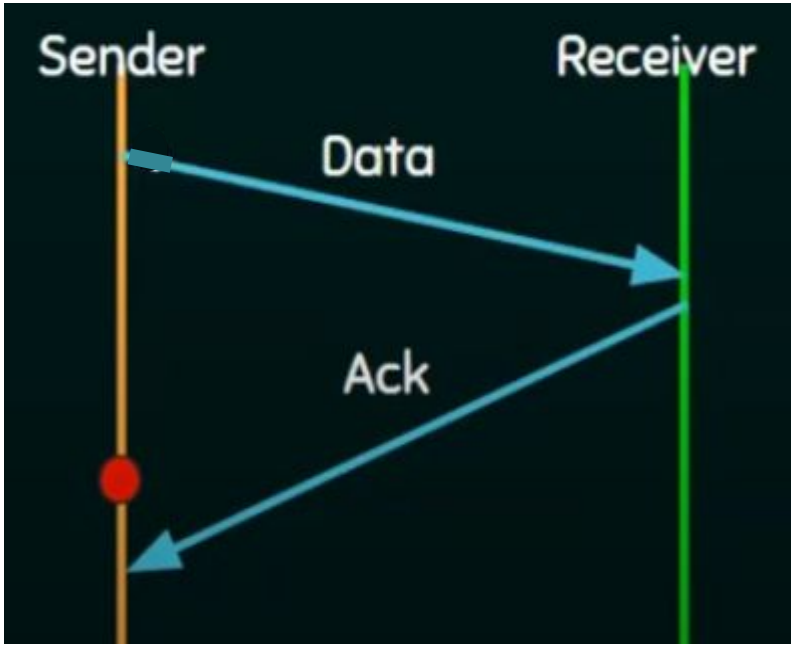
a. **Sender** waits for an **infinite** amount of time for Ack.

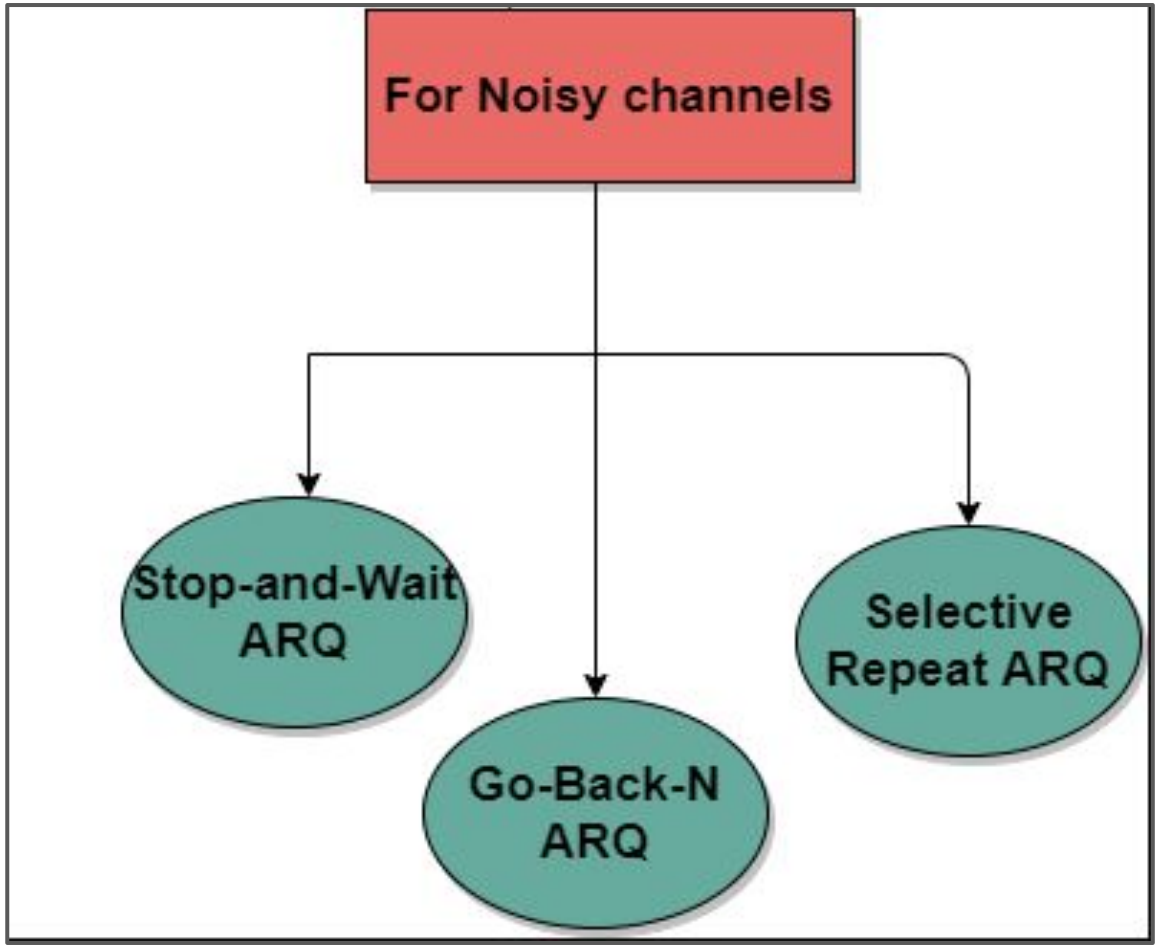


Problems of Stop and Wait Protocol

3. Problems due to Delayed Ack / Data.

a. After timeout on sender side, a delayed Ack might be wrongly considered as an Ack of some other data packet.




For Noisy channels
Stop-and-Wait ARQ
Go-Back-N ARQ
Selective Repeat ARQ

Stop and Wait ARQ [Stop-and-Wait Automatic Repeat Request]

★ Stop-and-Wait ARQ is a specific variation of the Stop-and-Wait protocol that incorporates error detection and retransmission mechanisms to ensure reliable data transmission in the presence of **errors or noise in the channel.**

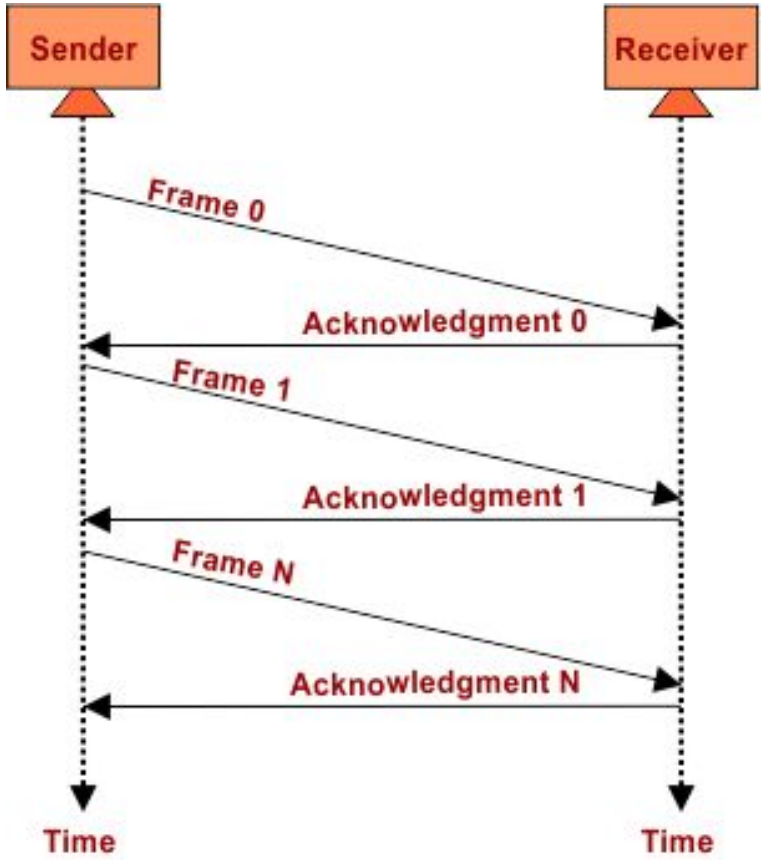
Sender:

- The sender sends a frame to the receiver. After sending the frame, it starts a timer.
- If the sender receives an acknowledgment (ACK) from the receiver within the timeout period, it assumes that the frame was successfully received and proceeds to send the next frame.
- If the timer expires before receiving the ACK, the sender assumes that the frame was lost or damaged and retransmits the same frame.

Receiver:

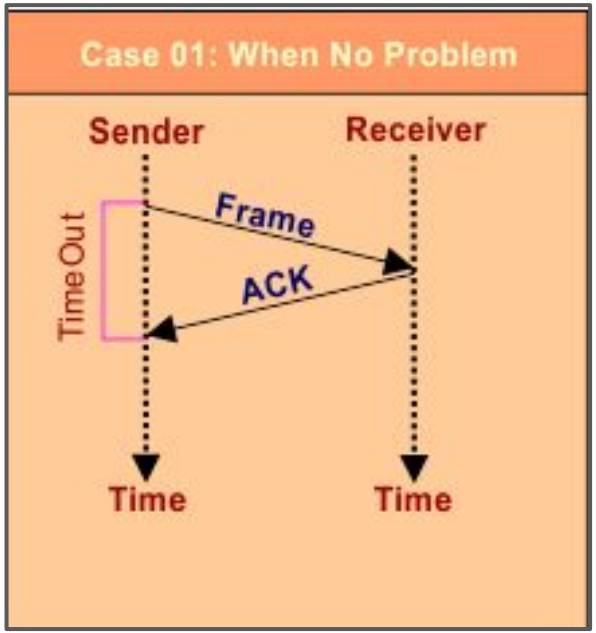
- The receiver receives a frame from the sender.
- If the frame contains errors or is damaged, the receiver discards the frame and does not send an ACK.
- If the frame is error-free and successfully received, the receiver sends an ACK back to the sender indicating successful reception.
- The receiver waits for the sender to retransmit the frame if it does not receive a valid frame.

- **Stop and wait ARQ = Stop and wait + Timeout Timer + Sequence NUmber**

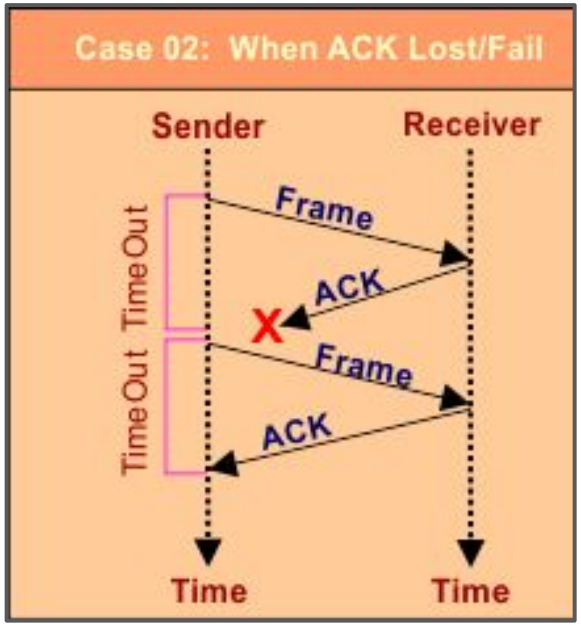


All possible scenarios of this protocol are explain under

[1] The Ack is received before the timer Expires

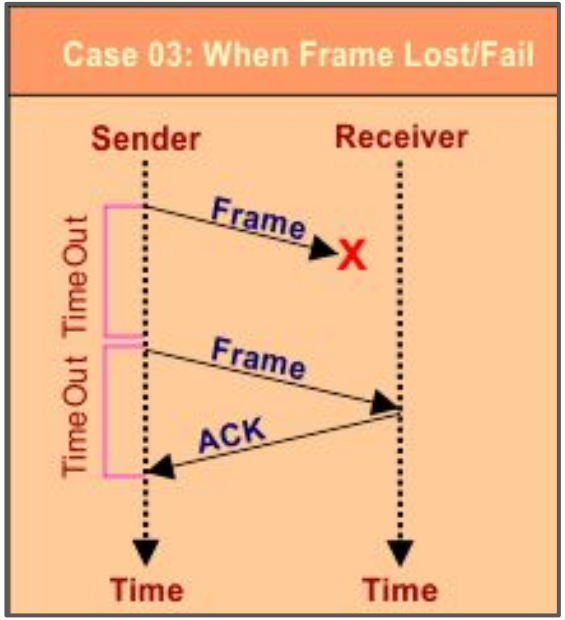


[2] The Ack is lost

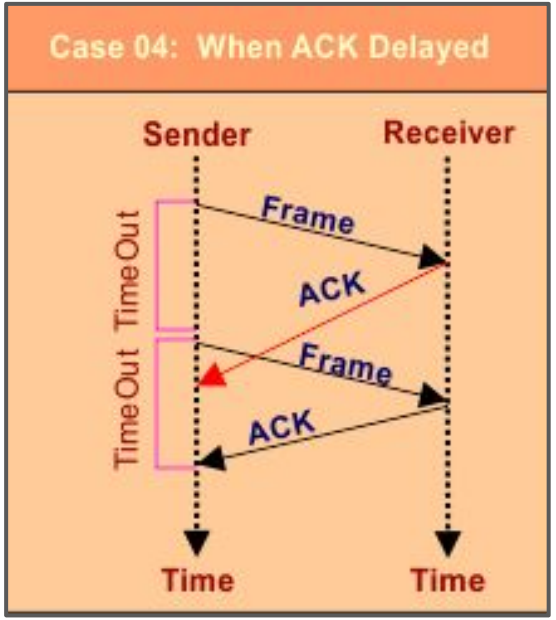


All possible scenarios of this protocol are explain under

[3] When Frame Lost



[4] The Ack is Delayed



Sliding Window Protocol

Sliding Window Protocol

★ The Sliding Window Protocol is a flow control protocol used in data communication to allow **efficient and reliable transmission of data between a sender and a receiver.**

★ It is an **extension of the Stop-and-Wait protocol** and overcomes its limitations by allowing **multiple frames** to be in transit **without waiting for individual acknowledgments**.

★ The number of frames to be sent based on the **Window Size**.

★ **Sender:**

- The sender maintains a **"window"** that represents the range of acceptable sequence numbers for the frames it can send.
- The sender can transmit **multiple frames** within the window **without waiting for acknowledgments.**
- As frames are sent, the sender **slides the window forward**.
- The sender starts **a timer** for the first frame in the window.

Sliding Window Protocol

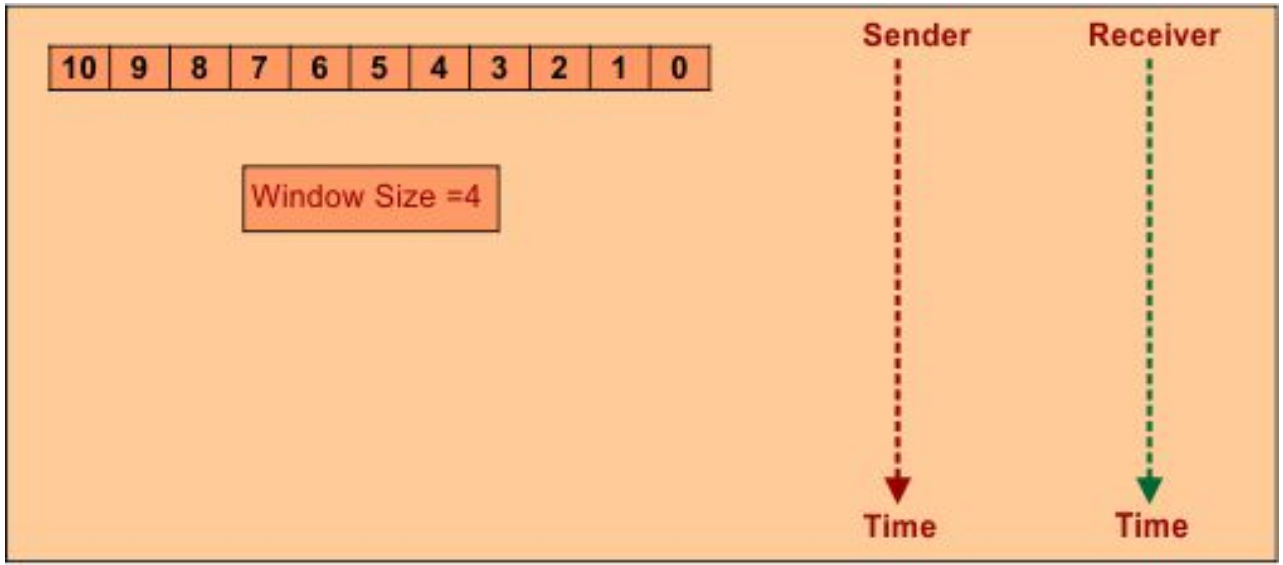
★ Receiver:

- The receiver maintains a corresponding **"window"** that represents the range of acceptable sequence numbers for the frames it expects to receive.
- The receiver acknowledges the frames it successfully receives, indicating the next expected sequence number.
- If a frame is received out of order or contains errors, the receiver discards the frame and does not send an acknowledgment.
- The receiver can accept frames within its window and slides the window forward as it receives valid frames.

Working of Sliding Window Protocol

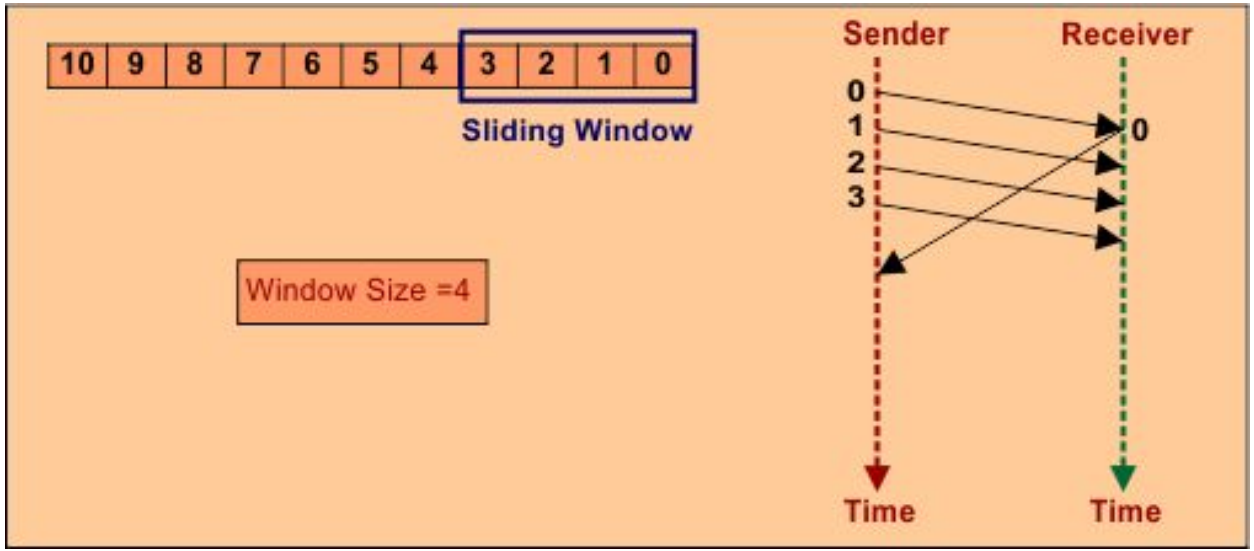
Example :
For example, suppose there are 11 frames required transmitting and sender window size is 4 then the sequence number will be 0,1,2,3, 0,1,2,3, 0, 1 and 2.

Step 1: 11 frames (0-10), window size, sender and receiver are shown below



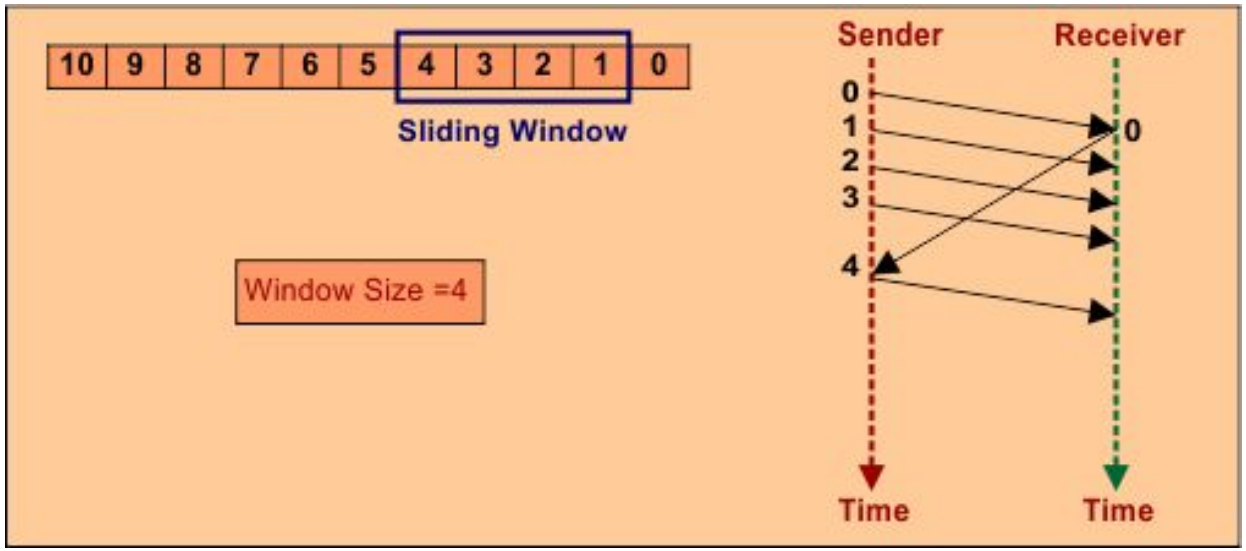
Working of Sliding Window Protocol

Step 2:After sending all frames equal to window size, Sender wait for ACK from receiver of first frame (i.e. 0).



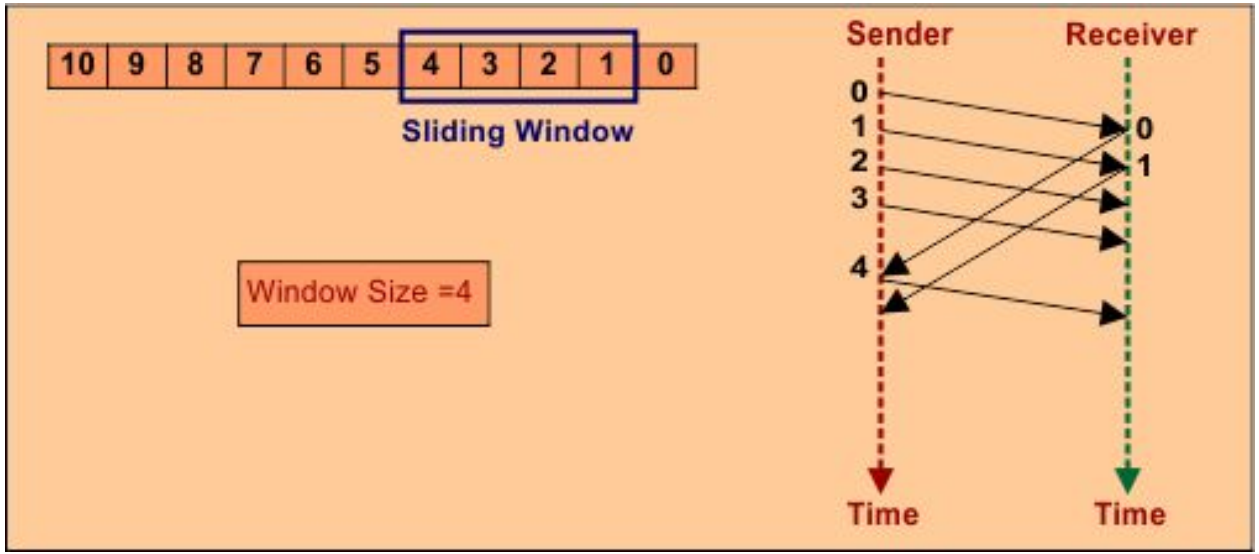
Working of Sliding Window Protocol

Step 3:After receiving the ACK the Sliding window moves one position next and transmit frame 4 and frame No.4 is transmitted.



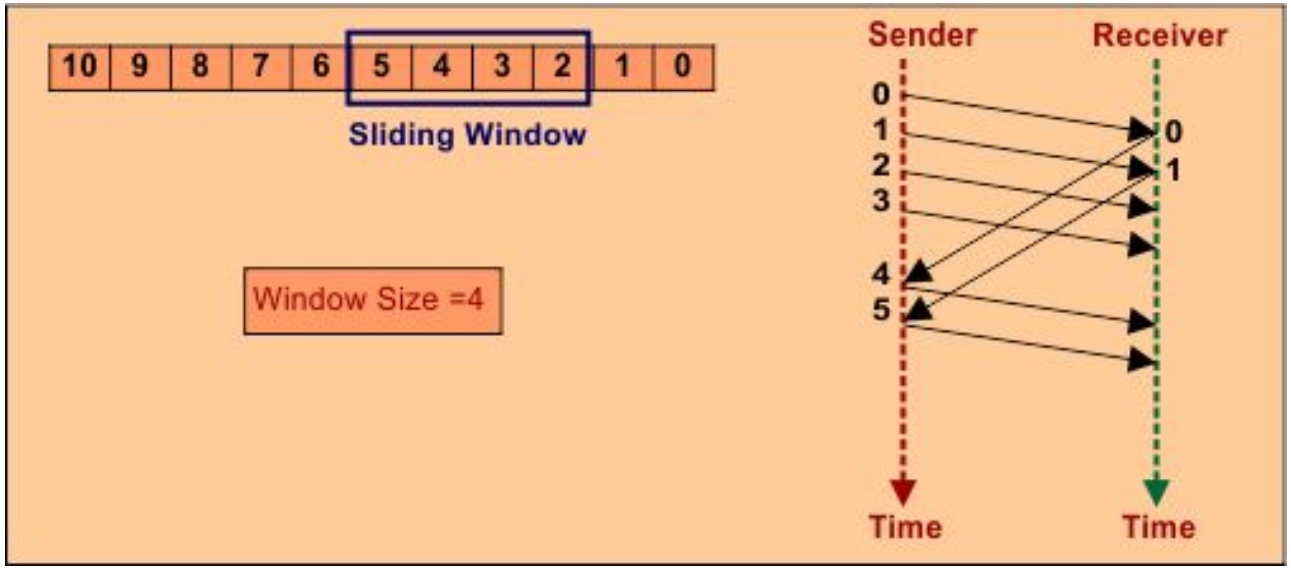
Working of Sliding Window Protocol

Step 4:After transmission of frame No.4, Sender waits for ACK from receiver of Second frame (i.e. 1).

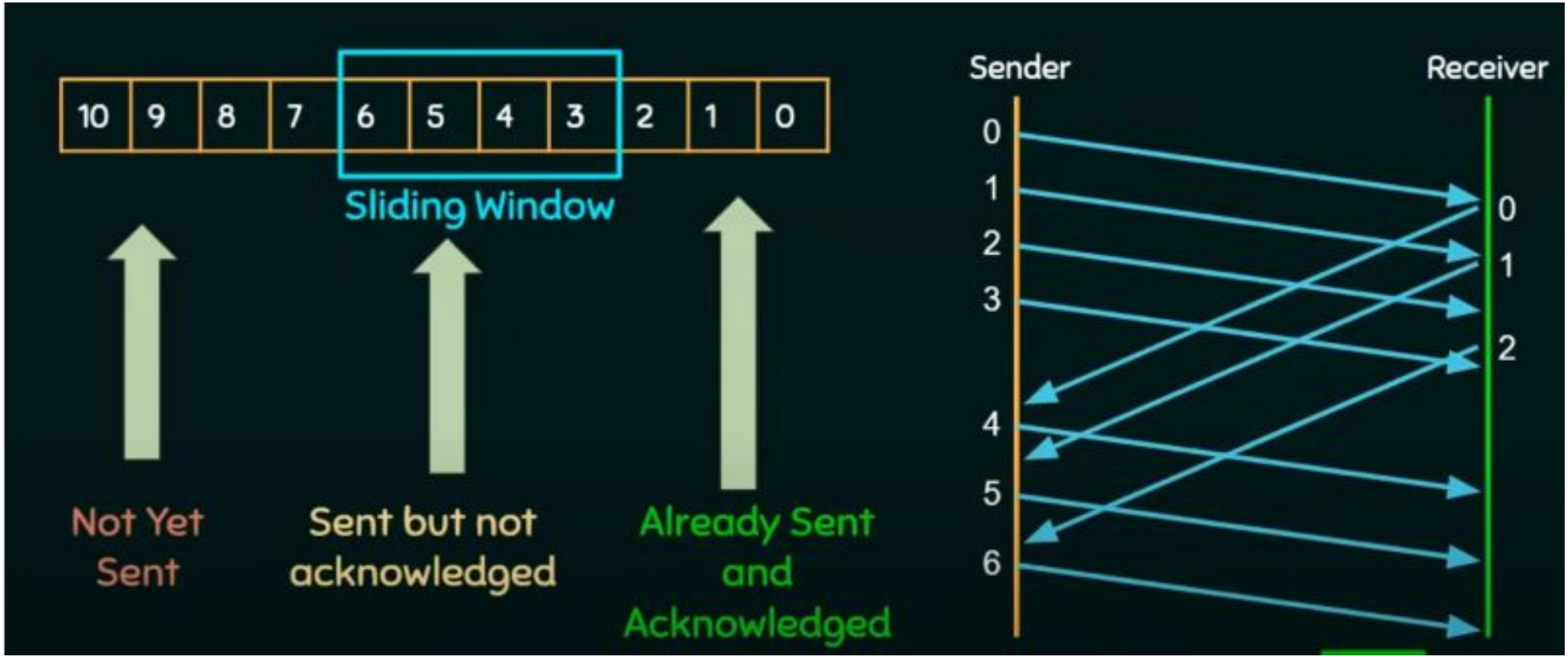


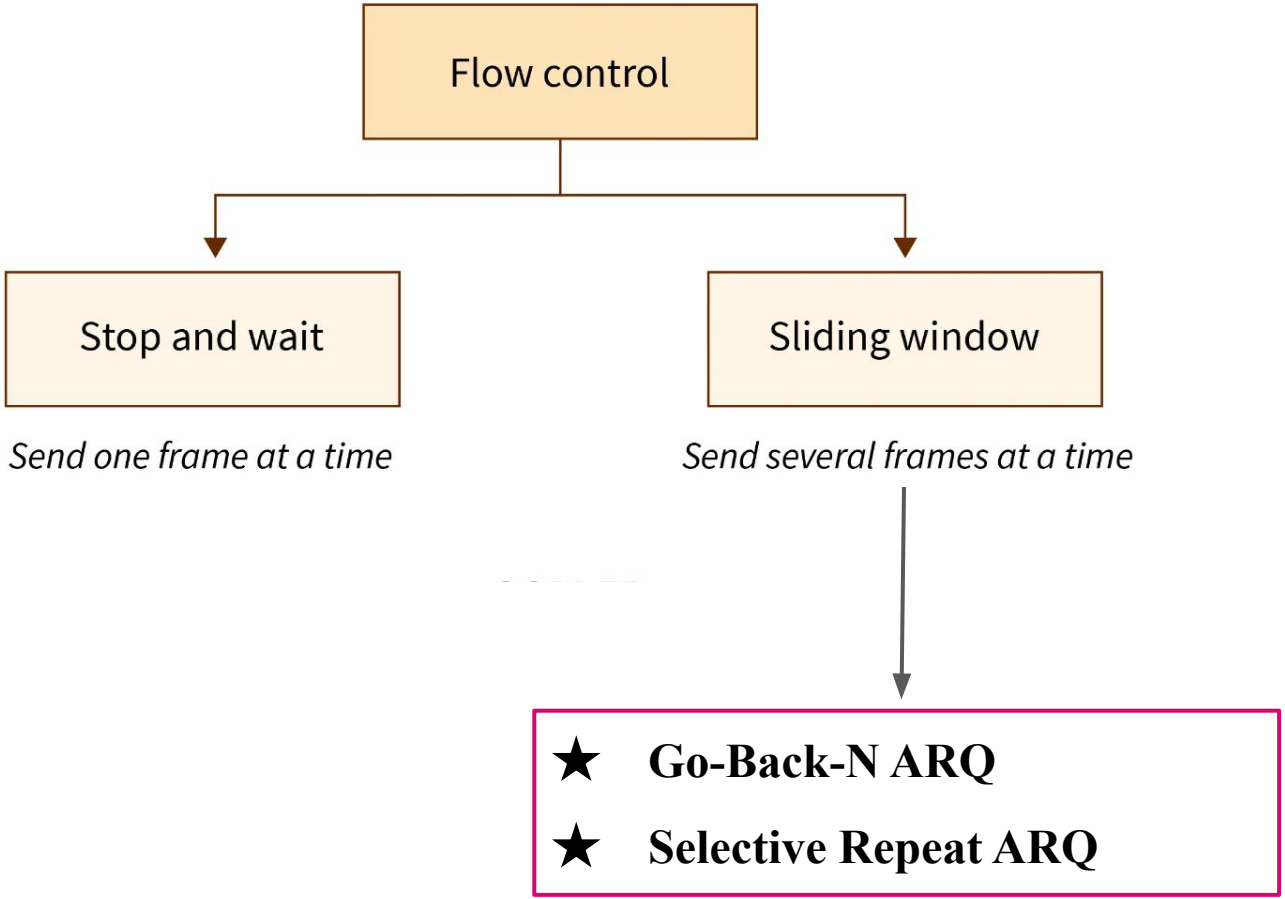
Working of Sliding Window Protocol

Step 5:After receiving ACK for Frame No.1 Sender transmit the Next frame No.5 by moving the sliding window one position next and so on.



Working of Sliding Window Protocol




Flow control
Stop and wait
Sliding window
Send one frame at a time
Send several frames at a time
★ Go-Back-N ARQ
★ Selective Repeat ARQ

Go-Back-N ARQ

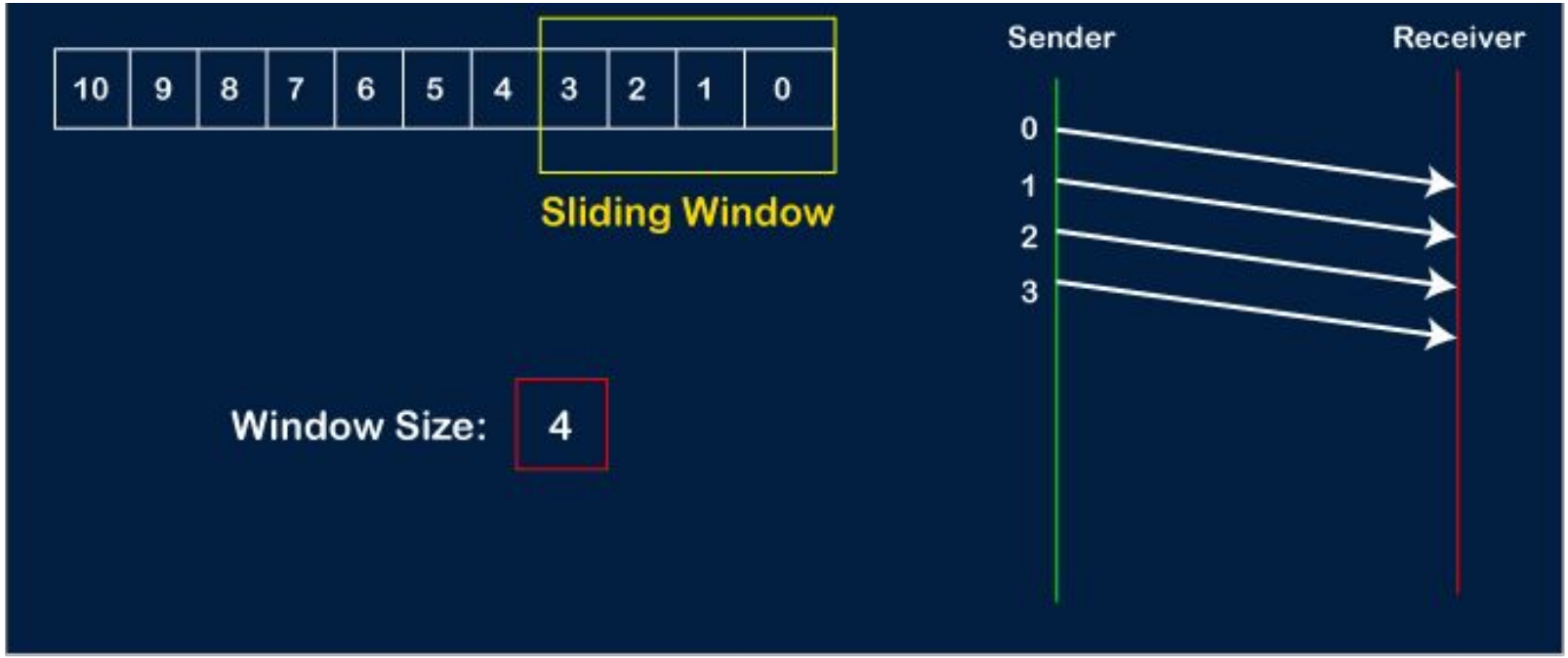
- In Go-Back-N ARQ, **N is the sender's window size**. Suppose we say that **Go-Back-3**, which means that the **three frames can be sent at a time before expecting the acknowledgment from the receiver.**
- It uses the principle of protocol **pipelining** in which the multiple frames can be sent before receiving the acknowledgment of the first frame. If we have **five frames** and the concept is **Go-Back-3**, which means that **the three frames can be sent,** i.e., **frame no 1, frame no 2, frame no 3** can be sent before expecting the acknowledgment of **frame no 1**.
- In Go-Back-N ARQ, the frames are numbered sequentially as Go-Back-N ARQ sends the multiple frames at a time that requires **the numbering approach** to distinguish the frame from another frame, and these numbers are known as **the sequential numbers.**

- The number of frames that can be sent at a time totally depends on the size of the sender's window.
- So, we can say that 'N' is the number of frames that can be sent at a time before receiving the acknowledgment from the receiver.
- If the acknowledgment of a frame is not received within an agreed-upon time period, then all the frames available in the current window will be retransmitted.
    - Suppose we have sent the frame no 5, but we didn't receive the acknowledgment of frame no 5, and the current window is holding three frames, then these three frames will be retransmitted.

Working of Go-Back-N ARQ

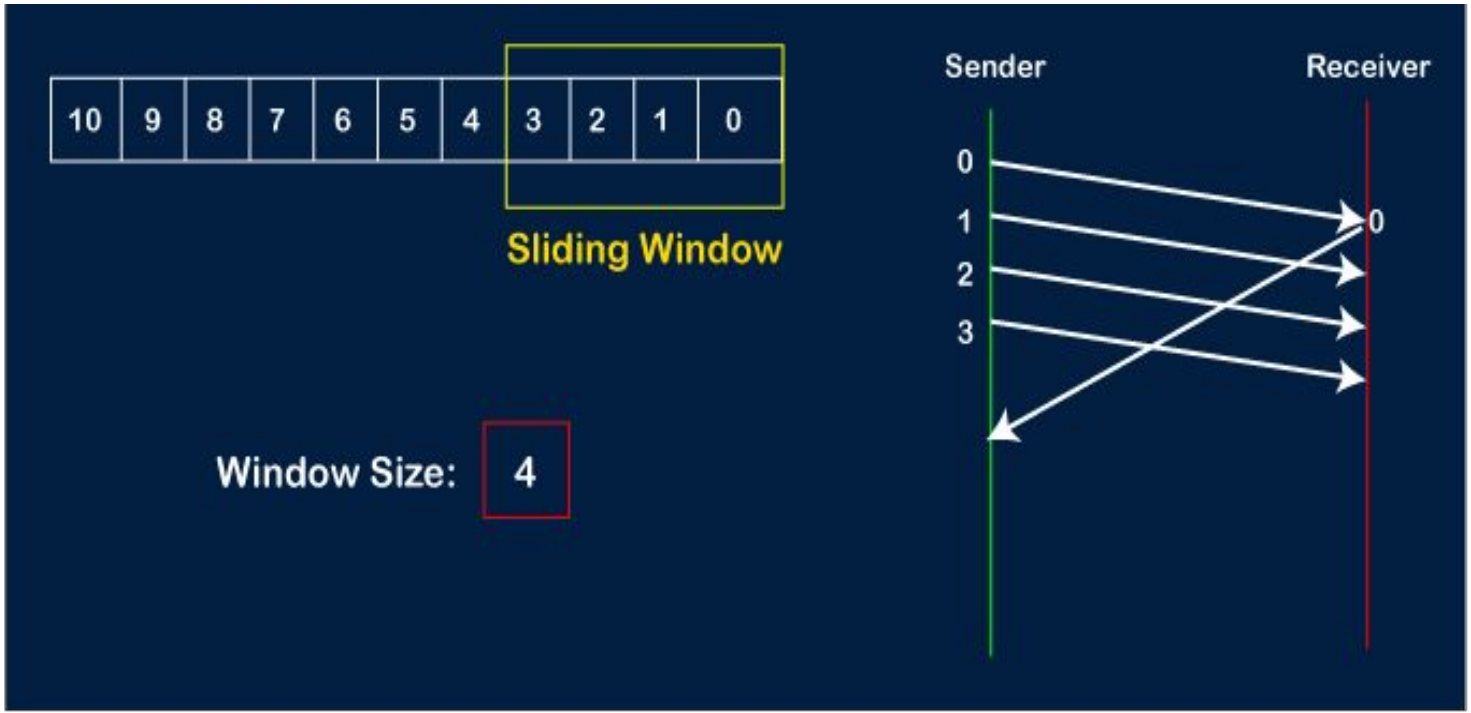
Example : For example, suppose there are 11 frames required transmitting and sender window size is 4

Step 1: Step 1: Firstly, the sender will send the first four frames to the receiver, i.e., 0,1,2,3, and now the sender is expected to receive the acknowledgment of the 0th frame.



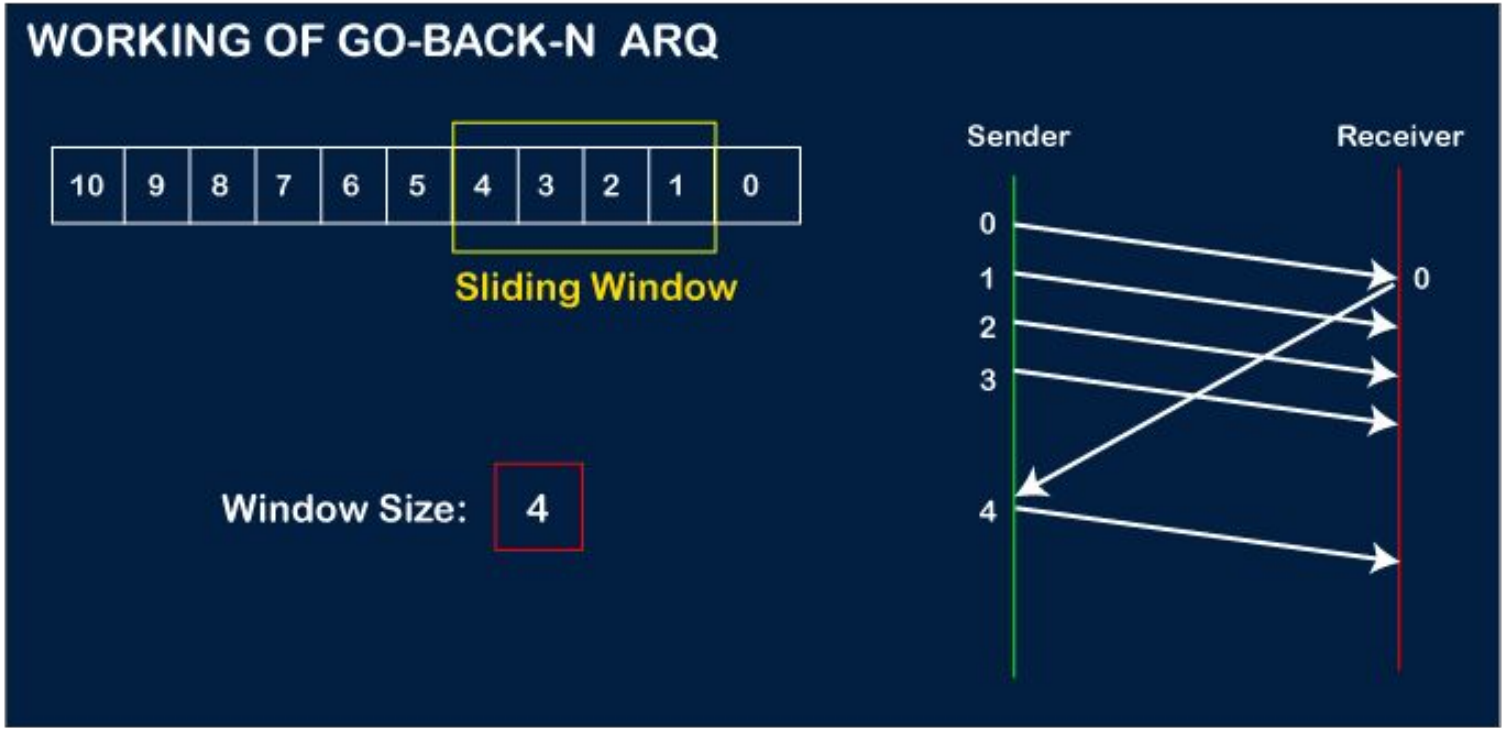
Working of Go-Back-N ARQ

Step 2: Let's assume that the receiver has sent the acknowledgment for the 0 frame, and the receiver has successfully received it.



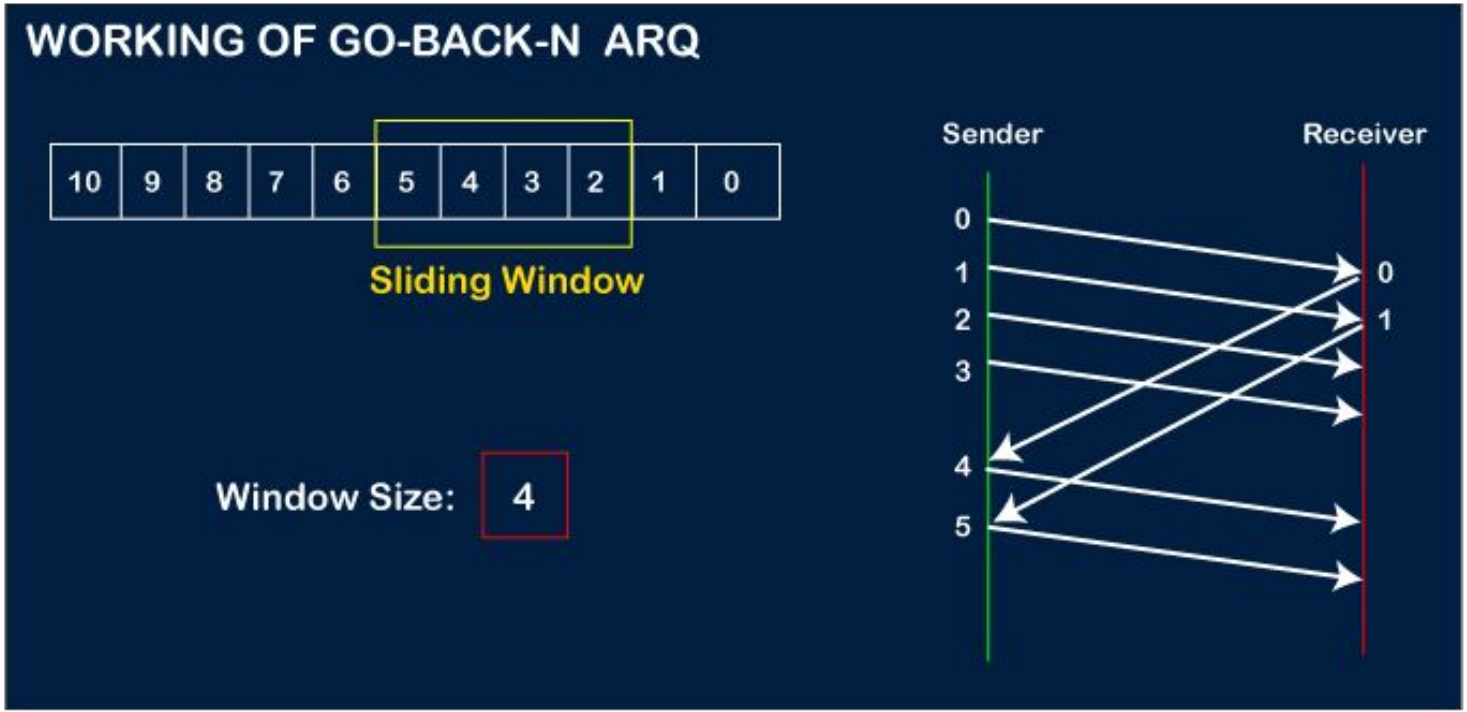
Working of Go-Back-N ARQ

Step 3: The sender will then send the next frame, i.e., 4, and the window slides containing four frames (1,2,3,4).

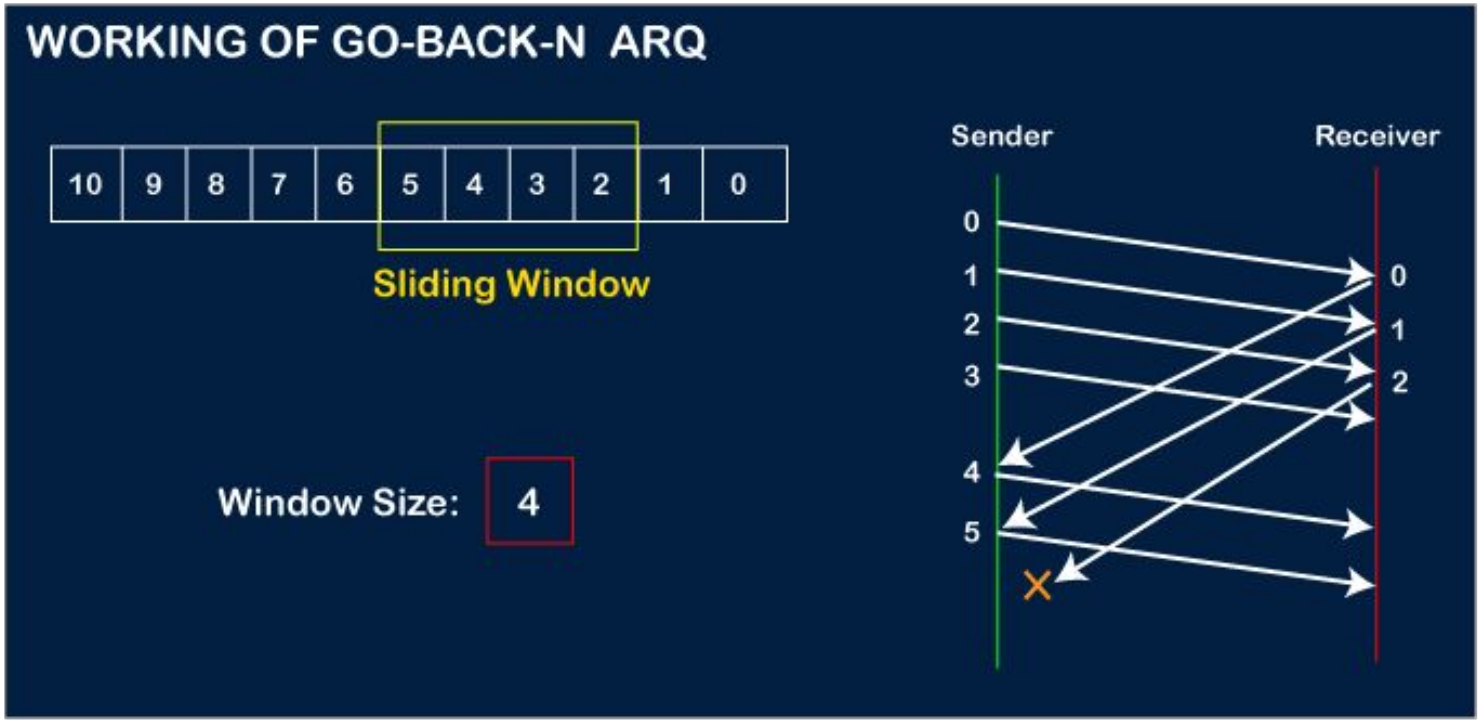


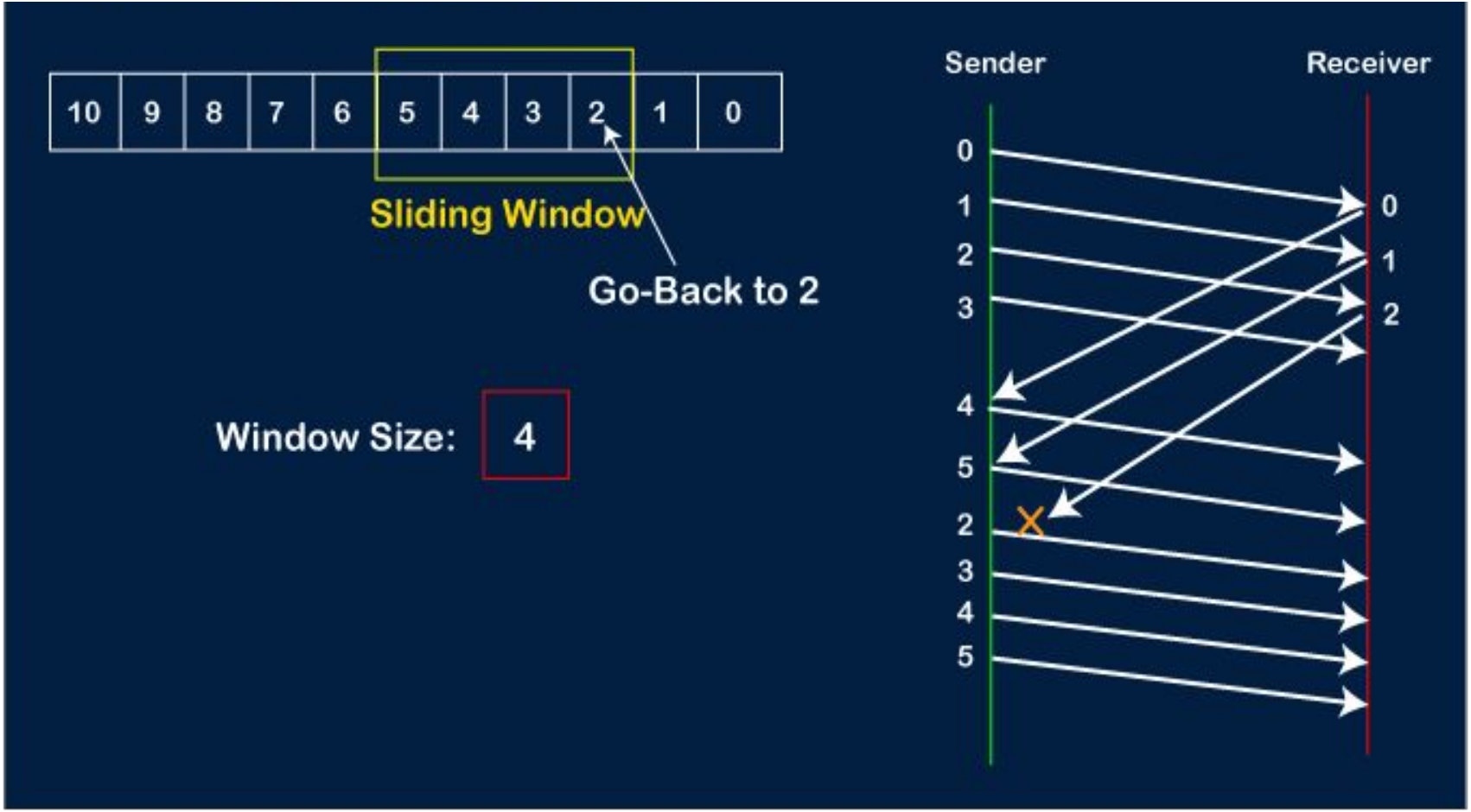
Working of Go-Back-N ARQ

Step 4: The receiver will then send the acknowledgment for the frame no 1. After receiving the acknowledgment, the sender will send the next frame, i.e., frame no 5, and the window will slide having four frames (2,3,4,5).



Step 5: Now, let's assume that the receiver is not acknowledging the frame no 2, either the frame is lost, or the acknowledgment is lost. Instead of sending the frame no 6, the sender Go-Back to 2, which is the first frame of the current window, retransmits all the frames in the current window, i.e., 2,3,4,5.



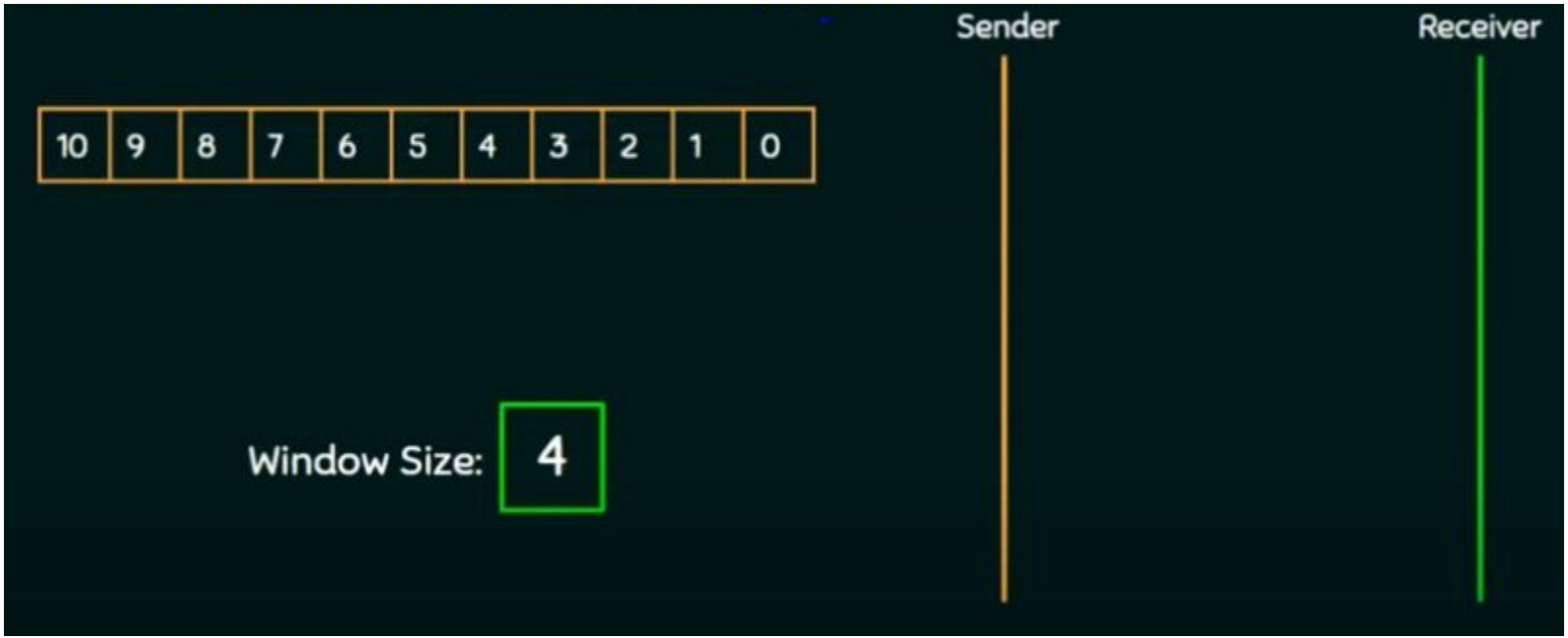

10
9
8
7
6
5
4
3
2
1
0
Sliding Window
Go-Back to 2
Window Size:
4
Sender
Receiver
0
1
2
3
4
5
2
3
4
5
0
1
2
X

Selective Repeat ARQ

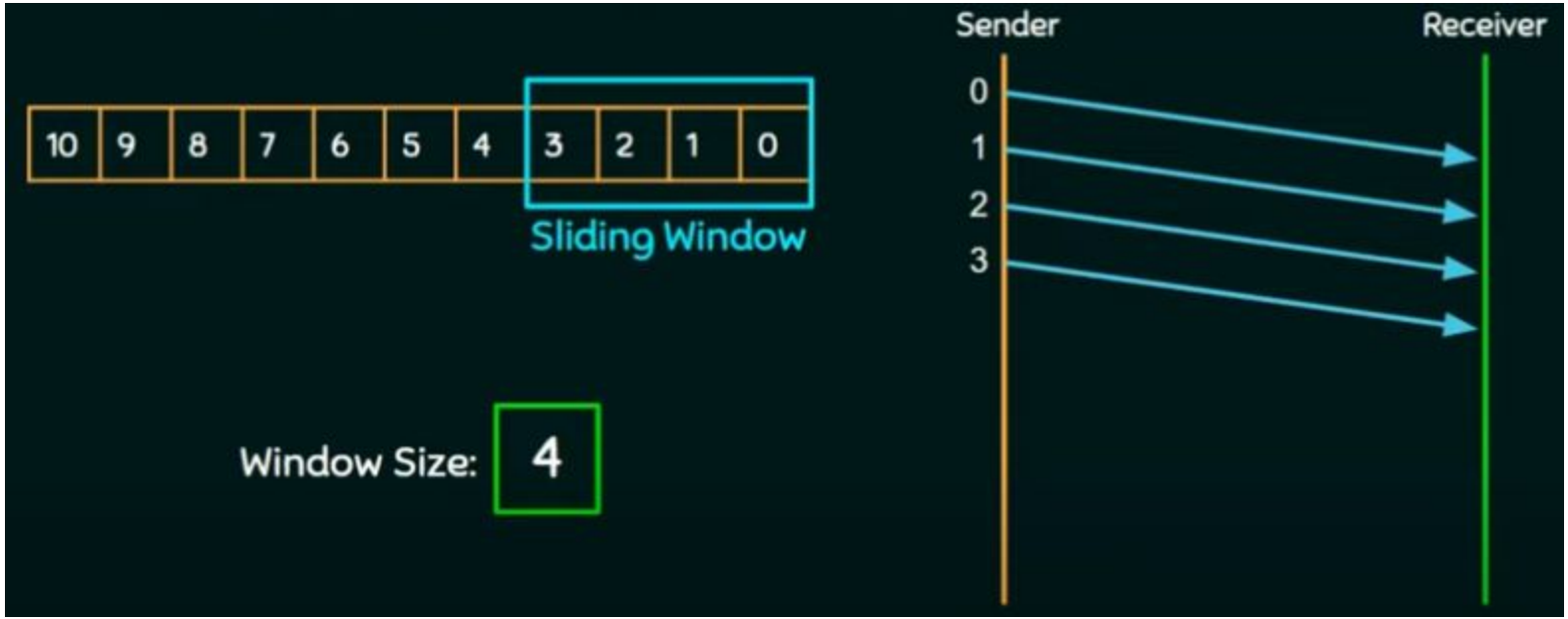
- In selective repeat, both the sender and receiver maintain a window. The sender can transmit multiple packets within the window, and the receiver acknowledges each correctly received packet.
- **If the sender does not receive an acknowledgment for a specific packet within a timeout period, it retransmits only that packet, instead of retransmitting the entire window.**
- Selective repeat reduces retransmissions and improves efficiency by eliminating unnecessary retransmissions of already received packets.

Working of **Selective Repeat ARQ**

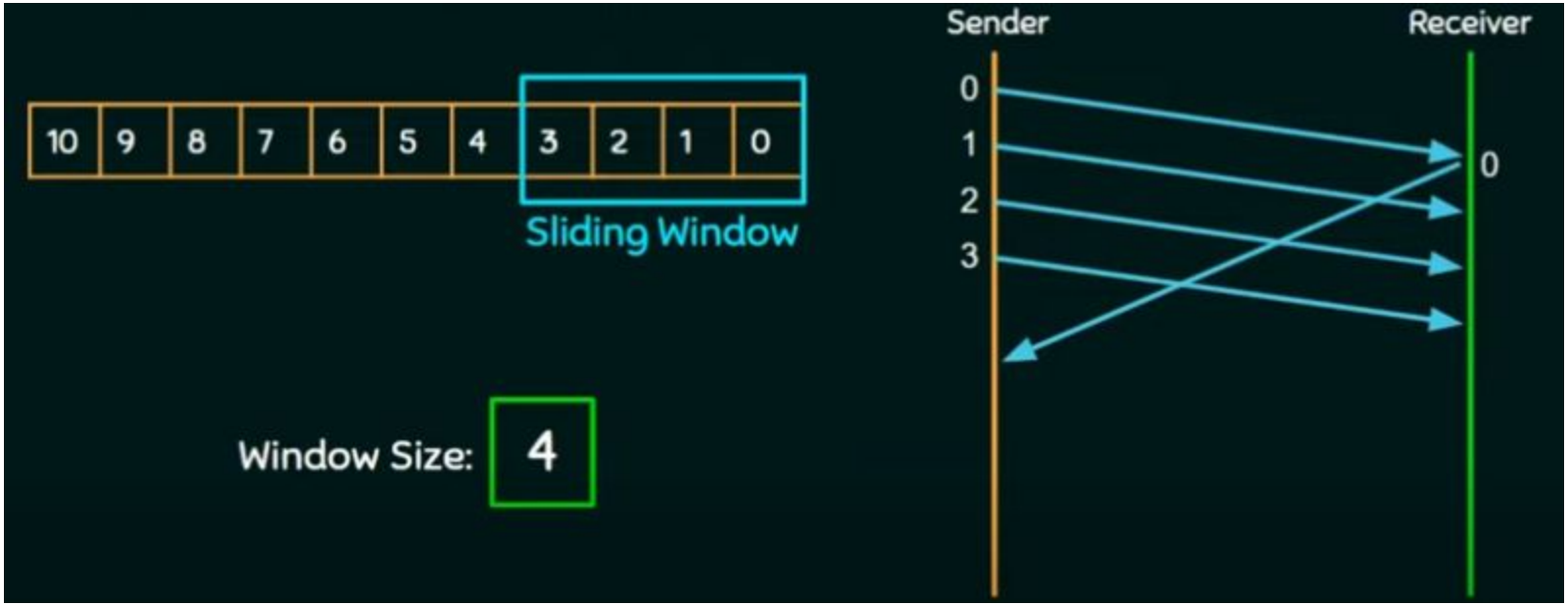
Example : **Suppose there are 11 frames required transmitting and sender window size is 4**



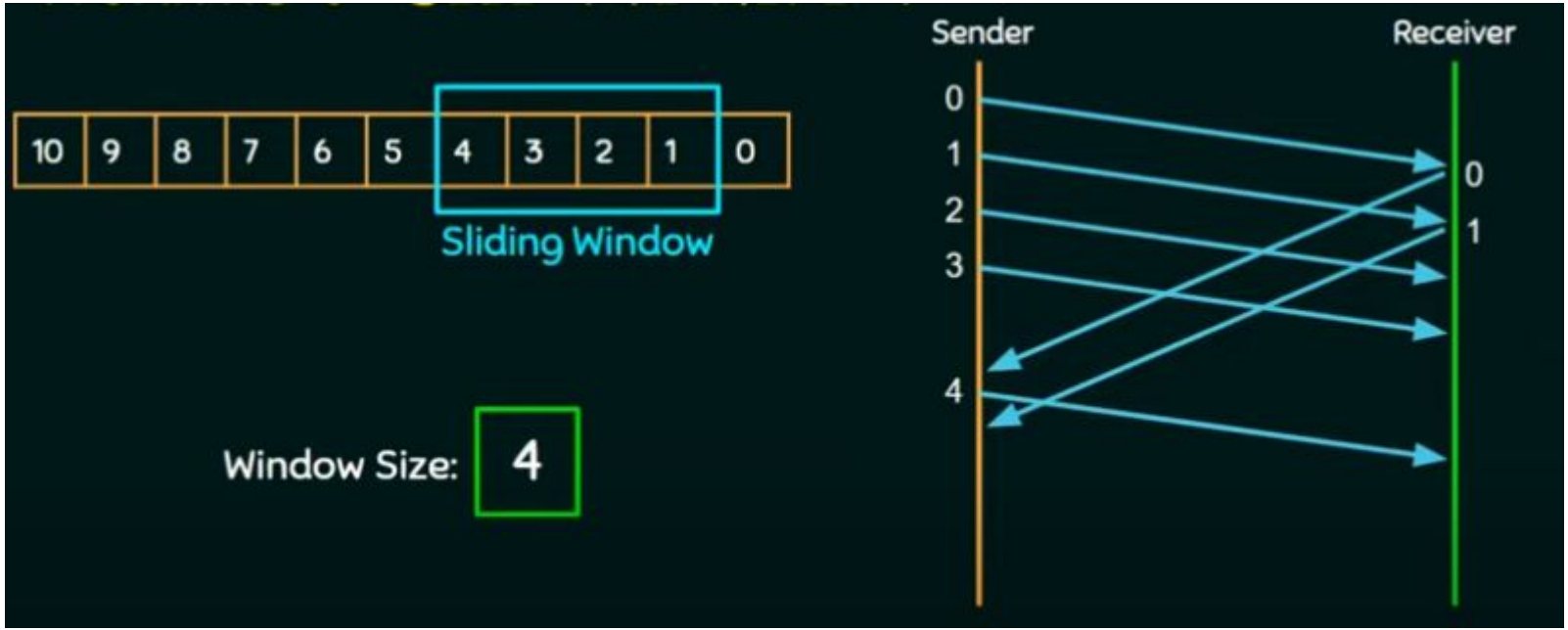
Step 1: Step 1: Firstly, the sender will send the first four frames to the receiver, i.e., 0,1,2,3, and now the sender is expected to receive the acknowledgment of the 0th frame.



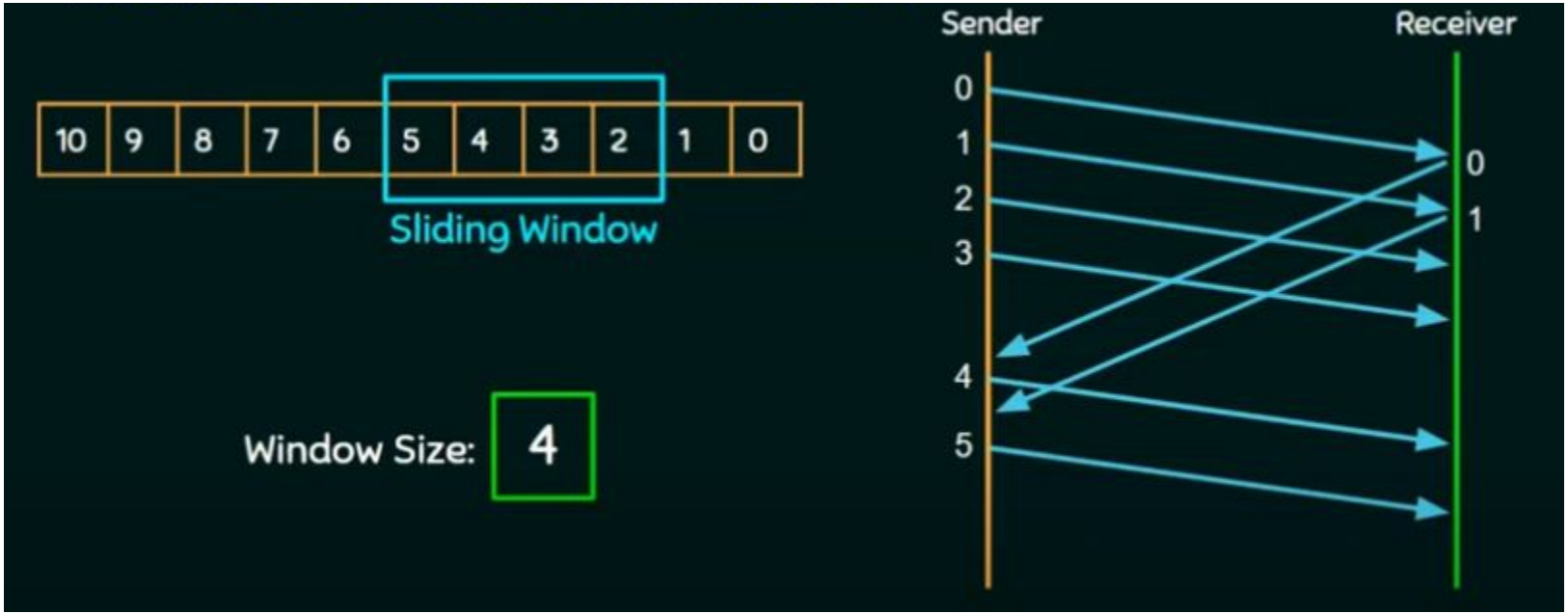
Step 2: Let's assume that the receiver has sent the acknowledgment for the 0 frame, and the receiver has successfully received it.



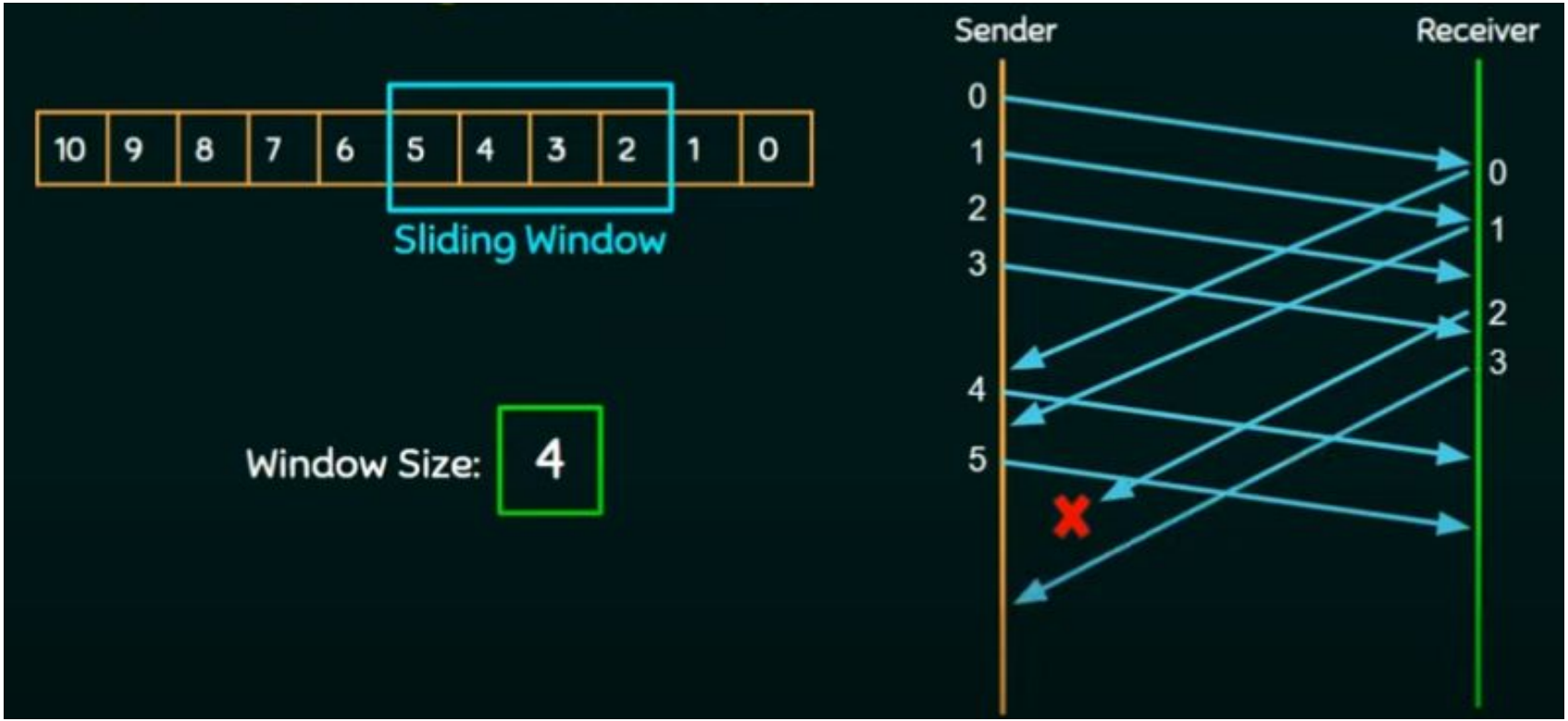
Step 3: The sender will then send the next frame, i.e., 4, and the window slides containing four frames (1,2,3,4).



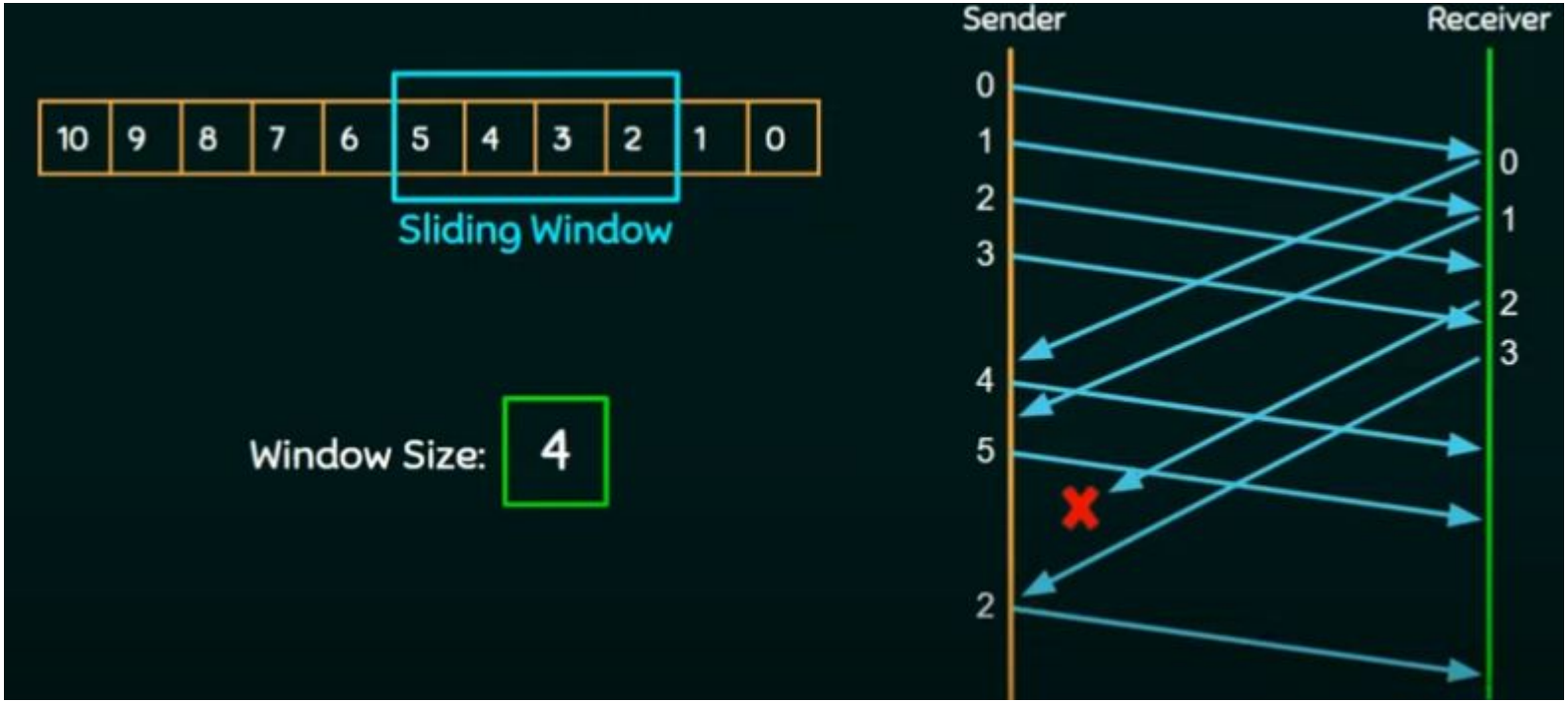
Step 4: The receiver will then send the acknowledgment for the frame no 1. After receiving the acknowledgment, the sender will send the next frame, i.e., frame no 5, and the window will slide having four frames (2,3,4,5).



Step 5: Now, let's assume that the receiver is not acknowledging the frame no 2, either the frame is lost, or the acknowledgment is lost. In Selective Repeat ARQ only the lost or error frames are retransmitted, whereas correct frames are received and buffered.


10
9
8
7
6
5
4
3
2
1
0
Sliding Window
Window Size:
4
Sender
Receiver
0
1
2
3
4
5
0
1
2
3

Sender will **retransmit frame 2 alone** and as usual other frames are transmitted.



Go-Back-N ARQ	Selective Repeat ARQ
If a frame is corrupted or lost in it,all subsequent frames have to be sent again.	In this, only the frame is sent again, which is corrupted or lost.
If it has a high error rate,it wastes a lot of bandwidth.	There is a loss of low bandwidth.
It is less complex.	It is more complex because it has to do sorting and searching as well. And it also requires more storage.
It does not require sorting.	In this, sorting is done to get the frames in the correct order.
It does not require searching.	The search operation is performed in it.
It is used more.	It is used less because it is more complex.

HDLC(High-Level Data Link Control)

Outcomes:

- Understand the bit-oriented protocol.
- Importance of HDLC.
- Know the frame format of HDLC.
- Know the types of HDLC frames.

Bit -Oriented Approach:

- It simply views frames as collection of bits.

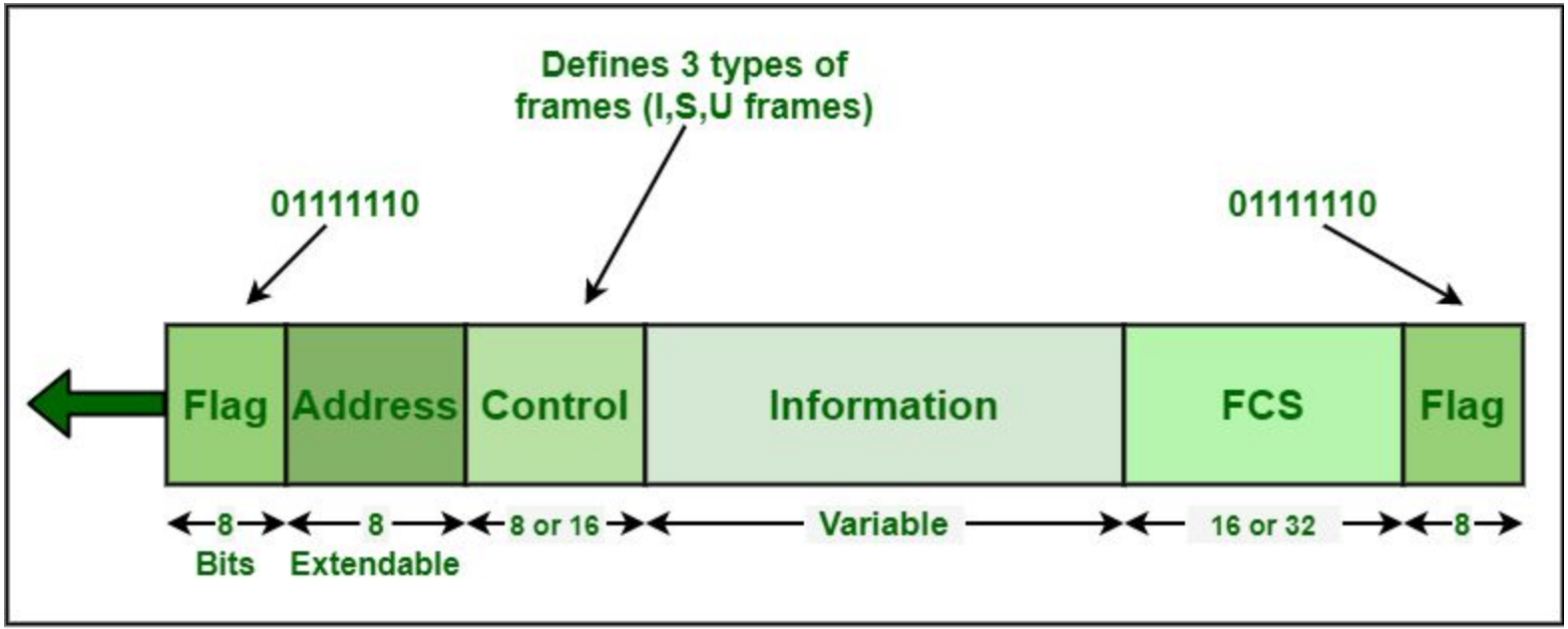
Bit -Oriented protocol:

- HDLC (High-Level Data Link Control)

What is HDLC ?

- Synchronous Data Link Control **(SDLC)** developed by **IBM** is an examples of bit-oriented protocol.
- SDLC was later standardized by **ISO** as **HDLC protocol.**
- High-Level Data Link Control (HDLC) basically provides **reliable delivery** of data frames over a network or communication link.
- HDLC provides various operations such as **framing, data transparency, error detection, and correction, and even flow control.**
- Primary stations simply transmit commands that contain address of secondary stations. The secondary station then simply transmits responses that contain its own address.

Frame Format of HDLC



Basic Frame Structure

Frame Format of HDLC

- Flag: The frame starts and ends with a flag delimiter, which is an 8-bit sequence. The most commonly used flag sequence is 01111110.
- Address: The address field is used to specify the source and destination devices. It can be 1 to 8 bits long, depending on the configuration of the HDLC variant.
- Control: The control field is used to indicate the frame type and control information. It is typically 8 bits long and contains various control bits for different purposes, such as indicating supervisory frames, information frames, or commands.
- Information: The information field carries the actual data being transmitted. Its length can vary depending on the needs of the application.
- Cyclic Redundancy Check (CRC): a CRC (Cyclic Redundancy Check) value, which is calculated based on the frame's data and control fields. The receiver uses the CRC value to check for errors in the received frame.
- Flag: The frame ends with another flag delimiter, which is the same 8-bit sequence used at the beginning of the frame.

Types of HDLC Frames

Control Field –

- HDLC generally uses this field to determine **how to control process of communication**.
- The control field is different for different types of frames in HDLC protocol.
- The types of frames can be
  - **Information frame (I-frame)**
  - **Supervisory frame (S-frame)**
  - **Un-numbered frame (U-frame).**

I-Frame	**In the control field if the first bit is 0**	**Carrying Information**
S-Frame	**In the control field if the first bit is 10**	**Used for Error Control and Flow Control**
U-Frame	**In the control field if the first bit is 11**	**Used for Link Management**

The data link layer in the internet

I. In the Internet, the data link layer is responsible for the reliable transmission of data over a physical network link.

II. It sits above the physical layer and below the network layer in the networking protocol stack. While the Internet Protocol (IP) primarily operates at the network layer, the data link layer is essential for the proper functioning of IP-based communication.

III. The data link layer in the Internet encompasses various protocols and technologies, including:

A. **Ethernet**: Ethernet is the most commonly used technology at the data link layer in local area networks (LANs). It defines the standards for wired communication, specifying the physical and data link layer protocols for transmitting data between devices connected to the same LAN.

B. **Wi-Fi (Wireless LAN)**: Wi-Fi is a wireless data link layer technology that enables devices to connect to a network wirelessly. It uses protocols such as IEEE 802.11 to establish wireless connections and handle data transmission over the air.

C. Point-to-Point Protocol (PPP): PPP is a data link layer protocol used for establishing direct point-to-point connections over various physical links, including dial-up connections and dedicated serial links. It provides authentication, error detection, and framing capabilities.

D. Asynchronous Transfer Mode (ATM): Although less commonly used in modern networks, ATM was a data link layer technology that provided high-speed transmission of fixed-size cells. It was used in wide area networks (WANs) and for certain specialized applications.

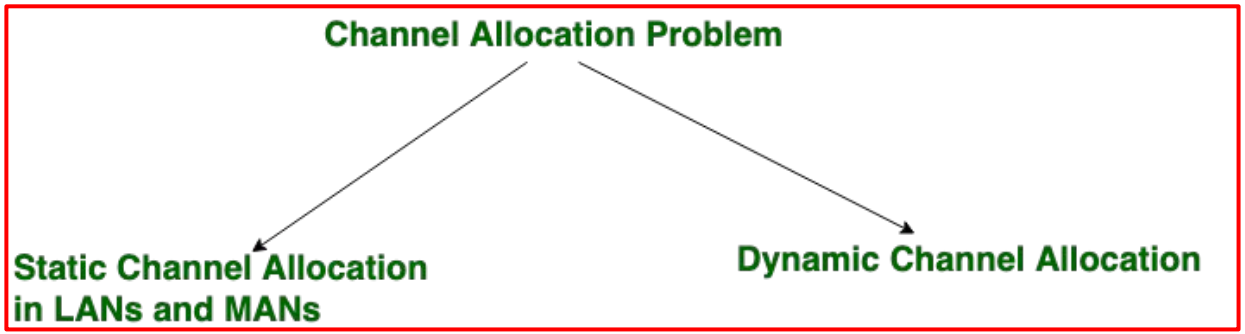
E. MPLS (Multi-Protocol Label Switching): MPLS is a protocol that operates at both the network and data link layers. It adds a label to IP packets at the data link layer, allowing routers to make forwarding decisions based on the label rather than examining the entire IP header.

Syllabus

- **The Media Access Sub Layer**
  - **Channel allocation problem**
  - **Multiple access protocols.**

Channel allocation problem

- Channel allocation is a process in which **a single channel is divided and allotted to multiple users** in order to carry user specific tasks.
- There are user's quantity may vary every time the process takes place.
- If there are N number of users and channel is divided into N equal-sized sub channels, Each user is assigned one portion.
- If the number of users are small and don't vary at times, then Frequency Division Multiplexing can be used as it is a simple and efficient channel bandwidth allocating technique.
- Solved by two schemes:



Static Channel Allocation in LANs and MANs:

- It is the classical or traditional approach of allocating a single channel among multiple competing users using Frequency Division Multiplexing (FDM).
- If there are N users, the frequency channel is divided into N equal sized portions (bandwidth), each user being assigned one portion. since each user has a private frequency band, there is no interference between users.
- It is **not** efficient to divide into **fixed number of chunks**.

$$T = 1/(U*C-L)$$

$$T(FDM) = N*T(1/U(C/N)-L/N)$$

Where:

T = Mean time delay,
C = Capacity of channel,
L = Arrival rate of frames,
1/U = bits/frame,
N = Number of sub channels,
T(FDM) = Frequency Division Multiplexing Time

2. Dynamic Channel Allocation

1.Independent Traffic

★ The model consists of N independent stations (e.g., computers, telephones), each with a program or user that generates frames for transmission.

★ The expected number of frames generated in an interval of length Δt is $\lambda \Delta t$, where λ is a constant (the arrival rate of new frames).

★ Once a frame has been generated, the station is blocked and does nothing until the frame has been successfully transmitted.

2.Single Channel.

★ A single channel is available for all communication.All stations can transmit on it and all can receive from it.

★ The stations are assumed to be equally capable, though protocols may assign them different roles (e.g., priorities).

2. Dynamic Channel Allocation

3.Observable Collisions.

★ If two frames are transmitted simultaneously, they overlap in time and the resulting signal is garbled.

★ This event is called a collision. All stations can detect that a collision has occurred.

★ A collided frame must be transmitted again later. No errors other than those generated by collisions occur.

4.Continuous or Slotted Time

★ Time may be assumed continuous, in which case frame transmission can begin at any instant. Alternatively,time may be slotted or divided into discrete intervals (called slots).

★ Frame transmissions must then begin at the start of a slot.

★ A slot may contain 0, 1, or more frames, corresponding to an idle slot, a successful transmission, or a collision, respectively.

2. Dynamic Channel Allocation

5. Carrier Sense or No Carrier Sense

★ With the carrier sense assumption, stations can tell if the channel is in use before trying to use it.

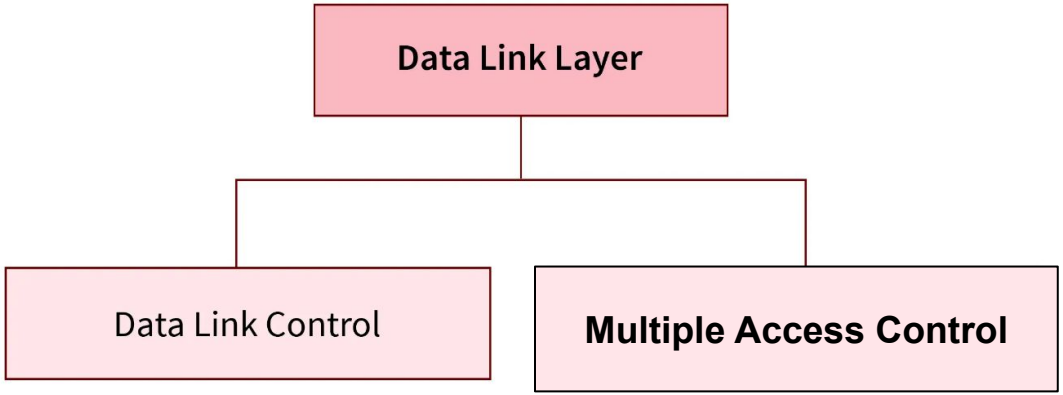
★ No station will attempt to use the channel while it is sensed as busy.

★ If there is no carrier sense, stations cannot sense the channel before trying to use it.

★ They just go ahead and transmit. Only later can they determine whether the transmission was successful.

Multiple Access Protocols

- The Data Link Layer is responsible for transmission of data between two nodes. Its main functions are-



Data Link control :

➔ Data Link Control is a sublayer of the Data Link Layer (Layer 2) in the OSI model.

➔ **The data link control protocols** offer **reliable communication** when there is **a dedicated link** between the communicating stations.

➔ **If there is a dedicated link**, the data link control protocols are self-sufficient to handle the **framing, flow and error control**.

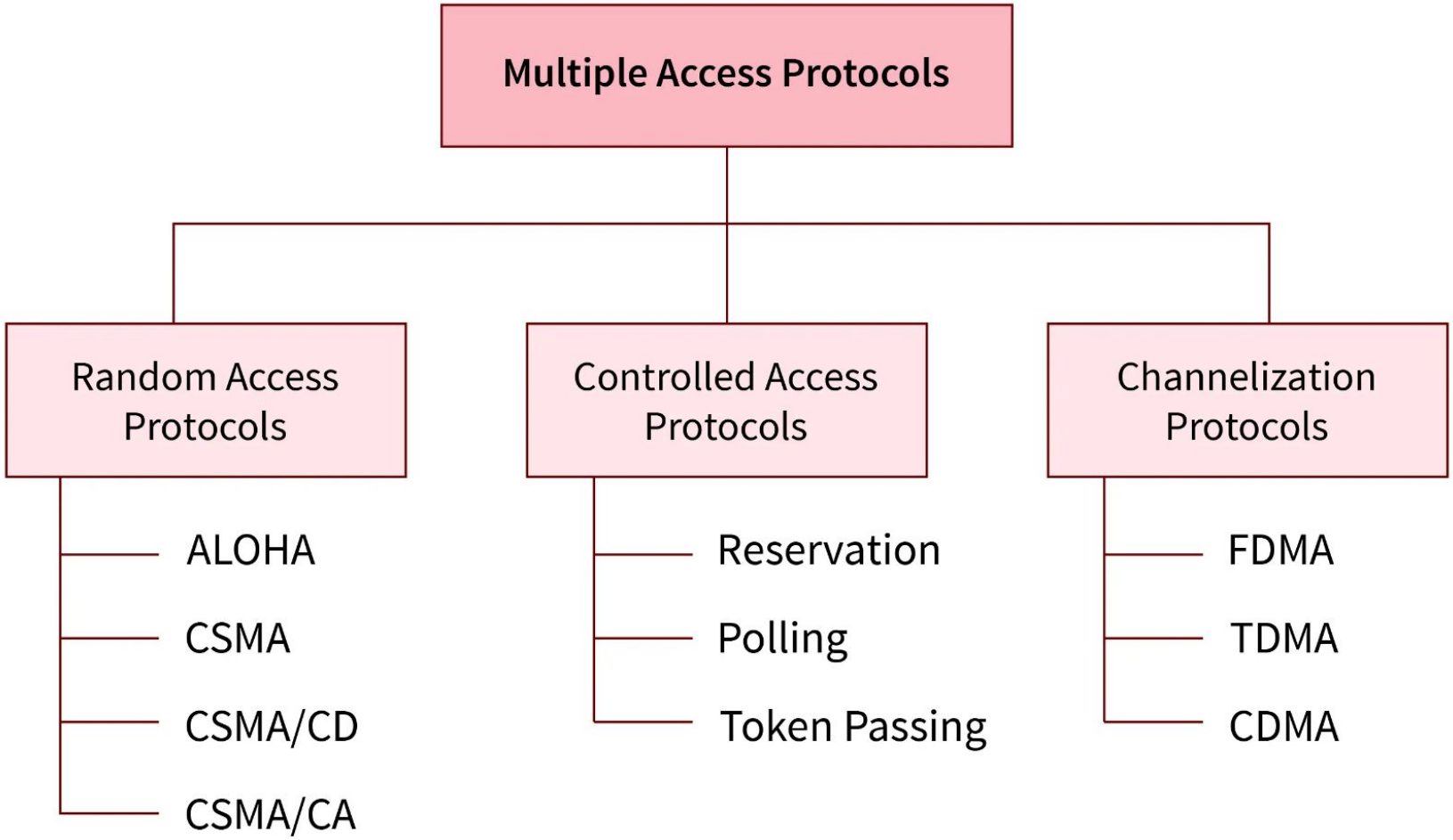
Why Multiple Access Control:

- Multiple access protocols are a set of protocols operating in the **Medium Access Control sublayer (MAC sublayer)** of the Open Systems Interconnection (OSI) model.
- If there is **a dedicated link** between the sender and the receiver then **data link control layer is sufficient.**
- However if there is **no dedicated link** present then **multiple stations** can access the channel simultaneously. Hence **multiple access protocols** are required **to decrease collision and avoid crosstalk.**

Analogy:

Example:

- [1] For example, in a classroom full of students, when a teacher asks a question and all the students (or stations) start answering simultaneously (send data at same time) then a lot of chaos is created(data overlap or data lost) then it is the job of the teacher (multiple access protocols) to manage the students and make them answer one at a time.
- [2] A Real-time example of a Medium Access Control (MAC) protocol is the Wi-Fi protocol (IEEE 802.11 standard). Wi-Fi is a widely used wireless communication technology that allows devices to connect to the internet or a local network. Within the Wi-Fi protocol, the MAC layer is responsible for controlling access to the shared wireless medium.


Multiple Access Protocols
Random Access Protocols
Controlled Access Protocols
Channelization Protocols
ALOHA
CSMA
CSMA/CD
CSMA/CA
Reservation
Polling
Token Passing
FDMA
TDMA
CDMA

1. Random Access Protocol:

★ In this, **all stations** have **same superiority** that is no station has more priority than another station.

★ Any station can send data depending on medium's state(idle or busy).

★ It has two features:

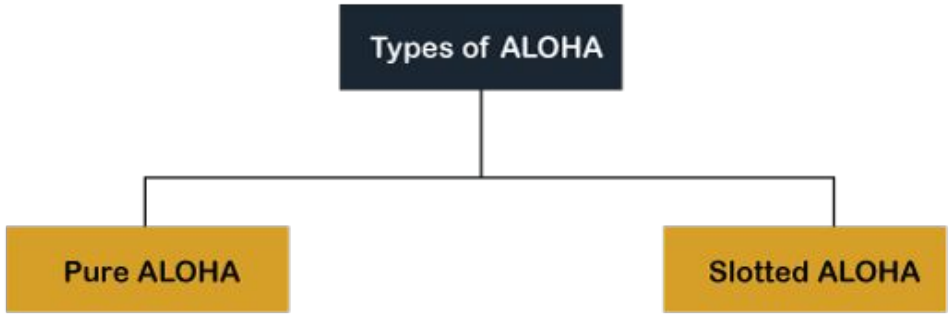
- There is **no fixed time** for sending data
- There is **no fixed sequence of stations** sending data

The Random access protocols are further subdivided as:

1. **ALOHA**
2. **Carrier Sense Multiple Access (CSMA)**
3. **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**
4. **Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)**

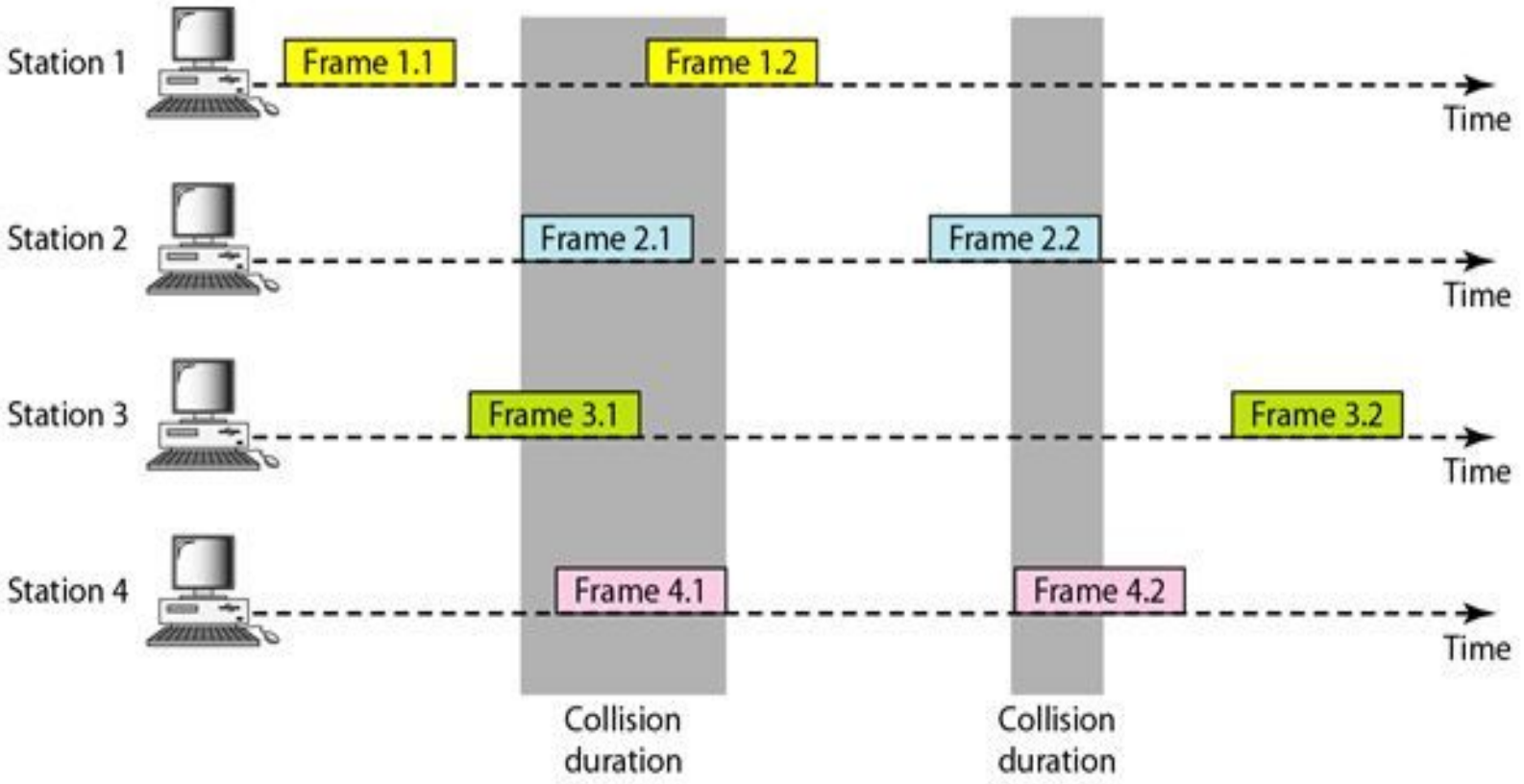
ALOHA

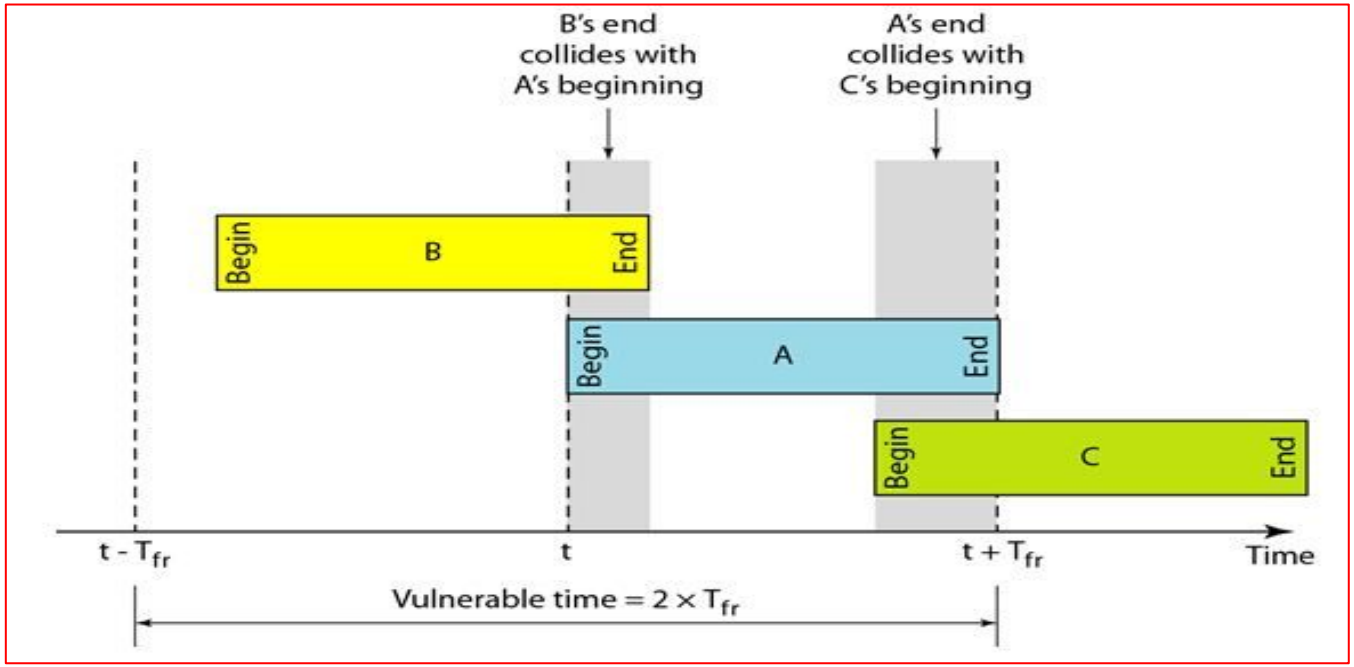
- ALOHA is an early **random access protocol** used in computer networks.
- It was first used in the ALOHAnet network at the University of Hawaii in the 1970s.
- The basic operation of the ALOHA protocol is as follows:
  - *Devices can transmit data whenever they have a message to send.*
  - *If two or more devices transmit simultaneously, their messages will **collide and be corrupted.***
  - *Devices that detect a collision will **wait for a random amount of time** before trying to transmit again.*



PURE ALOHA

- Pure Aloha allows stations to transmit whenever they have data to be sent.
- **In pure Aloha, when each station transmits data to a channel without checking whether the channel is idle or busy, the chances of collision may occur, and the data frame can be lost.**
- When any station transmits the data frame to a channel, the pure Aloha waits for the receiver's acknowledgment.
- If it does not acknowledge the receiver end within the specified time, the station waits for a **random amount of time**, called **the backoff time (Tb)**. And the station may assume the frame has been lost or destroyed.
- Therefore, it retransmits the frame until all the data are successfully transmitted to the receiver.
- Since different stations wait for different amount of time, the probability of further collision decreases.
- The throughput of pure aloha is maximized when frames are of uniform length.


Station 1
Frame 1.1
Frame 1.2
Time
Station 2
Frame 2.1
Frame 2.2
Time
Station 3
Frame 3.1
Frame 3.2
Time
Station 4
Frame 4.1
Frame 4.2
Time
Collision duration
Collision duration



- Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be garbled.
- If the first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted later.

- **The time required to send a frame** is called **Frame Transmission Time**(T_{fr}).
- **Vulnerable time** (the length of time in which there is a possibility of collision) for pure ALOHA is:

➔ **Vulnerable Time = 2* T_{fr}**

(T_{fr} Frame transmission time)

➔ **Throughput (S) = G x e-2G**

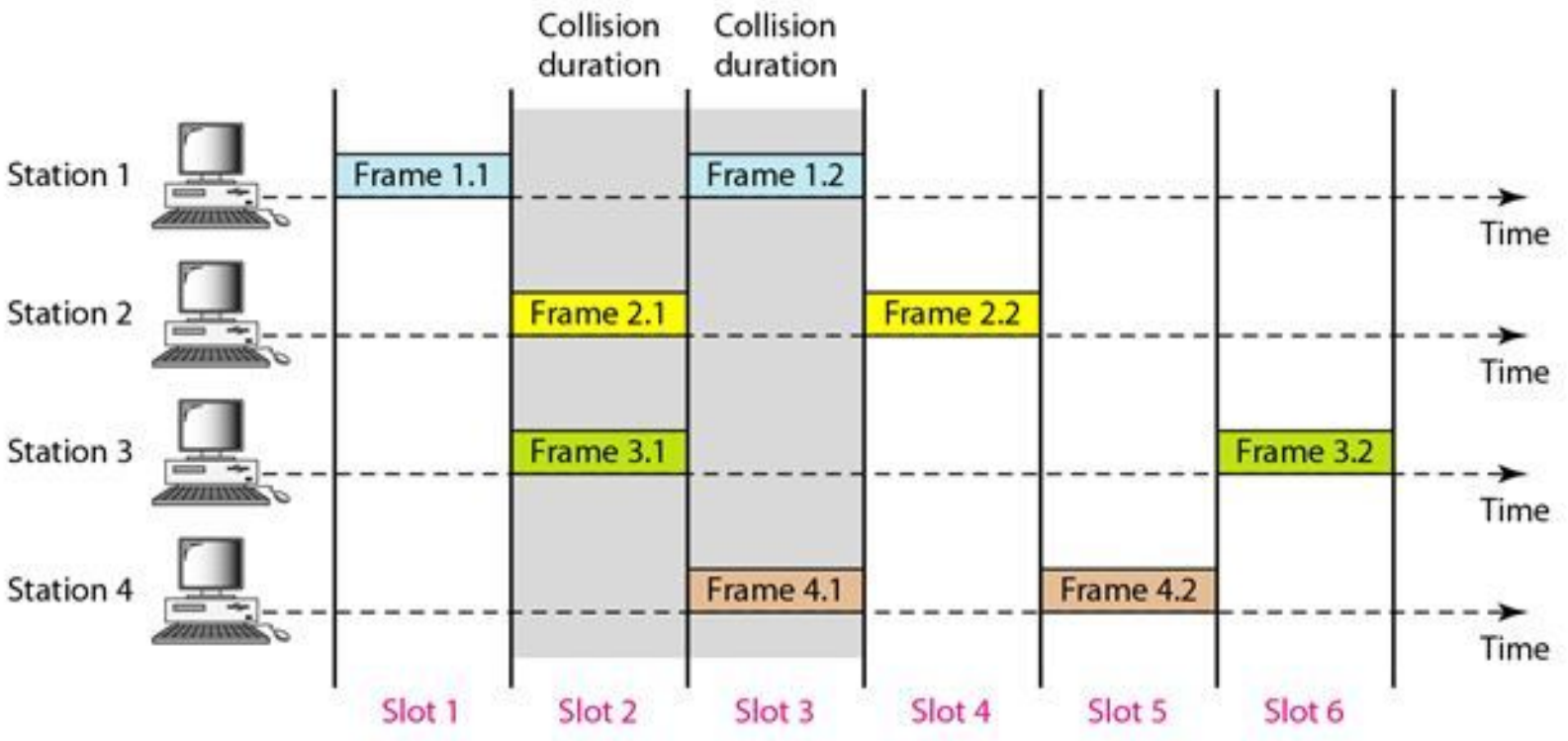
Where **G** is the average number of transmission attempts per unit of time and **e** is the mathematical constant approximately equal to 2.71828. The maximum throughput occurs when G = 0.5,

➔ **Maximum throughput = 0.184**

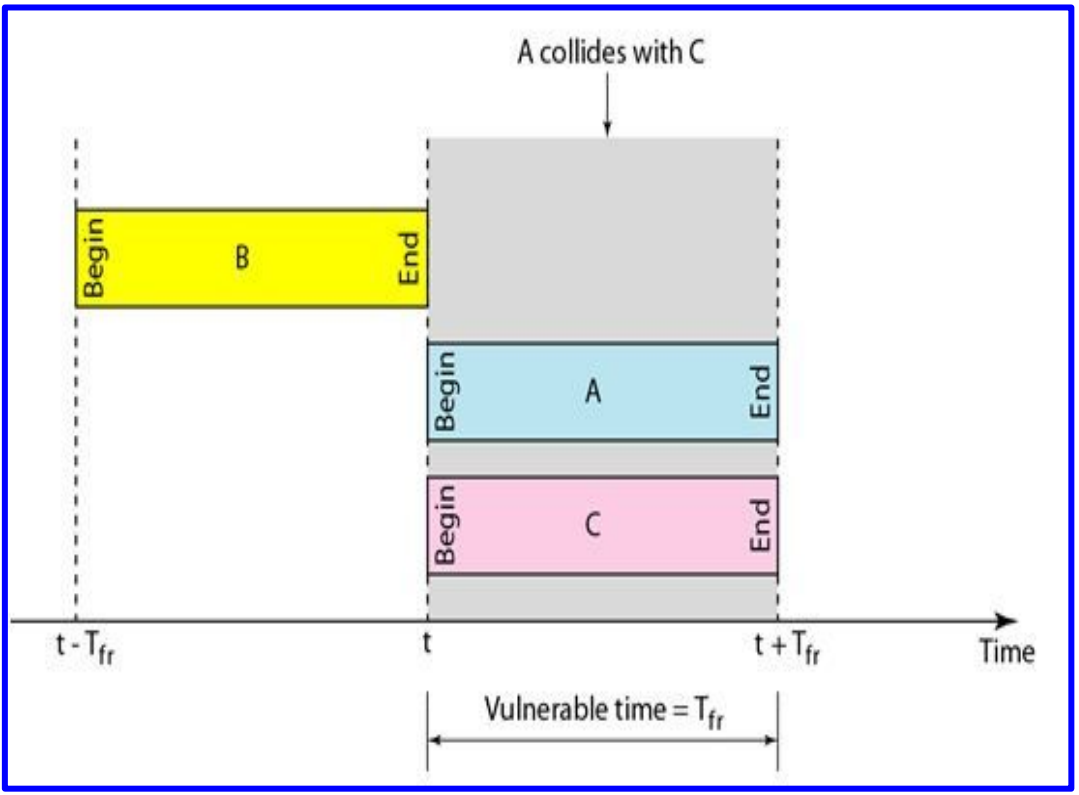
SLOTTED ALOHA

- Slotted ALOHA is an improvement over ALOHA and Pure ALOHA, introducing a **synchronized time-slot-based approach**.
- In Slotted ALOHA, **the time is divided into discrete slots**.
- Devices are allowed to transmit only **at the beginning of each time slot.**
- This synchronization **reduces the chances of collisions**.
- If a collision occurs, the devices involved wait for the **next time slot to retransmit**.
- If a station misses out the allowed time, it must wait for the next slot, This **reduces the probability of collision.**

Slotted Aloha



Vulnerable Time

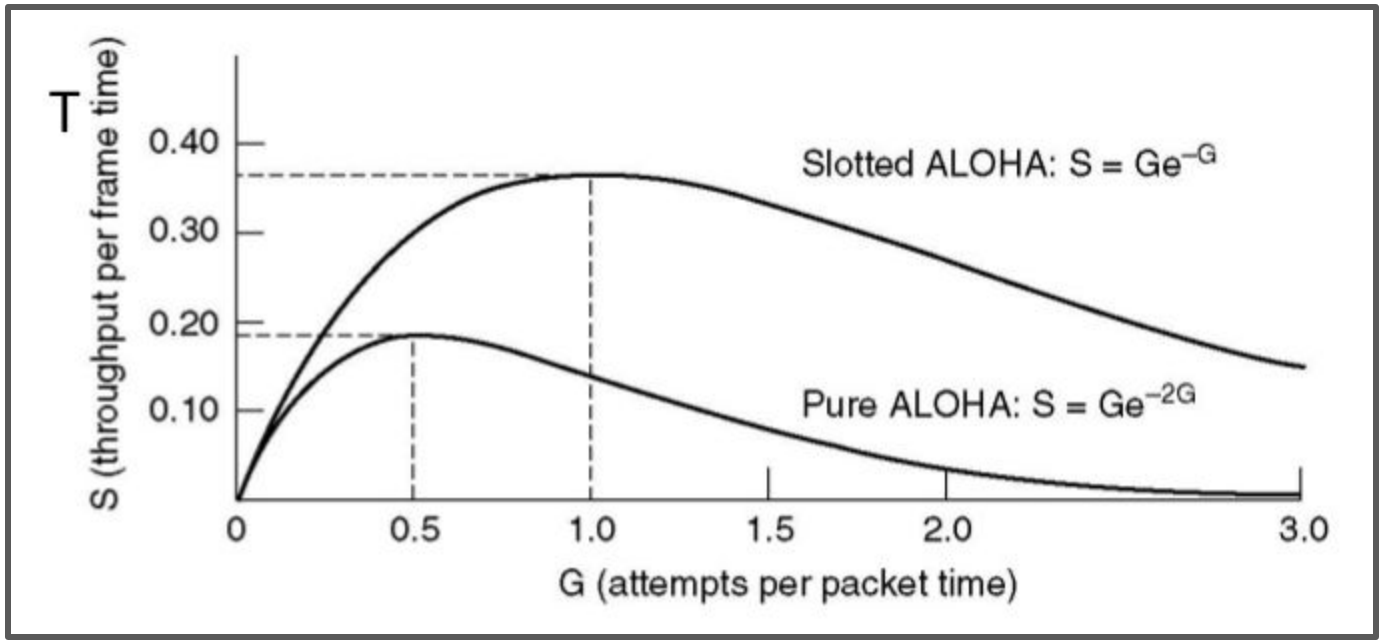


- ➔ Vulnerable Time = T_{fr}

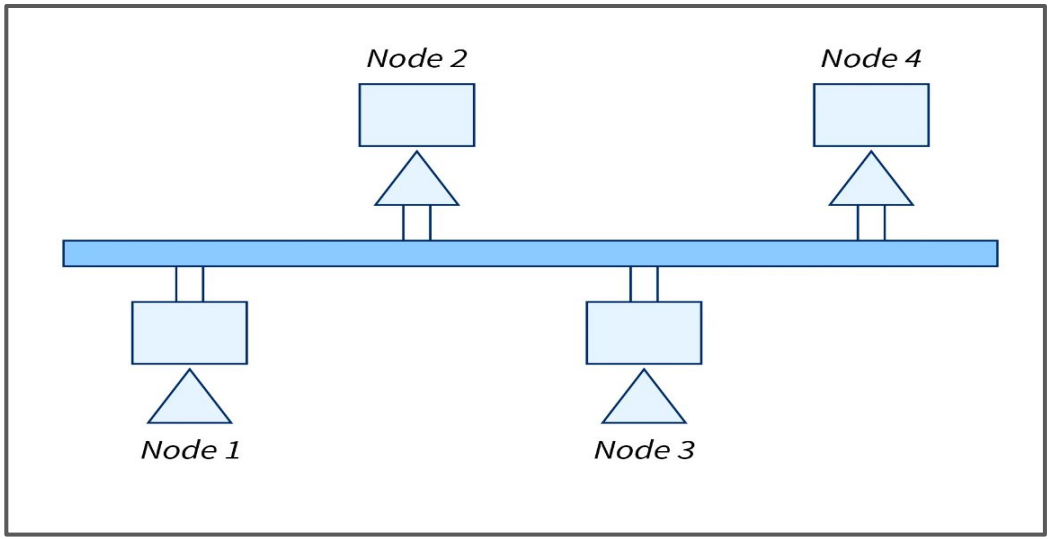
  Where T_{fr} —>Frame Transmission Time
- ➔ Throughput = **S = G x e-G**
- ➔ Maximum throughput = **0.368** for G=1

Pure Aloha	Slotted Aloha
Any station can send data at any moment.	Any station can send data at the start of any time period.
The time is not globally synchronised and is continuous.	The time is discrete and synced worldwide.
Vulnerable time for a collision to occur = 2 x Tt.	Vulnerable time for a collision to occur = Tt
The biggest benefit of pure aloha is its ease of implementation.	The primary benefit of slotted aloha is that it lowers collisions by half and doubles the efficiency of pure aloha.

Throughput versus offered traffic for ALOHA System



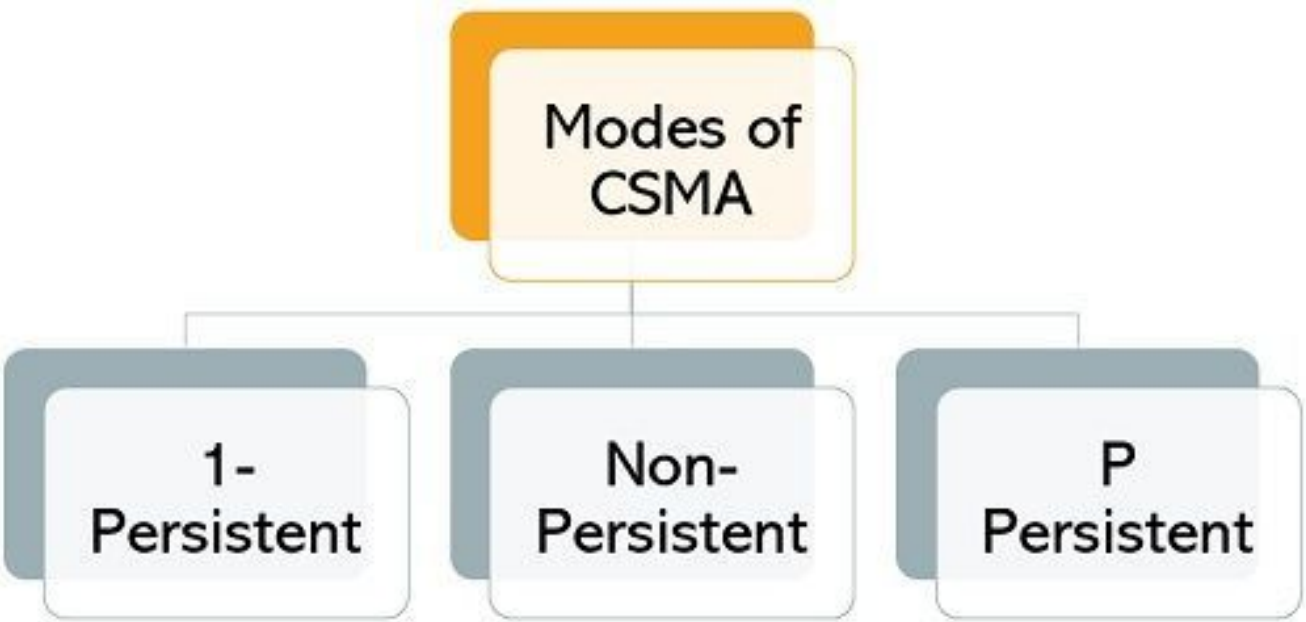
CSMA



Carrier Sense Multiple Access

- Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data.
- If it is idle then it sends data, otherwise it waits till the channel becomes idle.
- **Principle of CSMA** : **“ Sense before transmit “ or “ Listen before talk”.**
- **Carrier Busy** = Transmission is taking place
- **Carrier Idle** = No transmission currently taking place
- However there is still chance of collision in CSMA due to **propagation delay.**
  - For example, if station A wants to send data, it will first sense the medium.If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

Types of CSMA



1-Persistent CSMA

- In this, if the station wants to transmit the data. Then the station first senses the medium.
- If the medium is busy then the station waits until the channel becomes idle. And the station **continuously senses the channel** until the medium becomes idle.
- If the station detected the channel as idle then the station will immediately send the data frame with 1 probability that's why the name of this method is 1-persistent.
- Refer to the below image to show the flow diagram of the 1-persistent method of CSMA

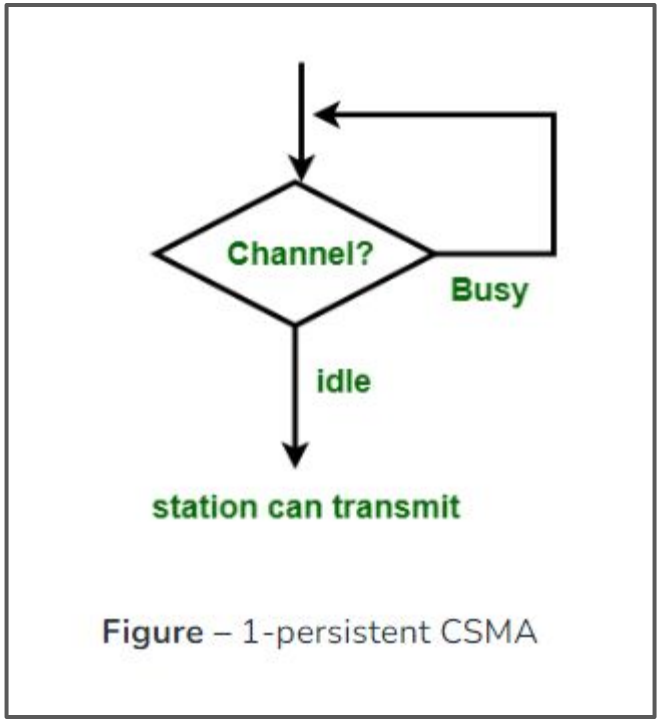
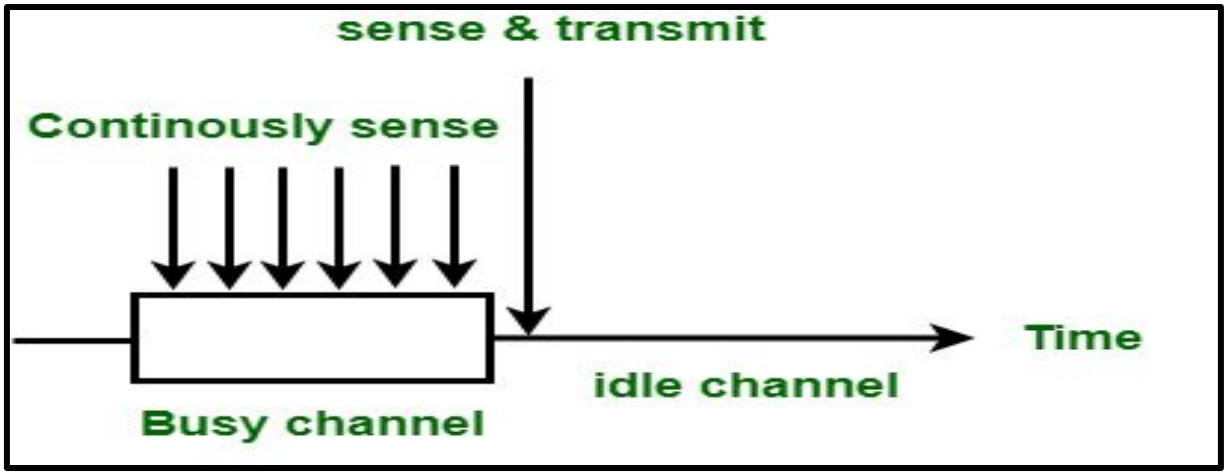


Figure – 1-persistent CSMA

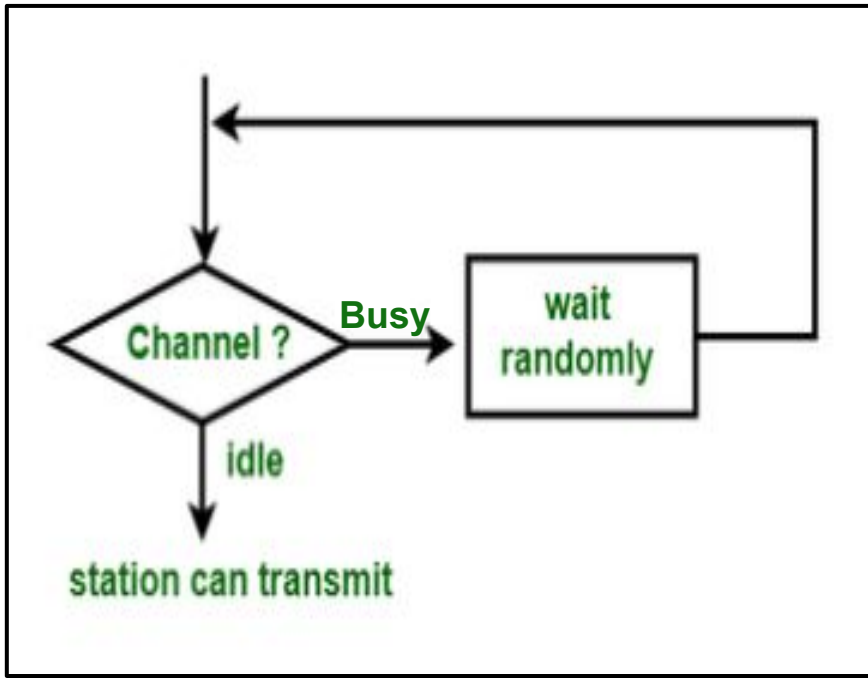


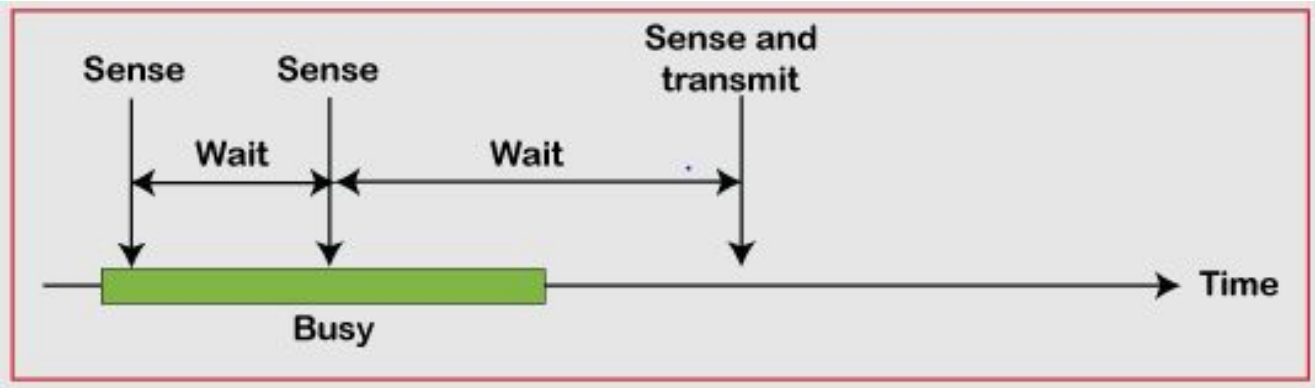
Problems :

- In this method there is **a high possibility of collision** as two or more station sense the channel idle at the same time and transmits data simultaneously which may lead to a collision.
- In this method, once the station finds that the medium is idle then it immediately sends the frame.
- By using this method there are higher chances for collision because **it is possible that two or more stations find the shared medium idle at the same time** and then they send their frames immediately.

Non-Persistent CSMA

- If the station wants to transmit the data then first of all it will sense the medium.
- If the medium is idle then the station will immediately send the data.
- Otherwise, if the medium is busy then the station **waits for a random amount of time and then again senses the channel** after waiting for a random amount of time.
- In Non-persistent there is less chance of collision in comparison to the 1-persistent method as this station will not continuously sense the channel but the channel waiting for a random amount of time.





Advantage of non-persistent

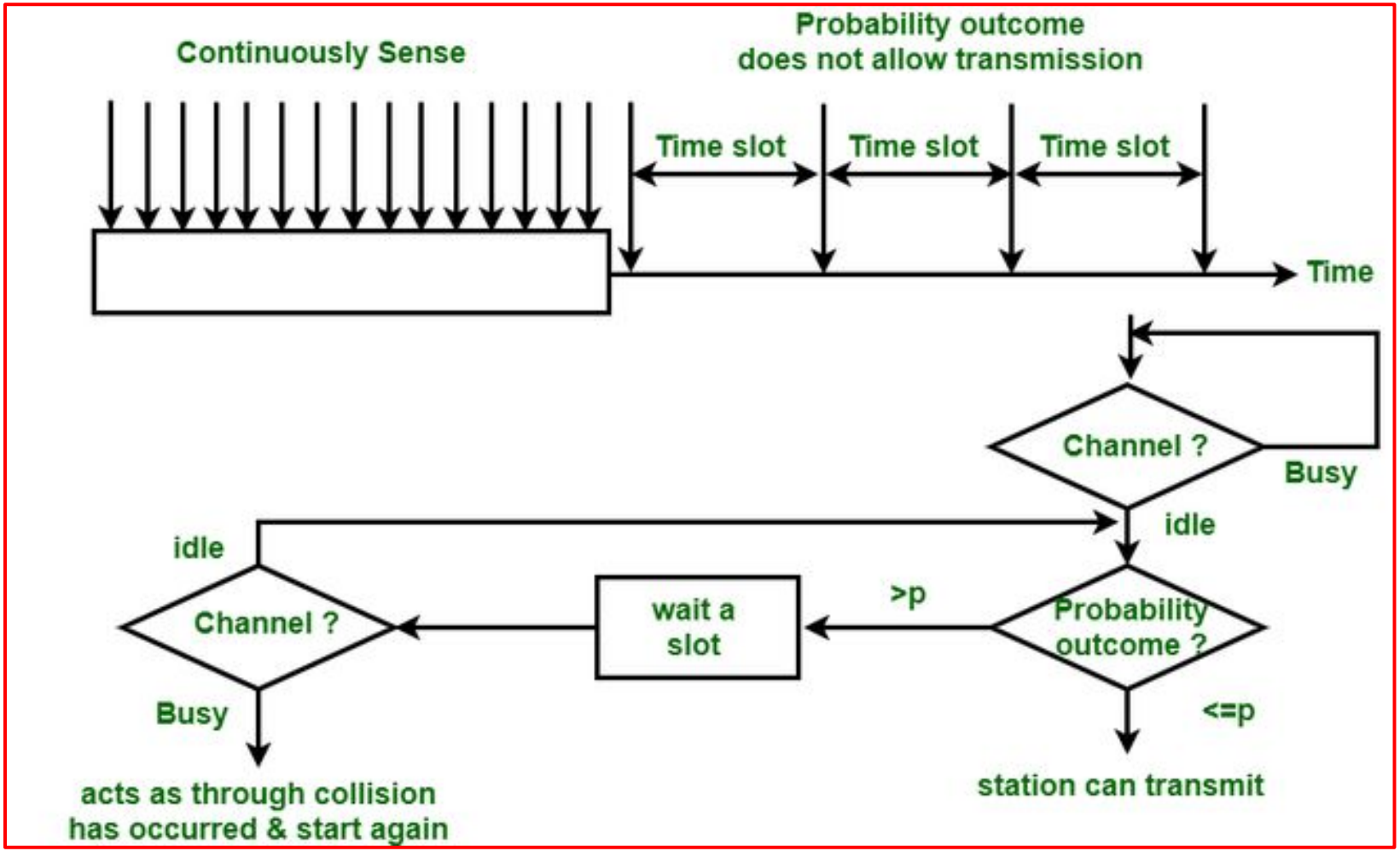
- It reduces the chance of collision because the stations **wait a random amount of time**.
- It is unlikely that two or more stations will wait for same amount of time and will retransmit at the same time.

Disadvantage of non-persistent

- It reduces the efficiency of network because the channel remains idle when there may be stations with frames to send.
- This is due to the fact that the stations wait a random amount of time after the collision.

P-Persistent

- It is the combination of **1-Persistent and Non-persistent modes.**
- This method is used when **channel has time slots** such that the time slot duration is equal to or greater than the maximum propagation delay time.
- Whenever a station becomes ready to send, it senses the channel.
- If channel is **busy**, station **waits until next slot**.
- If channel is **idle**, it **transmits with a probability p**.
- With the probability q=l-p, the station then waits for the beginning of the next time slot.
- If the next slot is also idle, it either transmits or waits again with probabilities p and q.
- This process is repeated till either frame has been transmitted or another station has begun transmitting.


Continuously Sense
Probability outcome
does not allow transmission
Time slot
Time slot
Time slot
Time
Channel ?
Busy
idle
idle
Channel ?
wait a
slot
>p
Probability
outcome ?
Busy
<=p
acts as through collision
has occurred & start again
station can transmit

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

CSMA/CD stands for Carrier Sense Multiple Access with Collision Detection.

It is a protocol used in **Ethernet networks** to control access to the shared transmission medium and manage collisions when multiple devices attempt to transmit data simultaneously.

Here's how CSMA/CD works:

Carrier Sense:

Before a device starts transmitting data, it listens to the network to check if the medium is idle. If it detects other devices transmitting, it waits for the network to become idle.

Multiple Access:

If the medium is **idle**, the device can start transmitting its data.

Collision Detection:

- ★ Sender transmits its data on the link. CSMA/CD does not use an ‘acknowledgment’ system.
- ★ It checks for successful and unsuccessful transmissions through collision signals.
- ★ During transmission, if **a collision signal** is received by the node, transmission is stopped.
- ★ The station then transmits **a jam signal** onto the link and waits for **random time** intervals before it re-sends the frame.
- ★ After some random time, it again attempts to transfer the data and repeats the above process.
- ★ The station’s hardware must listen to the channel while it is transmitting.
- ★ **If the signal it reads back is different from the signal it is putting out, it knows that a collision is occurring.**

Backoff and Retransmission:

After a collision is detected, the device waits for a random amount of time (backoff) before attempting to retransmit the data. The random backoff helps to reduce the probability of another collision. The device retransmits the data once the backoff time elapses and the medium is idle.

- ➔ CSMA/CD was **widely used** in early Ethernet networks based on **coaxial or twisted-pair cables.**
- ➔ However, with the advent of faster Ethernet technologies and the prevalence of switched networks, CSMA/CD has become less common.

CSMA/CD

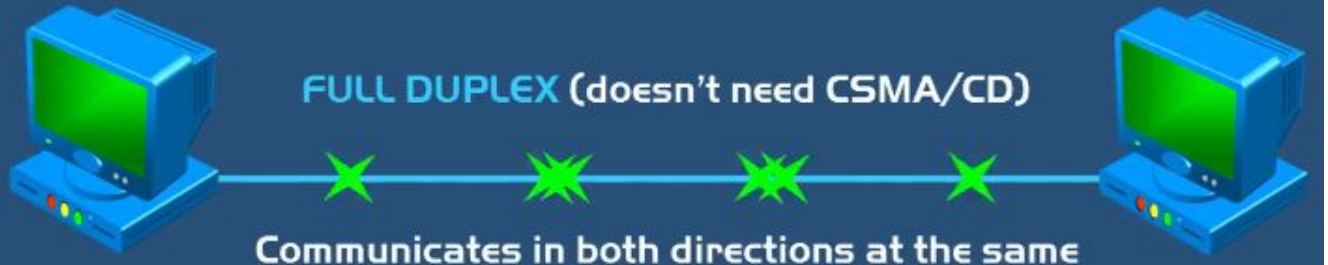
Carrier Sense Multiple Access with Collision Detection

CSMA/CD was used on early Ethernet networks.

Not as relevant today.

HALF DUPLEX (uses CSMA/CD)

Communicates in both directions, but not at the same time.



FULL DUPLEX (doesn't need CSMA/CD)

Communicates in both directions at the same time.

The Conceptual Model for CSMA/CD

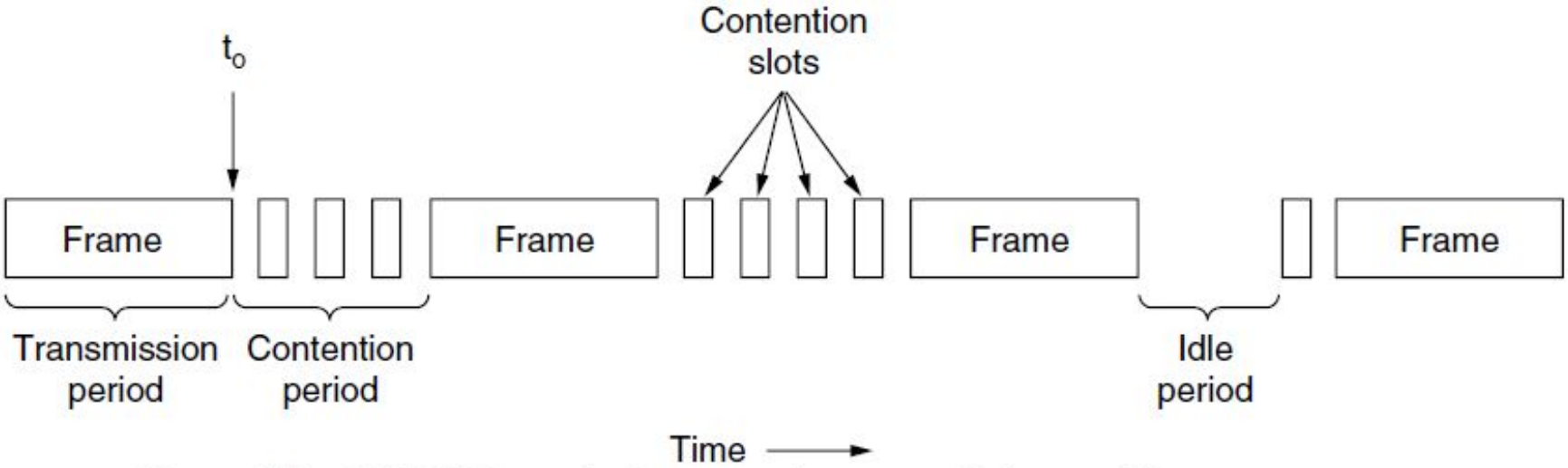


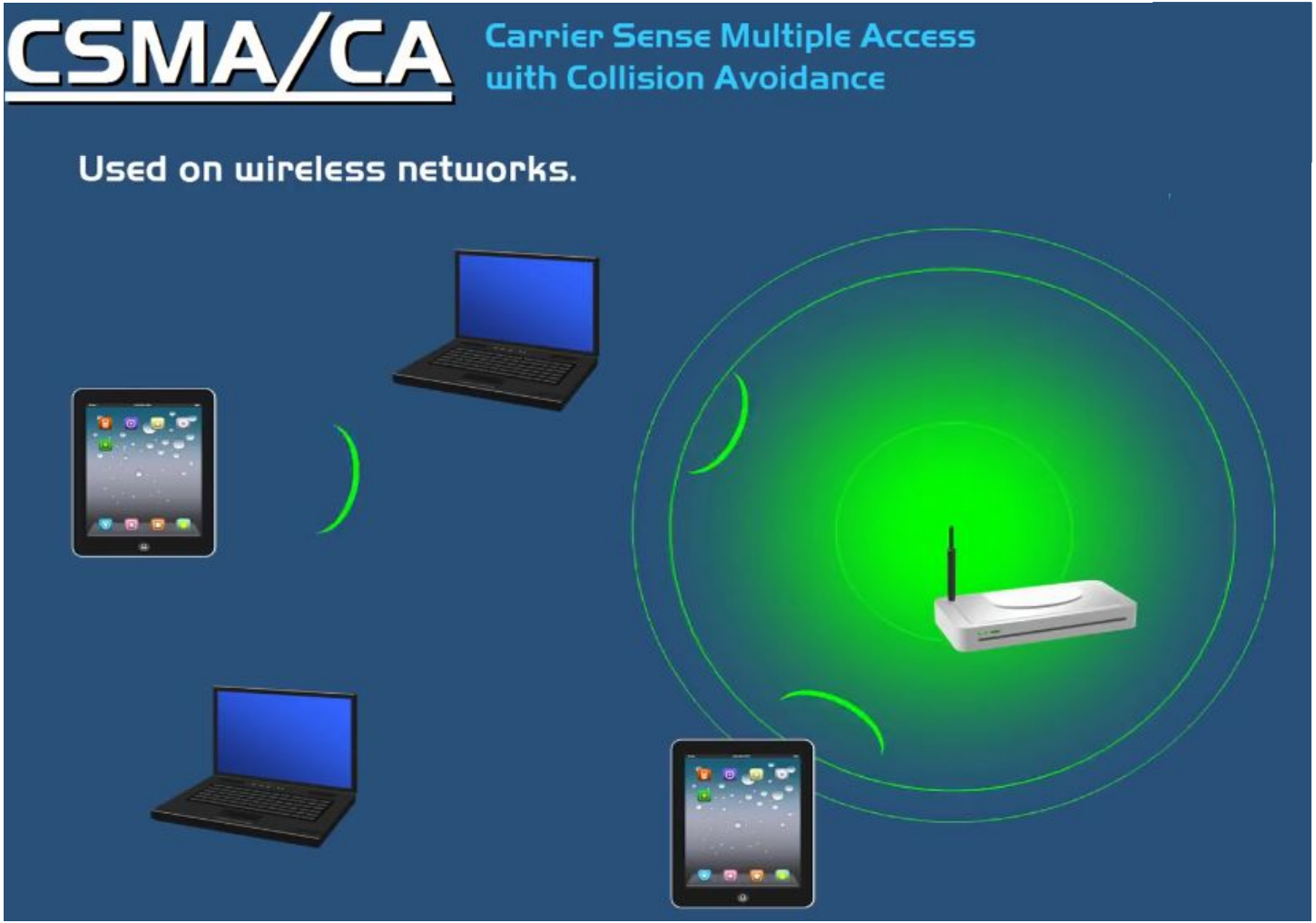
Figure 4-5. CSMA/CD can be in contention, transmission, or idle state.

- ★ CSMA/CD, as well as many other LAN protocols, uses the conceptual model of Fig. 4-5.
- ★ At the point marked t 0, a station has finished transmitting its frame.
- ★ Any other station having a frame to send may now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision.
- ★ If a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again (assuming that no other station has started transmitting in the meantime).
- ★ Therefore, our model for CSMA/CD will consist of alternating **contention and transmission periods, with idle periods** occurring when all stations are quiet (e.g., for lack of work).

Contention Periods:

Contending devices (those wanting to transmit) use contention periods, These contention periods are time intervals during which **a device listens to the network to determine if it's clear for transmission.**

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)


CSMA/CA
Carrier Sense Multiple Access
with Collision Avoidance
Used on wireless networks.

- ➔ CSMA/CA stands for *Carrier Sense Multiple Access with Collision Avoidance*.
- ➔ It is a protocol used in **wireless networks**, particularly **Wi-Fi networks**, to manage access to the shared wireless medium and prevent collisions.
- ➔ **In wired networks**, if a collision has occurred then the energy of the received signal almost doubles, and the station can sense the possibility of collision.
- ➔ In the case of wireless networks, most of the energy is used for transmission, and the energy of the received signal increases by only 5-10% if a collision occurs. It can't be used by the station to sense collision. Therefore **CSMA/CA has been specially designed for wireless networks.**

Here's how CSMA/CA works:

1. **Carrier Sense:** Before a device starts transmitting data, it listens to the wireless medium to check if it is idle. If the medium is busy, the device waits for it to become idle.

2. Collision Avoidance:

Rather than relying on collision detection, CSMA/CA uses a technique called collision avoidance. The device sends **a Request to Send (RTS)** frame to the Wireless Access Point(WAP), indicating its intention to transmit data. The RTS frame includes information about the duration of the transmission.

3. Clear to Send (CTS): Upon receiving the RTS frame, the access point responds with a Clear to Send (CTS) frame, granting permission to the device to transmit data. The CTS frame also includes the duration of the transmission.

4.Transmission: Once the device receives the CTS frame, it can start transmitting its data. Other devices within range of the access point listen to the CTS frame and defer their own transmissions to avoid collisions.