Chapter 3
Data Link Layer

DLL (Data Link Layer)

Data Link Layer is second layer of OSI Layered Model. The data link layer is responsible for the node-to-node delivery of the message. The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer. When a packet arrives in a network, it is the responsibility of DLL to transmit it to the Host using its MAC address.

Functions of DLL

- **Framing:** Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, it sends each frame bit-by-bit on the hardware. At receiver’ end, data link layer picks up signals from hardware and assembles them into frames.

- **Physical addressing:** After creating frames, the Data link layer adds physical addresses (MAC address) of the sender and/or receiver in the header of each frame.

- **Error control:** Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.

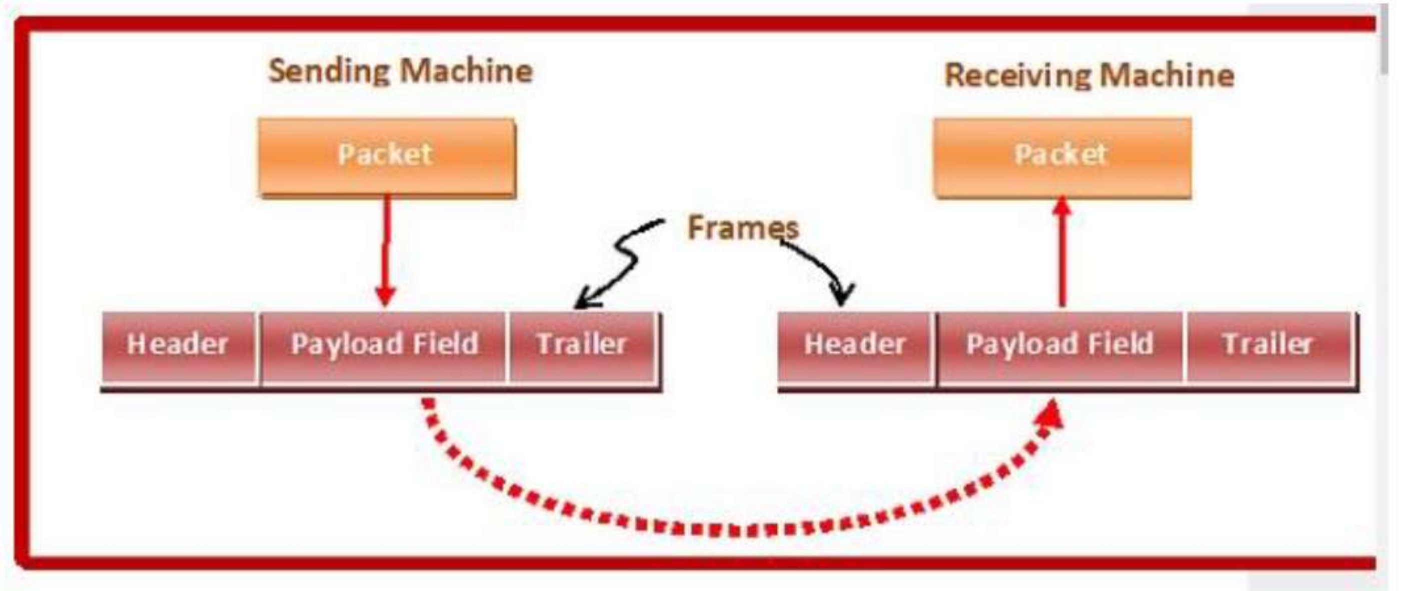
Functions of DLL

- **Access Control:** When more than two or two devices are connected to the common link, data link layer protocols are necessary to determine which device has control over the link at any point of time.

- **Flow control:** If the rate at which the data are consumed by the receiver is less than the rate produced by the sender, the data link layer deals with a flow control mechanism to prevent overrun the receiver.

Functions of DLL

- **Framing:** Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, it sends each frame bit-by-bit on the hardware. At receiver’ end, data link layer picks up signals from hardware and assembles them into frames.



Functions of DLL

Parts of a Frame

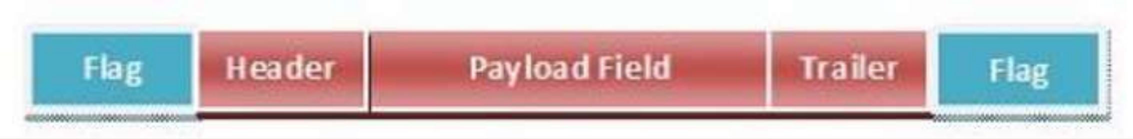
A frame has the following parts −

Frame Header − It contains the source and the destination addresses of the frame.

Payload field − It contains the message to be delivered.

Trailer − It contains the error detection and error correction bits.

Flag − It marks the beginning and end of the frame.



Functions of DLL

Types of Framing

Framing can be of two types, fixed sized framing and variable sized framing.

Fixed-sized Framing

Here the size of the frame is fixed and so the frame length acts as delimiter of the frame. Consequently, it does not require additional boundary bits to identify the start and end of the frame.

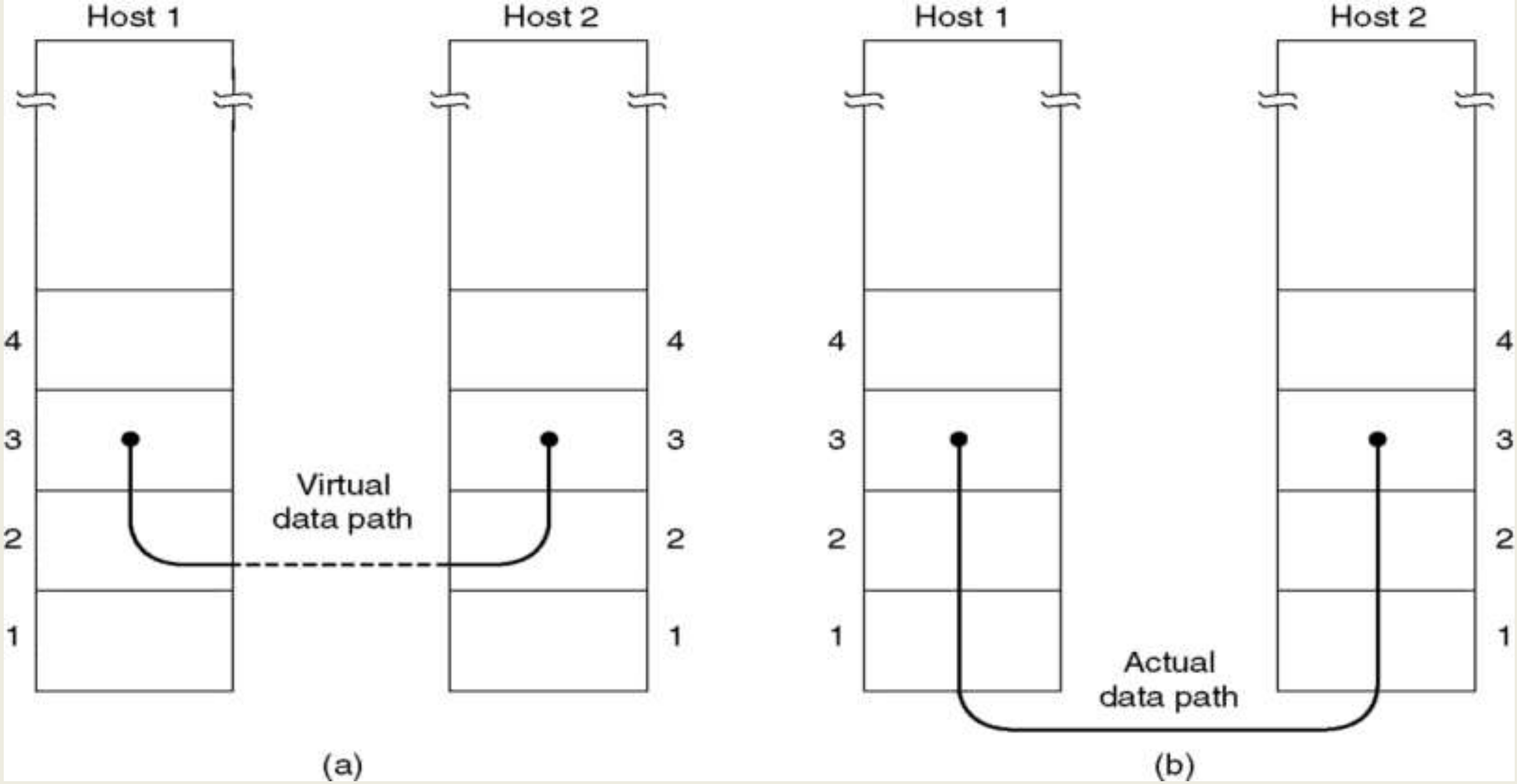
Variable – Sized Framing

Here, the size of each frame to be transmitted may be different. So additional mechanisms are kept to mark the end of one frame and the beginning of the next frame.

It is used in local area networks.

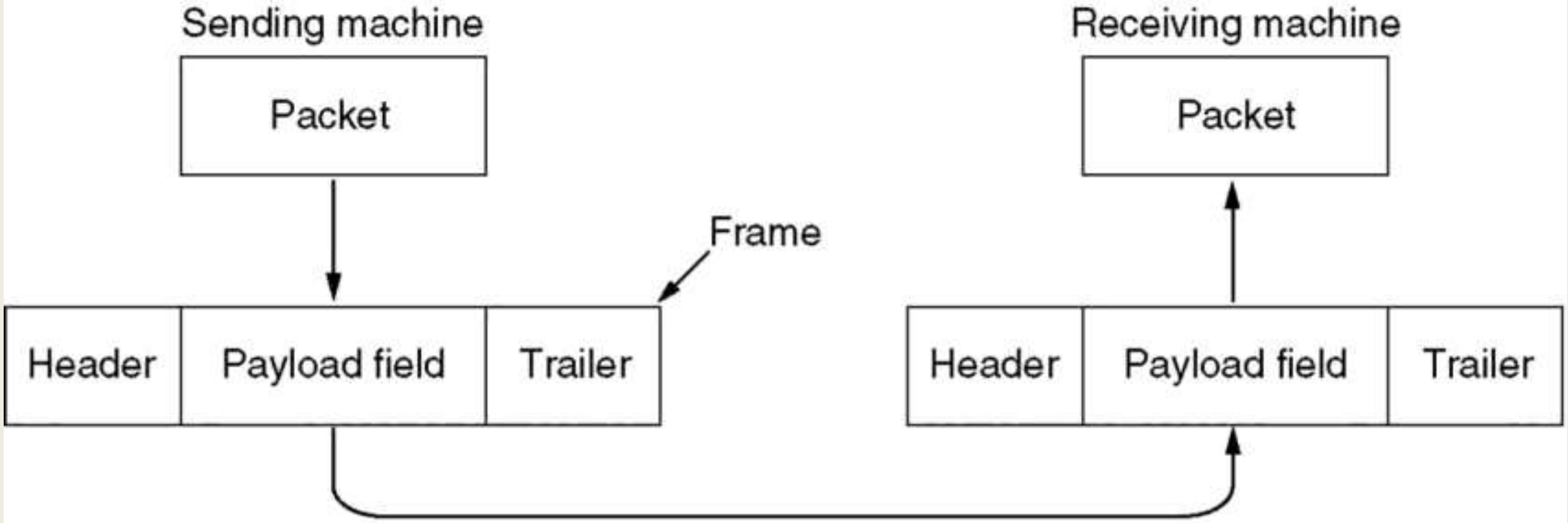
Data Flow

- To the Network Layer, it looks as though the path to the new machine happens at the DLL level, when it is really happening at the physical level.



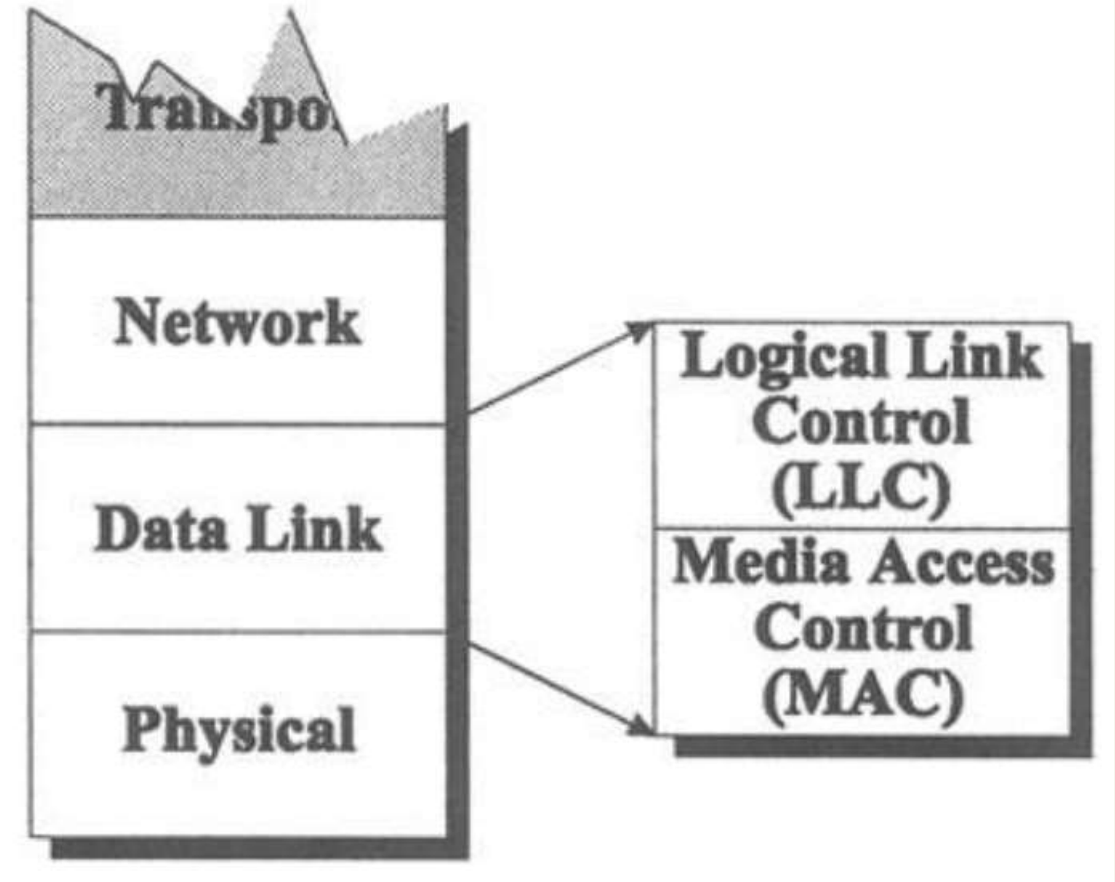
How the data is sent?

- Takes the **packets** of information from the Network Layer.
- Convert them into **frames** for transmission.
- Each frame holds the payload(packet) plus a header and a trailer (overhead).
- Transmit frames over the physical layer.



Overview of Logical Link Control (LLC) and Media Access Control (MAC):

The data link layer is divided into two sublayers:



LLC(Logical Link Control) and MAC (Media Access Control)

- The LLC sublayer acts as an interface between the media access control (MAC) sublayer and the network layer. Logical Link Control (LLC) sublayer provides the logic for the data link. It controls the synchronization, flow control, and error checking functions of the data link layer.

- Media Access Control (MAC) sublayer provides control for accessing the transmission medium. It is responsible for moving data packets from one network interface to another, across a shared transmission medium. Physical addressing is handled at the MAC sublayer. When sending data to another device on the network, the MAC sublayer encapsulates higher-level frames into frames appropriate for the transmission medium. It deals with actual control of media

Framing Approaches

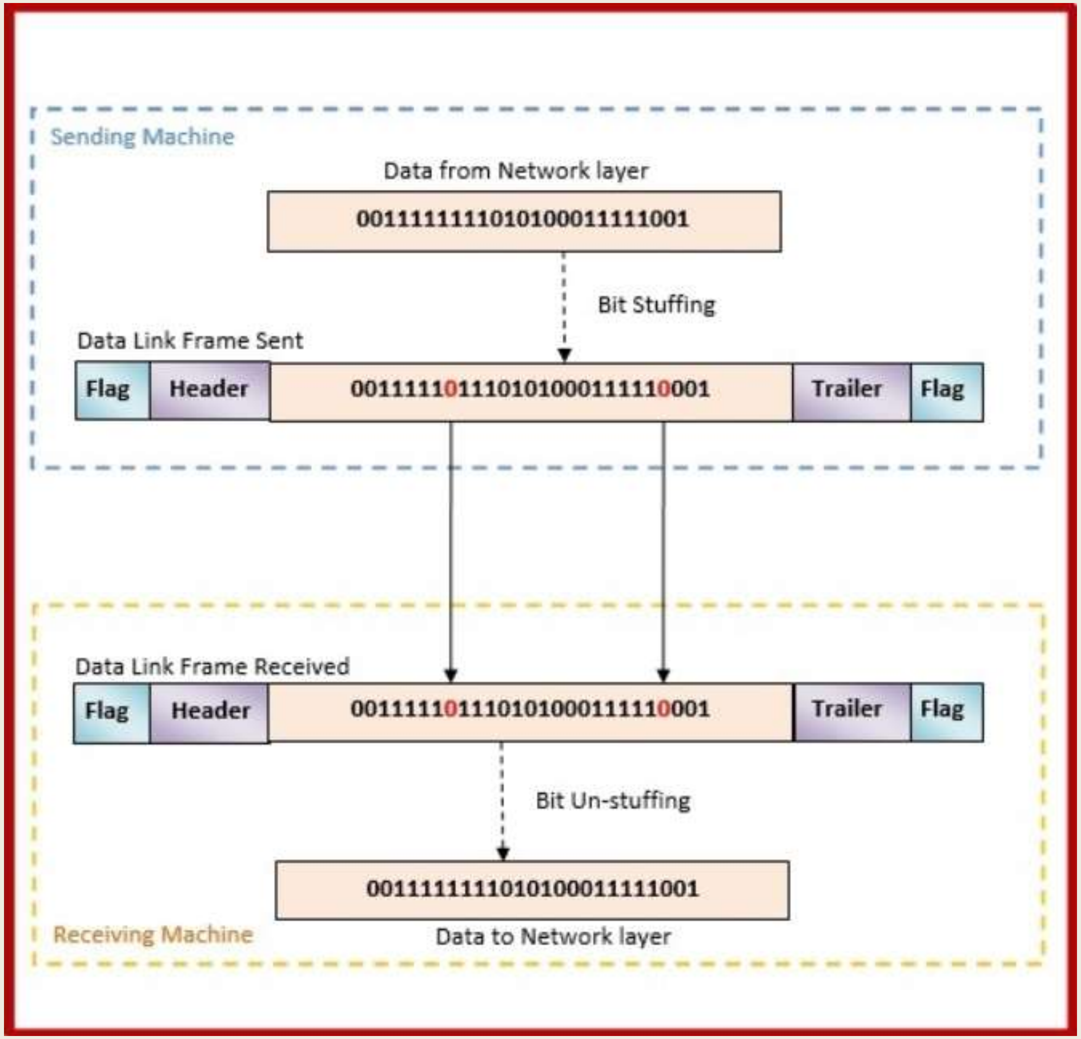
- Bit Stuffing
- Byte Stuffing.

Bit Stuffing Mechanism

In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.

Bit Stuffing Mechanism

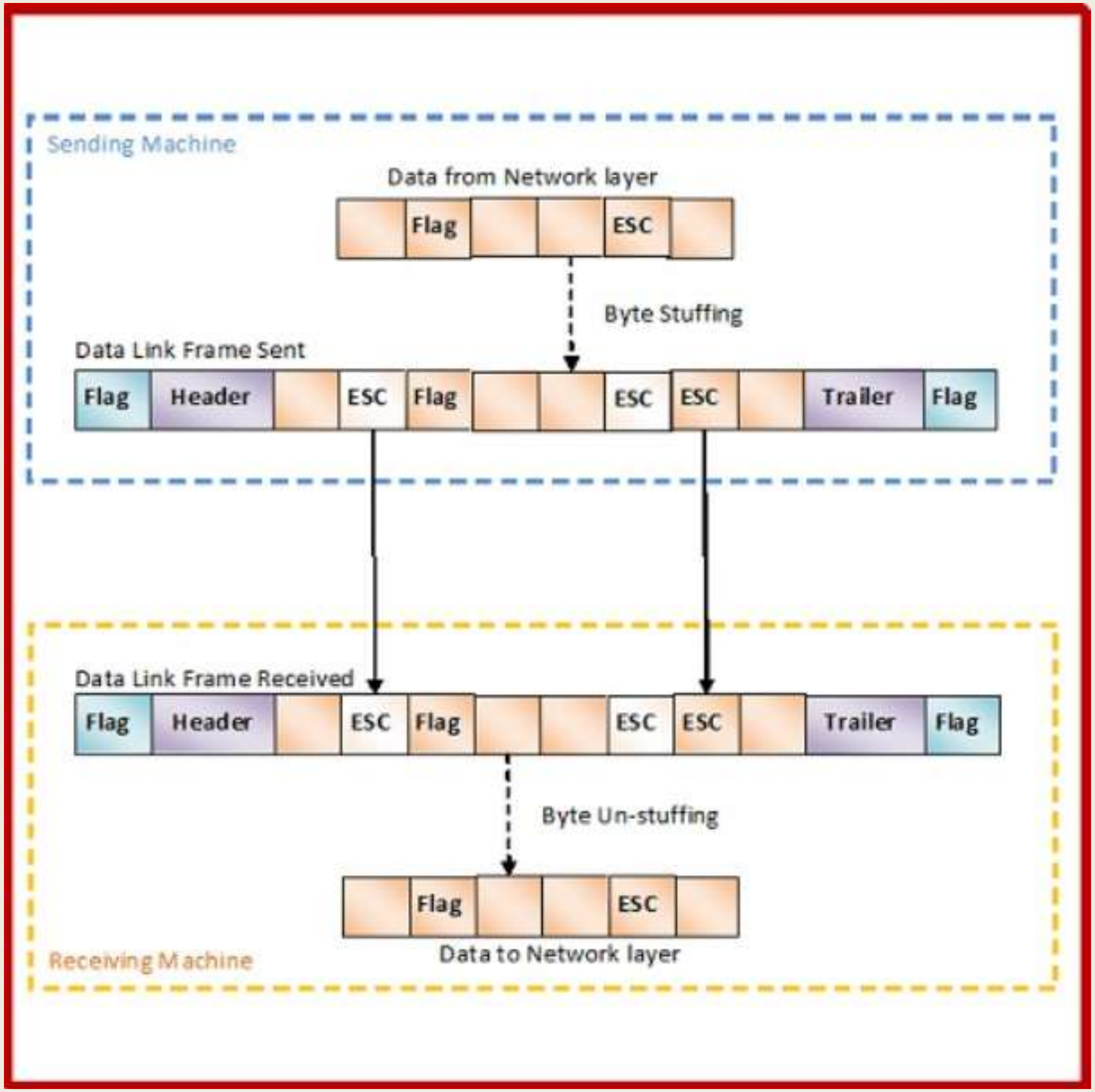


Byte Stuffing Mechanism

If the pattern of the flag byte is present in the message byte, there should be a strategy so that the receiver does not consider the pattern as the end of the frame. In character – oriented protocol, the mechanism adopted is byte stuffing.

In byte stuffing, a special byte called the escape character (ESC) is stuffed before every byte in the message with the same pattern as the flag byte. If the ESC sequence is found in the message byte, then another ESC byte is stuffed before it.

Byte Stuffing Mechanism



Error detection and Control

- Data-link layer uses the techniques of error control simply to ensure and confirm that all the data frames or packets, i.e. bit streams of data, are transmitted or transferred from sender to receiver with certain accuracy. Using or providing error control at this data link layer is an optimization, it was never requirement. Error control is basically process in data link layer of detecting or identifying and re-transmitting data frames that might be lost or corrupted during transmission.
- In both of these cases, receiver or destination does not receive correct data-frame and sender or source does not even know anything about any such loss regarding data frames. Therefore, in such type of cases, both sender and receiver are provided with some essential protocols that are required to detect or identify such type of errors like loss of data frames.

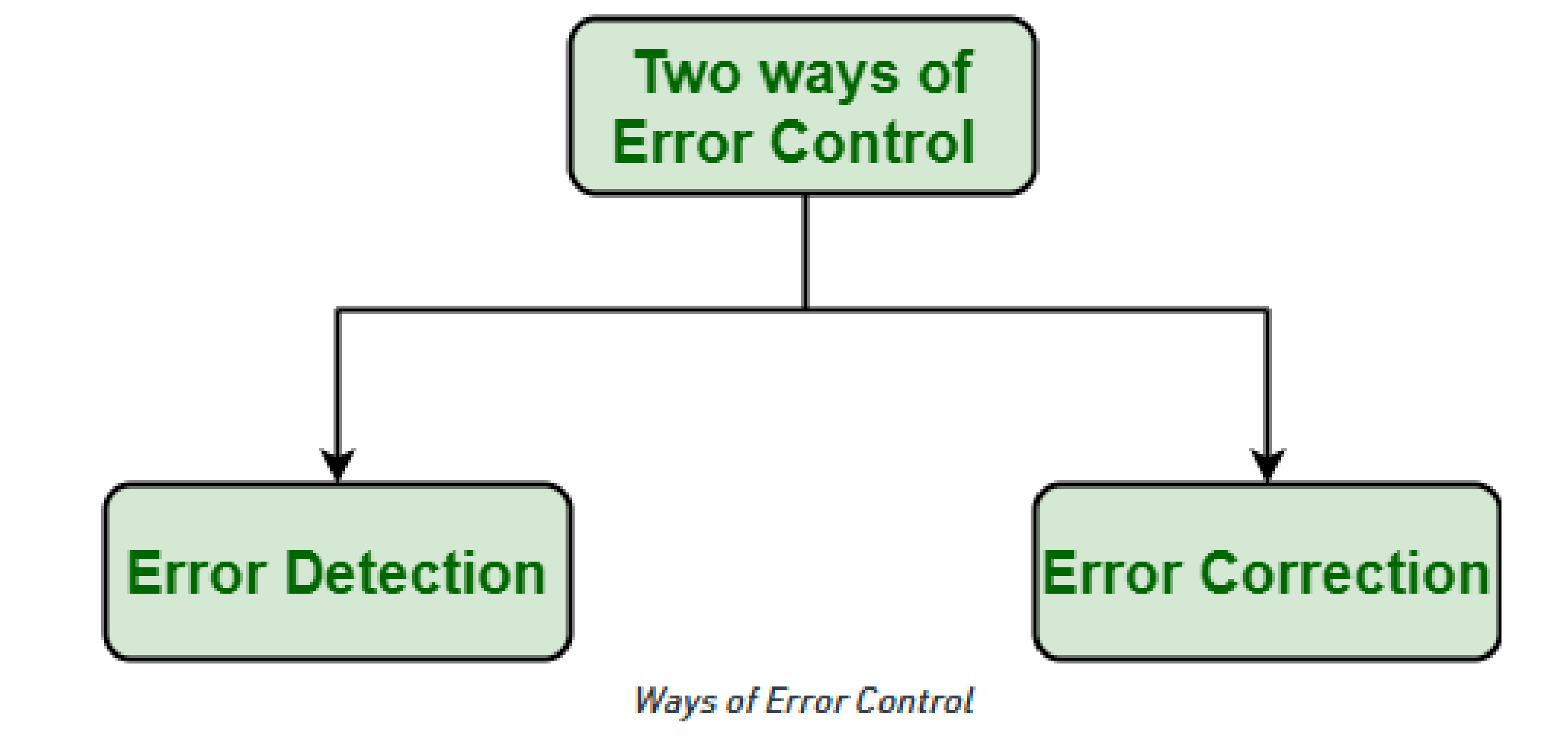
Error detection and Control

- The Data-link layer follows technique known as re-transmission of frames to detect or identify transit errors and also to take necessary actions that are required to reduce or remove such errors. Each and every time an effort is detected during transmission, particular data frames retransmitted and this process is known as ARQ (Automatic Repeat Request).

Error detection and Control

- **Ways of doing Error Control :**

There are basically two ways of doing Error control as given below :



Ways of Error Control

Error detection and Control

Error Detection :

Error detection, as name suggests, simply means detection or identification of errors. These errors may cause due to noise or any other impairments during transmission from transmitter to the receiver, in communication system. It is class of technique for detecting garbled i.e. unclear and distorted data or message.

Error Correction :

Error correction, as name suggests, simply means correction or solving or fixing of errors. It simply means reconstruction and rehabilitation of original data that is error-free. But error correction method is very costly and is very hard.

Error detection and Control

Error Control

- Error control includes both error detection and error correction.
- It allows the receiver to inform the sender if a frame is lost or damaged during transmission and coordinates the retransmission of those frames by the sender.
- Error control in the data link layer is based on automatic repeat request (ARQ). Whenever an error is detected, specified frames are retransmitted.

Error detection and Control

Phases in Error Control

The error control mechanism in data link layer involves the following phases –

Detection of Error – Transmission error, if any, is detected by either the sender or the receiver.

Acknowledgment – acknowledgment may be positive or negative.

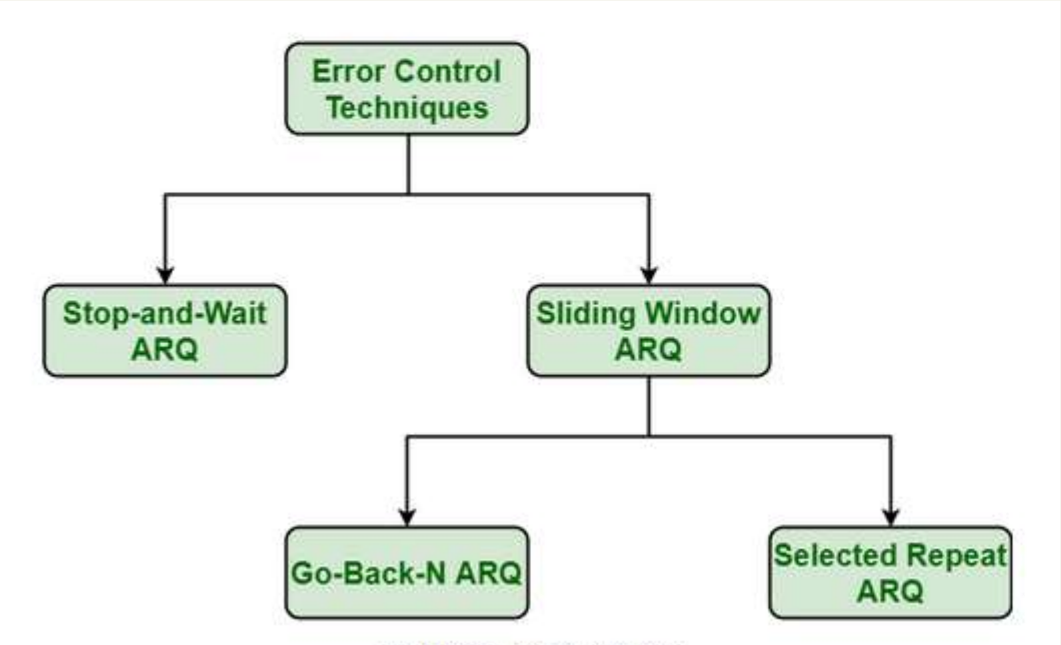
> **Positive ACK** – On receiving a correct frame, the receiver sends a positive acknowledge.
>
> **Negative ACK** – On receiving a damaged frame or a duplicate frame, the receiver sends a negative acknowledgment back to the sender.

Retransmission – The sender maintains a clock and sets a timeout period. If an acknowledgment of a data-frame previously transmitted does not arrive before the timeout, or a negative acknowledgment is received, the sender retransmits the frame.

Error detection and Control

Various Techniques for Error Control :

There are various techniques of error control as given below :



Techniques of Error Control

Stop and Wait ARQ

- This protocol involves the following transitions −
- A timeout counter is maintained by the sender, which is started when a frame is sent.
- If the sender receives acknowledgment of the sent frame within time, the sender is confirmed about successful delivery of the frame. It then transmits the next frame in queue.
- If the sender does not receive the acknowledgment within time, the sender assumes that either the frame or its acknowledgment is lost in transit. It then retransmits the frame.
- If the sender receives a negative acknowledgment, the sender retransmits the frame.

Go-Back-N ARQ

The working principle of this protocol is –

- The sender has buffers called sending window.
- The sender sends multiple frames based upon the sending-window size, without receiving the acknowledgment of the previous ones.
- The receiver receives frames one by one. It keeps track of incoming frame’s sequence number and sends the corresponding acknowledgment frames.
- After the sender has sent all the frames in window, it checks up to what sequence number it has received positive acknowledgment.
- If the sender has received positive acknowledgment for all the frames, it sends next set of frames.
- If sender receives NACK or has not receive any ACK for a particular frame, it retransmits all the frames after which it does not receive any positive ACK.

Selective Repeat ARQ

- Both the sender and the receiver have buffers called sending window and receiving window respectively.
- The sender sends multiple frames based upon the sending-window size, without receiving the acknowledgment of the previous ones.
- The receiver also receives multiple frames within the receiving window size.
- The receiver keeps track of incoming frame’s sequence numbers, buffers the frames in memory.
- It sends ACK for all successfully received frames and sends NACK for only frames which are missing or damaged.
- The sender in this case, sends only packet for which NACK is received.

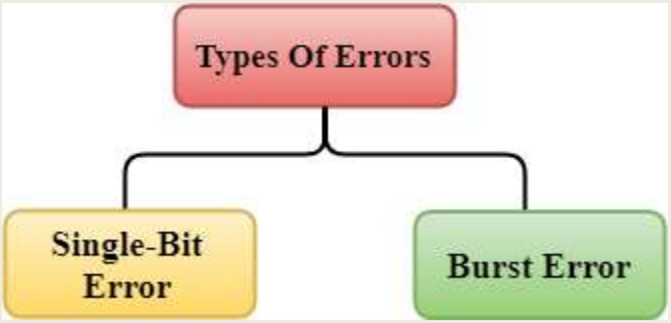
Error Detection

When data is transmitted from one device to another device, the system does not guarantee whether the data received by the device is identical to the data transmitted by another device. An Error is a situation when the message received at the receiver end is not identical to the message transmitted.

Types of Error

Errors can be classified into two categories:

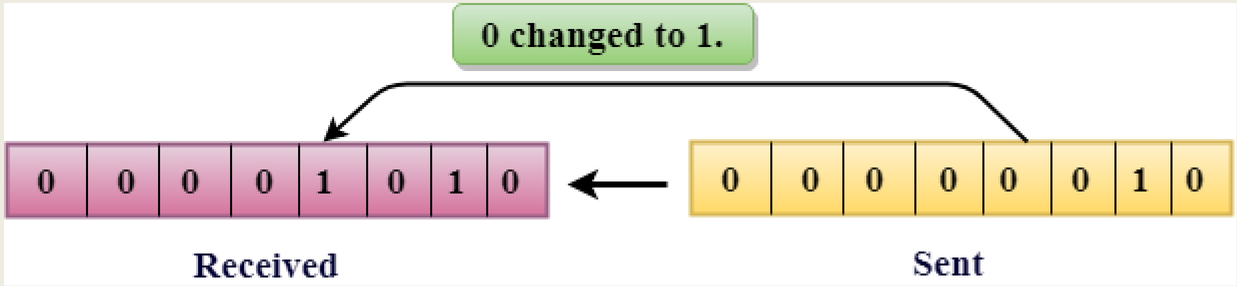
- Single-Bit Error
- Burst Error



Error Detection

Single-Bit Error

The only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.



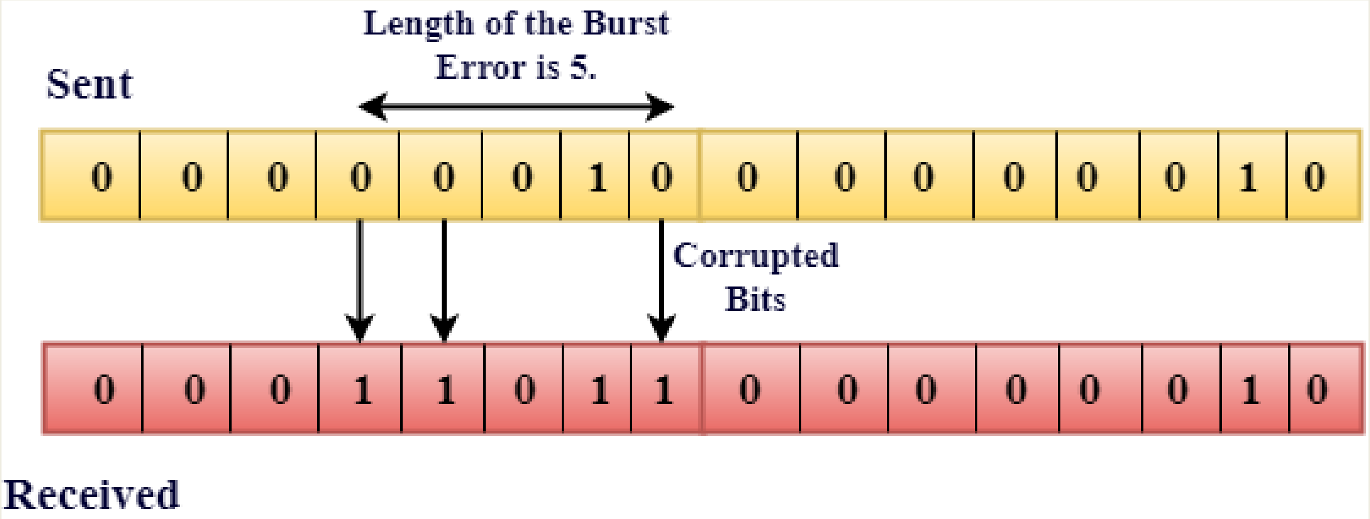
In the above figure, the message which is sent is corrupted as single-bit, i.e., 0 bit is changed to 1.

Error Detection

Burst Error:

The two or more bits are changed from 0 to 1 or from 1 to 0 is known as Burst Error.

The Burst Error is determined from the first corrupted bit to the last corrupted bit.



Error Detection Techniques

There are four main techniques for detecting errors in frames:

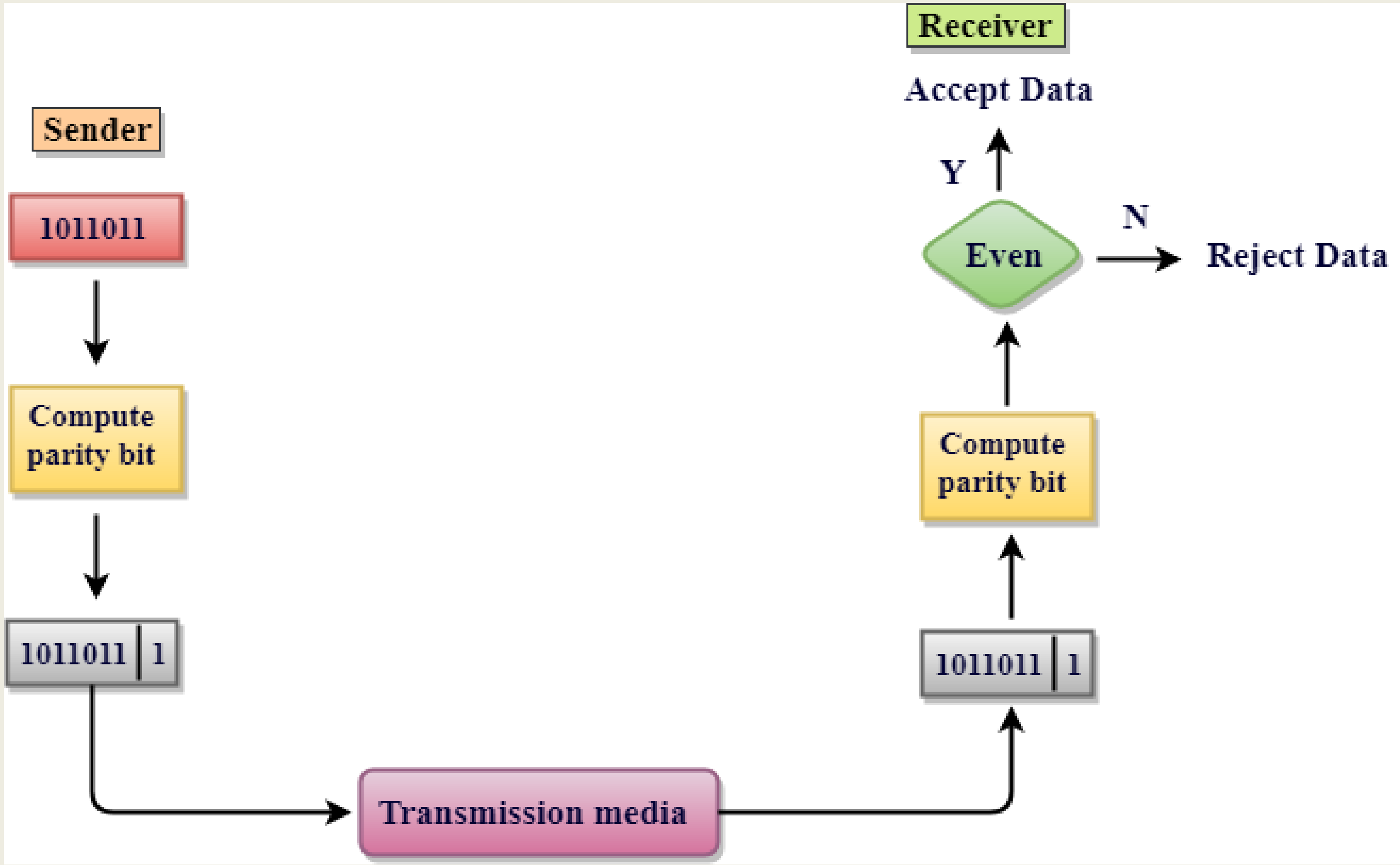
1. **Parity Check**
    1. **Single parity Check**
    2. **Two-Dimensional Parity Check**

2. **Checksum**
3. **Cyclic Redundancy Check (CRC)**

Parity Check

- The most common and least expensive mechanism for error- detection is the parity check. The parity check is done by adding an extra bit, called parity bit to the data to make a number of 1s either even in case of even parity or odd in 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.
- The parity check is suitable for single bit error detection only.

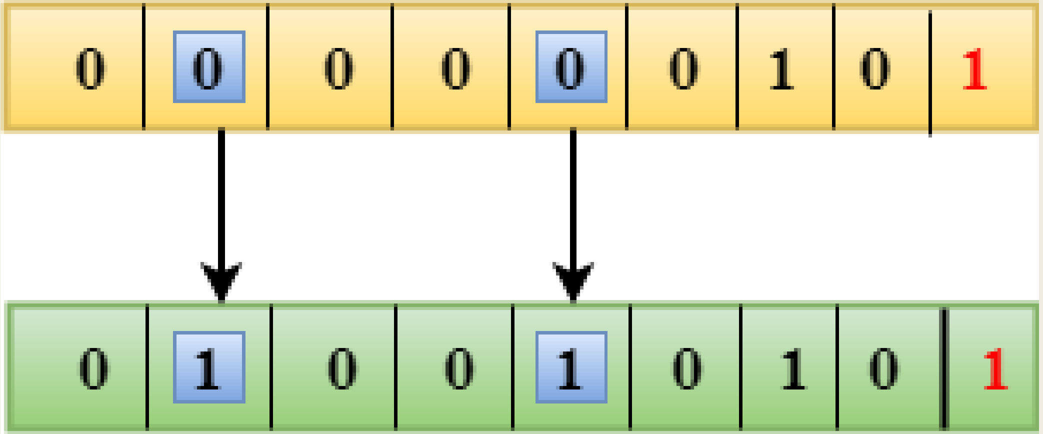
Parity Check



Parity Check

Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors

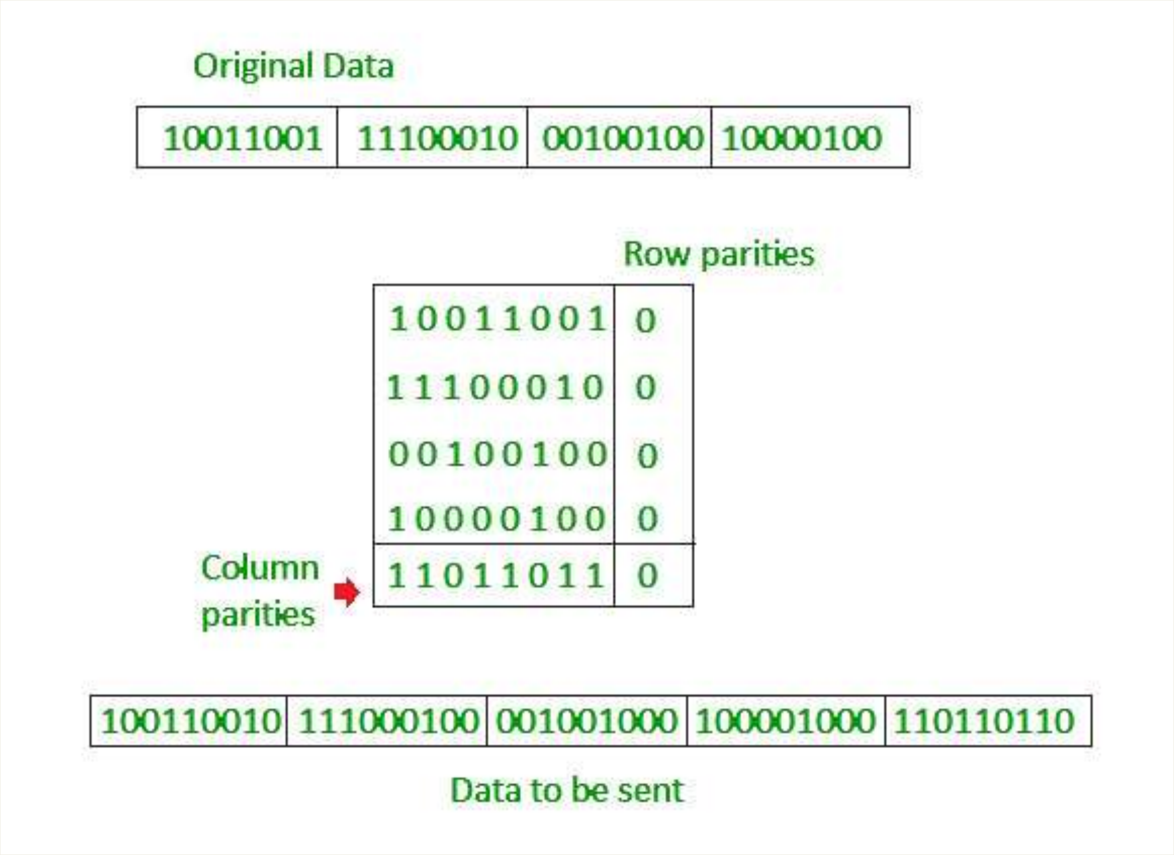


0	0	0	0	0	0	1	0	1
0	1	0	0	1	0	1	0	1

Two-Dimensional Parity Check

- Performance can be improved by using **Two-Dimensional Parity Check** which organizes the data in the form of a table.
- Parity check bits are computed for each row, which is equivalent to the single-parity check.
- In Two-Dimensional Parity check, a block of bits is divided into rows, and the redundant row of bits is added to the whole block.
- At the receiving end, the parity bits are compared with the parity bits computed from the received data.

Two-Dimensional Parity Check



Original Data

10011001	11100010	00100100	10000100

Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column parities

100110010	111000100	001001000	100001000	110110110

Data to be sent

Two-Dimensional Parity Check

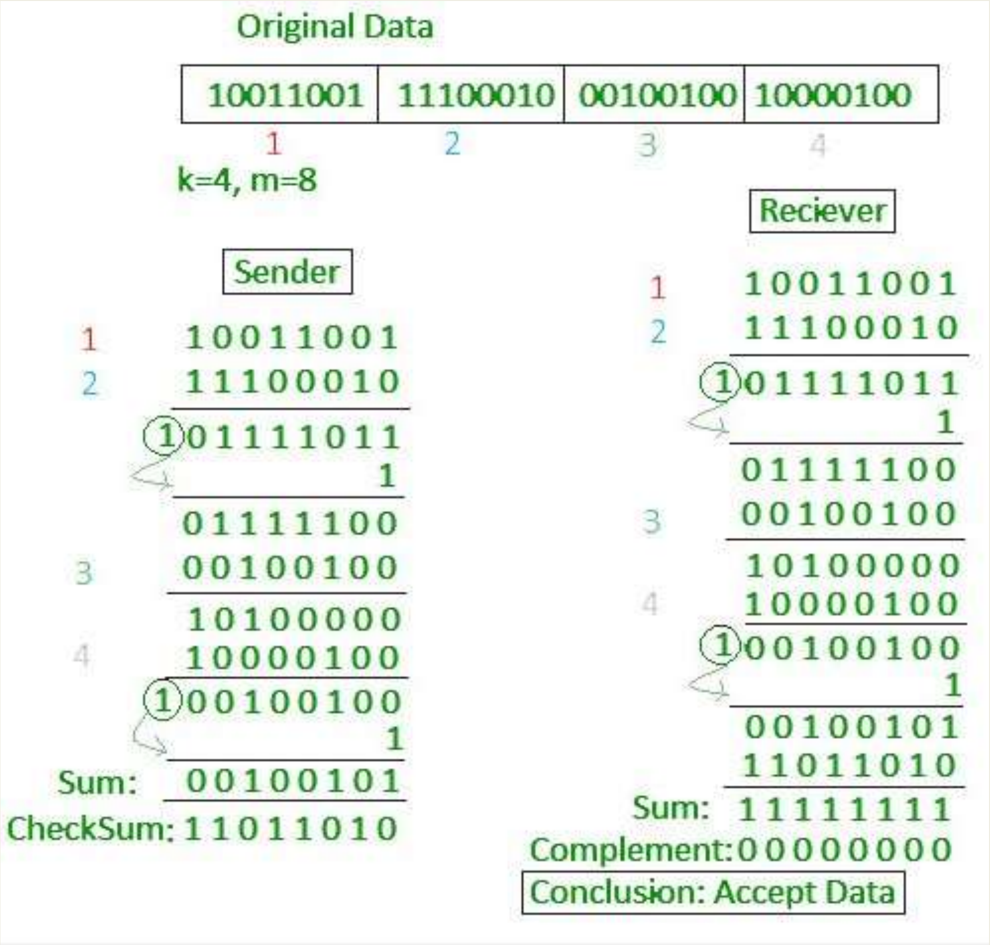
Drawbacks Of 2D Parity Check

- If two bits in one data unit are corrupted and two bits exactly the same position in another data unit are also corrupted, then 2D Parity checker will not be able to detect the error.
- This technique cannot be used to detect the 4-bit errors or more in some cases.

Checksum

- In this error detection scheme, the following procedure is applied
  - Data is divided into fixed sized frames or segments.
  - The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.
  - The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.
  - If the result is zero, the received frames are accepted; otherwise, they are discarded.

Checksum



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
```

Reciever

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

Conclusion: Accept Data

Checksum

Drawbacks Of 2D Parity Check

If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect the error.

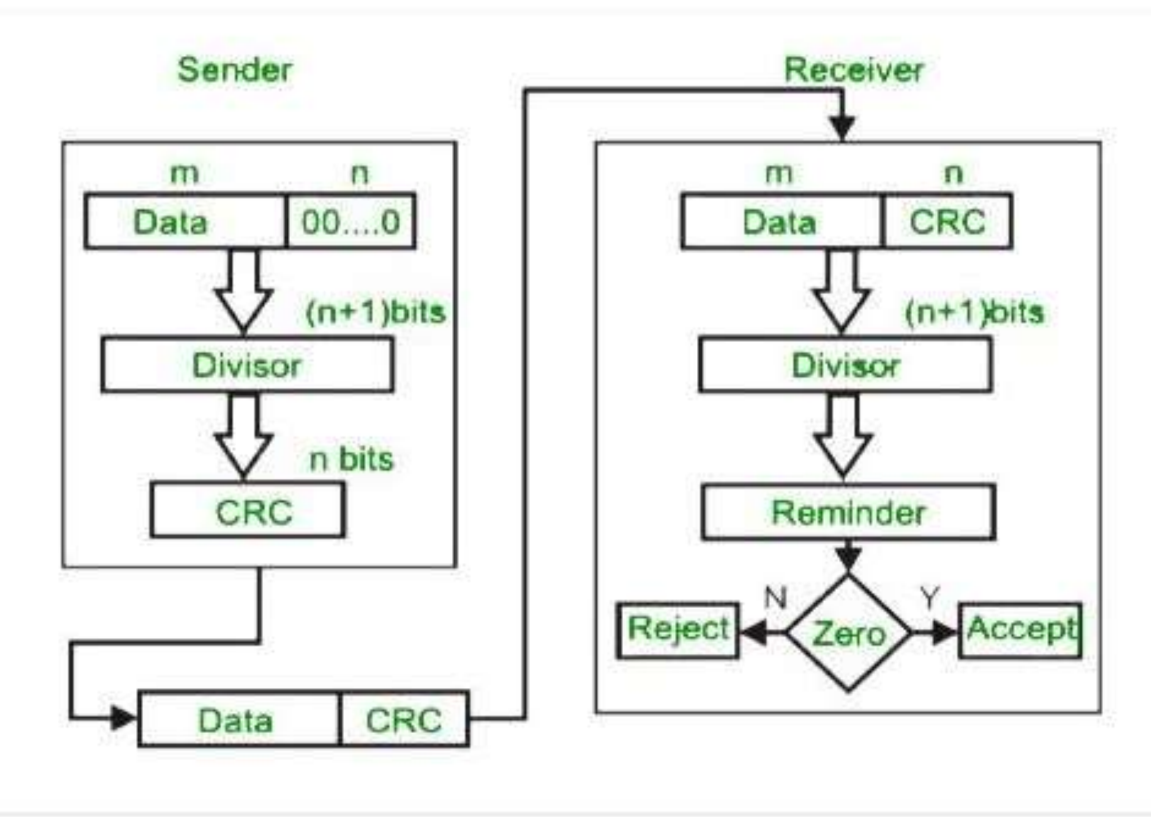
Cyclic Redundancy Check (CRC)

A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data. Unlike checksum scheme, which is based on addition, CRC is based on binary division. In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number. At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted. A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

Cyclic Redundancy Check (CRC)

- Cyclic Redundancy Check (CRC) involves binary division of the data bits being sent by a predetermined divisor agreed upon by the communicating system. The divisor is generated using polynomials.
- Here, the sender performs binary division of the data segment by the divisor. It then appends the remainder called CRC bits to the end of the data segment. This makes the resulting data unit exactly divisible by the divisor.
- The receiver divides the incoming data unit by the divisor. If there is no remainder, the data unit is assumed to be correct and is accepted. Otherwise, it is understood that the data is corrupted and is therefore rejected.

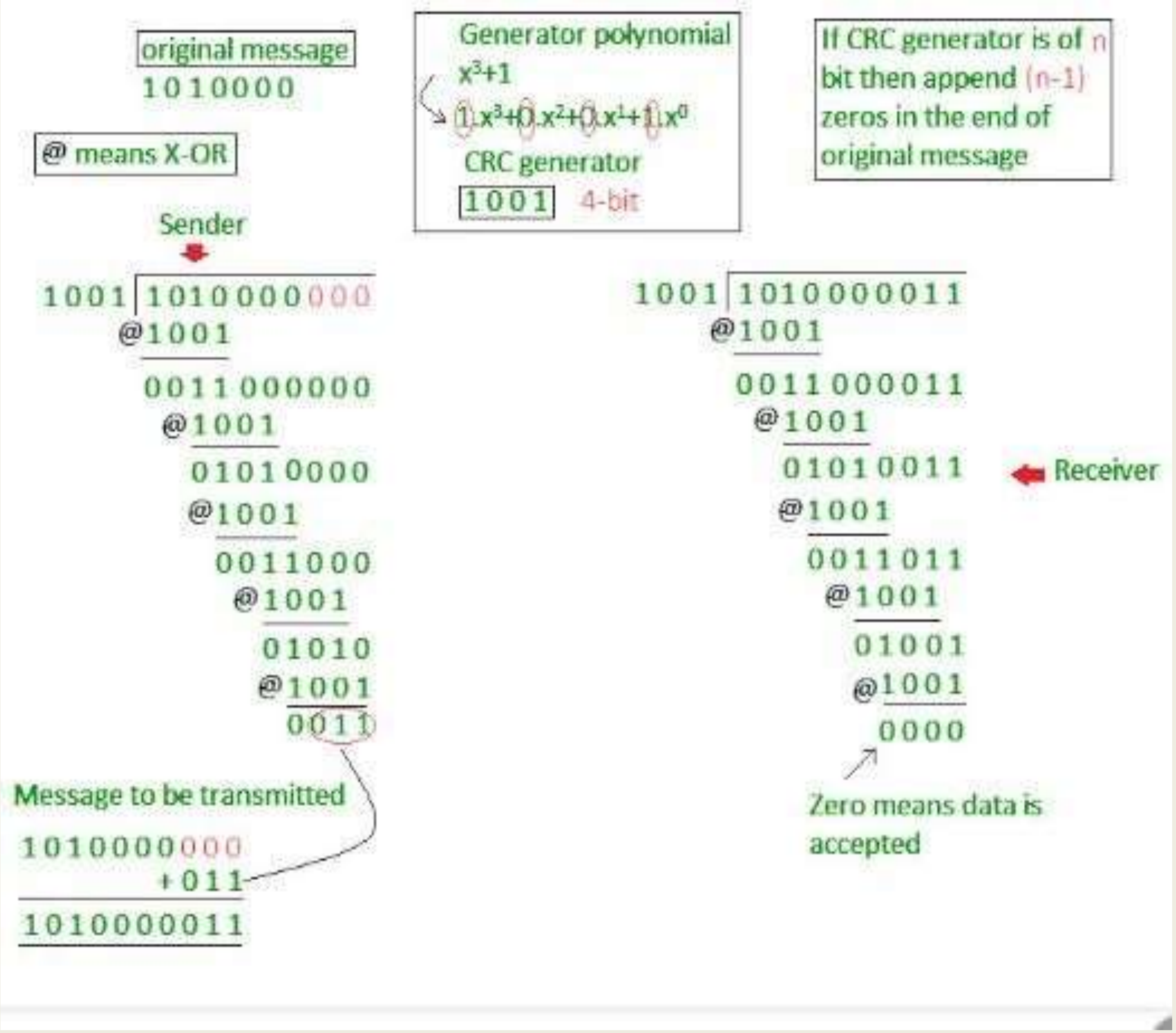
Cyclic Redundancy Check (CRC)



Cyclic Redundancy Check (CRC)

At the sender side, the data unit to be transmitted IS divided by a predetermined divisor (binary number) in order to obtain the remainder. This remainder is called CRC. The CRC has one bit less than the divisor. It means that if CRC is of n bits, divisor is of n+ 1 bit. The sender appends this CRC to the end of data unit such that the resulting data unit becomes exactly divisible by predetermined divisor i.e. remainder becomes zero. At the destination, the incoming data unit i.e. data + CRC is divided by the same number (predetermined binary divisor). If the remainder after division is zero then there is no error in the data unit & receiver accepts it. If remainder after division is not zero, it indicates that the data unit has been damaged in transit and therefore it is rejected. This technique is more powerful than the parity check and checksum error detection.

Cyclic Redundancy Check (CRC)



original message
1010000

@ means X-OR

Generator polynomial
x^3+1
$1.x^3+0.x^2+0.x^1+1.x^0$
CRC generator
1001 4-bit

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

Sender

```
1001 | 1010000000
      @1001
       0011000000
        @1001
         01010000
         @1001
          0011000
           @1001
            01010
           @1001
            0011
```

Message to be transmitted

```
1010000000
     +011
1010000011
```

Receiver

```
1001 | 1010000011
      @1001
       0011000011
        @1001
         01010011
         @1001
          0011011
           @1001
            01001
            @1001
             0000
```

Zero means data is accepted

#Assignment

- Generate CRC code for data 110010101. The divisor is 10101. Also check whether there are errors in the received codeword 1100101011011.
- Generate the CRC code for 1110010101 with the divisor x^3+x^2+1
- Generate the CRC for 11001001 with the divisor x^3+1.Corrupt the left most third bit of the transmitted message and show that the error is detected by the receiver using CRC technique.

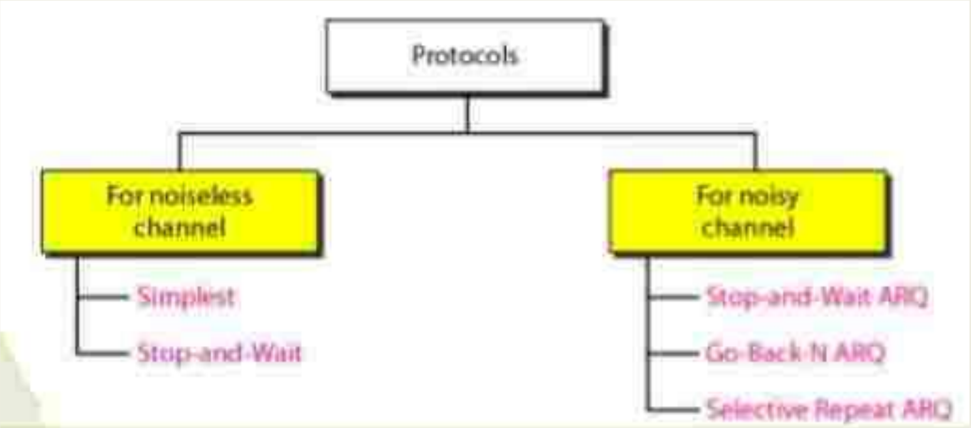
Flow Control Mechanisms:

- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

Flow and Error Control

- **Data link control = flow control + error control**
- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send
before waiting for an acknowledgment.
- Error control in the data link layer is based on automatic repeat request (ARQ), which is the
retransmission of data.
- ACK, NAK(Negative ACK), Piggybacking (ACKs and NAKs in data frames)

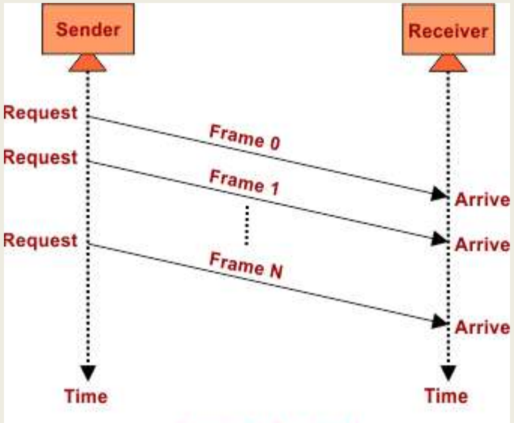
Flow Control Protocol



Simplest Protocol

Simplest Protocol is one that has no flow or error control and it is a unidirectional protocol in which data frames are traveling in only one direction from the sender to the receiver. We assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible. The data link layer of the receiver immediately removes the header from the frame and hands the data packet to its network layer, which can also accept the packet immediately.

The following figure shows an example of communication using this protocol. It is very simple. The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site. Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.



Simplest Protocol

Stop-and-Wait Protocol

Stop and wait is a protocol that is used for reliable data transmission in a noiseless channel. In this protocol, the sender sends a single packet at a time and waits for an acknowledgment (ACK) from the receiver before sending the next packet. This way, the sender can ensure that each packet is received by the receiver and has been successfully processed.

The stop-and-wait protocol is simple and efficient, but it has one major drawback. Because only one packet can be transmitted at a time, the overall data transmission rate is relatively slow. To overcome this limitation, the sliding window protocol was developed. In the sliding window protocol, multiple packets can be transmitted at the same time, allowing for faster data transmission.

Stop-and-Wait Protocol

The flow diagram of the Stop-and-wait protocol in a noiseless channel involves the following steps:

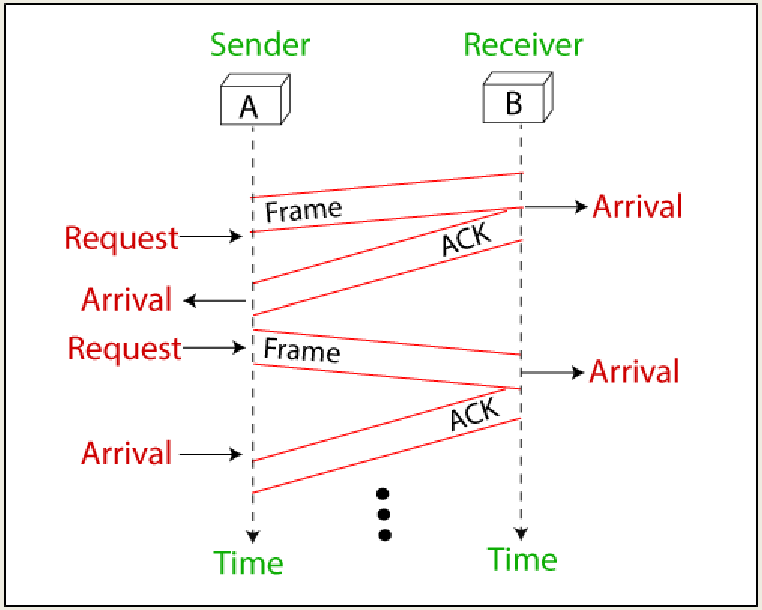
1.The sender transmits a data frame to the receiver.

2.The sender waits for an acknowledgment (ACK) from the receiver.

3.The receiver processes the received data frame.

4.The receiver sends an ACK to the sender to confirm receipt of the data frame.

5.The sender continues to transmit the next data frame, repeating the process from step i.



Stop-and-Wait ARQ Protocol

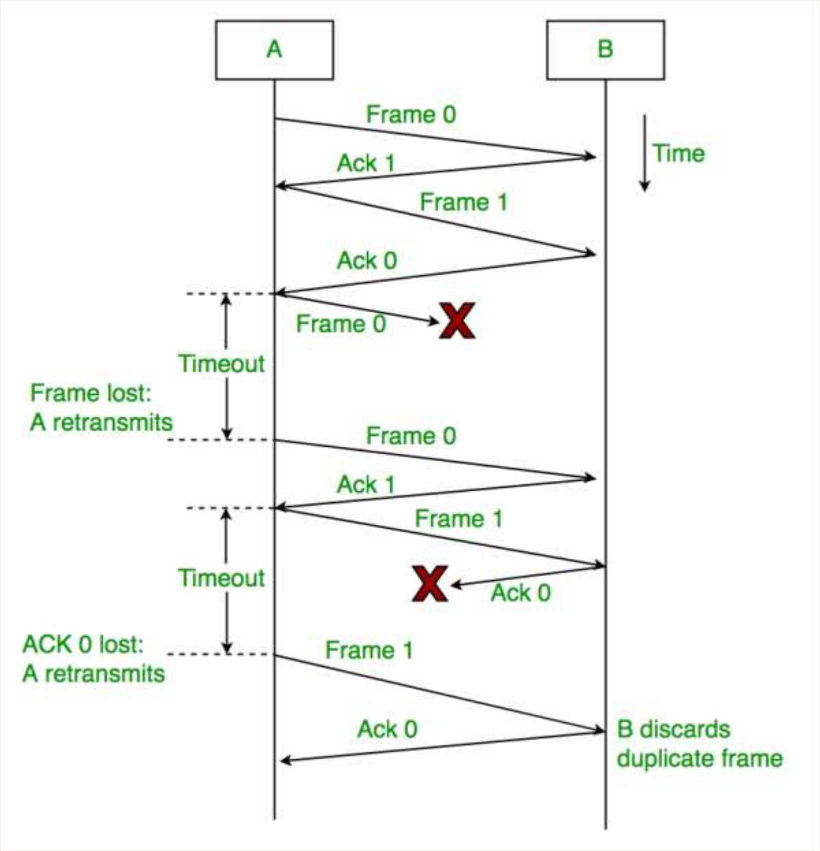
The Stop and Wait protocol is a protocol used for reliable data transmission over a noisy channel. In this protocol, the sender only sends one frame at a time and waits for an acknowledgment (ACK) from the receiver before sending the next frame. This helps to ensure that the receiver receives the data correctly and eliminates the need for retransmission in the case of errors caused by the noisy channel. The sender continuously monitors the channel for errors, and if an error is detected, it waits for the next ACK before resending the frame. This protocol adds error control to the basic unidirectional communication of data frames and ACK frames in the opposite direction.

Stop-and-Wait ARQ Protocol

A data flow diagram in the Stop-and-Wait protocol in a noisy channel can be used to describe the flow of data between the sender and the receiver. This diagram generally includes the following components:

1. ***Sender*****:** The sender sends data frames one at a time, and waits for a response (ACK or NACK) from the receiver before sending the next data frame.
2. ***Receiver*****:** The receiver receives the data frames and processes them. If the frame is received correctly, the receiver sends an ACK signal to the sender. If the frame is not received correctly, the receiver sends a NACK signal to the sender.
3. ***Noisy Channel*****:** The noisy channel is the medium through which the data frames are transmitted from the sender to the receiver. The channel can add noise to the data frames, resulting in errors and corruption of the data.
4. ***Error Detection*****:** The receiver uses error detection techniques such as checksums to detect errors in the received data frames.
5. ***Error Correction*****:** If an error is detected, the receiver sends a NACK signal to the sender, requesting a retransmission of the frame.

Stop-and-Wait ARQ Protocol



GO-BACK-N ARQ Protocol

The Go-Back-N Automatic Repeat Request (ARQ) protocol is a type of error-control protocol used in data communication to ensure reliable delivery of data over a noisy channel. In a noisy channel, the probability of errors in the received packets is high, and hence, there is a need for a mechanism to detect and correct these errors.

The Go-Back-N ARQ protocol is a type of sliding window protocol where the sender transmits a window of packets to the receiver, and the receiver sends back an acknowledgment (ACK) to the sender indicating successful receipt of the packets. In case the sender does not receive an ACK within a specified timeout period, it retransmits the entire window of packets.

GO-BACK-N ARQ Protocol

The flow diagram that illustrates the operation of the Go-Back-N ARQ protocol in a noisy channel:

Sender Side:

1.The sender transmits a window of packets to the receiver, starting with sequence number i and ending with sequence number i + N - 1, where N is the window size.

2.The sender sets a timer for each packet in the window.

3.The sender waits for an acknowledgment (ACK) from the receiver.

Receiver Side:

1.The receiver receives the packets and checks for errors.

2.If a packet is received correctly, the receiver sends an ACK back to the sender with the sequence number of the next expected packet.

3.If a packet is received with errors, the receiver discards the packet and sends a negative acknowledgment (NAK) to the sender with the sequence number of the next expected packet.

GO-BACK-N ARQ Protocol

Sender Side (in case of no ACK received):

1.If the sender does not receive an ACK before the timer for a packet expires, the sender retransmits the entire window of packets starting with the packet whose timer expired.

2.The sender resets the timer for each packet in the window.

3.The sender waits for an ACK from the receiver.

Sender Side (in case of NAK received):

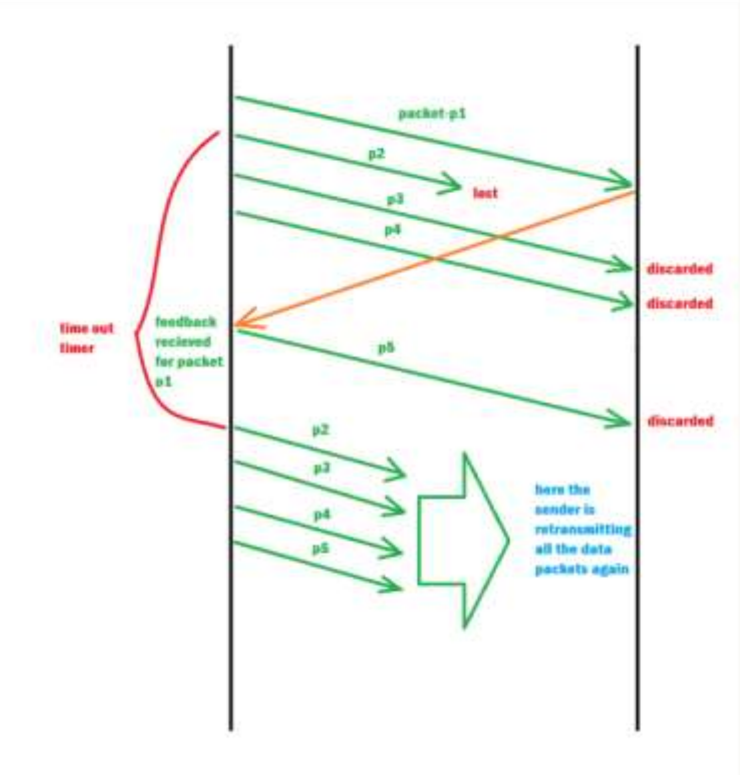
1.If the sender receives a NAK from the receiver, the sender retransmits only the packets that were not correctly received by the receiver.

2.The sender resets the timer for each packet that was retransmitted.

3.The sender waits for an ACK from the receiver.

The above steps are repeated until all packets have been successfully received by the receiver. The Go-Back-N ARQ protocol provides a reliable mechanism for transmitting data over a noisy channel while minimizing the number of retransmissions required.

GO-BACK-N ARQ Protocol



Selective Repeat ARQ Protocol

The Selective Repeat ARQ protocol is a type of error-control protocol used in data communication to ensure reliable delivery of data over a noisy channel. Unlike the Go-Back-N ARQ protocol which retransmits the entire window of packets, the Selective Repeat ARQ protocol retransmits only the packets that were not correctly received.

In the Selective Repeat ARQ protocol, the sender transmits a window of packets to the receiver, and the receiver sends back an acknowledgment (ACK) to the sender indicating successful receipt of the packets. If the receiver detects an error in a packet, it sends a negative acknowledgment (NAK) to the sender requesting retransmission of that packet. In the Selective Repeat ARQ protocol, the sender maintains a timer for each packet in the window. If the sender does not receive an ACK for a packet before its timer expires, the sender retransmits only that packet.

On the receiver side, if a packet is received correctly, the receiver sends back an ACK with the sequence number of the next expected packet. However, if a packet is received with errors, the receiver discards the packet and sends back an NAK with the sequence number of the packet that needs to be retransmitted.

Unlike Go-Back-N ARQ, in Selective Repeat ARQ, the receiver buffer is maintained for all packets that are not in sequence. When a packet with a sequence number different from the expected sequence number arrives at the receiver, it is buffered, and the receiver sends an ACK for the last in-order packet it has received. If a packet with a sequence number that the receiver has already buffered arrives, it is discarded, and the receiver sends an ACK for the last in-order packet it has received.

Selective Repeat ARQ Protocol

The flow diagram that illustrates the operation of the Selective Repeat ARQ protocol in a noisy channel:

Sender Side:

1.The sender transmits a window of packets to the receiver, starting with sequence number i and ending with sequence number i + N - 1, where N is the window size.

2.The sender sets a timer for each packet in the window.

3.The sender waits for an acknowledgment (ACK) from the receiver.

Receiver Side:

1.The receiver receives the packets and checks for errors.

2.If a packet is received correctly and is in order, the receiver sends an ACK back to the sender with the sequence number of the next expected packet.

3.If a packet is received with errors or is out of order, the receiver discards the packet and sends a negative acknowledgment (NAK) to the sender with the sequence number of the packet that needs to be retransmitted.

4.The receiver buffers out-of-order packets and sends an ACK for the last in-order packet it has received.

Selective Repeat ARQ Protocol

Sender Side (in case of no ACK received):

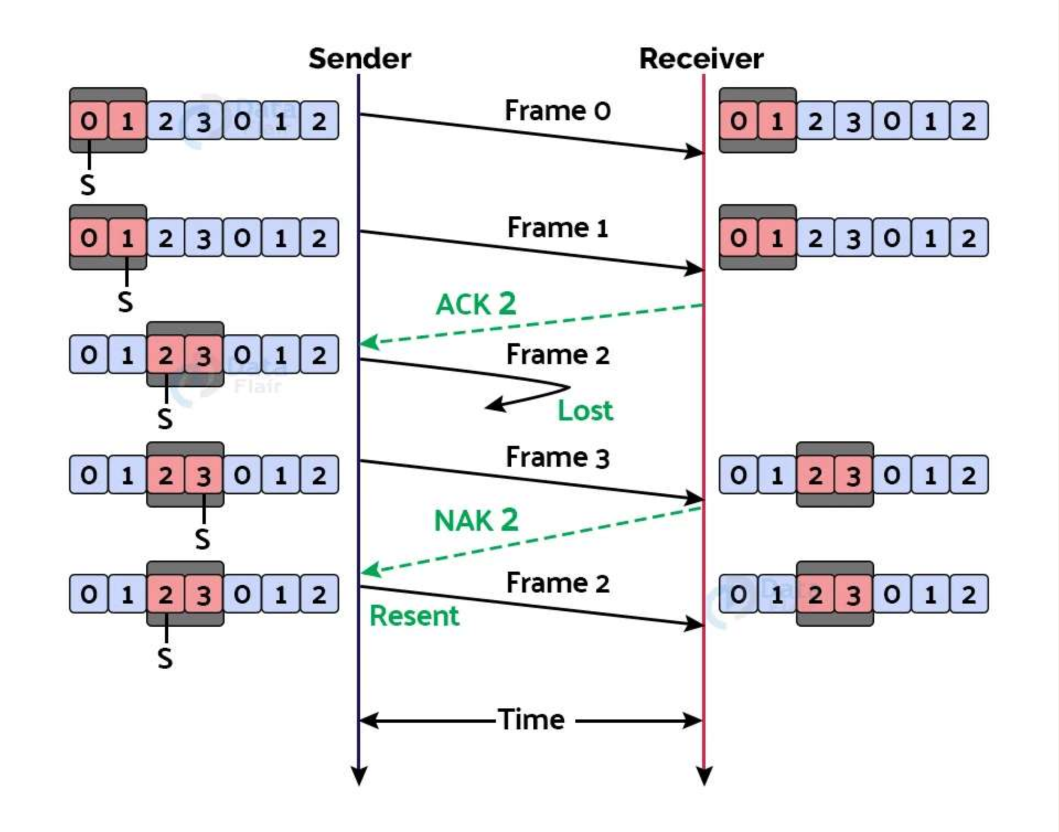
1. If the sender does not receive an ACK before the timer for a packet expires, the sender retransmits only that packet.
2. The sender resets the timer for the retransmitted packet.
3. The sender waits for an ACK from the receiver.

Sender Side (in case of NAK received):

1. If the sender receives a NAK from the receiver, the sender retransmits only the packets that were not correctly received.
2. The sender resets the timer for each packet that was retransmitted.
3. The sender waits for an ACK from the receiver.

The above steps are repeated until all packets have been successfully received by the receiver. The Selective Repeat ARQ protocol provides a reliable mechanism for transmitting data over a noisy channel while minimizing the number of retransmissions required. It retransmits only the packets that were not correctly received, and buffers out-of-order packets to reduce the number of retransmissions required.

Selective Repeat ARQ Protocol



Piggybacking

Piggybacking is delaying outgoing acknowledgment and attaching it to the next data packet.

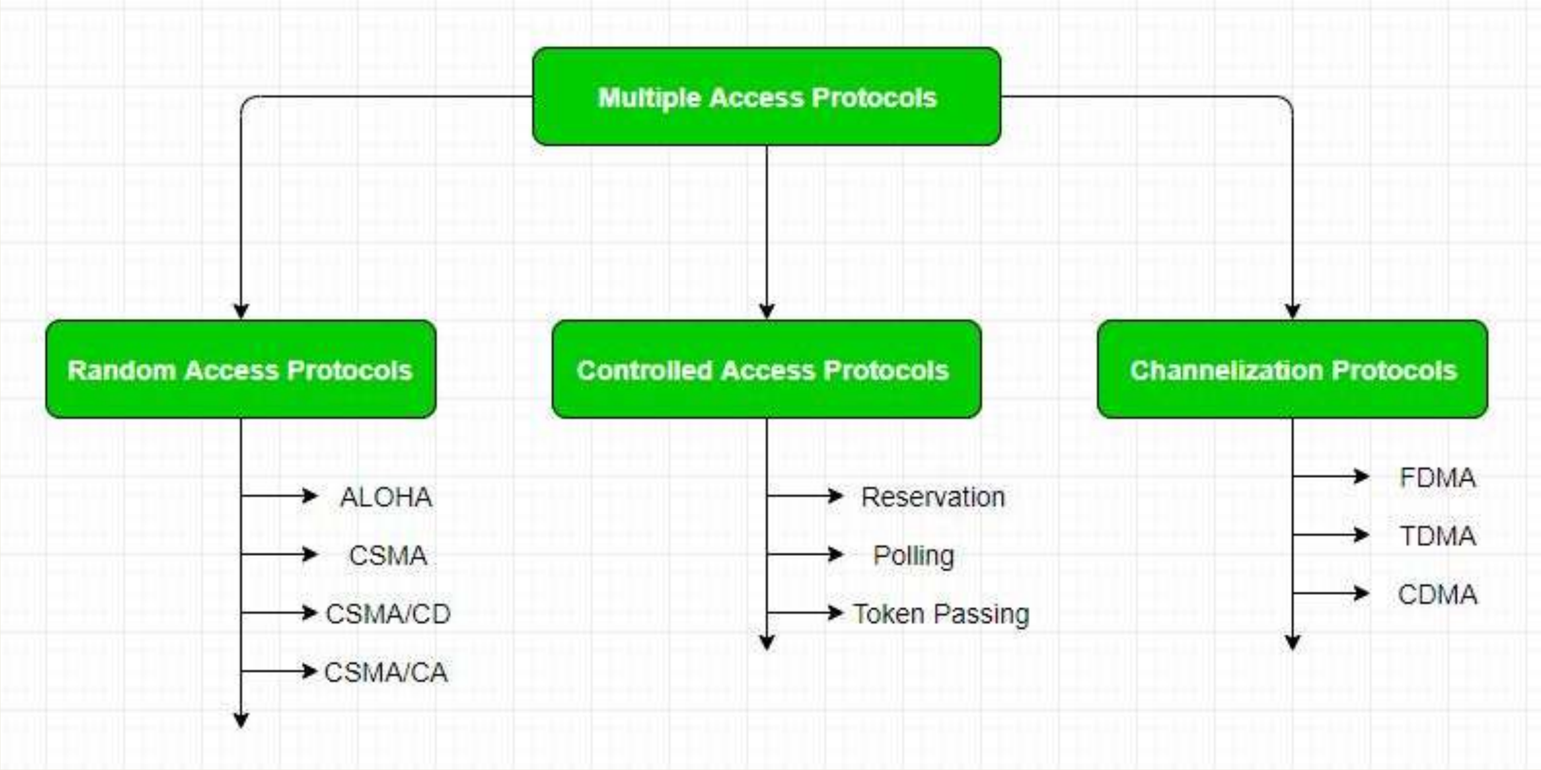
When a data frame arrives, the receiver waits and does not immediately send the control frame (acknowledgment) back. The receiver waits until its network layer moves to the next data packet. Acknowledgment is associated with this outgoing data frame. Thus, the acknowledgment travels along with the next data frame. This technique in which the outgoing acknowledgment is delayed temporarily is called Piggybacking.

Multiple Access Protocol (Channel Allocation Techniques):

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. 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(confusion) 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.

Multiple Access Protocol (Channel Allocation Techniques):

Thus, protocols are required for sharing data on non dedicated channels. Multiple access protocols can be subdivided further as –



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. **CSMA**
3. **CSMA/CD**
4. **CSMA/CA**

ALOHA

ALOHA is the earliest random-access method developed for wireless LAN but can be used on any shared medium. In this, multiple stations can transmit data at the same time and can hence lead to collision. The data from the two stations collide.

ALOHA

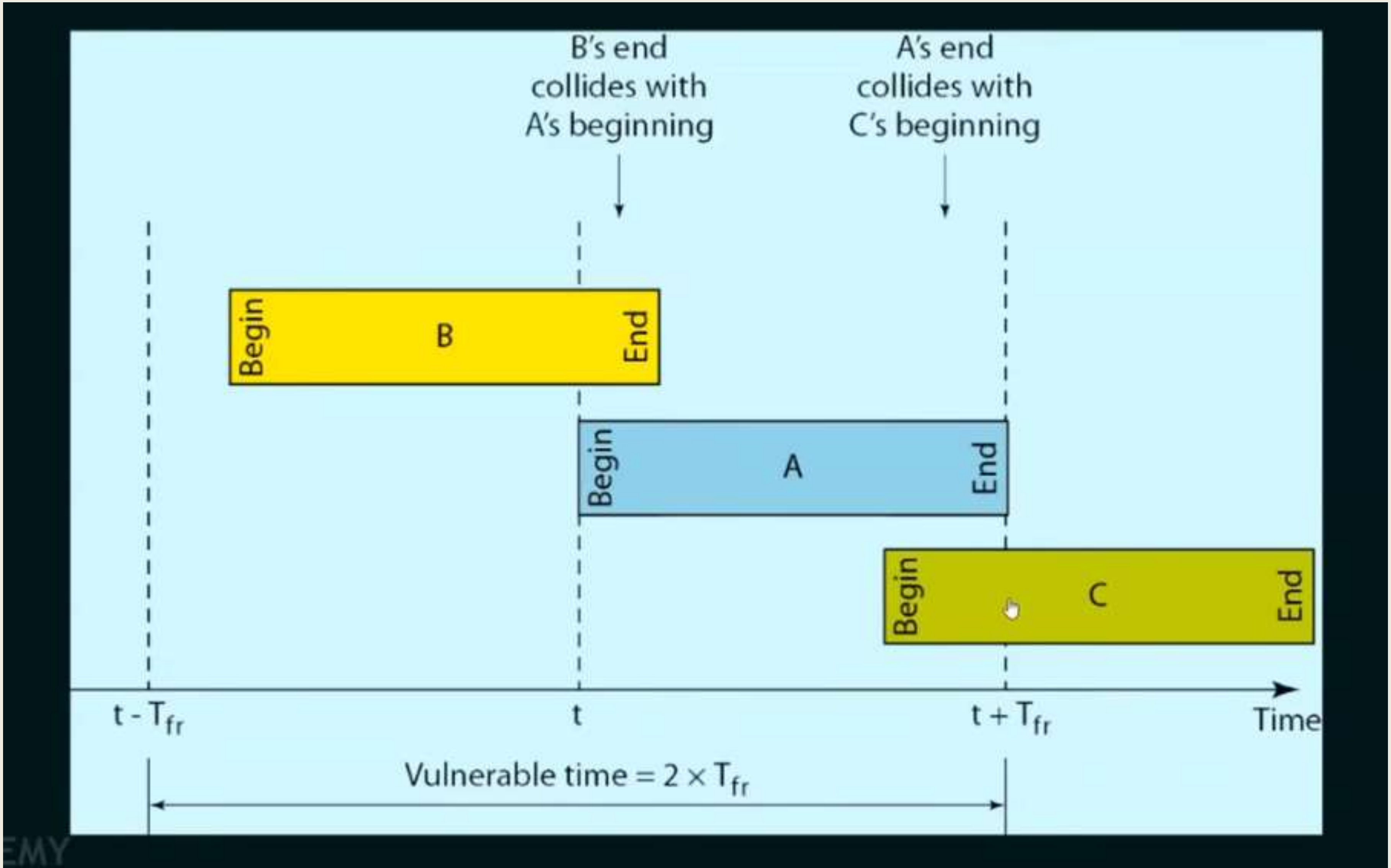
Pure ALOHA: When a station sends data it waits for an acknowledgement. If the acknowledgement doesn’t come within the allotted time, then the station waits for a random amount of time called back-off time (Tb) and re-sends the data. Since different stations wait for different amount of time, the probability of further collision decreases. Even if one bit of a frame coexists on the channel with one bit from another frame, there is a collision and both will be destroyed.

Maximum Throughput=18.4%or 0.184

Meaning only **18.4% of the total channel capacity** is effectively used.

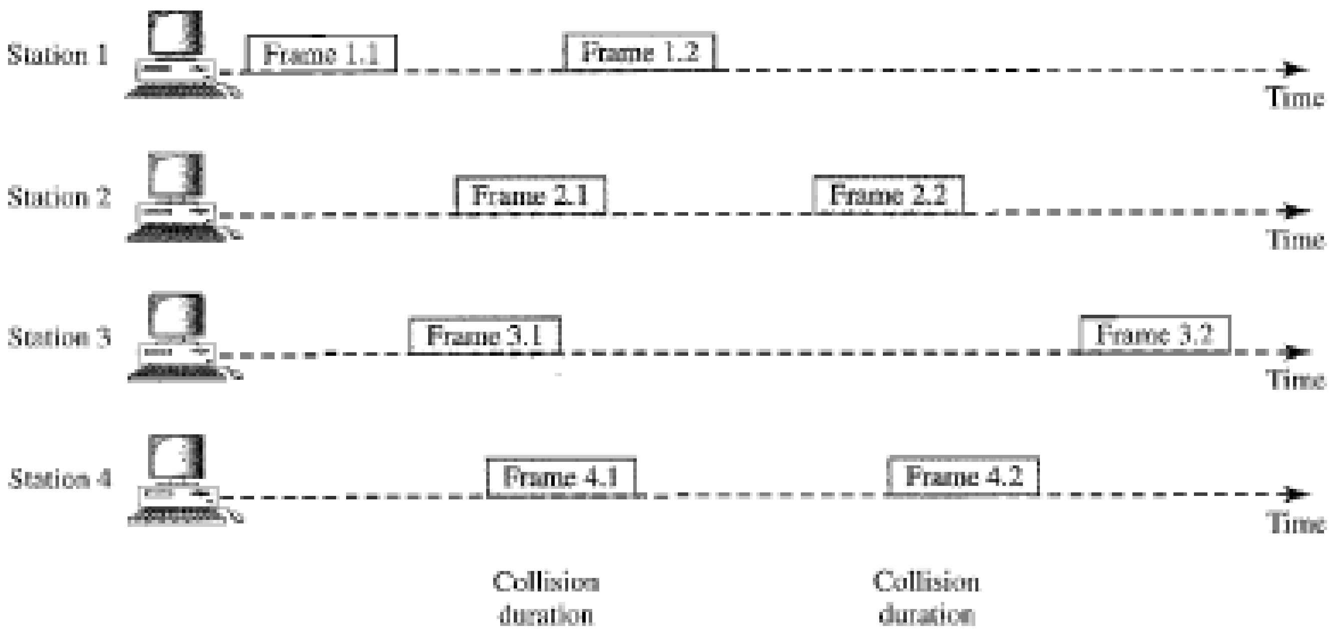
Most of the time, frames **collide or get lost**.

ALOHA


B's end collides with A's beginning
A's end collides with C's beginning
Begin
B
End
Begin
A
End
Begin
C
End
t - T_fr
t
t + T_fr
Time
Vulnerable time = 2 × T_fr
EMY

ALOHA

Figure 12.3 *Frames in a pure ALOHA network*



ALOHA

Slotted Aloha:

Slotted ALOHA is a time-based protocol where time is divided into equal slots, and stations can only send data at the beginning of these time slots. This reduces the chance of collisions compared to Pure ALOHA.

How it Works:

1.Time is divided into fixed-size slots equal to the transmission time of one frame.

2.A station must wait for the next time slot to begin transmission.

3.If no other station transmits in the same slot, the transmission succeeds.

4.If two or more stations transmit in the same slot, a collision occurs, and all involved stations retransmit after a random delay.

ALOHA

Slotted Aloha:

Why It's Better than Pure ALOHA:

•Collisions can only happen at the beginning of a slot.

•The vulnerable period is reduced from 2 × frame time (in Pure ALOHA) to 1 × frame time in Slotted ALOHA.

Efficiency (Throughput):

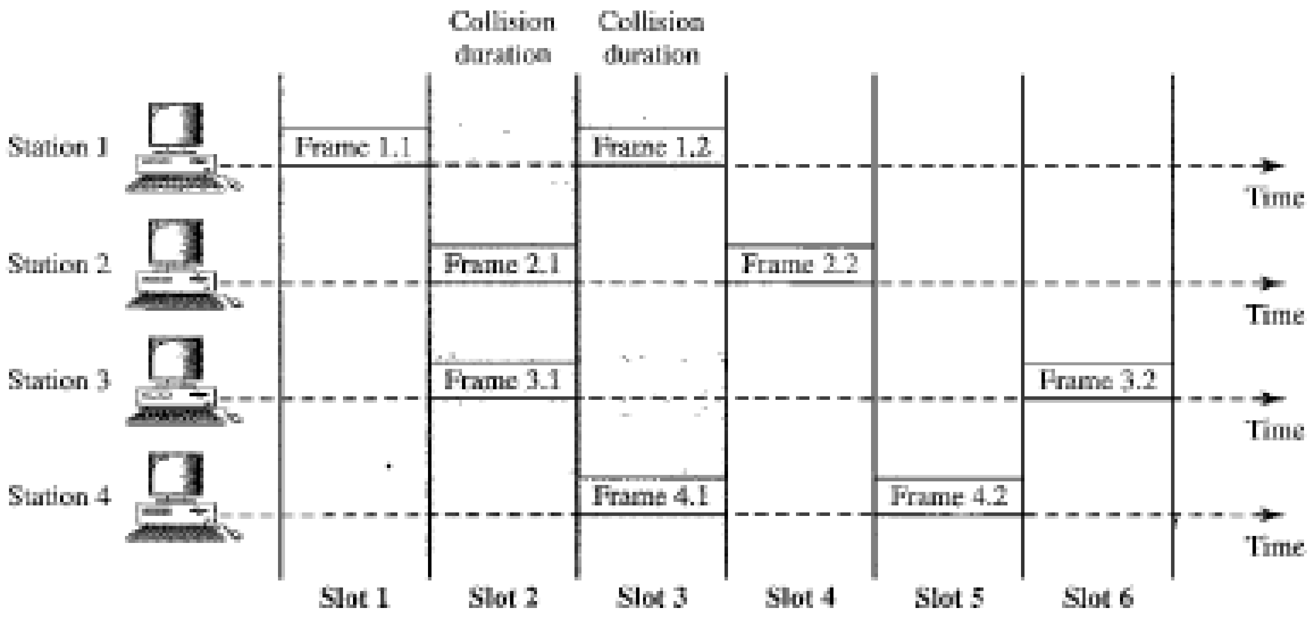
Max Throughput=36.8% or 0.368

This means about **36.8% of the time slots** carry successful transmissions.

ALOHA

Slotted Aloha:

Figure 12.6 *Frames in a slotted ALOHA network*



CSMA

Carrier Sense Multiple Access (CSMA) is a method used in computer networks to manage how devices share a communication channel to transfer the data between two devices. In this protocol, each device first sense the channel before sending the data. If the channel is busy, the device waits until it is free. CSMA is commonly used in technologies like Ethernet and Wi-Fi.

How CSMA Works (Steps):
Carrier Sense:
Check if the communication channel is **idle**.

If Idle:
→ Start transmitting the data.

If Busy:
→ Wait for the channel to become free.

Collision Possibility:
Even if the channel is free when checked, two stations might transmit **at the same time** (nearly), causing a **collision**.

CSMA

CSMA access modes-

❑ **1-Persistent CSMA**

- The station listens to the channel.
- If idle, it immediately transmits (with 100% probability).
- If busy, it keeps sensing until the channel is free, then sends immediately.

Problem:

If multiple stations are waiting, they all transmit as soon as the channel is free, causing collisions.

CSMA

CSMA access modes-

❑ **Non-Persistent CSMA**

- The station senses the channel.
- **If busy**, it **waits a random time**, then checks again.
- **If idle**, it transmits.

Advantage:

Reduces chances of collision by **not crowding** the channel right after it becomes free.

CSMA

CSMA access modes-

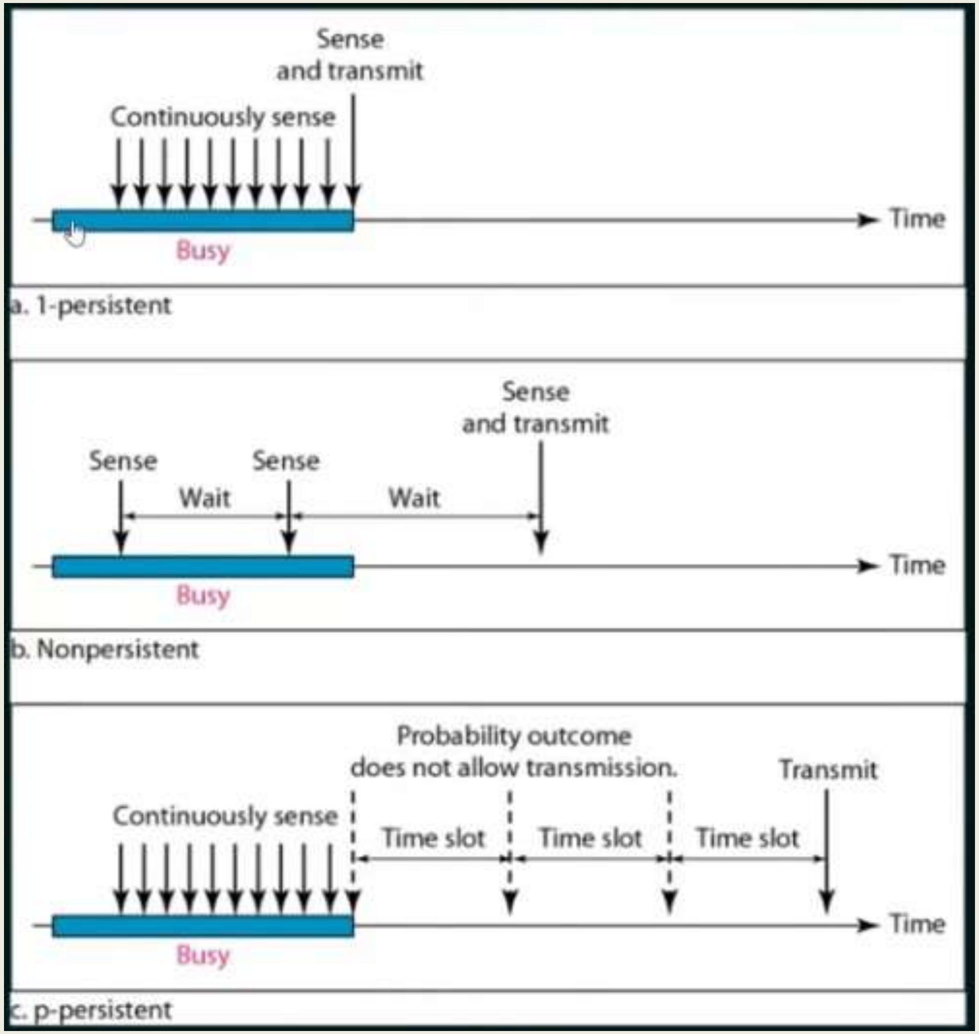
❑ P-Persistent CSMA

- It senses the channel.
- If the channel is **busy**:
  - Wait until it becomes **idle**.
- When the channel becomes **idle**, and a new slot starts:
  - Transmit the frame with **probability p**
  - Or wait for the **next time slot** with **probability 1 - p**
- If it waits, repeat step 3 in the next time slot.

Advantage:

Choosing the right value of p helps balance between **collision rate** and **delay**.

Carrier sense multiple access with collision detection (CSMA):



Carrier sense multiple access with collision detection (CSMA/CD):

1. **Carrier Sense (CS)**
   Before sending data, a device **listens** to the network to check if the channel is **idle** (no one else is transmitting).
2. **Multiple Access (MA)**
   If the channel is idle, multiple devices can access the same medium, but only **one** can transmit at a time.
3. **Transmit Data**
   If the channel is free, the device **sends** the data.
4. **Collision Detection (CD)**
   While transmitting, the device continues to monitor the channel.
   If it **detects a collision** (two devices transmitting at the same time), it immediately stops.
5. **Jam Signal**
   After detecting a collision, the device sends a **jam signal** to inform all devices that a collision occurred.
6. **Backoff Algorithm**
   Each device waits a **random amount of time** (using Binary Exponential Backoff) before attempting to retransmit.

Carrier sense multiple access with collision detection (CSMA/CD):

[Device A] ----> Checks line ---> Free ---> Transmits ---> Success (if no collision)

[Device A] ----> Checks line ---> Free ---> Transmits

[Device B] ----> Checks line ---> Also Free ---> Transmits (at same time)

>>> Collision occurs <<<

Both devices detect collision ---> Send jam signal ---> Wait (random backoff) ---> Try again

Collision occurs

↓

[Device A] and [Device B] both detect it

↓

[Device A] and [Device B] send jam signal

↓

All devices hear the jam signal

↓

A and B wait for random backoff time (Binary Exponential Backoff)

↓

Try retransmission later

Carrier sense multiple access with collision avoidance (CSMA/CA):

CSMA/CA is a network access method used to avoid collisions in wireless networks, especially Wi-Fi (IEEE 802.11). It is the wireless counterpart of CSMA/CD, but since collision detection is difficult in wireless, CSMA/CA tries to prevent collisions before they happen.

How it works?
1. Carrier Sense
A device **listens to the channel** to check if it's **idle** (no one is transmitting).

2. Wait for IFS (Interframe Space)
Even if the channel is idle, the device waits a small amount of time (called **Interframe Space**) before proceeding.

3. Random Backoff Timer
A **random backoff time** is selected to further avoid collision.
The timer **counts down only when the channel is idle**.

4. Transmit Data
When the backoff timer reaches 0, the device transmits the data.

5. Acknowledgment (ACK)
Receiver sends back an **ACK** if data is received successfully.
If no ACK is received (implying possible collision or error), the sender tries again.

Carrier sense multiple access with collision avoidance (CSMA/CA):

[Sender]

↓ (channel idle?)

[Wait IFS]

↓

[Random Backoff countdown]

↓

[Transmit Data]

↓

[Receiver sends ACK]

Controlled Access Protocols

In controlled access, the right to transmit is regulated, so collisions are avoided. Only one device transmits at a time.

1. Reservation

Devices **reserve time slots** in advance before transmitting.
Time is divided into **slots**.
A device sends a **reservation request** in a special control slot.
If the reservation is accepted, it is assigned a slot to transmit.

2. Polling

A **central controller (master)** asks each device (slave) one-by-one if it wants to send data.
Master → "Device 1, do you want to send?"
If yes, device sends. If no, move to next device.

3. Token Passing

A special message called a **token** circulates in the network.
Only the device **holding the token** can transmit.
After transmitting, it **passes the token** to the next device.

Channelization Protocol

Channelization is a **multiple access method** that divides the channel into **separate logical parts**, so **many users can transmit at the same time** using different channels.

1. FDMA (Frequency Division Multiple Access)
Idea: Divide bandwidth into **separate frequency bands**.
Each user gets a **dedicated frequency channel**.
Example:
User A → 890–891 MHz
User B → 891–892 MHz

2. TDMA (Time Division Multiple Access)
Idea: Divide time into **slots**.
Each user transmits in **their own time slot** on the same frequency.
Example:
Time Slot 1 → User A
Time Slot 2 → User B

3. CDMA (Code Division Multiple Access)
Idea: All users transmit **at the same time & frequency**, but use **unique codes** to separate data.
Example:
User A → Code 1
User B → Code 2
Same time & frequency

IEEE Standards

- The Institute of Electrical and Electronic Engineers (IEEE) has developed standard for LANs. These standards are collectively known as IEEE802 or Project 802.
- The IEEE 802 divides data link layer into two sublayers i.e. LLC (Logical Link Control) and MAC (Media Access Control)

IEEE Standards

1. Logical Link Control(LLC)
    - In IEEE 802, flow control, error control and part of framing duties are performed both by LLC & MAC sublayer.
    - It provides one single data link control protocol for all IEEE LANs.
    - This single LLC protocol can provide interconnectivity between different LANs.

IEEE Standards

1. Medium Access Control(MAC)
   - MAC sublayer of IEEE 802 defines the specific access methods for each LANs.
   - For example, it defines CSMA/CD as the media access method for Ethernet LANs and token passing method for Token Ring and Token Bus LANs

Various IEEE Standards

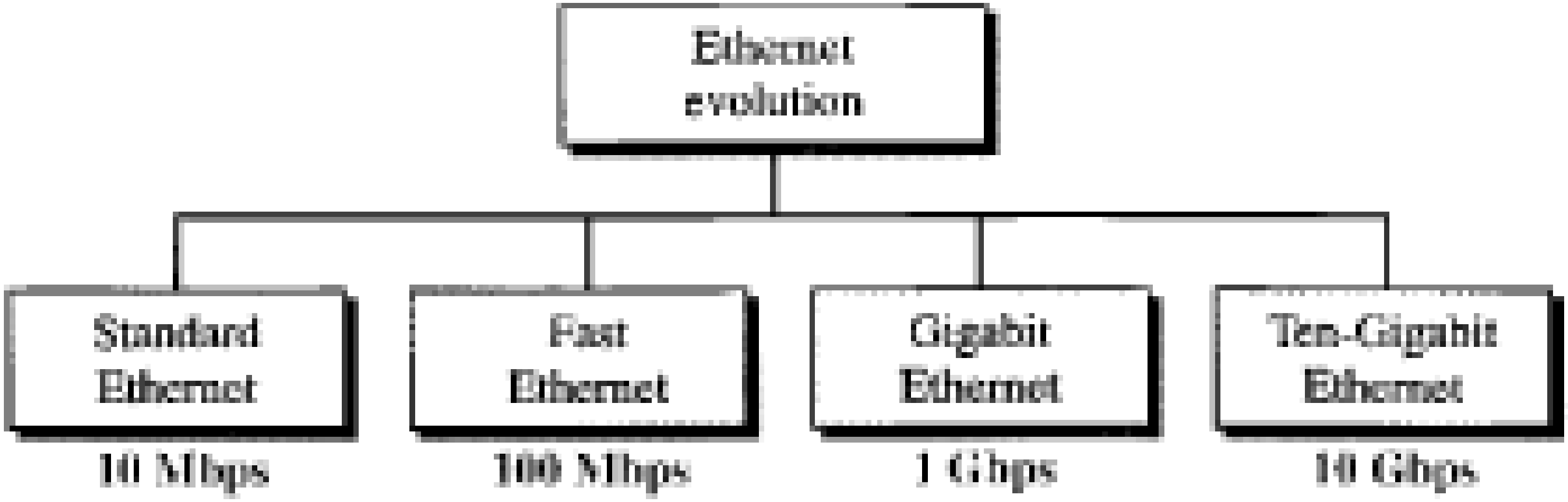
IEEE 802	LAN/MAN
IEEE 802.1	Standards for LAN/MAN bridging and management and remote media access control (MAC) bridging
IEEE 802.2	Standards for Logical Link Control (LLC) standards for connectivity
IEEE 802.3	Ethernet Standards for Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
IEEE 802.4	Standard for Token Passing Bus Access
IEEE 802.5	Standard for Token Ring Access
IEEE 802.6	Standard for information exchange between systems

IEEE 802.3 Ethernet for CSMA/CD

- IEEE 802.3 is a set of standards and protocols that define Ethernet-based networks. Ethernet technologies are primarily used in LANs, though they can also be used in MANs and even WANs.
- Ethernet is a multi-access network in which set of nodes share a common link.
- CSMA/CD as the media access method for Ethernet LANs.
- Whenever a station wants to transmit , it sense the carrier to determine the channel is idle or busy.
- The stations can detect the collision i.e. whenever two or more stations transmit simultaneously and their frames collide; the stations abort their transmission.

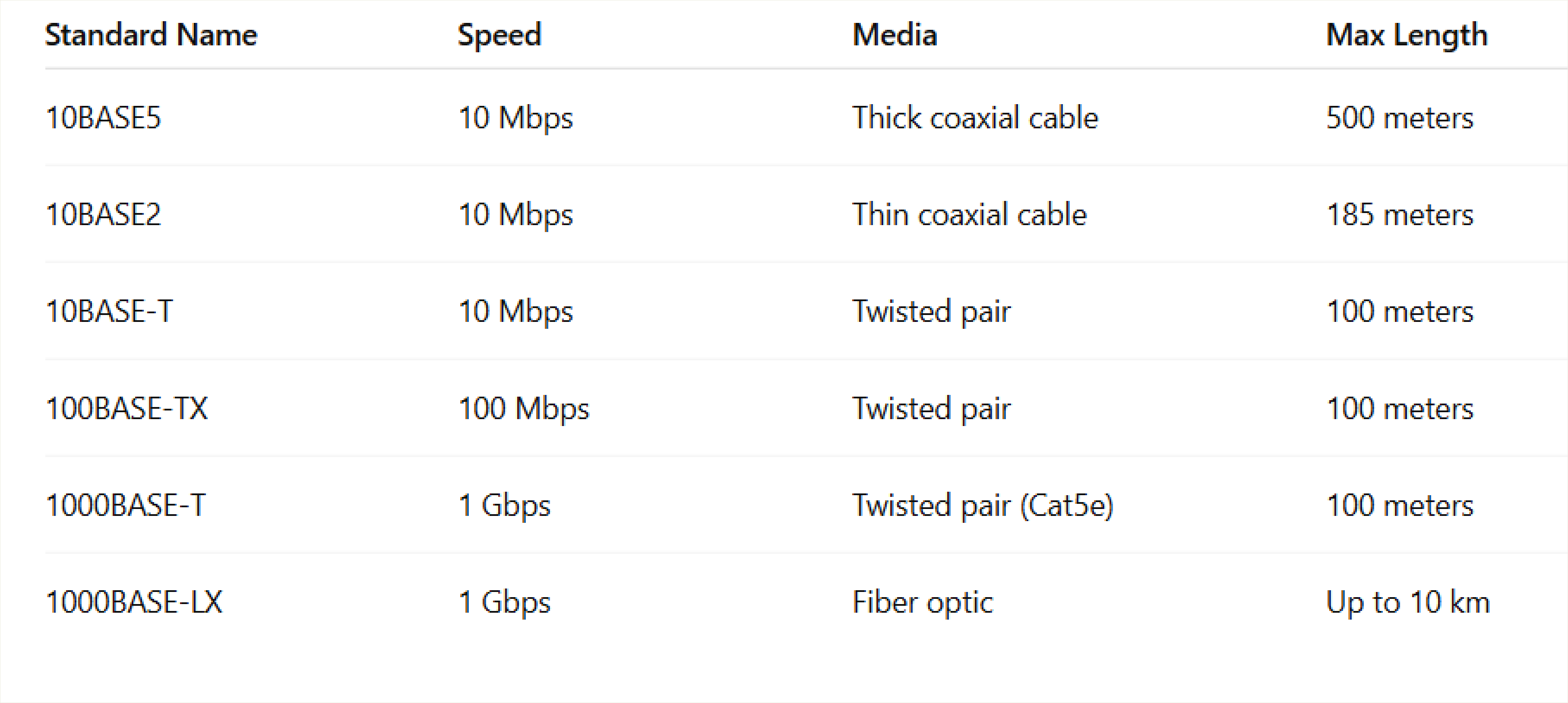
IEEE 802.3 Ethernet for CSMA/CD

Ethernet evolution through four generations



IEEE 802.3 Ethernet for CSMA/CD

Ethernet Standards (under IEEE 802.3):



Standard Name	Speed	Media	Max Length
10BASE5	10 Mbps	Thick coaxial cable	500 meters
10BASE2	10 Mbps	Thin coaxial cable	185 meters
10BASE-T	10 Mbps	Twisted pair	100 meters
100BASE-TX	100 Mbps	Twisted pair	100 meters
1000BASE-T	1 Gbps	Twisted pair (Cat5e)	100 meters
1000BASE-LX	1 Gbps	Fiber optic	Up to 10 km

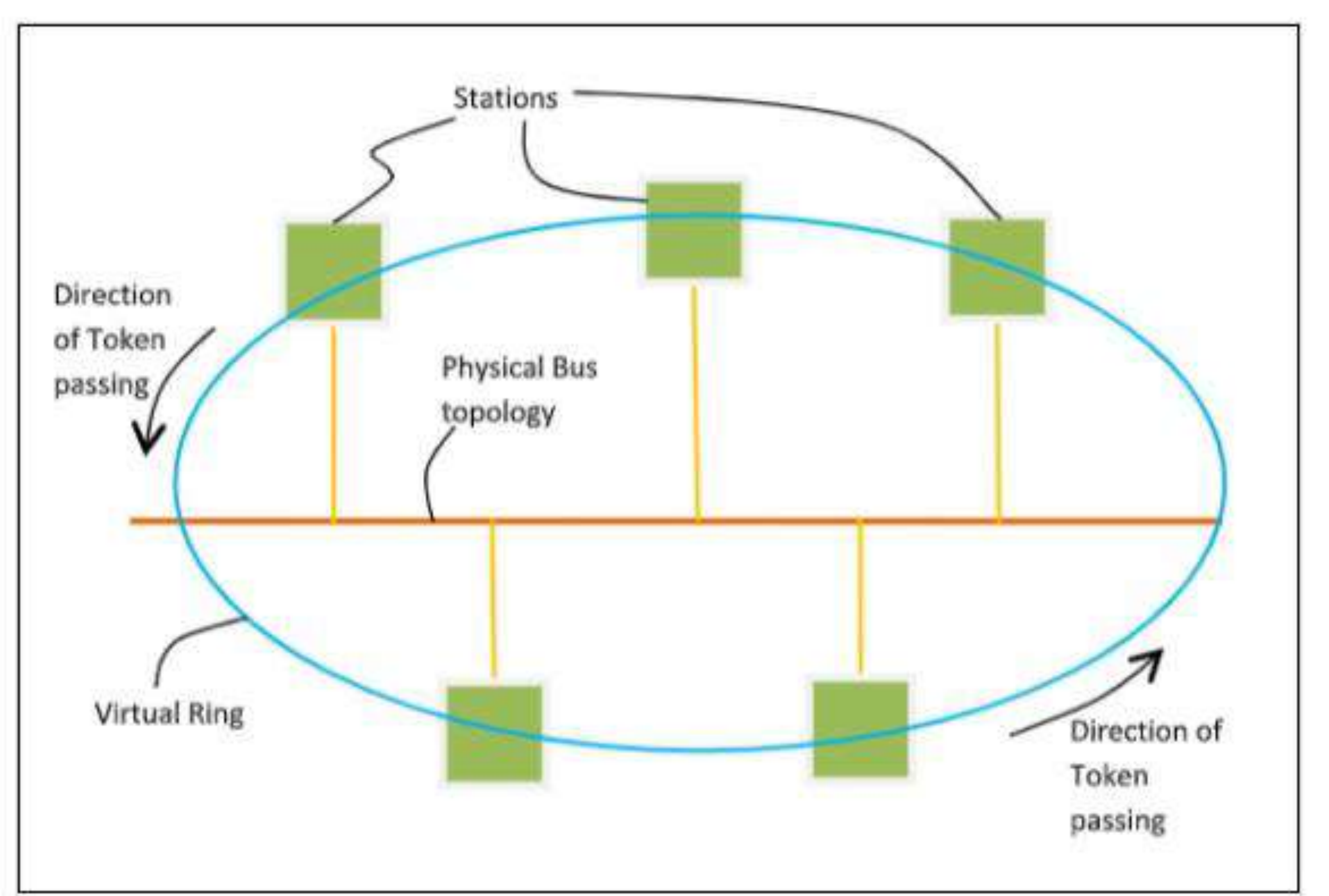
Token Bus / (IEEE 802.4)

Token Bus (IEEE 802.4) is a standard for implementing token ring over virtual ring in LANs. The physical media has a bus or a tree topology and uses coaxial cables. A virtual ring is created with the nodes/stations and the token is passed from one node to the next in a sequence along this virtual ring. Each node knows the address of its preceding station and its succeeding station. A station can only transmit data when it has the token. The working principle of token bus is similar to Token Ring.

Token Passing Mechanism in Token Bus

A token is a small message that circulates among the stations of a computer network providing permission to the stations for transmission. If a station has data to transmit when it receives a token, it sends the data and then passes the token to the next station; otherwise, it simply passes the token to the next station.

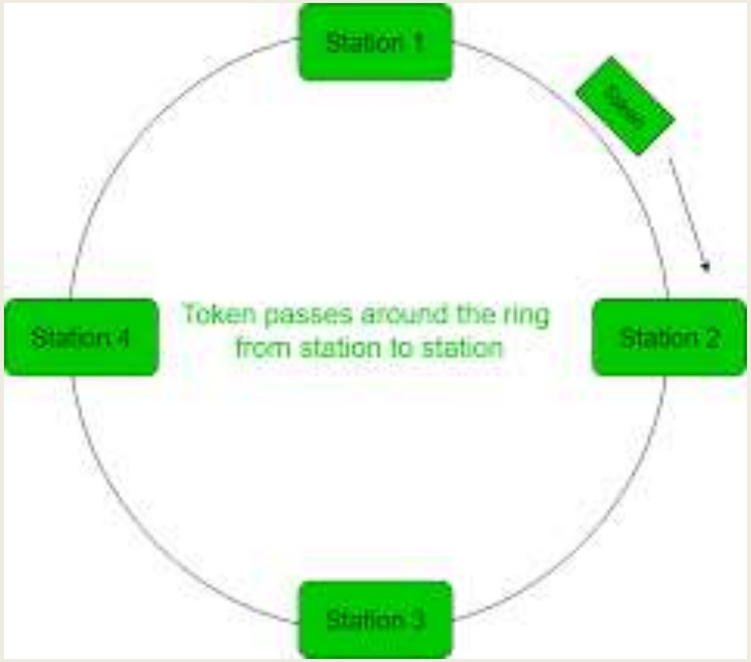
Token Bus / (IEEE 802.4)

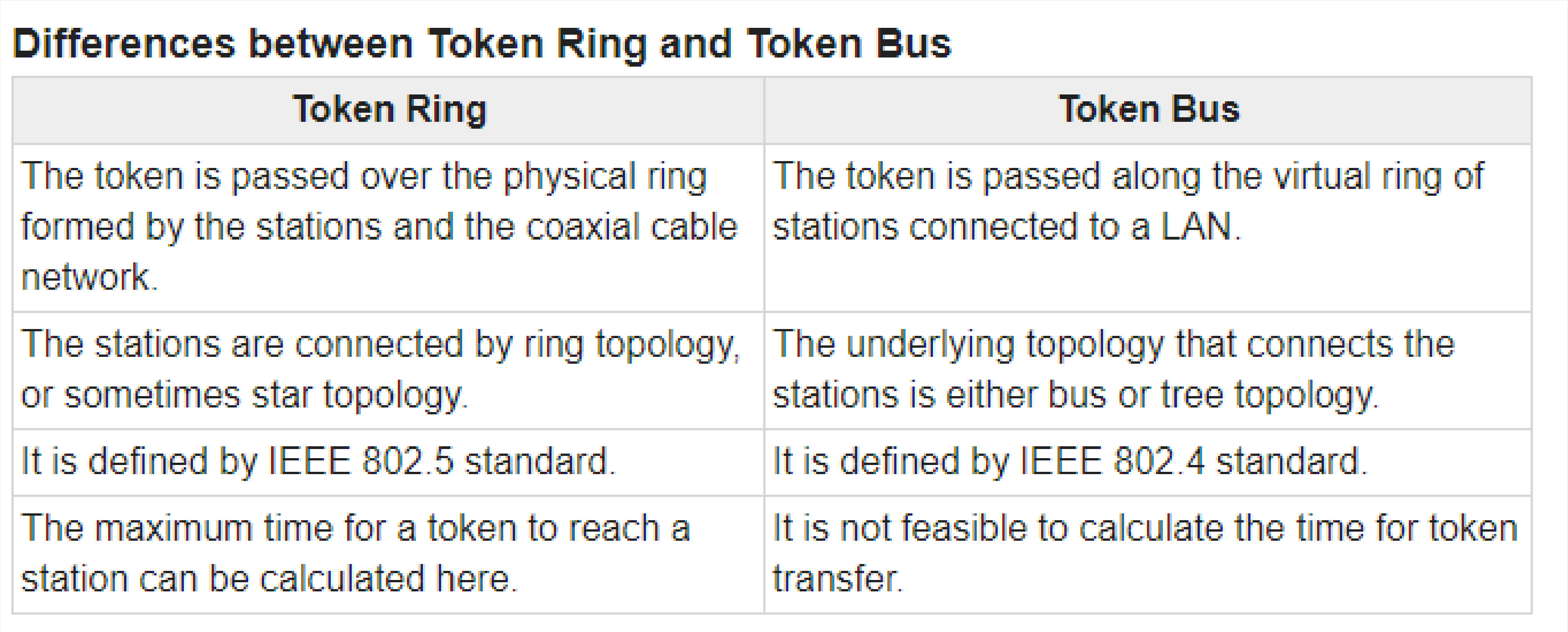


Token Ring / (IEEE 802.5)

- Ring Topology is used
- Access control method used is token passing.
- Token ring is unidirectional
- Data Rate used in 4Mbps and 16Mbps.
- Piggybacking acknowledgement is used
- Differential Manchester encoding is used.
- Variable size framing.

Token Ring / (IEEE 802.5)





Differences between Token Ring and Token Bus

Token Ring	Token Bus
The token is passed over the physical ring formed by the stations and the coaxial cable network.	The token is passed along the virtual ring of stations connected to a LAN.
The stations are connected by ring topology, or sometimes star topology.	The underlying topology that connects the stations is either bus or tree topology.
It is defined by IEEE 802.5 standard.	It is defined by IEEE 802.4 standard.
The maximum time for a token to reach a station can be calculated here.	It is not feasible to calculate the time for token transfer.

Wireless LAN

A **Wireless Local Area Network (WLAN)** allows devices to connect and communicate over a **wireless medium** (radio waves) within a **limited area** like homes, offices, or campuses. Replaces physical (wired) Ethernet connections. Uses **Wi-Fi technology**, based on the **IEEE 802.11** standard

Advantages of WLAN:

- No cabling required → easy and low-cost setup
- Mobile access for laptops, phones, etc.
- Easy to expand the network

Disadvantages:

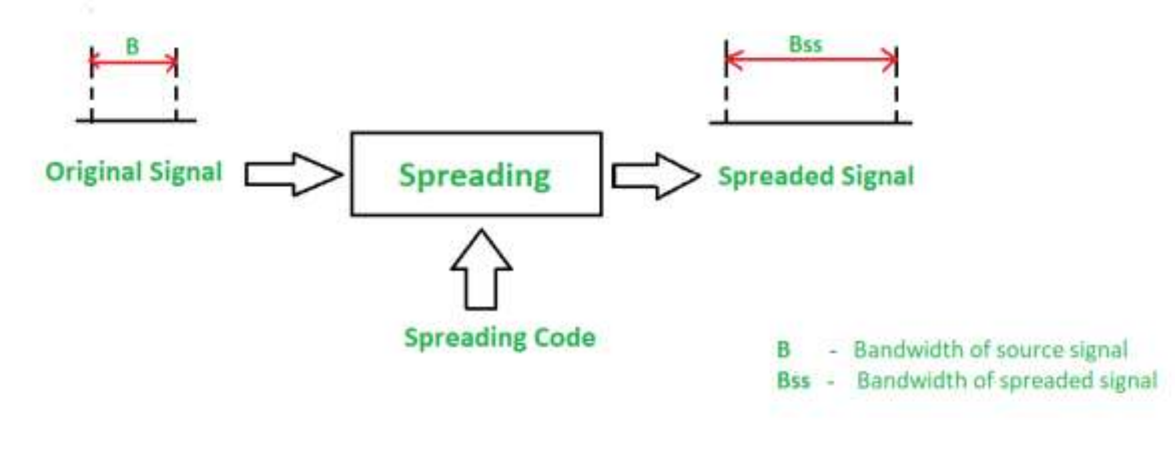
- **Security risks** (hacking)
- **Slower** than wired LAN (varies with standard)
- **Interference** from walls, devices, etc.
- **Limited range**

Spread Spectrum

Spread Spectrum is a **signal transmission technique** used in wireless communication to reduce **interference**, enhance **security**, and improve **signal reliability**. Spread spectrum is a method of transmitting radio signals over a wide range of frequencies. It spreads the signal over a broader bandwidth than the minimum required to send the information, which provides advantages such as increased resistance to interference, improved security, and enhanced privacy.

Two Main Types of Spread Spectrum:

1. **Frequency Hopping Spread Spectrum (FHSS)**
2. **Direct Sequence Spread Spectrum (DSSS)**


B
Bss
Original Signal
Spreading
Spreaded Signal
Spreading Code
B - Bandwidth of source signal
Bss - Bandwidth of spreaded signal

Spread Spectrum

Frequency Hopping Spread Spectrum (FHSS)
The transmitter and receiver **hop between different frequency channels** based on a pre-determined pattern.
Reduces interference and increases security.
Example:
Bluetooth uses FHSS to avoid interference with other devices.

Direct Sequence Spread Spectrum (DSSS)
Each bit of data is **spread** into a sequence of bits (called chips).
Uses a **pseudo-random code** to modulate the data.
More robust against interference.
Example:
Wi-Fi (IEEE 802.11b) uses DSSS.

Bluetooth

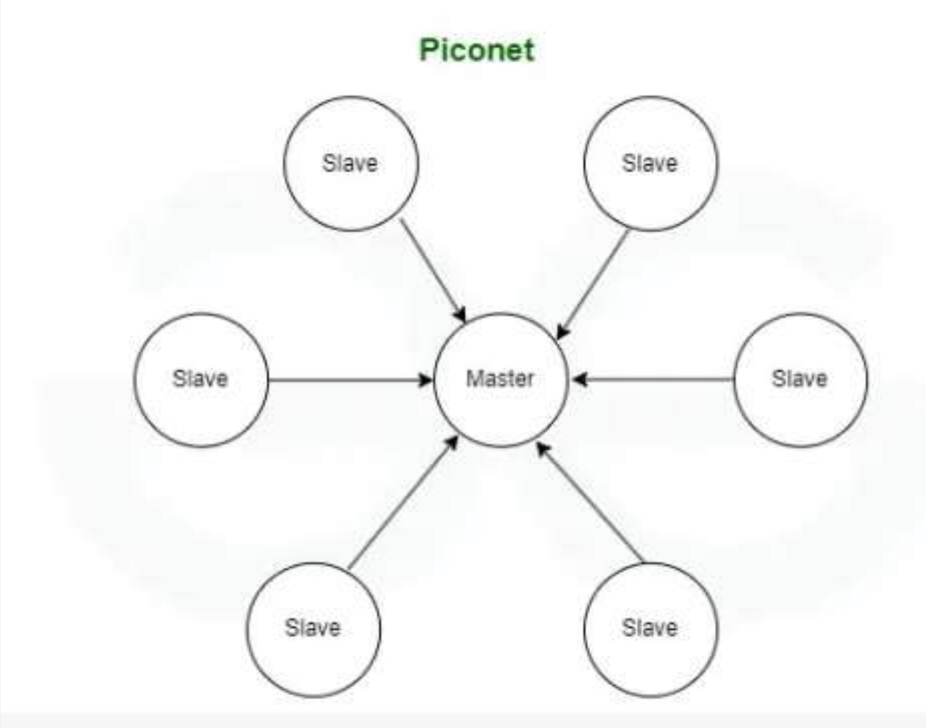
Bluetooth is a wireless technology that lets devices like phones, tablets, and headphones connect to each other and share information without needing cables. Bluetooth simply follows the principle of transmitting and receiving data using radio waves. It can be paired with the other device which has also Bluetooth but it should be within the estimated communication range to connect. When two devices start to share data, they form a network called piconet which can further accommodate more than five devices.

The architecture of Bluetooth defines two types of networks:

1. **Piconet**
2. **Scatternet**

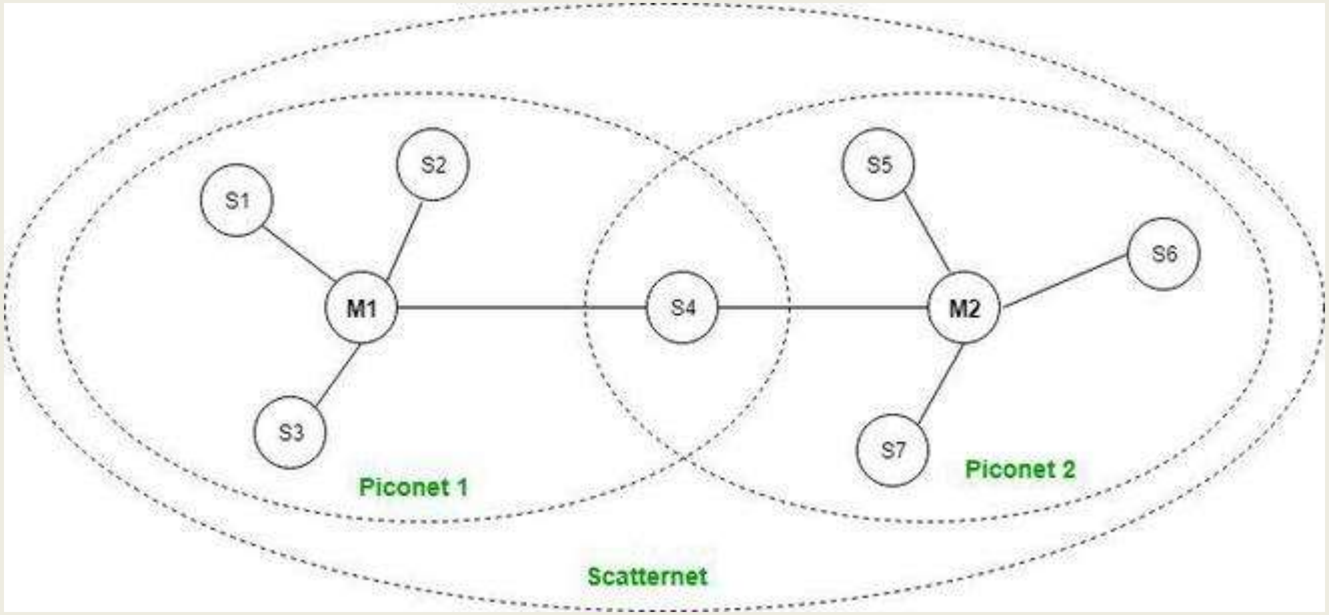
Bluetooth-Piconet

Piconet is a type of Bluetooth network that contains one primary node called the master node and seven active secondary nodes called slave nodes. Thus, we can say that there is a total of 8 active nodes which are present at a distance of 10 meters. The communication between the primary and secondary nodes can be one-to-one or one-to-many.



Bluetooth - Scatternet

It is formed by using various piconets. A slave that is present in one piconet can act as master or we can say primary in another piconet. This kind of node can receive a message from a master in one piconet and deliver the message to its slave in the other piconet where it is acting as a master. This type of node is referred to as a bridge node. A station cannot be mastered in two piconets.



WIFI

Wi-Fi is a wireless technology that allows electronic devices to connect to the internet and communicate with each other without a physical cable. This uses radio waves to transmit the data between a Wi-Fi router and compatible devices like smartphones, computers, and smart home gadgets. These Wi-Fi networks are common in homes, offices, and public spaces providing convenient internet access and local connectivity. This technology has become an essential part of modern digital life enabling wireless internet browsing, file sharing, and device communication in various settings.

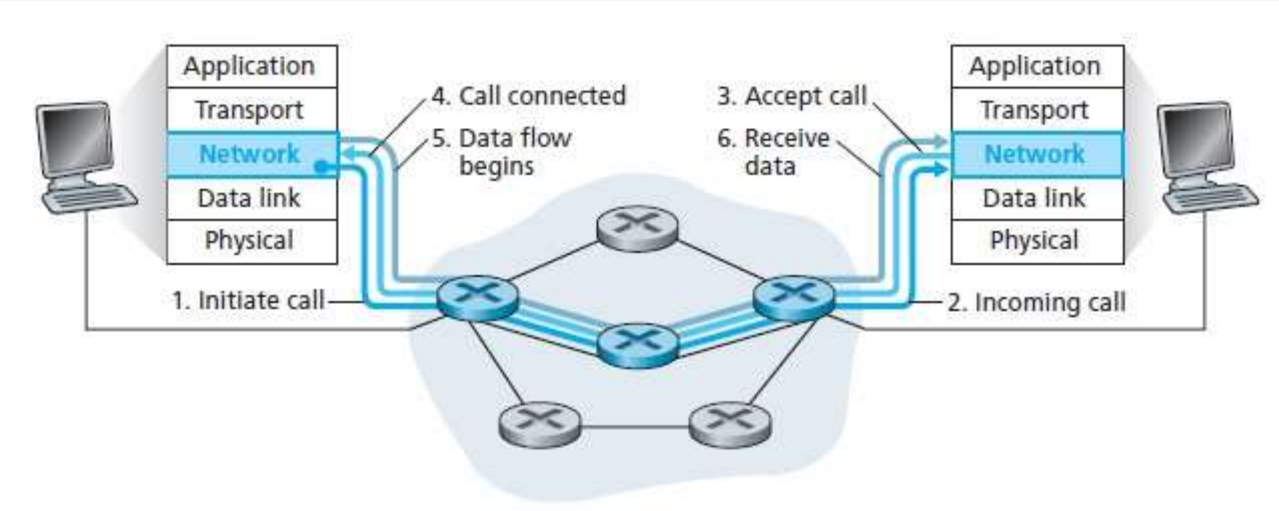
Virtual circuit switching

It is a network where a virtual connection is established between source and the destination. Through this network, packets will be transferred during any call. The path established between two points appears as a dedicated physical circuit. Therefore, it is called a virtual circuit. It is a type of packet switching.

It is a connection-oriented service, where the first packet goes and reserves the resources for the subsequent packets.

For examples − ATM and frame relay.

The pictorial representation of virtual circuit connection over a telephone call is as follows −



Virtual circuit switching

Advantages

- Packets are delivered in the same order as they all follow the same route between the source & the destination.

- The overhead is smaller as full address is not required on each packet as they all follow the same established path.

- The connection is more reliable as it is one to one connection.

- Less chances of data loss.

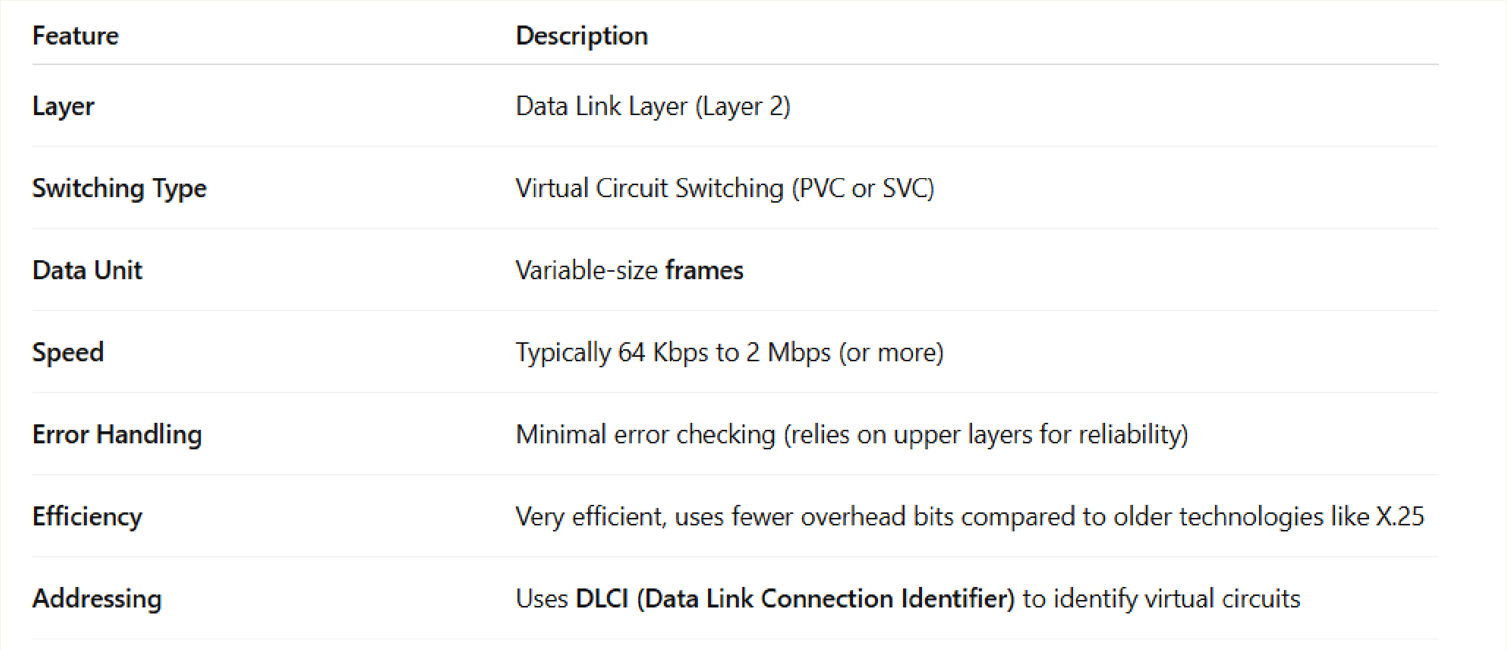
Disadvantages:

- The switching equipment should be powerful.

- Re-establishment of the network is difficult as if there is any failure. All calls need to be re-established.

Frame Relay

Frame Relay is a high-performance Wide Area Network (WAN) technology that uses virtual circuit switching to transmit data efficiently between LANs over long distances.
It operates at the Data Link Layer (Layer 2) of the OSI model and is designed for cost-effective, high-speed data communication.



Feature	Description
Layer	Data Link Layer (Layer 2)
Switching Type	Virtual Circuit Switching (PVC or SVC)
Data Unit	Variable-size **frames**
Speed	Typically 64 Kbps to 2 Mbps (or more)
Error Handling	Minimal error checking (relies on upper layers for reliability)
Efficiency	Very efficient, uses fewer overhead bits compared to older technologies like X.25
Addressing	Uses **DLCI (Data Link Connection Identifier)** to identify virtual circuits

Frame Relay

Types of Virtual Circuits in Frame Relay

1. **PVC (Permanent Virtual Circuit)** - Always available, pre-established by the service provider
2. **SVC (Switched Virtual Circuit)** - Temporary, created and deleted as needed (rarely used in Frame Relay)

❑ **How Frame Relay Works: Step-by-Step**

- Data is broken into **frames** (small packets).
- These frames travel through a **virtual circuit** established between sender and receiver.
- Virtual circuits can be **permanent (PVC)** or **switched (SVC)**.
- Frame Relay routers forward frames based on **Data Link Connection Identifiers (DLCIs)**.

Frame Relay

Advantages of Frame Relay

- Cost-effective for WANs
- Supports multiple virtual circuits
- Efficient bandwidth usage
- Scalable for growing networks

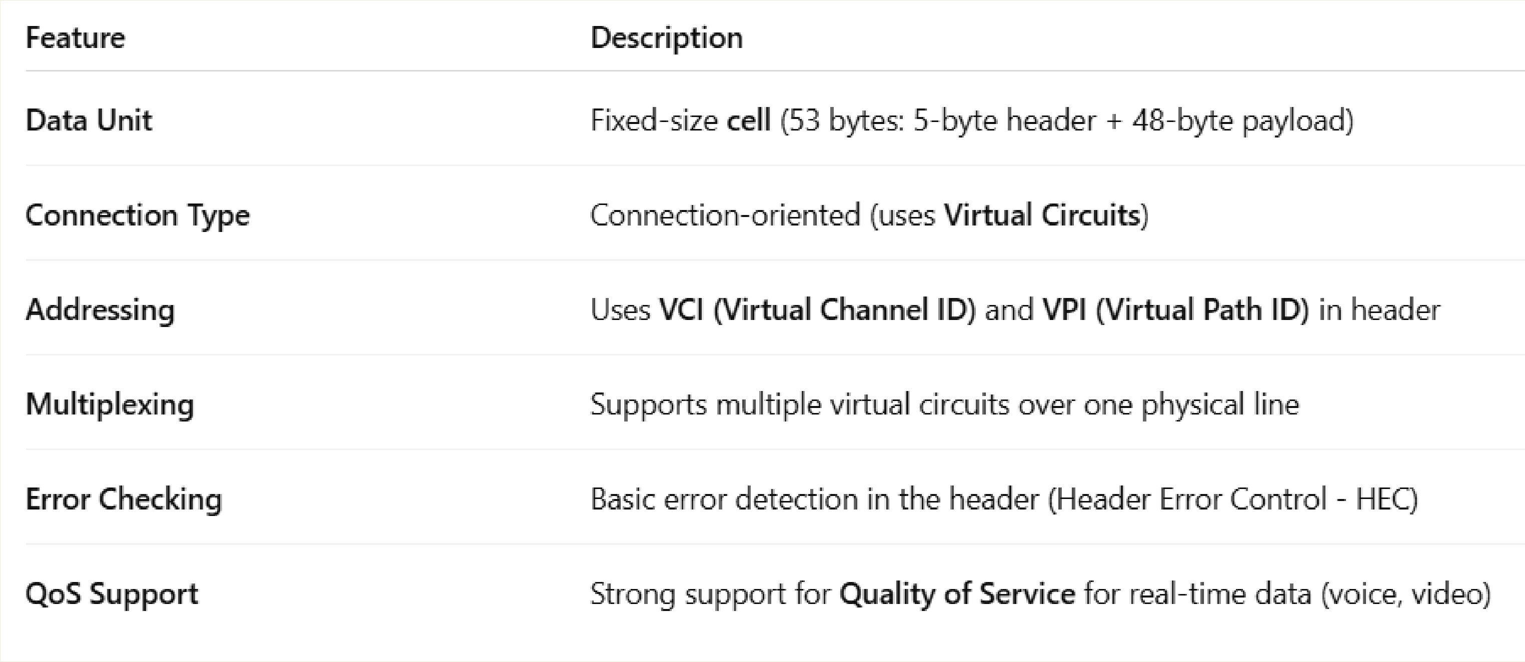
Disadvantages

- Minimal error control
- Not ideal for real-time voice/video
- Outdated

ATM

ATM (Asynchronous Transfer Mode) is a high-speed, connection-oriented switching technology used for transmitting data, voice, and video over telecom and computer networks.

It operates mainly at the Data Link Layer (Layer 2) of the OSI model, but also includes functions of the Physical and Network layers.



Feature	Description
Data Unit	Fixed-size **cell** (53 bytes: 5-byte header + 48-byte payload)
Connection Type	Connection-oriented (uses **Virtual Circuits**)
Addressing	Uses **VCI (Virtual Channel ID)** and **VPI (Virtual Path ID)** in header
Multiplexing	Supports multiple virtual circuits over one physical line
Error Checking	Basic error detection in the header (Header Error Control - HEC)
QoS Support	Strong support for **Quality of Service** for real-time data (voice, video)

ATM

Advantages of ATM

High Speed: Another characteristic of ATM is that it has high data transfer rates, which is good for applications that require high processing speed of data.

Fixed Packet Size: ATM also has a more stable transmission due to the fixed packet size, thus a lower packet delay is observed.

Reliable: ATM incorporates error control and flow control features that place it more appealing than frame relay.

Disadvantages of ATM

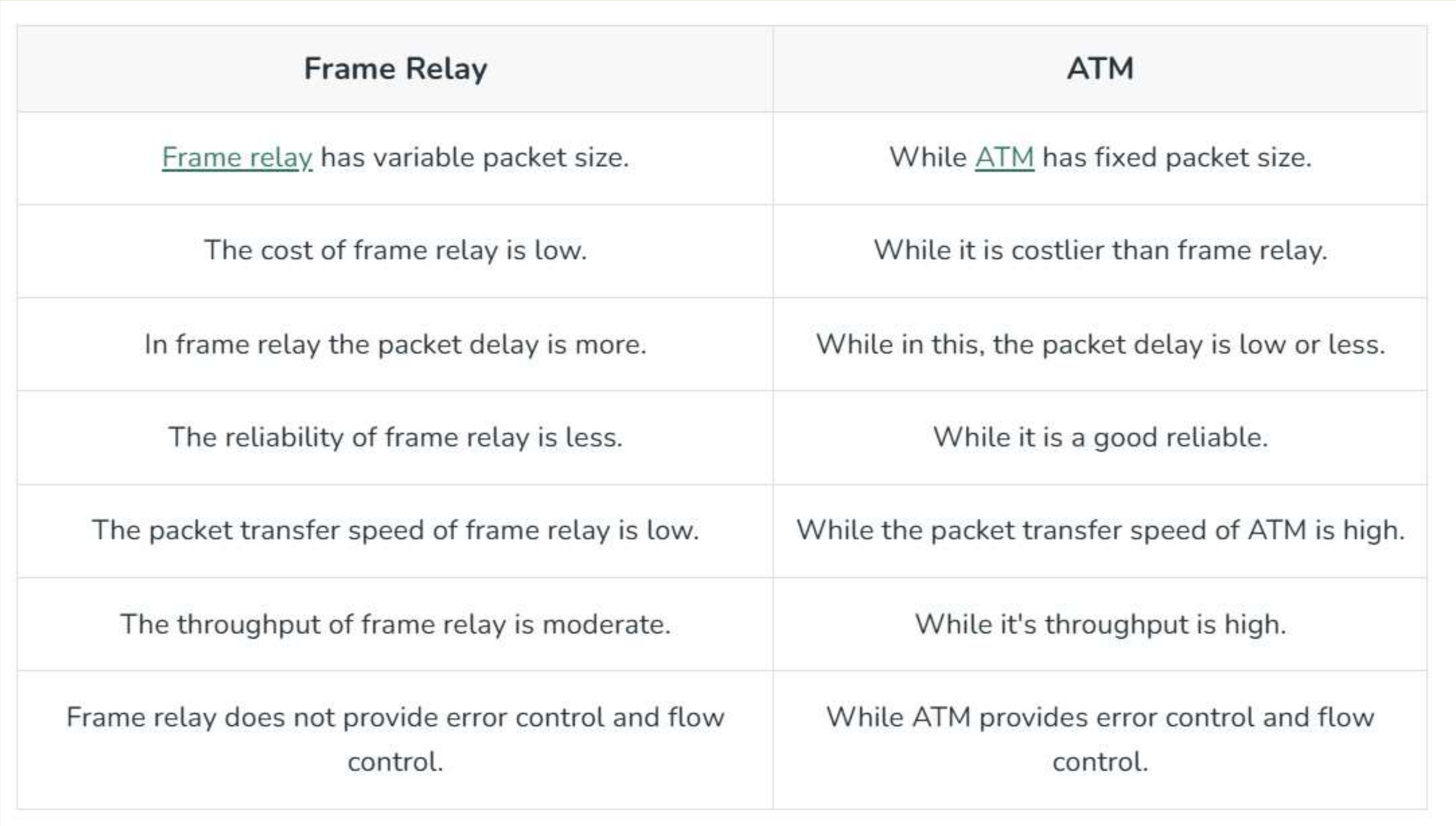
Costly: The data link implementation and maintenance costs are comparatively higher in case of ATM as compared to Frame Relay.

Complexity: Because of its features, and faster connectivity, ATM is more challenging to implement and manage than the traditional systems.

ATM vs Frame Relay

- **ATM** uses small, fixed-size packets (cells) and is better for real-time voice and video.
- **Frame Relay** uses variable-sized packets and is mainly used for data in wide-area networks.
- ATM is more complex and costly; Frame Relay is simpler and cheaper.

Difference Between Frame Relay and ATM



Frame Relay	ATM
Frame relay has variable packet size.	While ATM has fixed packet size.
The cost of frame relay is low.	While it is costlier than frame relay.
In frame relay the packet delay is more.	While in this, the packet delay is low or less.
The reliability of frame relay is less.	While it is a good reliable.
The packet transfer speed of frame relay is low.	While the packet transfer speed of ATM is high.
The throughput of frame relay is moderate.	While it's throughput is high.
Frame relay does not provide error control and flow control.	While ATM provides error control and flow control.

Data Link Layer Protocols

- HDLC (High-level Data Link Control)
- PPP(Point-to-Point Protocol

High-level Data Link Control (HDLC)

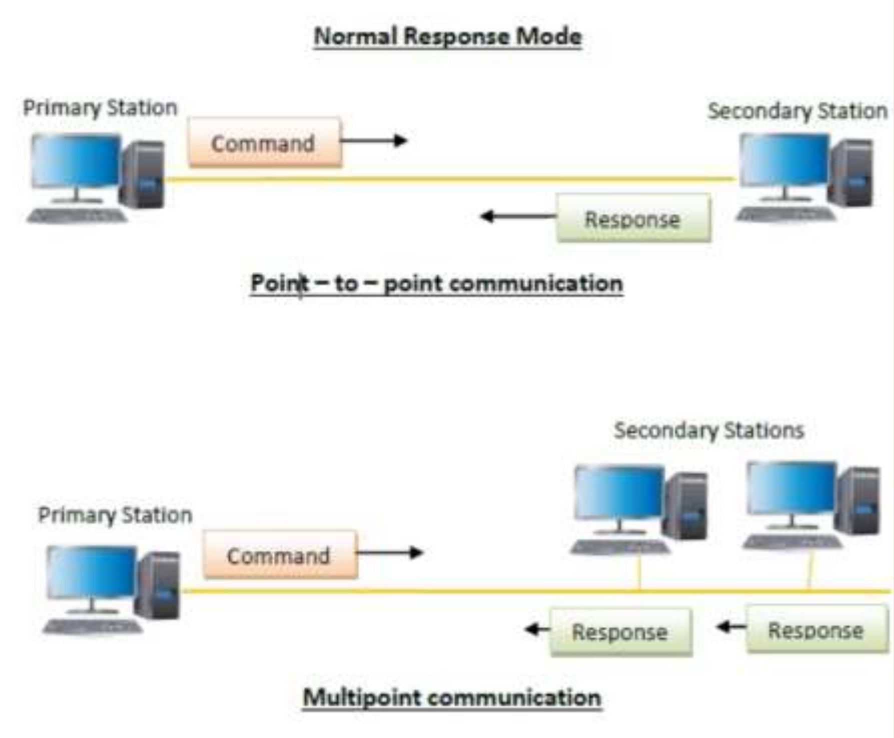
High-level Data Link Control (HDLC) is a group of communication protocols of the data link layer for transmitting data between network points or nodes. Since it is a data link protocol, data is organized into frames. A frame is transmitted via the network to the destination that verifies its successful arrival. It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

High-level Data Link Control (HDLC)

HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

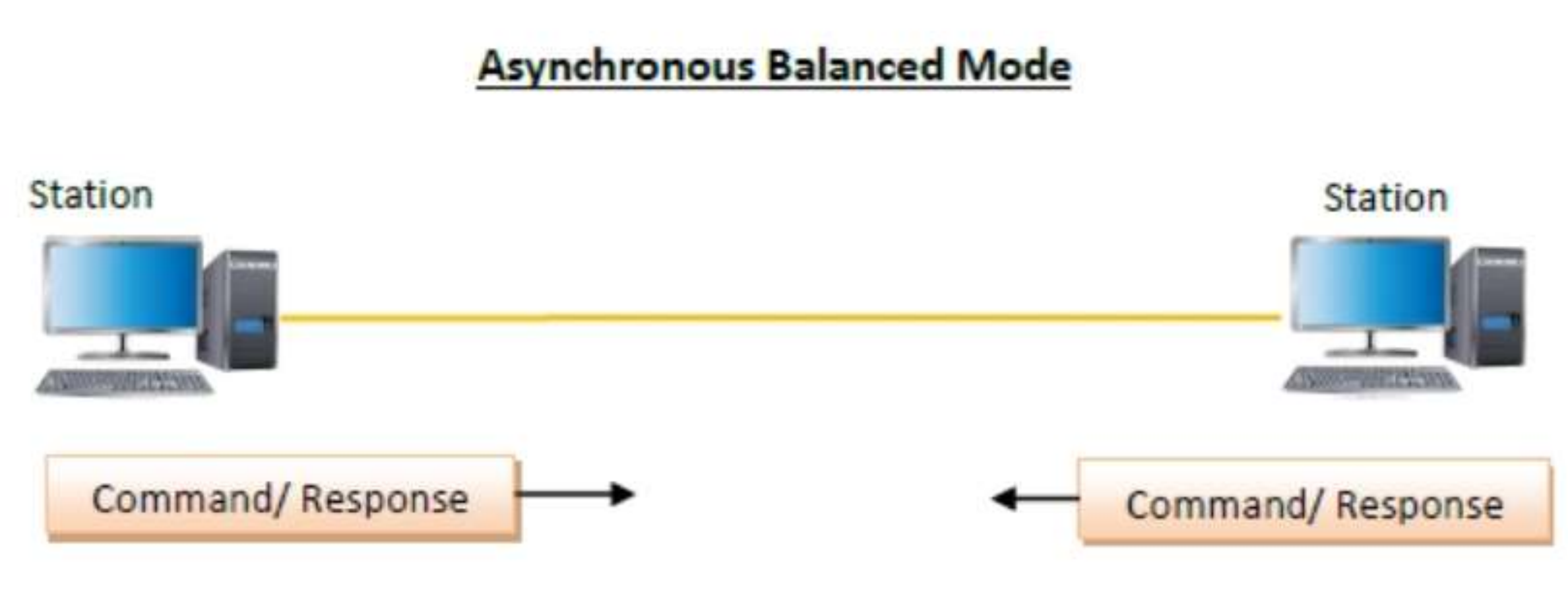
1. **Normal Response Mode (NRM)** − Here, two types of stations are there, a primary station that send commands and secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



High-level Data Link Control (HDLC)

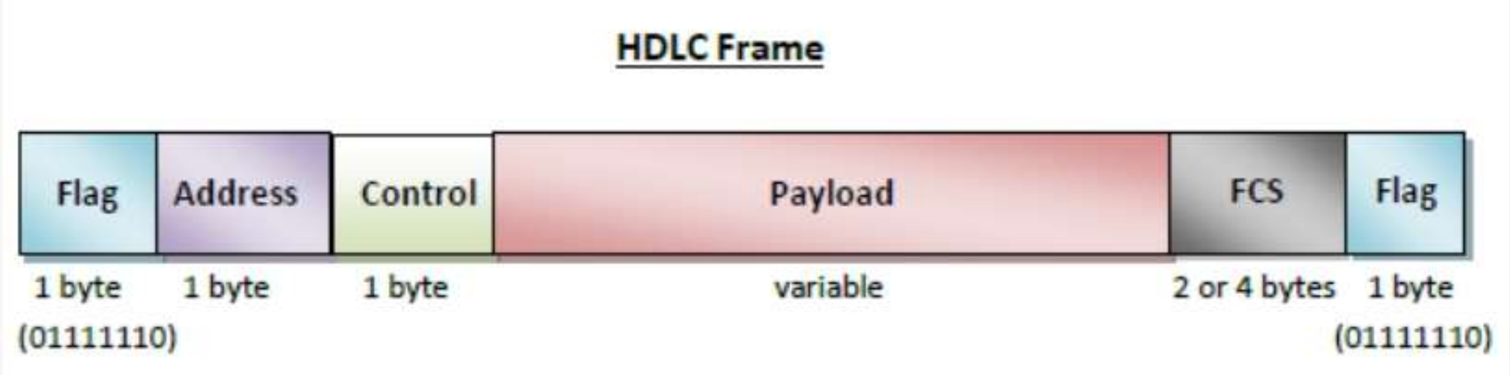
HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

2. **Asynchronous Balanced Mode (ABM)** – Here, the configuration is balanced, i.e. each station can both send commands and respond to commands. It is used for only point - to - point communications.



High-level Data Link Control (HDLC)

HDLC is a bit - oriented protocol where each frame contains up to six fields. The structure varies according to the type of frame. The fields of a HDLC frame are –



HDLC Frame

Flag	Address	Control	Payload	FCS	Flag
1 byte (01111110)	1 byte	1 byte	variable	2 or 4 bytes	1 byte (01111110)

High-level Data Link Control (HDLC)

Flag − It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.

Address − It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.

Control − It is 1- or 2-bytes containing flow and error control information.

Payload − This carries the data from the network layer. Its length may vary from one network to another.

FCS − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)

High-level Data Link Control (HDLC)

Types of HDLC Frames

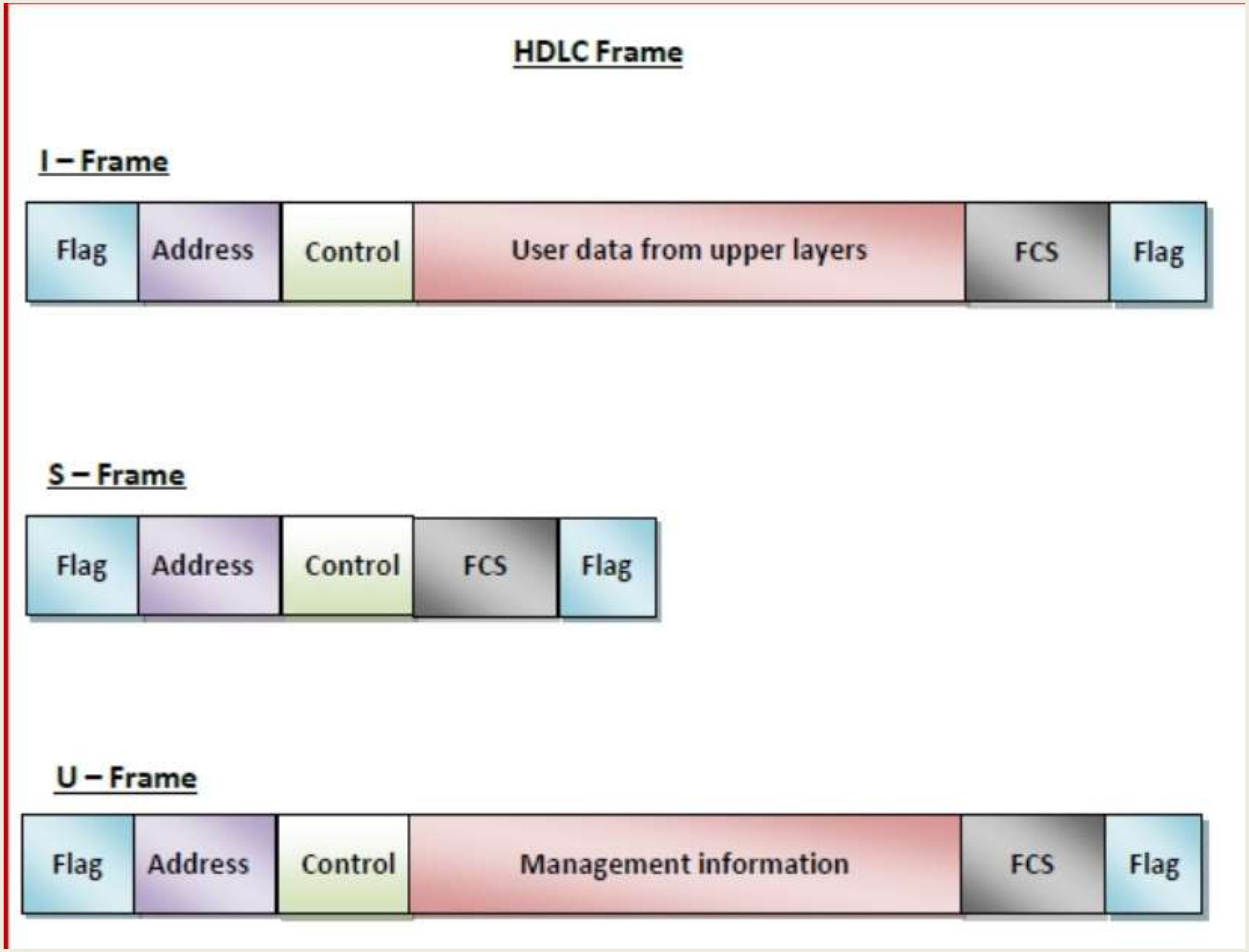
There are three types of HDLC frames. The type of frame is determined by the control field of the frame.

I-frame − I-frames or Information frames carry user data from the network layer. They also include flow and error control information that is piggybacked on user data. The first bit of control field of I-frame is 0.

S-frame − S-frames or Supervisory frames do not contain information field. They are used for flow and error control when piggybacking is not required. The first two bits of control field of S-frame is 10.

U-frame − U-frames or Un-numbered frames are used for myriad miscellaneous functions, like link management. It may contain an information field, if required. The first two bits of control field of U-frame is 11.

High-level Data Link Control (HDLC)



HDLC Frame

I – Frame

Flag | Address | Control | User data from upper layers | FCS | Flag

S – Frame

Flag | Address | Control | FCS | Flag

U – Frame

Flag | Address | Control | Management information | FCS | Flag

Point to Point Protocol (PPP)

- The Internet needs a point-to-point protocol for a variety of purposes, including router-to-router traffic and home user-to-ISP traffic.
- PPP handles error detection, supports multiple protocols, allows IP addresses to be negotiated at connection time, permits authentication, and has many other features.

PPP features

- Packet framing - encapsulation of network-layer datagram in data link frame
- Multi-protocol - carry network layer data of any network layer protocol (not just IP) ***at same time*** ability to demultiplex upwards
- Bit transparency - must carry any bit pattern in the data field (even if underlying channel can't)
- Error detection - not correction
- Connection liveness: detect, signal link failure to network layer
- Authentication: Identify the User

PPP Frame format

Bytes 1 1 1 1 or 2 Variable 2 or 4 1

Flag	Address	Control	Protocol	Payload	Checksum	Flag
01111110	11111111	00000011				01111110

- **Flag :** always begins and end with standard HDLC flag i.e 01111110
- **Address :** since PPP is used for point-to-point connection, it uses the broadcast address 11111111.
- **Control** : The value is 00000011 to show that the frame does not contain any sequence numbers and there is no flow or error control
- **Protocol** : Identify Upper Layer Protocol
- **Payload / Data Field:** carries user data and other information
- **Checksum / FCS**(Frame check Sequence) - This field usually contains checksum simply for identification of errors.

Thank You !!!