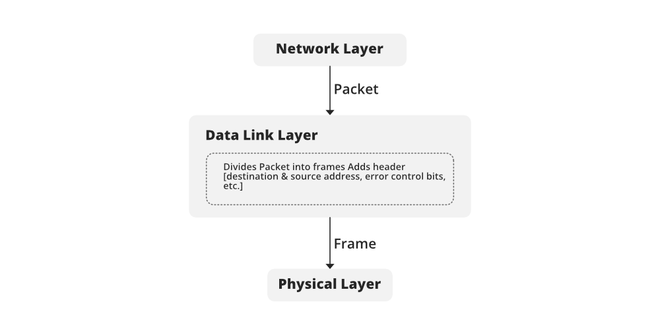
UNIT-2

**MODULE II:** Data link layer Functionalities of Data link layer - Introduction, Framing, Error Detection and Correction – Parity – LRC – CRC- Hamming code, Flow and Error Control, Noiseless Channels, Noisy Channels, HDLC, Point to Point Protocols. Random access, Controlled access, Channelization, Collision Free Protocols. LAN - LAN - Ethernet IEEE 802.3 - IEEE 802.4 - IEEE 802.5 - IEEE 802.11

**Data link layer Functionalities**

**Functions of the Data-link Layer:**



**1. Framing:**The packet received from the Network layer is known as a frame in the Data link layer. At the sender’s side, DLL receives packets from the Network layer and divides them into small frames, then, sends each frame bit-by-bit to the physical layer. It also attaches some special bits (for error control and addressing) at the header and end of the frame. At the receiver’s end, DLL takes bits from the Physical layer organize them into the frame, and sends them to the Network layer.

**2. Addressing:**The data link layer encapsulates the source and destination’s [MAC address](https://www.geeksforgeeks.org/introduction-of-mac-address-in-computer-network/)/ physical address in the header of each frame to ensure node-to-node delivery. MAC address is the unique hardware address that is assigned to the device while manufacturing.

**3. Error Control:**Data can get corrupted due to various reasons like noise, attenuation, etc. So, it is the responsibility of the data link layer, to detect the error in the transmitted data and correct it using [error detection](https://www.geeksforgeeks.org/error-detection-in-computer-networks/) and [correction](https://www.geeksforgeeks.org/hamming-code-in-computer-network/) techniques respectively. DLL adds error detection bits into the frame’s header, so that receiver can check received data is correct or not.

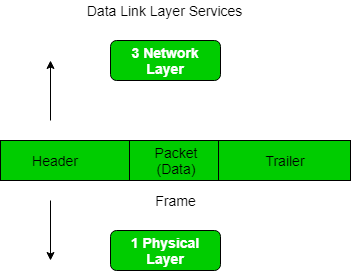
**4. Flow Control:**If the receiver’s receiving speed is lower than the sender’s sending speed, then this can lead to an overflow in the receiver’s buffer and some frames may get lost. So, it’s the responsibility of DLL to synchronize the sender’s and receiver’s speeds and establish flow control between them.

**5. Access Control:**When multiple devices share the same communication channel there is a high probability of collision, so it’s the responsibility of DLL to check which device has control over the channel and [CSMA/CD](https://www.geeksforgeeks.org/collision-detection-csmacd/) and [CSMA/CA](https://www.geeksforgeeks.org/difference-between-csma-ca-and-csma-cd/) can be used to avoid collisions and loss of frames in the channel.

**Framing**

Frames are the units of digital transmission, particularly in computer networks and telecommunications. Frames are comparable to the packets of energy called photons in the case of light energy. Frame is continuously used in Time Division Multiplexing process.

Framing is a point-to-point connection between two computers or devices consists of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into discernible blocks of information. Framing is a function of the data link layer. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.



At the data link layer, it extracts the message from the sender and provides it to the receiver by providing the sender’s and receiver’s addresses. The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.

**Problems in Framing –**

* **Detecting start of the frame:**When a frame is transmitted, every station must be able to detect it. Station detects frames by looking out for a special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimiter).
* **How does the** **station detect a frame:**Every station listens to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.
* **Detecting end of frame:**When to stop reading the frame.

**Types of framing –** There are two types of framing:

**1. Fixed size –** The frame is of fixed size and there is no need to provide boundaries to the frame, the length of the frame itself acts as a delimiter.

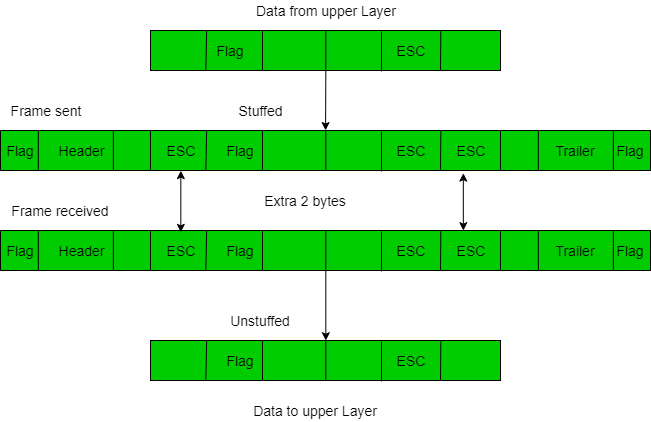
* **Drawback:**It suffers from internal fragmentation if the data size is less than the frame size
* **Solution:**Padding

**2. Variable size –** In this, there is a need to define the end of the frame as well as the beginning of the next frame to distinguish. This can be done in two ways: 

1. **Length field –** We can introduce a length field in the frame to indicate the length of the frame. Used in **Ethernet(802.3)**. The problem with this is that sometimes the length field might get corrupted.
2. **End Delimiter (ED) –** We can introduce an ED(pattern) to indicate the end of the frame. Used in **Token Ring**. The problem with this is that ED can occur in the data. This can be solved by:

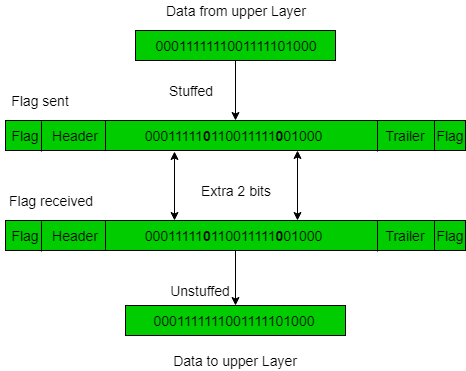
**1. Character/Byte Stuffing:**Used when frames consist of characters. If data contains ED then, a byte is stuffed into data to differentiate it from ED.

Let ED = “$” –> if data contains ‘$’ anywhere, it can be escaped using ‘\O’ character.   
–> if data contains ‘\O$’ then, use ‘\O\O\O$'($ is escaped using \O and \O is escaped using \O).



**Disadvantage –** It is very costly and obsolete method.

**2. Bit Stuffing:**Let ED = 01111 and if data = 01111   
**–>** Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 0111**0**1.   
**–>** Receiver receives the frame.   
**–>** If data contains 011101, receiver removes the 0 and reads the data.



**Examples –**

* If Data –> 011100011110 and ED –> 0111 then, find data after bit stuffing?

–> 011**0**100011**0**11**0**0 

* If Data –> 110001001 and ED –> 1000 then, find data after bit stuffing?

–> 1100**1**0100**1**1

**Error Detection and Correction**

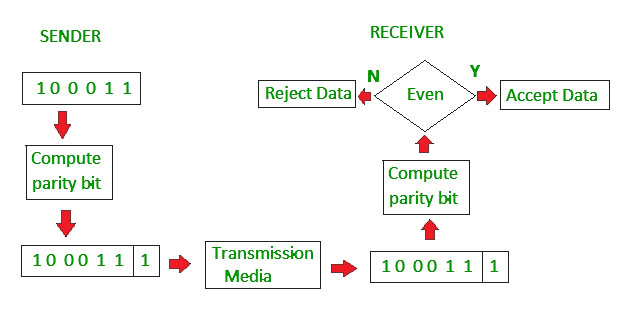
**Error**  
A condition when the receiver’s information does not match with the sender’s information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.  
  
   
**Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model)**  
Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.  
  
   
Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.

Some popular techniques for error detection are:  
1. Simple Parity check  
2. Two-dimensional Parity check  
3. Checksum  
4. Cyclic redundancy check

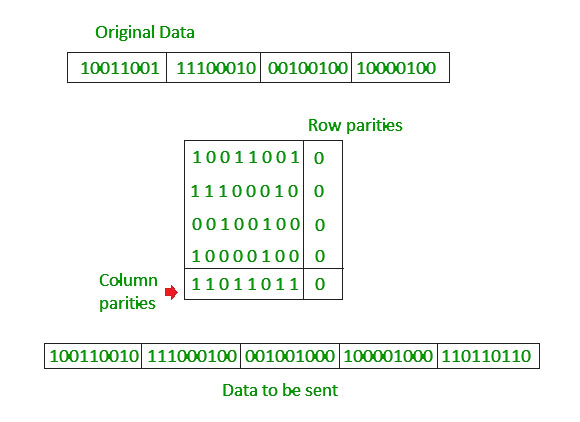
**1. Simple Parity check**  
Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

* 1 is added to the block if it contains odd number of 1’s, and
* 0 is added if it contains even number of 1’s

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

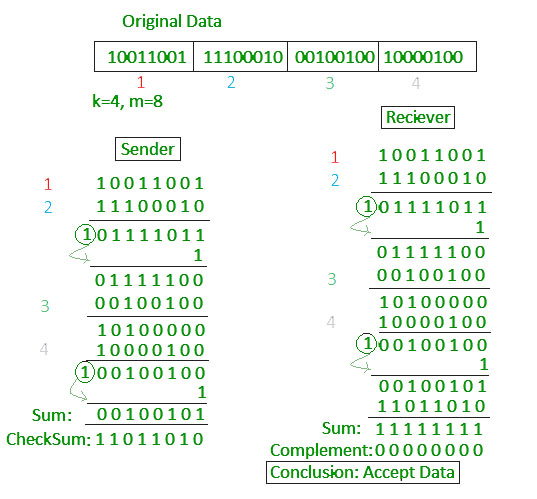
[](https://media.geeksforgeeks.org/wp-content/uploads/detect12.jpg)

**2. Two-dimensional Parity check**  
Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

[](https://media.geeksforgeeks.org/wp-content/uploads/detect11.jpg)

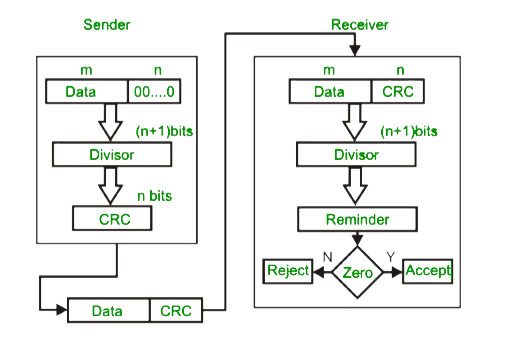
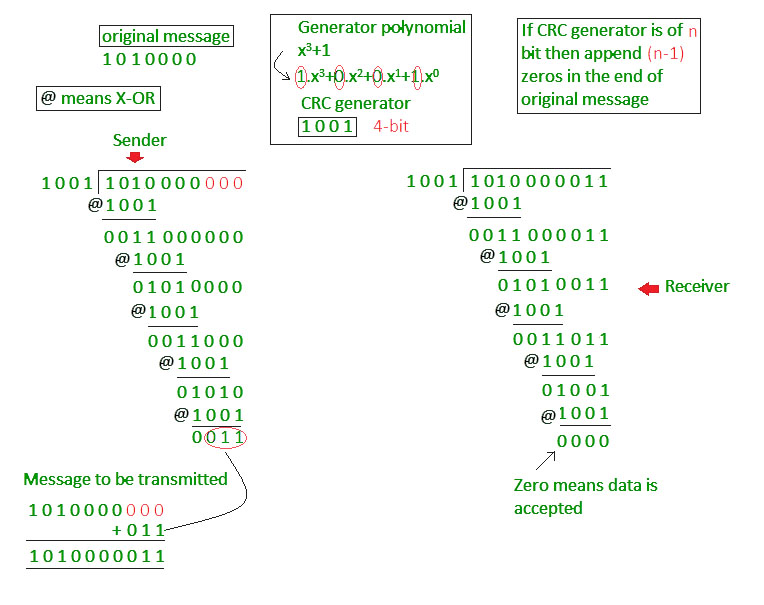
**3. Checksum**

* In checksum error detection scheme, the data is divided into k segments each of m bits.
* In the sender’s end the segments are added using 1’s complement arithmetic to get the sum. The sum is complemented to get the checksum.
* The checksum segment is sent along with the data segments.
* At the receiver’s end, all received segments are added using 1’s complement arithmetic to get the sum. The sum is complemented.
* If the result is zero, the received data is accepted; otherwise discarded.

[](https://media.geeksforgeeks.org/wp-content/uploads/detect13.jpg)

**4. Cyclic redundancy check (CRC)**

* 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.

[](https://media.geeksforgeeks.org/wp-content/uploads/detect14.jpg)  
   
**Example :**  
[](https://media.geeksforgeeks.org/wp-content/uploads/detect15.jpg)

# Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:

* **Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.
* **Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors.

A single additional bit can detect the error, but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.Hello Java Program for Beginners

Suppose r is the number of redundant bits and d is the total number of the data bits. The number of redundant bits r can be calculated by using the formula:

2r>=d+r+1

The value of r is calculated by using the above formula. For example, if the value of d is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

## Hamming Code

**Parity bits:** The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

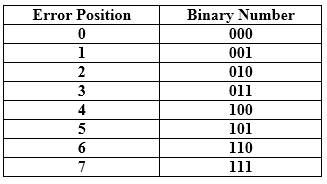
**Even parity:** To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

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

### Algorithm of Hamming code:

* An information of 'd' bits are added to the redundant bits 'r' to form d+r.
* The location of each of the (d+r) digits is assigned a decimal value.
* The 'r' bits are placed in the positions 1,2,.....2k-1.
* 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.



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 :** 2r >= d+r+1

2r>= 4+r+1

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

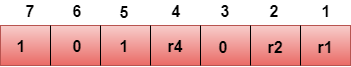
**Total number of bits = d+r = 4+3 = 7;**

## 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, 21, 22**.

1. The position of r1 = 1
2. The position of r2 = 2
3. The position of r4 = 4

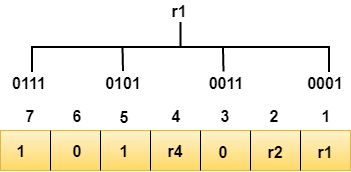
Representation of Data on the addition of parity bits:



## Determining the Parity bits

### Determining the r1 bit

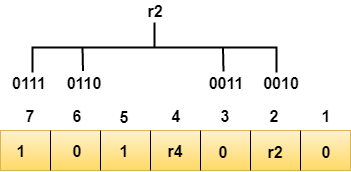
The r1 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the first position.



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 r2 bit

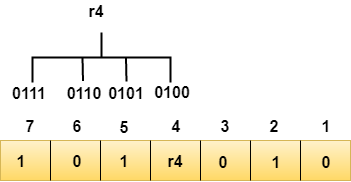
The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position.



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**.

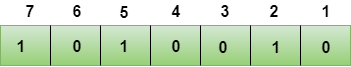
### Determining r4 bit

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**.

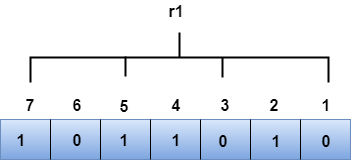
**Data transferred is given below:**



Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

### R1 bit

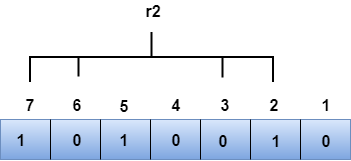
The bit positions of the r1 bit are 1,3,5,7



We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

### R2 bit

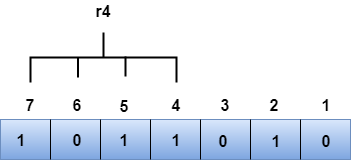
The bit positions of r2 bit are 2,3,6,7.



We observe from the above figure that the binary representation of r2 is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the r2 bit is an even number. Therefore, the value of r2 is 0.

### R4 bit

The bit positions of r4 bit are 4,5,6,7.



We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

**Flow and Error Control**

Flow control and Error control are the two main responsibilities of the Data link layer. Let us understand what these two terms specify. For the node-to-node delivery of the data, the flow and error control are done at the data link layer.

**Flow Control** mainly coordinates with the amount of data that can be sent before receiving an acknowledgment from the receiver and it is one of the major duties of the data link layer.

* For most of the protocols,**flow control**is a set of procedures that mainly tells the sender how much data the sender can send before it must **wait for an acknowledgment** from **the receiver**.
* The data flow must not be allowed to overwhelm the receiver; because any receiving device has a very limited speed at which the device can process the incoming data and the limited amount of memory to store the incoming data.
* The processing rate is slower than the transmission rate; due to this reason each receiving device has a block of memory that is commonly known as **buffer,**that is used to store the incoming data until this data will be processed. In case the buffer begins to fillup then the receiver must be able to tell the sender to halt the transmission until once again the receiver become able to receive.

Thus the flow control makes the sender; wait for the acknowledgment from the receiver before the continuation to send more data to the receiver.

Some of the common flow control techniques are: Stop-and-Wait and sliding window technique.

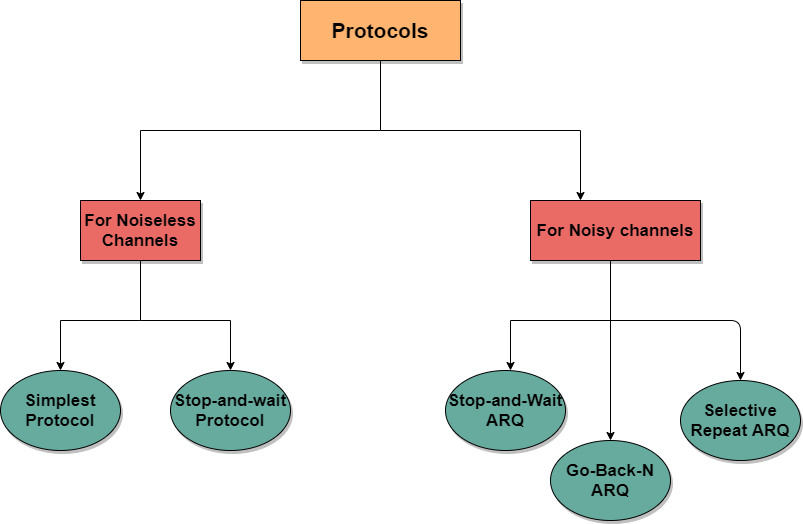
**Error Control** contains both error detection and error correction. It mainly allows the receiver to inform the sender about any damaged or lost frames during the transmission and then it coordinates with the retransmission of those frames by the sender.

The term Error control in the data link layer mainly refers to the methods of error detection and retransmission. Error control is mainly implemented in a simple way and that is whenever there is an error detected during the exchange, then specified frames are retransmitted and this process is also referred to as **Automatic Repeat request (ARQ).**

Protocols

The implementation of protocols is mainly implemented in the software by using one of the common programming languages. The classification of the protocols can be mainly done on the basis of where they are being used.

Protocols can be used for **noiseless channels**(that is **error-free**) and also used for noisy channels(that is**error-creating**). The protocols used for noiseless channels mainly cannot be used in real-life and are mainly used to serve as the basis for the protocols used for**noisy channels.**

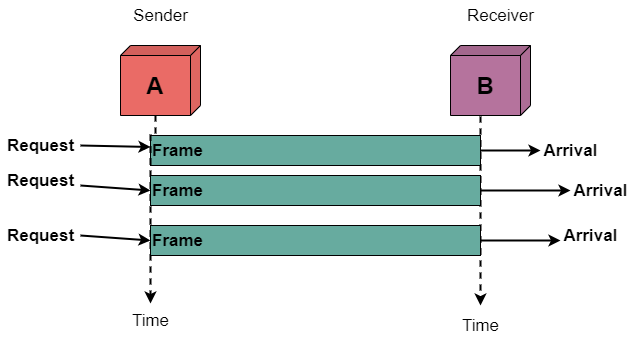
****

All the above-given protocols are unidirectional in the sense that the data frames travel from one node i.e Sender to the other node i.e receiver.

The special frames called acknowledgment (ACK) and negative acknowledgment (NAK) both can flow in opposite direction for flow and error control purposes and the data can flow in only one direction.

But in the real-life network, the protocols of the data link layer are implemented as bidirectional which means the flow of the data is in both directions. And in these protocols, the flow control and error control information such as ACKs and NAKs are included in the data frames in a technique that is commonly known as **piggybacking.**

# Simplest Protocol

* The simplest protocol is basically a **unidirectional protocol** in which data frames only travel in one direction; from the sender to the receiver.
* In this, the receiver can immediately handle the frame it receives whose processing time is small enough to be considered as negligible.
* Basically, the data link layer of the receiver immediately removes the header from the frame and then hand over the data packet to the network layer that also accepts the data packet immediately.
* We can also say that in the case of this protocol the receiver never gets overwhelmed with the incoming frames from the sender.
* Flow Diagram for Simplest Protocol
* Using the simplest protocol the sender A sends a sequence of frames without even thinking about receiver B.
* 
* In order to send the three frames, there will be an occurrence of three events at sender A and three events at the receiver B.
* It is important to note that in the above figure the data frames are shown with the help of boxes.
* The height of the box mainly indicates the transmission time difference between the first bit and the last bit of the frame.

# Stop-and-Wait Protocol

In this tutorial, we will be covering another protocol used in the **Noiseless** channels in the Data link layer.

Stop-and-wait Protocol is used in the data link layer for the transmission in the noiseless channels. Let us first understand why there is a need to use this protocol then we will cover this protocol in detail.

We have studied the simplest protocol in the previous tutorial, suppose there is a scenario in which the data frames arrive at the receiver's site faster than they can be processed means the rate of transmission is more than the processing rate of the frames. Also, it is normal that the receiver does not have enough space, and the data is also coming from multiple sources. Then due to all these, there may occur discarding of frames or denial of service.

In order to prevent the receiver from overwhelming, there is a need to tell the sender to slow down the transmission of frames. We can make use of feedback from the receiver to the sender.

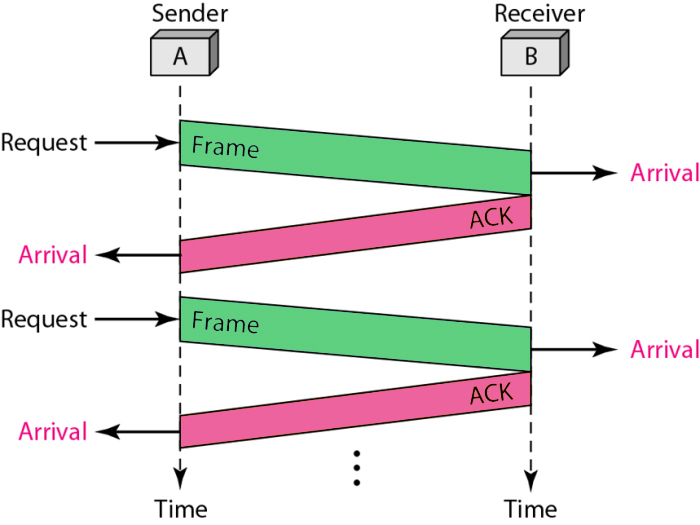
Now from the next section, we will cover the **concept of the Stop-and-wait protocol.**

As the name suggests, when we use this protocol during transmission, then the sender sends one frame, then stops until it receives the confirmation from the receiver, after receiving the confirmation sender sends the next frame.

* There is **unidirectional communication** for the data frames, but the acknowledgment or ACK frames travel from the other direction. Thus the flow control is added here.
* Thus the stop-and-wait is one of the flow control protocol which makes the use of flow control service provided by the data link layer.
* For every sent frame, the acknowledgment is needed and it takes the same amount of time for propagation in order to get back to the sender.

Flow diagram of the stop-and-wait protocol

Given below is the flow diagram of the stop-and-wait protocol:



Advantages

One of the main advantages of the stop-and-wait protocol is the accuracy provided. As the transmission of the next frame is only done after receiving the acknowledgment of the previous frame. Thus there is no chance for data loss.

Disadvantages

Given below are some of the drawbacks of using the stop-and-wait Protocol:

* Using this protocol only one frame can be transmitted at a time.
* Suppose in a case, the frame is sent by the sender but it gets lost during the transmission and then the receiver can neither get it nor can send an acknowledgment back to the sender. Upon not receiving the acknowledgment the sender will not send the next frame. Thus there will occur two situations and these are: The receiver has to wait for an infinite amount of time for the data and the sender has to wait for an infinite amount of time in order to send the next frame.
* In the case of the transmission over a long distance, this is not suitable because the propagation delay becomes much longer than the transmission delay.
* In case the sender sends the data and this data has also been received by the receiver. After receiving the data the receiver then sends the acknowledgment but due to some reasons, this acknowledgment is received by the sender after the timeout period. Now as this acknowledgment is received too late; thus it can be wrongly considered as the acknowledgment of another data packet.
* The time spent waiting for the acknowledgment for each frame also adds up in the total transmission time.

# Go-Back-N Automatic Repeat request

In this tutorial, we will be covering the Go-Back-N Automatic Repeat Request protocol for Noisey channels in the data link layer.

**Go-Back-N ARQ** is mainly a specific instance of Automatic Repeat Request **(ARQ) protocol** where the sending process continues to send a number of frames as specified by the window size even without receiving an acknowledgement**(ACK) packet** from the receiver.

The sender keeps a copy of each frame until the arrival of acknowledgement.

This protocol is a practical approach to the sliding window.

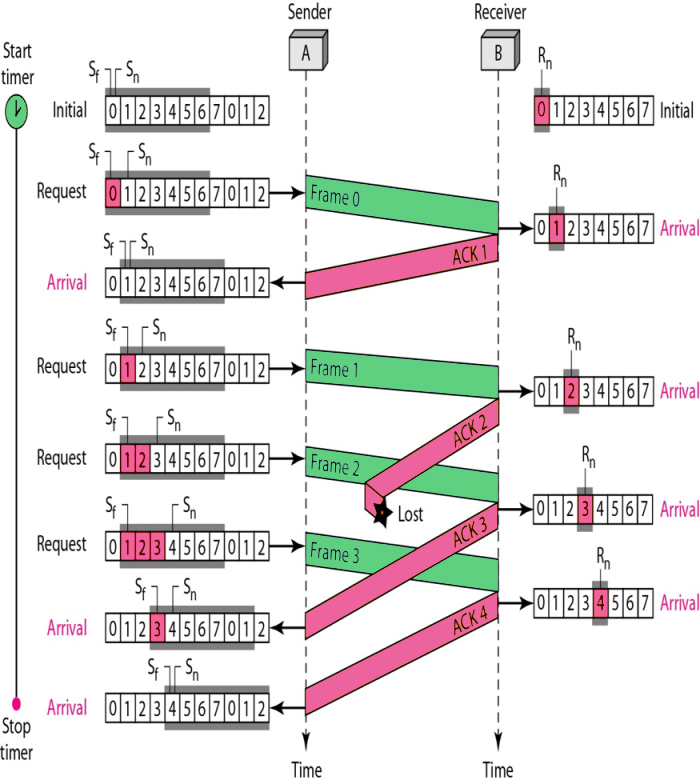
* In Go-Back-N ARQ, the size of the sender is N and the size of the receiver window is always 1.
* This protocol makes the use of **cumulative acknowledgements** means here the receiver maintains an acknowledgement timer; whenever the receiver receives a new frame from the sender then it starts a new acknowledgement timer. When the timer expires then the receiver sends the **cumulative acknowledgement** for all the frames that are unacknowledged by the receiver at that moment.
* It is important to note that the new acknowledgement timer only starts after the receiving of a new frame, it does not start after the expiry of the**old acknowledgement timer**.
* If the receiver receives a corrupted frame, then it silently discards that corrupted frame and the correct frame is retransmitted by the sender after the timeout timer expires. Thus receiver silently discards the corrupted frame. By discarding silently we mean that: “Simply rejecting the frame and not taking any action for the frame".
* In case after the expiry of the acknowledgement timer, suppose there is only one frame that is left to be acknowledged. In that case, the receiver sends the independent acknowledgement for that frame.
* In case if the receiver receives the out of order frame then it simply discards all the frames.
* In case if the sender does not receive any acknowledgement then the entire window of the frame will be retransmitted in that case.
* Using the Go-Back-N ARQ protocol leads to the retransmission of the lost frames after the expiry of the **timeout timer.**

## The Need of Go-Back-N ARQ

This protocol is used to send more than one frame at a time. With the help of Go-Back-N ARQ, there is a reduction in the waiting time of the sender.

With the help of the Go-Back-N ARQ protocol the efficiency in the transmission increases.

Flow Diagram



Advantages

Given below are some of the benefits of using the Go-Back-N ARQ protocol:

* The efficiency of this protocol is more.
* The waiting time is pretty much low in this protocol.
* With the help of this protocol, the timer can be set for many frames.
* Also, the sender can send many frames at a time.
* Only one ACK frame can acknowledge more than one frame.

Disadvantages

Given below are some drawbacks:

* Timeout timer runs at the receiver side only.
* The transmitter needs to store the last N packets.
* The retransmission of many error-free packets follows an erroneous packet.

# Sliding Window Protocol

In this tutorial, we will be covering the concept of **Sliding window Protocol** in Computer Networks.

Sliding Window protocols are those protocols that are used as a method of flow control in networks for the transfer of data.

* With the help of the sliding window technique, multiple frames can be sent at a time by the sender before receiving any acknowledgment from the receiver.
* Sliding Window protocols make the use of **TCP** (transmission control protocol).
* The receiver can send the acknowledgment of **multiple frames** transmitted by the sender using a single **ACK frame**.
* In the Sliding Window protocols, the term **sliding window** mainly refers to the imaginary box that can hold the frames of both the sender side as well as receiver side.

## Sliding Window

The Sliding Window mainly provides the upper limit on the number of frames that can be transmitted before the requirement of an acknowledgment.

* The frames get acknowledged by the receiver at any point even when the window is not completely full on the receiver side.
* Also, the Frames may be transmitted by the source side even when at the time the window is not yet full on the sender side.
* There is the specific size of the window, where the frames are numbered modulo- n, which simply means frames are numbered from **0 to n-1**. For e.g. if **n = 10**, the frames are numbered 0, 1,2,3,4,5,6, 7,8,9, 0, 1,2,3,4,5,6, 7, 8,9,0, 1, ….
* Whenever the receiver sends an acknowledgment (ACK) it also includes the number of the next frame that it expects to receive. For example, in the order to acknowledge the group of frames that ends in frame 6, the receiver needs to send the ACK that contains the number 7. When the sender sees an ACK with the number 7, then the sender comes to know that all the frames up to number 6 have been successfully received.

Let us now understand the working of the sliding window protocols:

There is a finite size buffer on the **sender side** as well as on the **receiver side**. The buffer on the**sending side** is also known as **sending window** while the buffer on the **receiving side** is known as **receiving window**.

* Mainly the size of the sending window is used to determine the sequence number of the outbound frames.
* Suppose the sequence number of the frames is a field of size **n bit**, and then the assigned range of the sequence numbers is **0 to 2?????1**.
* As a result, the size of the sending window is **2?????1**. In the order to accommodate a sending window whose size is 2?????1, there must be an n-bit sequence number that is chosen.

Each and every packet sent by the sender must get acknowledged by the receiver. There is a timer maintained by the sender for each sent packet and in case if there is a packet that is left unacknowledged in a certain time then the packet will be resent.

Also, the sender can send a whole window of packets before receiving an acknowledgment of the first packet in the window. And this will lead to higher transfer rates because the sender may send multiple packets without waiting for the acknowledgment of each packet.

Also, the receiver advertises its window size, which mainly indicates to the sender how much data it can receive. So that sender does not fill up the buffers of the receiver.

With the help of the **modulo-n** technique, the sequence numbers are mainly numbered. Let us take an example here if the size of the sending window size is 5, then the sequence numbers will be 0, 1, 2, 3, 4,0, 1, 2, 3, 4, 0, and so on.

The size of the receiving window indicates the maximum number of frames that the receiver can accept at a time. This size also determines the maximum number of frames that can be sent by the sender before receiving an acknowledgment.

In the sliding window, we make use of a technique known as **Piggybacking**. With the help of piggybacking technique, the acknowledgment is attached with the data frame.

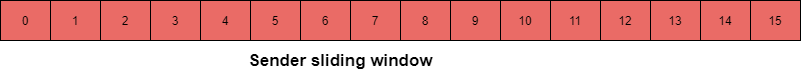
Given below are two protocols that lie under the category of sliding window are as follows;

1. **Go-Back-N ARQ Protocol**
2. **Selective Repeat ARQ protocol**

## Sliding Window (Sender side)

In order to keep the track of the frames, the sender station sends the sequentially numbered frames.

* As the sequence number to be used should occupy a field in the frame. Thus sequence number should be of a limited size.
* If the k bits are allowed by the header of the frame, then the sequence number ranges from 0 to 2k-1.
* There is a list of sequence numbers that is mainly maintained by the sender and on;y these are allowed to send by the sender.
* The size of the sender window is at most 2k-1.
* For example; if 4 bits are allowed by the frame then the size of the window is 2 raised to the power 4 -1 16-1=15.
* The buffer is provided to the sender that has the size equal to the size of the window.



## Sliding Window (Receiver side)

The size of the window on the receiver side is always 1.

* Acknowledgment of a frame is done by sending an ACK frame by the receiver to the sender along with the sequence number of the next expected frame.
* It is explicitly announced by the receiver that it is prepared to receive N next frames, which begins with the number that is specified.
* And this scheme is used in order to acknowledge multiple frames.
* The window at the receiver side can hold 2,3,4 frame but holding the ACK frame until frame 4 has arrived. After the arrival, it will send the ACK along with **sequence number 5** with which the acknowledgment of 2,3,4 is done at a time.
* The buffer size needed by the receiver is 1.

### Advantages of Sliding window

* The transmission of multiple packets can be done without receiving an acknowledgment.
* Using full duplex lines piggybacking can be done.

### Disadvantages of the Sliding window

* There is no limitation on sequence numbers that are required by this protocol.
* There may be wastage of bandwidth in some situations.

# Point-to-Point Protocol (PPP)

Point - to - Point Protocol (PPP) is a communication protocol of the data link layer that is used to transmit multiprotocol data between two directly connected (point-to-point) computers. It is a byte - oriented protocol that is widely used in broadband communications having heavy loads and high speeds. Since it is a data link layer protocol, data is transmitted in frames. It is also known as RFC 1661.

Services Provided by PPP

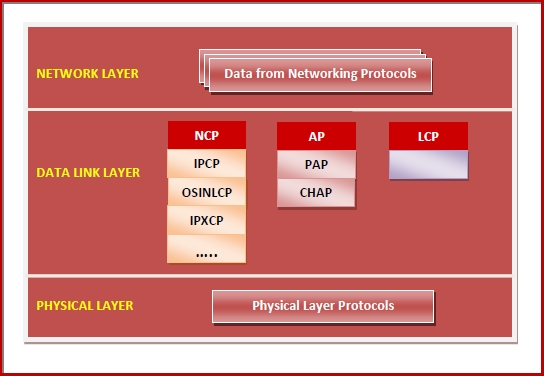
The main services provided by Point - to - Point Protocol are −

* Defining the frame format of the data to be transmitted.
* Defining the procedure of establishing link between two points and exchange of data.
* Stating the method of encapsulation of network layer data in the frame.
* Stating authentication rules of the communicating devices.
* Providing address for network communication.
* Providing connections over multiple links.
* Supporting a variety of network layer protocols by providing a range os services.

Components of PPP

Point - to - Point Protocol is a layered protocol having three components −

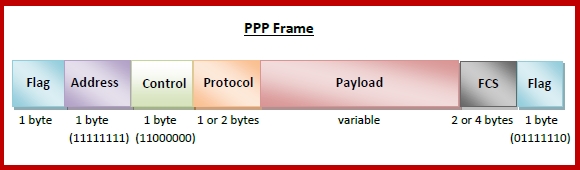
* Encapsulation Component − It encapsulates the datagram so that it can be transmitted over the specified physical layer.
* Link Control Protocol (LCP) − It is responsible for establishing, configuring, testing, maintaining and terminating links for transmission. It also imparts negotiation for set up of options and use of features by the two endpoints of the links.
* Authentication Protocols (AP) − These protocols authenticate endpoints for use of services. The two authentication protocols of PPP are −
  + Password Authentication Protocol (PAP)
  + Challenge Handshake Authentication Protocol (CHAP)
* Network Control Protocols (NCPs) − These protocols are used for negotiating the parameters and facilities for the network layer. For every higher-layer protocol supported by PPP, one NCP is there. Some of the NCPs of PPP are −
  + Internet Protocol Control Protocol (IPCP)
  + OSI Network Layer Control Protocol (OSINLCP)
  + Internetwork Packet Exchange Control Protocol (IPXCP)
  + DECnet Phase IV Control Protocol (DNCP)
  + NetBIOS Frames Control Protocol (NBFCP)
  + IPv6 Control Protocol (IPV6CP)



**PPP Frame**

PPP is a byte - oriented protocol where each field of the frame is composed of one or more bytes. The fields of a PPP frame are −

* Flag − 1 byte that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
* Address − 1 byte which is set to 11111111 in case of broadcast.
* Control − 1 byte set to a constant value of 11000000.
* Protocol − 1 or 2 bytes that define the type of data contained in the payload field.
* Payload − This carries the data from the network layer. The maximum length of the payload field is 1500 bytes. However, this may be negotiated between the endpoints of communication.
* FCS − It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)



Byte Stuffing in PPP Frame − Byte stuffing is used is PPP payload field whenever the flag sequence appears in the message, so that the receiver does not consider it as the end of the frame. The escape byte, 01111101, is stuffed before every byte that contains the same byte as the flag byte or the escape byte. The receiver on receiving the message removes the escape byte before passing it onto the network layer.

# Multiple Access Protocols

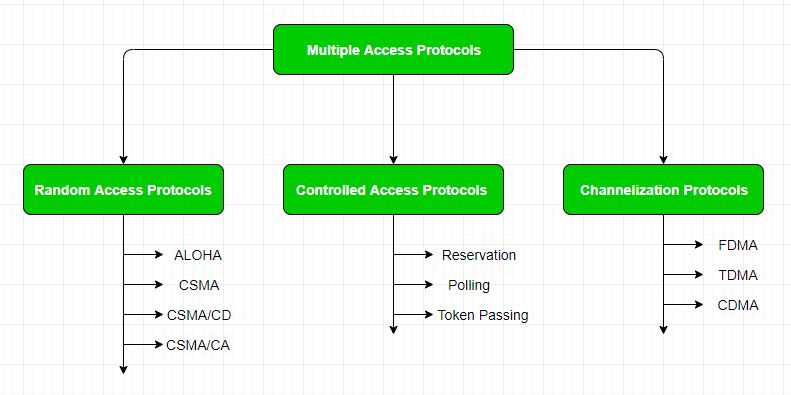
The Data Link Layer is responsible for transmission of data between two nodes. Its main functions are-

* Data Link Control
* Multiple Access Control

**Data Link control –**   
The data link control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control. For Data link control refer to – [Stop and Wait ARQ](https://www.geeksforgeeks.org/stop-and-wait-arq/)

**Multiple Access Control –**   
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 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.

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



**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:

1. There is no fixed time for sending data
2. There is no fixed sequence of stations sending data

The Random access protocols are further subdivided as:

Following are the different methods of random-access protocols for broadcasting frames on the channel.

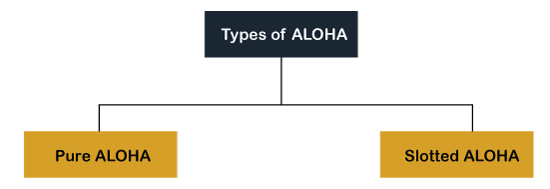
* Aloha
* CSMA
* CSMA/CD
* CSMA/CA

### ALOHA Random Access Protocol

It is designed for wireless LAN (Local Area Network) but can also be used in a shared medium to transmit data. Using this method, any station can transmit data across a network simultaneously when a data frameset is available for transmission.

**Aloha Rules**

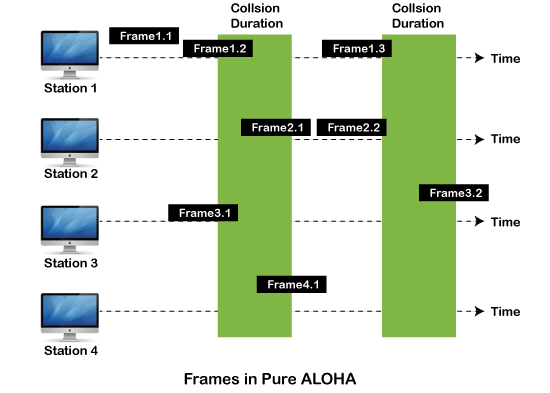
1. Any station can transmit data to a channel at any time.
2. It does not require any carrier sensing.
3. Collision and data frames may be lost during the transmission of data through multiple stations.
4. Acknowledgment of the frames exists in Aloha. Hence, there is no collision detection.
5. It requires retransmission of data after some random amount of time.



**Pure Aloha**

Whenever data is available for sending over a channel at stations, we use Pure Aloha. In pure Aloha, when each station transmits data to a channel without checking whether the channel is idle or not, 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.

1. The total vulnerable time of pure Aloha is 2 \* Tfr.
2. Maximum throughput occurs when G = 1/ 2 that is 18.4%.
3. Successful transmission of data frame is S = G \* e ^ - 2 G.

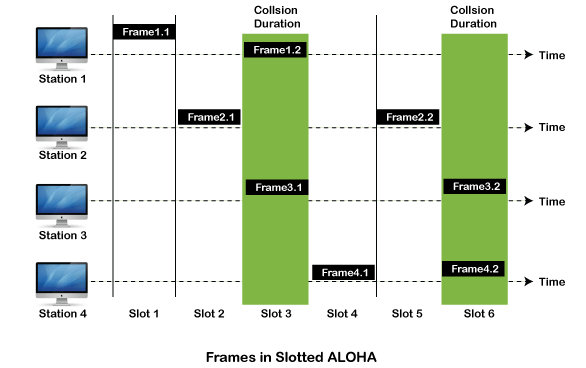


As we can see in the figure above, there are four stations for accessing a shared channel and transmitting data frames. Some frames collide because most stations send their frames at the same time. Only two frames, frame 1.1 and frame 2.2, are successfully transmitted to the receiver end. At the same time, other frames are lost or destroyed. Whenever two frames fall on a shared channel simultaneously, collisions can occur, and both will suffer damage. If the new frame's first bit enters the channel before finishing the last bit of the second frame. Both frames are completely finished, and both stations must retransmit the data frame.

**Slotted Aloha**

The slotted Aloha is designed to overcome the pure Aloha's efficiency because pure Aloha has a very high possibility of frame hitting. In slotted Aloha, the shared channel is divided into a fixed time interval called **slots**. So that, if a station wants to send a frame to a shared channel, the frame can only be sent at the beginning of the slot, and only one frame is allowed to be sent to each slot. And if the stations are unable to send data to the beginning of the slot, the station will have to wait until the beginning of the slot for the next time. However, the possibility of a collision remains when trying to send a frame at the beginning of two or more station time slot.

1. Maximum throughput occurs in the slotted Aloha when G = 1 that is 37%.
2. The probability of successfully transmitting the data frame in the slotted Aloha is S = G \* e ^ - 2 G.
3. The total vulnerable time required in slotted Aloha is Tfr.



### CSMA (Carrier Sense Multiple Access)

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

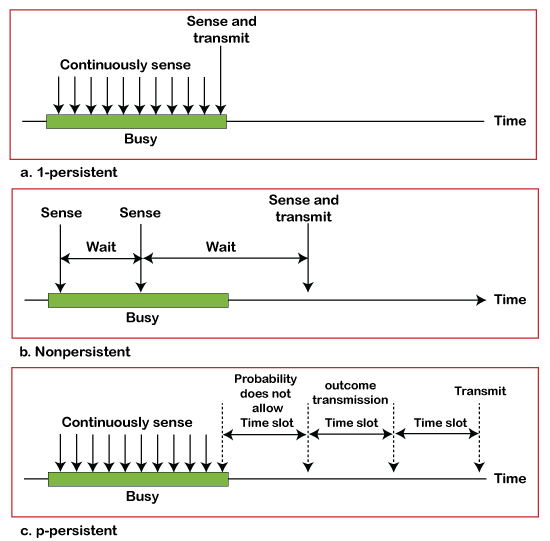
**CSMA Access Modes**

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

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

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

**O- Persistent:** It is an O-persistent method that defines the superiority of the station before the transmission of the frame on the shared channel. If it is found that the channel is inactive, each station waits for its turn to retransmit the data.



### CSMA/ CD

It is a **carrier sense multiple access/ collision detection** network protocol to transmit data frames. The CSMA/CD protocol works with a medium access control layer. Therefore, it first senses the shared channel before broadcasting the frames, and if the channel is idle, it transmits a frame to check whether the transmission was successful. If the frame is successfully received, the station sends another frame. If any collision is detected in the CSMA/CD, the station sends a jam/ stop signal to the shared channel to terminate data transmission. After that, it waits for a random time before sending a frame to a channel.

### CSMA/ CA

It is a **carrier sense multiple access/collision avoidance** network protocol for carrier transmission of data frames. It is a protocol that works with a medium access control layer. When a data frame is sent to a channel, it receives an acknowledgment to check whether the channel is clear. If the station receives only a single (own) acknowledgments, that means the data frame has been successfully transmitted to the receiver. But if it gets two signals (its own and one more in which the collision of frames), a collision of the frame occurs in the shared channel. Detects the collision of the frame when a sender receives an acknowledgment signal.

Following are the methods used in the [CSMA/ CA](https://www.javatpoint.com/csma-ca-vs-csma-cd) to avoid the collision:

**Interframe space**: In this method, the station waits for the channel to become idle, and if it gets the channel is idle, it does not immediately send the data. Instead of this, it waits for some time, and this time period is called the **Interframe** space or IFS. However, the IFS time is often used to define the priority of the station.

**Contention window**: In the Contention window, the total time is divided into different slots. When the station/ sender is ready to transmit the data frame, it chooses a random slot number of slots as **wait time**. If the channel is still busy, it does not restart the entire process, except that it restarts the timer only to send data packets when the channel is inactive.

**Acknowledgment**: In the acknowledgment method, the sender station sends the data frame to the shared channel if the acknowledgment is not received ahead of time.

### B. Controlled Access Protocol

In controlled access, the stations seek information from one another to find which station has the right to send. It allows only one node to send at a time, to avoid collision of messages on shared medium. The three controlled-access methods are:

1. Reservation
2. Polling
3. Token Passing

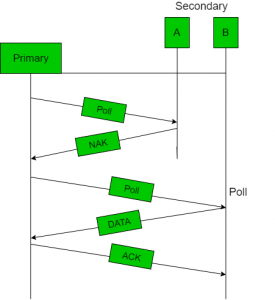
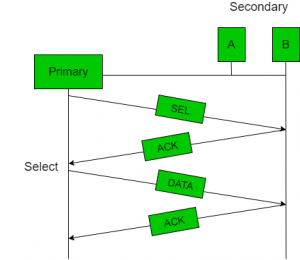
**Reservation**

* In the reservation method, a station needs to make a reservation before sending data.
* The time line has two kinds of periods:
  1. Reservation interval of fixed time length
  2. Data transmission period of variable frames.
* If there are M stations, the reservation interval is divided into M slots, and each station has one slot.
* Suppose if station 1 has a frame to send, it transmits 1 bit during the slot 1. No other station is allowed to transmit during this slot.
* In general, i th station may announce that it has a frame to send by inserting a 1 bit into i th slot. After all N slots have been checked, each station knows which stations wish to transmit.
* The stations which have reserved their slots transfer their frames in that order.
* After data transmission period, next reservation interval begins.
* Since everyone agrees on who goes next, there will never be any collisions.

The following figure shows a situation with five stations and a five-slot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation. https://media.geeksforgeeks.org/wp-content/uploads/re.png

**Polling**

* Polling process is similar to the roll-call performed in class. Just like the teacher, a controller sends a message to each node in turn.
* In this, one acts as a primary station(controller) and the others are secondary stations. All data exchanges must be made through the controller.
* The message sent by the controller contains the address of the node being selected for granting access.
* Although all nodes receive the message but the addressed one responds to it and sends data, if any. If there is no data, usually a “poll reject”(NAK) message is sent back.
* Problems include high overhead of the polling messages and high dependence on the reliability of the controller.

**Efficiency** Let Tpoll be the time for polling and Tt be the time required for transmission of data. Then,

Efficiency = Tt/(Tt + Tpoll)

**Token Passing**

* In token passing scheme, the stations are connected logically to each other in form of ring and access to stations is governed by tokens.
* A token is a special bit pattern or a small message, which circulate from one station to the next in some predefined order.
* In Token ring, token is passed from one station to another adjacent station in the ring whereas incase of Token bus, each station uses the bus to send the token to the next station in some predefined order.
* In both cases, token represents permission to send. If a station has a frame queued for transmission when it receives the token, it can send that frame before it passes the token to the next station. If it has no queued frame, it passes the token simply.
* After sending a frame, each station must wait for all N stations (including itself) to send the token to their neighbours and the other N – 1 stations to send a frame, if they have one.
* There exists problems like duplication of token or token is lost or insertion of new station, removal of a station, which need be tackled for correct and reliable operation of this scheme.

https://media.geeksforgeeks.org/wp-content/uploads/token.png

**Performance** Performance of token ring can be concluded by 2 parameters:-

1. **Delay**, which is a measure of time between when a packet is ready and when it is delivered. So, the average time (delay) required to send a token to the next station = a/N.
2. **Throughput**, which is a measure of the successful traffic.

Throughput, S = 1/(1 + a/N) for a<1

and

S = 1/{a(1 + 1/N)} for a>1.

where N = number of stations

a = Tp/Tt

(Tp = propagation delay and Tt = transmission delay)

### C. Channelization Protocols

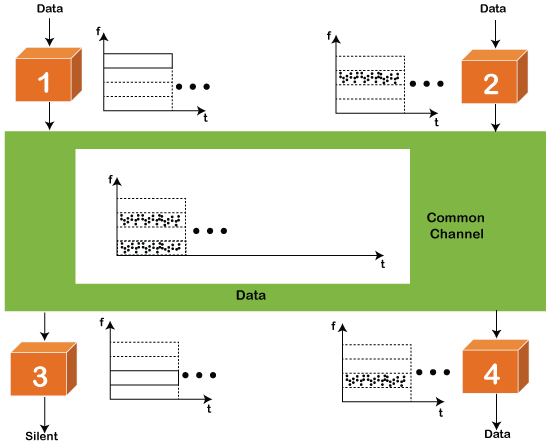
It is a channelization protocol that allows the total usable bandwidth in a shared channel to be shared across multiple stations based on their time, distance and codes. It can access all the stations at the same time to send the data frames to the channel.

Following are the various methods to access the channel based on their time, distance and codes:

1. FDMA (Frequency Division Multiple Access)
2. TDMA (Time Division Multiple Access)
3. CDMA (Code Division Multiple Access)

**FDMA**

It is a frequency division multiple access (**FDMA**) method used to divide the available bandwidth into equal bands so that multiple users can send data through a different frequency to the subchannel. Each station is reserved with a particular band to prevent the crosstalk between the channels and interferences of stations.



**TDMA**

Time Division Multiple Access (**TDMA**) is a channel access method. It allows the same frequency bandwidth to be shared across multiple stations. And to avoid collisions in the shared channel, it divides the channel into different frequency slots that allocate stations to transmit the data frames. The same **frequency** bandwidth into the shared channel by dividing the signal into various time slots to transmit it. However, TDMA has an overhead of synchronization that specifies each station's time slot by adding synchronization bits to each slot.

**CDMA**

The [code division multiple access (CDMA)](https://www.javatpoint.com/cdma-full-form) is a channel access method. In CDMA, all stations can simultaneously send the data over the same channel. It means that it allows each station to transmit the data frames with full frequency on the shared channel at all times. It does not require the division of bandwidth on a shared channel based on time slots. If multiple stations send data to a channel simultaneously, their data frames are separated by a unique code sequence. Each station has a different unique code for transmitting the data over a shared channel. For example, there are multiple users in a room that are continuously speaking. Data is received by the users if only two-person interact with each other using the same language. Similarly, in the network, if different stations communicate with each other simultaneously with different code language.

# Collision-Free Protocols

Almost collisions can be avoided in **CSMA/CD**.they can still occur during the contention period.the collision during contention period adversely affects the system performance, this happens when the cable is long and length of packet are short. This problem becomes serious as fiber optics network come into use. Here we shall discuss some protocols that resolve the collision during the contention period.

* Bit-map Protocol
* Binary Countdown
* Limited Contention Protocols
* The Adaptive Tree Walk Protocol

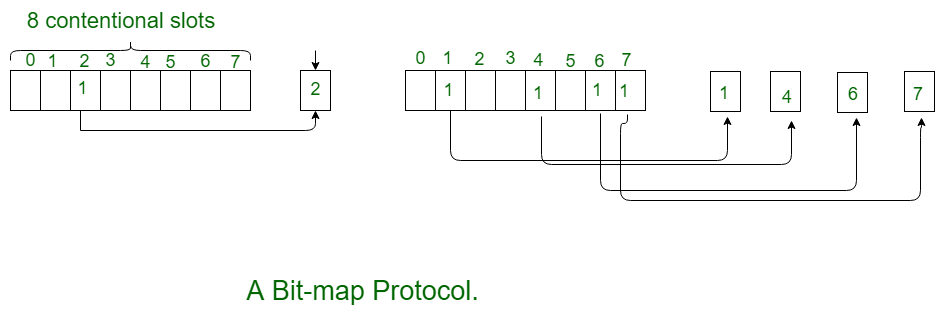
Pure and slotted Aloha, CSMA and CSMA/CD are **Contention based Protocols:**

* Try-if collide-Retry
* No guarantee of performance
  + - What happen if the network load is high?

**Collision Free Protocols:**

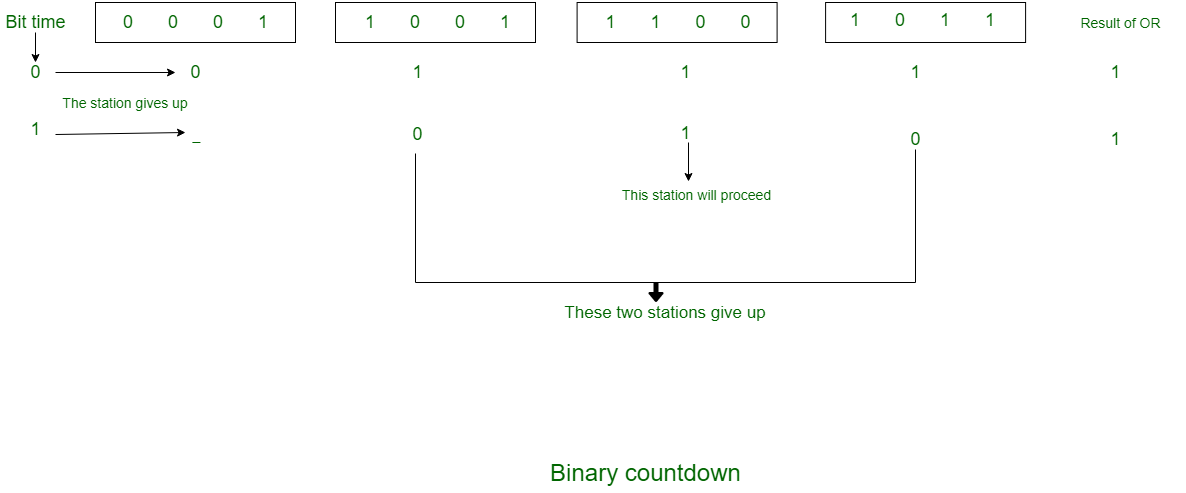
* + - Pay constant overhead to achieve performance guarantee
    - Good when network load is high

**1. Bit-map Protocol:**  
Bit map protocol is collision free Protocol in In bitmap protocol method, each contention period consists of exactly N slots. if any station has to send frame, then it transmits a 1 bit in the respective slot. For example if station 2 has a frame to send, it transmits a 1 bit during the second slot.

In general Station 1 Announce the fact that it has a frame questions by inserting a 1 bit into slot 1. In this way, each station has complete knowledge of which station wishes to transmit. There will never be any collisions because everyone agrees on who goes next. Protocols like this in which the desire to transmit is broadcasting for the actual transmission are called *Reservation Protocols*.  
  
  
  
  
  
For analyzing the performance of this protocol, We will measure time in units of the contention bits slot, with a data frame consisting of *d* time units. Under low load conditions, the bitmap will simply be repeated over and over, for lack of data frames.All the stations have something to send all the time at high load, the N bit contention period is prorated over N frames, yielding an overhead of only 1 bit per frame.

Generally, high numbered stations have to wait for half a scan before starting to transmit low numbered stations have to wait for half a scan(N/2 bit slots) before starting to transmit, low numbered stations have to wait on an average 1.5 N slots.  
  
  
**2. Binary Countdown:**  
Binary countdown protocol is used to overcome the overhead 1 bit per binary station. In binary countdown, binary station addresses are used. A station wanting to use the channel broadcast its address as binary bit string starting with the high order bit. All addresses are assumed of the same length. Here, we will see the example to illustrate the working of the binary countdown.

In this method, different station addresses are ORed together who decide the priority of transmitting. If these stations 0001, 1001, 1100, 1011 all are trying to seize the channel for transmission. All the station at first broadcast their most significant address bit that is 0, 1, 1, 1 respectively. The most significant bits are ORed together. Station 0001 see the 1MSB in another station addresses and knows that a higher numbered station is competing for the channel, so it gives up for the current round.

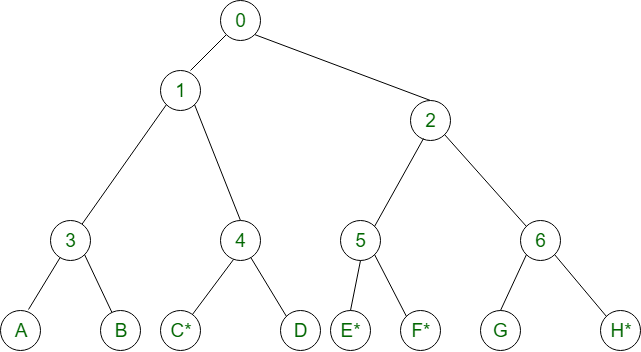
Other three stations 1001, 1100, 1011 continue. The next bit is 1 at station 1100, swiss station 1011 and 1001 give up. Then station 110 starts transmitting a frame, after which another bidding cycle starts.  
  
  
  
  
  
**Limited Contention Protocols:**

* + - Collision based protocols (pure and slotted ALOHA, CSMA/CD) are good when the network load is low.
    - Collision free protocols (bitmap, binary Countdown) are good when load is high.
    - How about combining their advantages
      1. Behave like the ALOHA scheme under light load
      2. Behave like the bitmap scheme under heavy load.

**Adaptive Tree Walk Protocol:**

* + - 1. partition the group of station and limit the contention for each slot.
      2. Under light load, everyone can try for each slot like aloha
      3. Under heavy load, only a group can try for each slot
      4. **How do we do it:**
         1. treat every stations as the leaf of a binary tree
         2. first slot (after successful transmission), all stations  
            can try to get the slot(under the root node).
         3. if no conflict, fine
         4. in case of conflict, only nodes under a subtree get to try for the next one. (depth first search)

**For Example:**



* + - 1. **Slot-0:** C\*, E\*, F\*, H\* (all nodes under node 0 can try which are going to send), conflict
      2. **Slot-1:** C\* (all nodes under node 1can try}, C sends
      3. **Slot-2:** E\*, F\*, H\*(all nodes under node 2 can try}, conflict
      4. **Slot-3:** E\*, F\* (all nodes under node 5 can try to send), conflict
      5. **Slot-4:** E\* (all nodes under E can try), E sends
      6. **Slot-5:** F\* (all nodes under F can try), F sends
      7. **Slot-6:** H\* (all nodes under node 6 can try to send), H sends.

# Ethernet

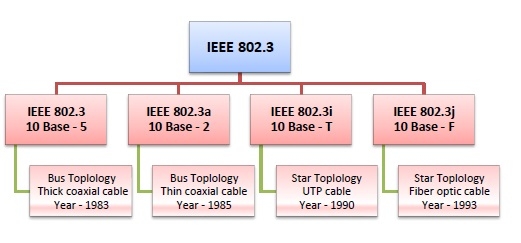
Ethernet is a type of communication protocol that is created at Xerox PARC in 1973 by Robert Metcalfe and others, which connects computers on a network over a wired connection. It is a widely used LAN protocol, which is also known as Alto Aloha Network. It connects computers within the local area network and wide area network. Numerous devices like printers and laptops can be connected by [LAN and WAN](https://www.javatpoint.com/lan-vs-wan) within buildings, homes, and even small neighborhoods.

# IEEE 802.3

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. IEEE 802.3 defines the physical layer and the medium access control (MAC) sub-layer of the data link layer for wired Ethernet networks.

IEEE 802.3 Popular Versions

There are a number of versions of IEEE 802.3 protocol. The most popular ones are.

* **IEEE 802.3**: This was the original standard given for 10BASE-5. It used a thick single coaxial cable into which a connection can be tapped by drilling into the cable to the core. Here, 10 is the maximum throughput, i.e. 10 Mbps, BASE denoted use of baseband transmission, and 5 refers to the maximum segment length of 500m.
* **IEEE 802.3a**: This gave the standard for thin coax (10BASE-2), which is a thinner variety where the segments of coaxial cables are connected by BNC connectors. The 2 refers to the maximum segment length of about 200m (185m to be precise).
* **IEEE 802.3i**: This gave the standard for twisted pair (10BASE-T) that uses unshielded twisted pair (UTP) copper wires as physical layer medium. The further variations were given by IEEE 802.3u for 100BASE-TX, 100BASE-T4 and 100BASE-FX.
* **IEEE 802.3i**: This gave the standard for Ethernet over Fiber (10BASE-F) that uses fiber optic cables as medium of transmission.

Frame Format of IEEE 802.3

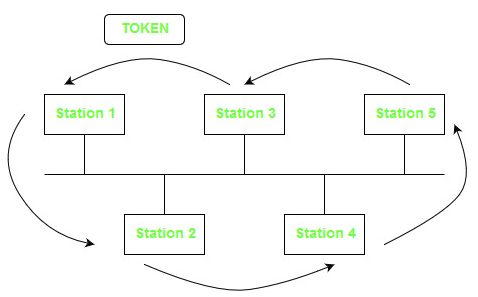
The main fields of a frame of classic Ethernet are -

* **Preamble**: It is a 7 bytes starting field that provides alert and timing pulse for transmission.
* **Start of Frame Delimiter**: It is a 1 byte field that contains an alternating pattern of ones and zeros ending with two ones.
* **Destination Address**: It is a 6 byte field containing physical address of destination stations.
* **Source Address**: It is a 6 byte field containing the physical address of the sending station.
* **Length**: It a 7 bytes field that stores the number of bytes in the data field.
* **Data**: This is a variable sized field carries the data from the upper layers. The maximum size of data field is 1500 bytes.
* **Padding**: This is added to the data to bring its length to the minimum requirement of 46 bytes.
* **CRC**: CRC stands for cyclic redundancy check. It contains the error detection information.

# Token Bus (IEEE 802.4)

**Token Bus (IEEE 802.4)** is a popular standard for token passing LANs. In a token bus LAN, the physical media is a bus or a tree, and a logical ring is created using a coaxial cable. The token is passed from one user to another in a sequence (clockwise or anticlockwise). Each station knows the address of the station to its “left” and “right” as per the sequence in the logical ring. A station can only transmit data when it has the token. The working of a token bus is somewhat similar to [Token Ring](https://www.geeksforgeeks.org/computer-network-efficiency-token-ring/).

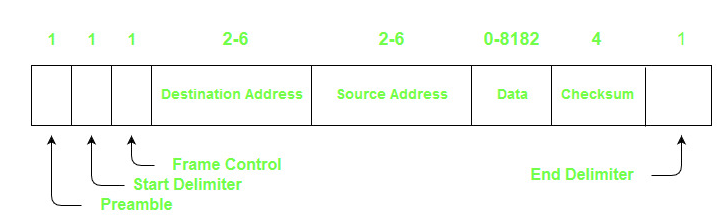
The Token Bus (IEEE 802.4) is a standard for deploying token rings in LANs over a virtual ring. The physical medium uses coaxial cables and has a bus or tree architecture. The nodes/stations form a virtual ring, and the token is transmitted from one node to the next in a sequence along the virtual ring. Each node knows the address of the station before it and the station after it. When a station has the token, it can only broadcast data. **The token bus works in a similar way as the Token Ring.**



The above diagram shows a logical ring formed in a bus-based token-passing LAN. The logical ring is shown with the arrows.

### Frame Format:

The various fields of the frame format are:



1. **Preamble –** It is used for bit synchronization. It is a 1-byte field.
2. **Start Delimiter –** These bits mark the beginning of the frame. It is a 1-byte field.
3. **Frame Control –** This field specifies the type of frame – data frame and control frames. It is a 1-byte field.
4. **Destination Address –** This field contains the destination address. It is a 2 to 6 bytes field.
5. **Source Address –** This field contains the source address. It is a 2 to 6 bytes field.
6. **Data –** If 2-byte addresses are used then the field may be up to 8182 bytes and 8174 bytes in the case of 6-byte addresses.
7. **Checksum –** This field contains the checksum bits which are used to detect errors in the transmitted data. It is 4 bytes field.
8. **End Delimiter –** This field marks the end of a frame. It is a 1-byte field.

### Ring topology has the following advantages:

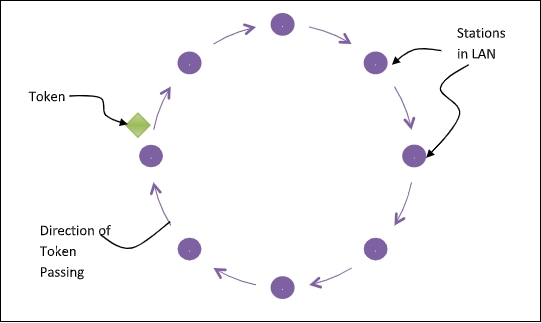
1. Data collisions are less likely because each node sends out a data packet after receiving the token.
2. Under heavy traffic, token passing makes ring topology perform better than bus topology.

## Token Ring (IEEE 802.5)

Token ring (IEEE 802.5) is a communication protocol in a local area network (LAN) where all stations are connected in a ring topology and pass one or more tokens for channel acquisition. A token is a special frame of 3 bytes that circulates along the ring of stations. A station can send data frames only if it holds a token. The tokens are released on successful receipt of the data frame.

### Token Passing Mechanism in Token Ring

If a station has a frame to transmit when it receives a token, it sends the frame and then passes the token to the next station; otherwise it simply passes the token to the next station. Passing the token means receiving the token from the preceding station and transmitting to the successor station. The data flow is unidirectional in the direction of the token passing. In order that tokens are not circulated infinitely, they are removed from the network once their purpose is completed. This is shown in the following diagram −



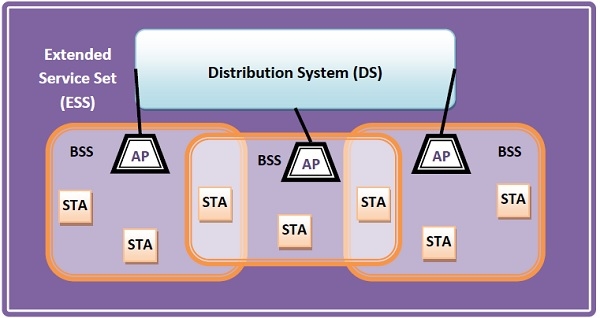
**IEEE 802.11**

IEEE 802.11 standard, popularly known as WiFi, lays down the architecture and specifications of wireless LANs (WLANs). WiFi or WLAN uses high-frequency radio waves instead of cables for connecting the devices in LAN. Users connected by WLANs can move around within the area of network coverage.

IEEE 802.11 Architecture

The components of an IEEE 802.11 architecture are as follows −

* **Stations (STA)** − Stations comprises of all devices and equipment that are connected to the wireless LAN. A station can be of two types−
  + Wireless Access Point (WAP) − WAPs or simply access points (AP) are generally wireless routers that form the base stations or access.
  + Client. Clients are workstations, computers, laptops, printers, smartphones, etc.
* Each station has a wireless network interface controller.
* **Basic Service Set (BSS)** − A basic service set is a group of stations communicating at the physical layer level. BSS can be of two categories depending upon the mode of operation−
  + Infrastructure BSS − Here, the devices communicate with other devices through access points.
  + Independent BSS − Here, the devices communicate in a peer-to-peer basis in an ad hoc manner.
* **Extended Service Set (ESS)** − It is a set of all connected BSS.
* **Distribution System (DS)** − It connects access points in ESS.



Frame Format of IEEE 802.11

The main fields of a frame of wireless LANs as laid down by IEEE 802.11 are −

* **Frame Control** − It is a 2 bytes starting field composed of 11 subfields. It contains control information of the frame.
* **Duration** − It is a 2-byte field that specifies the time period for which the frame and its acknowledgment occupy the channel.
* **Address fields** − There are three 6-byte address fields containing addresses of source, immediate destination, and final endpoint respectively.
* **Sequence** − It a 2 bytes field that stores the frame numbers.
* **Data** − This is a variable-sized field that carries the data from the upper layers. The maximum size of the data field is 2312 bytes.
* **Check Sequence** − It is a 4-byte field containing error detection information.

