Back To Course

Learn

Classroom

Theory

Quiz

Overview Learn Quiz Contest

We have combined Classroom and Theory tab and created a new Learn tab for easy access. You can access Classroom and Theory from the left panel.

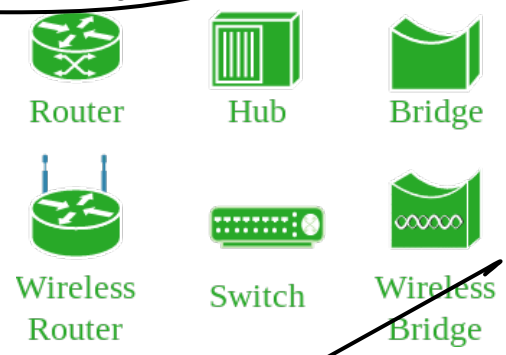
– Introduction to Computer Networks

Computer Networks is a communication network established between many electronic devices, not necessarily computers only for sharing resources and data. Such a network is established using physical links (such as cables, fiber, etc.) or can be wireless. (Wi-Fi, Bluetooth, etc.) We shall discuss the basic terminologies of computer networks in this tutorial, and then understand it's requirements and what goals it needs to attain.

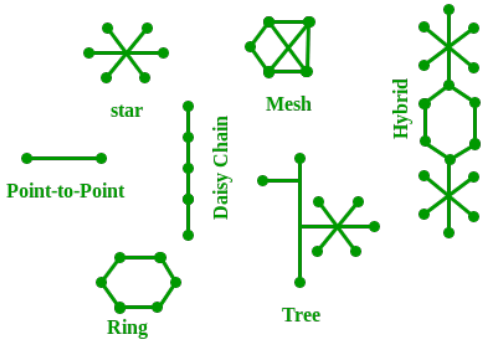
Open system - A system that is connected to the network and is ready for communication.

Closed system - A system that is not connected to the network and can't be communicated with.

Computer Network - It is the interconnection of multiple devices, generally termed as Hosts connected using multiple paths for the purpose of sending/receiving data or media.
There are also multiple devices or mediums which helps in the communication between two different devices which are known as **Network devices**. e.g. Router, Switch, Hub, Bridge



The layout pattern using which devices are interconnected is called as network topology. Such as Bus, Star, Mesh, Ring, Daisy chain.



OSI - OSI stands for **Open Systems Interconnection**. It is a reference model that specifies standards for communications protocols and also the functionalities of each layer.

Protocol - A protocol is the set of rules or algorithms which define the way how two entities can communicate across the network and there exists different protocol defined at each layer of the OSI model. A few of such protocols are TCP, IP, UDP, ARP, DHCP, FTP and so on.

Computer Network means an interconnection of autonomous (standalone) computers for information exchange. The connecting media could be a copper wire, optical fiber, microwave or satellite.

Networking Elements - The computer network includes the following networking elements:

1. At least two computers
2. Transmission medium either wired or wireless
3. Protocols or rules that govern the communication
4. Network software such as Network Operating System

Network Criteria - The criteria that have to be met by a computer network are:

1. **Performance** - It is measured in terms of transit time and response time.
    - Transit time is the time for a message to travel from one device to another
    - Response time is the elapsed time between an inquiry and a response.

    Performance is dependent on the following factors:
    - The number of users
    - Type of transmission medium
    - Capability of connected network
    - Efficiency of software
2. **Reliability** - It is measured in terms of
    - Frequency of failure
    - Recovery from failures
    - Robustness during catastrophe
3. **Security** - It means protecting data from unauthorized access.

Goals of Computer Networks - The following are some important goals of computer networks:

1. **Resource Sharing** - Many organizations have a substantial number of computers in operations, which are located apart. e.g. A group of office workers can share a common printer, fax, modem, scanner, etc.
2. **High Reliability** - If there are alternate sources of supply, all files could be replicated on two or, machines. If one of them is not available, due to hardware failure, the other copies could be used.

3. **Inter-process Communication** - Network users, located geographically apart, may converse in an interactive session through the network. In order to permit this, the network must provide almost error-free communications.

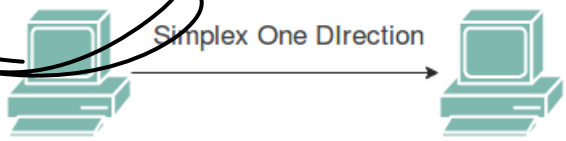
4. **Flexible access** - Files can be accessed from any computer in the network. The project can be begun on one computer and finished on another.

    Other goals include Distribution of processing functions, Centralized management, and allocation of network resources, Compatibility of dissimilar equipment and software, Good network performance, Scalability, Saving money, Access to remote information, Person to person communication, etc.

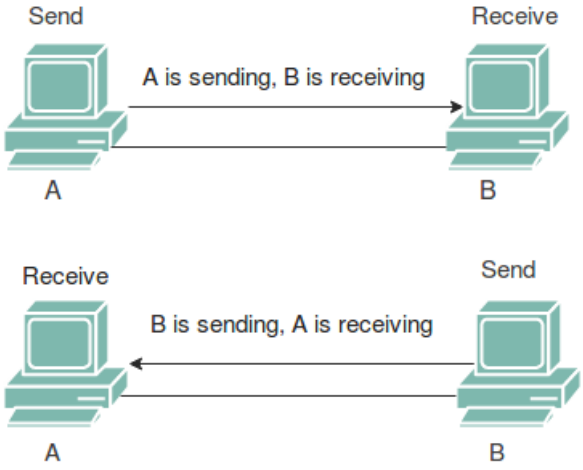
- Transmission Modes

Transmission Modes determine how data is transferred between two devices in a computer network. There are 3 transmission modes in computer networks given below:

Simplex Communication is uni-directional or one-way in Simplex Mode. i.e. Only one device is allowed to transfer data and the other device simply recieves it. e.g. Radio Station (The station transmits and the radio only recieves).



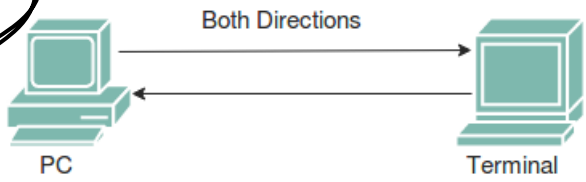
Half-Duplex Communication is possible both-ways but not simultaneously. i.e. Both the devices/stations can transmit and recieve data but not at the same time. At an instant, only communication in a single direction is allowed. e.g. Walkie-Talkie.



Full-Duplex Bidirectional communication is possible simultaneously. This can be possible in the following cases:

- Dedicated seperate channels for transmission and reception
- Capacity is divided between transmission and reception (if single channel is used)

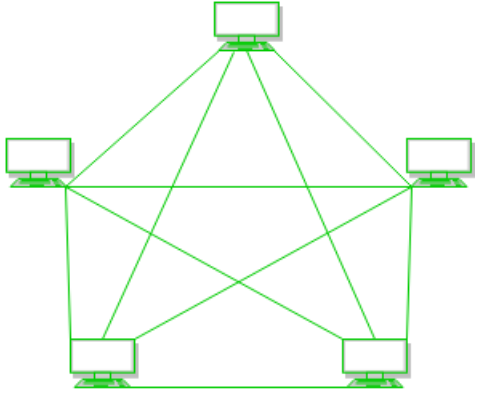
e.g. Cellular/Telephone Network.



- Network Topologies

The arrangement of nodes in a network generally follows some pattern or organization. Each of these patterns have their set of advantages/disadvantages. Such arrangements are called collectively referred to as network **topologies**. Some of the popular network topologies are as follows:

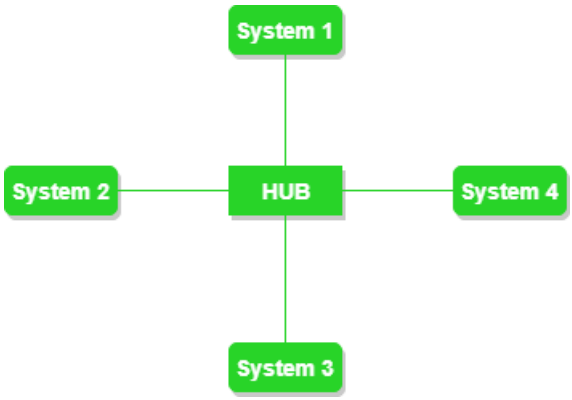
Mesh



Key points:

1. Robust & Easy fault-detection.
2. Installation is difficult & Expensive (fully-connected ~ lots of cable required $={}^{n}C_{2}$).

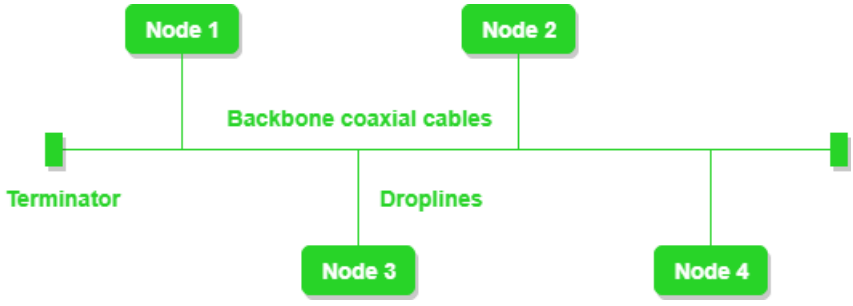
Star



Key points:

1. Easy & Cheap Installation (n cables required). Also device needs to have only 1 port.
2. Single point-of-failure (central node).

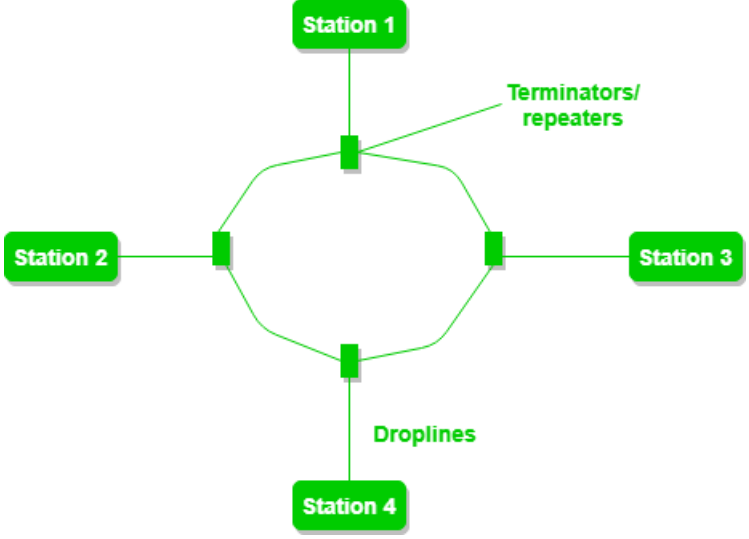
Bus



Key points:

1. Easy & Cheap Installation (n + 1(main-line) cables required).
2. Single line-of-failure (main-line).
3. Heavy Traffic causes collisions.

Ring



Key points:

1. Easy & Cheap Installation (1 line).
2. Difficulty in Troubleshooting.
3. Addition/Removal of nodes disturbs the topology.

Hybrid

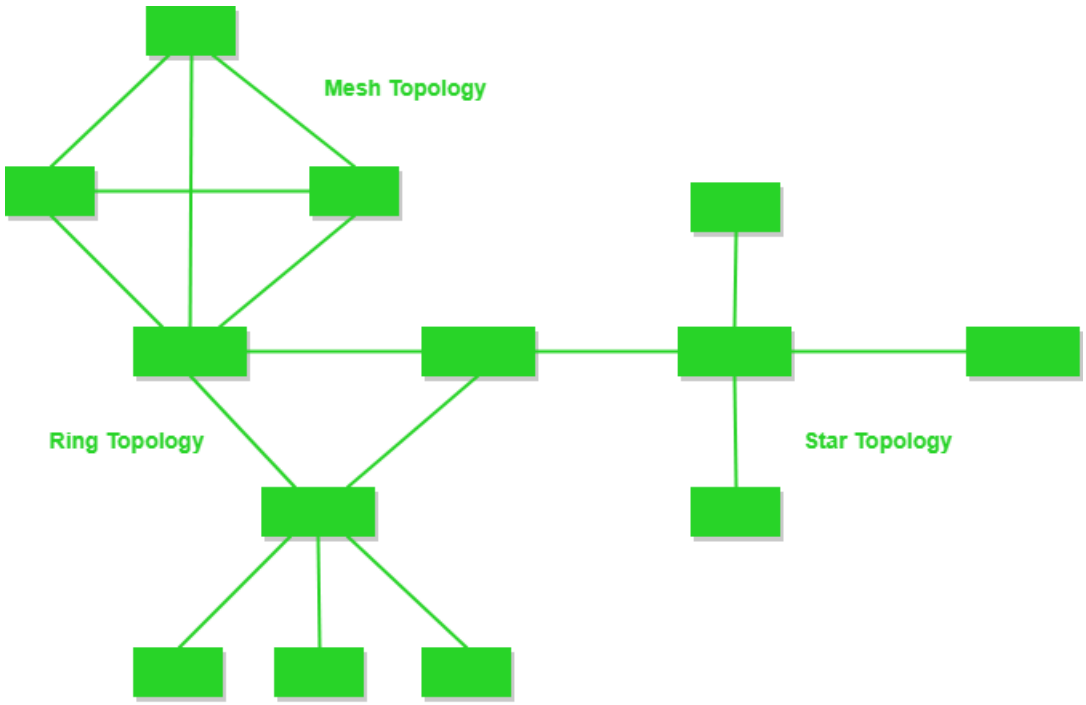


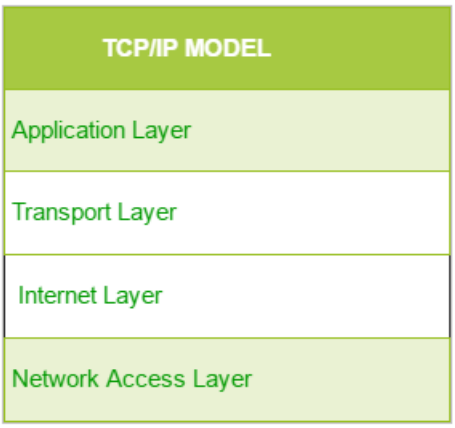
Figure - A Hybrid Topology

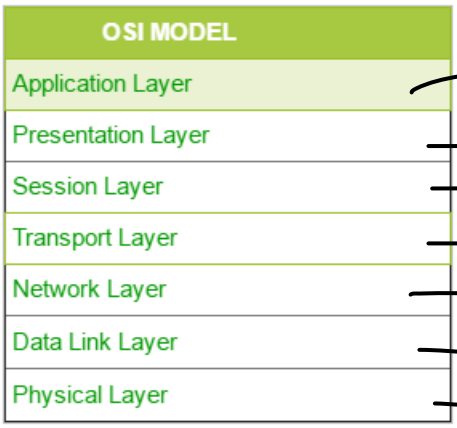
Key points:

1. Combination of all topologies (according to requirement).
2. This kind of topology is scalable and can serve a variety of requirements.
3. Due to intermixing of Ring, Bus, Star etc. topologies, it is difficult to develop. (As each of the individual topologies have their own rules and concepts ~ collision detection, protocols for data transfer etc.).

– TCP/IP vs OSI Model

TCP/IP and OSI are reference models which divides the various responsibilities into a logical separation of layers. Practical implementations in devices are based on these 2 popular models. The difference amongst them lies in the no. of layers and their respective responsibilities:





Brief description of each model:

OSI Model It stands for *Open Systems Interconnection*. It comprises of 7 layers with the following responsibilites (starting from the lowest layer):

- **Physical Layer**: It is responsible for actual physical transmission of data (through channels). It recieves/transmits signals and then converts it to physical bits (0 & 1). It handles *bit-synchronization* (using clock), *bit-rate* control (no.of bits/sec), physical *topology* and *transmission mode* (simplex, half-duplex, full-duplex).
- **Data-link Layer**: It is responsible for *Node-to-Node* delivery of packets, *Framing*, *Error control*, *Flow control*, *Physical Addressing (MAC)*. Upon recieving packets from network layer, it encapsulates it within a frame with the hardware (MAC) address of the reciever (obtained via *ARP* ~ Address Resolution Protocol).
- **Network Layer**: It is responsible for *Logical Addressing* (IPv4/v6) and *Routing*. Various routing algorithms are implemented at this layer, which determines the IP for the next hop in routing.
- **Transport Layer**: It is responsible for *End-to-end* delivery of packets. It also does *Segmentation & Reassembly* of packets(done if packet-size exceeds MTU ~ Max. Transmission Unit). It also does *multiplexing/de-multiplexing* of packets according to the application (using port no.). TCP/UDP (*Connection vs. Connection-less*) protocol is implemented at this layer.
- **Session Layer**: It is responsible for *Session Management* (Establishment, Maintenance, Termination), *Authentication*, *Security*, *Synchronization* & Restoration (check-points are established, such that upon re-connection state is resumed from the last saved point) and *Dialog Control* (synchronization when multiple parties are interacting ~ conference).
- **Presentation Layer**: It is responsible for *Translation* (e.g. ASCII to EBCDIC), *Encryption/Decryption* and *Compression*.
- **Application Layer**: Implements application-specific protocols (HTTP, HTTPS, FTP, SMTP etc.) They produce the data, interacts with the user (input and display of data). e.g. Browsers, Skype, Messaging Apps.

TCP/IP Model It comprises of 4 layers with the following responsibilities (starting from the lowest layer):

- **Network Access Layer**: It is a combination of the Physical and Data-Link Layer, and is responsible for data transmission and hardware addressing (MAC).
- **Internet Layer**: It is the counterpart of OSI's Network layer, and is responsible for routing and logical addressing. (IP, ICMP, ARP).
- **Transport Layer**: Maintains End-to-end connectivity. It is a counterpart of the OSI's transport layer, and has the same responsibilites (TCP vs. UDP).
- **Application Layer**: Application-specific protocols are implemented here. (HTTP, HTTPS, FTP, SMTP etc.)

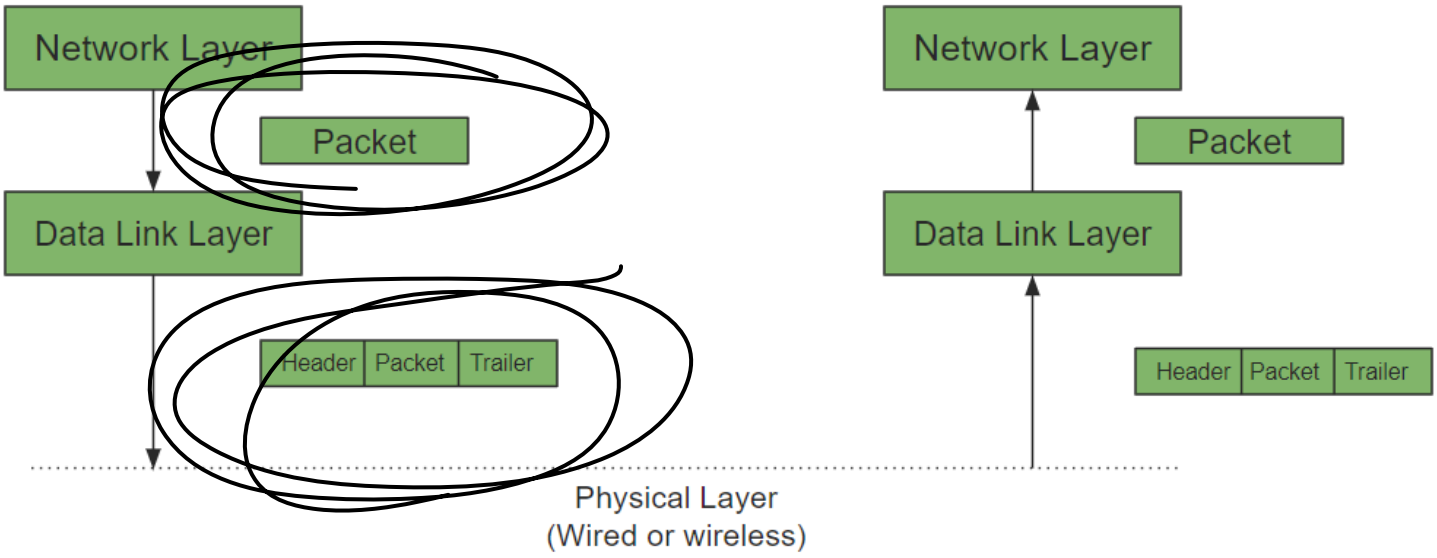
The differences amongst these two is tabulated as:

TCP/IP	OSI
TCP refers to Transmission Control Protocol.	OSI refers to Open Systems Interconnection.
TCP/IP has 4 layers.	OSI has 7 layers.
TCP/IP is more reliable	OSI is less reliable
TCP/IP does not have very strict boundaries.	OSI has strict boundaries
TCP/IP follow a horizontal approach.	OSI follows a vertical approach.
TCP/IP uses both session and presentation layer in the application layer itself.	OSI uses different session and presentation layers.
TCP/IP developed protocols then model.	OSI developed model then protocol.

– Data Link Layer

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. The working is as follows:



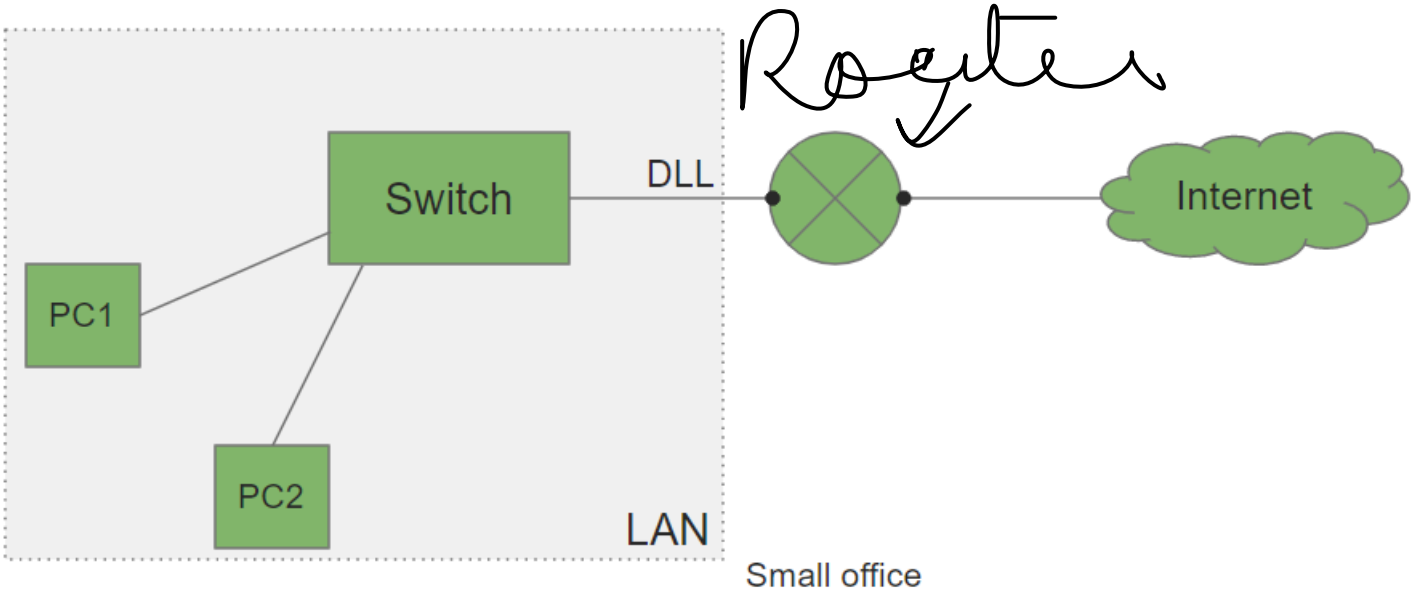
Data Link Layer is divided into two sub layers :

1. Logical Link Control (LLC)
2. Media Access Control (MAC)

The packet received from the Network layer is further divided into frames depending on the frame size of NIC(Network Interface Card). DLL also encapsulates Sender and Receiver's MAC address in the header.

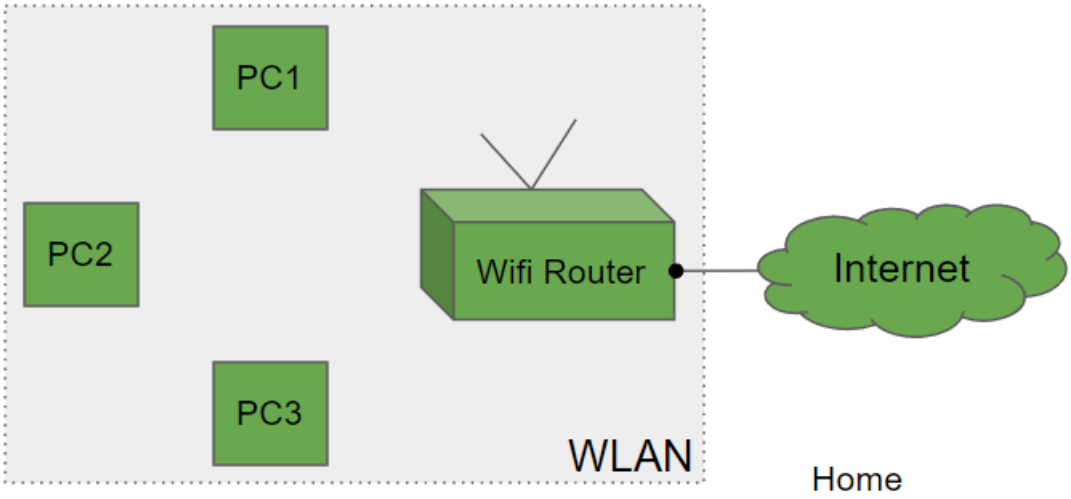
The Receiver's MAC address is obtained by placing an ARP(Address Resolution Protocol) request onto the wire asking "Who has that IP address?" and the destination host will reply with its MAC address.

Now let's see how the DLL works in a small office:



The switch is used to connect multiple computers or laptops which in turn is connected to a router. This is then connected to the internet. All the 1-to-1 connection is done using DLL. The setup is called LAN as they are all connected in Local Area Network.

Now let's see how the DLL works in a small office:



Here the router is used to convey the connection in wireless form. This is then connected to the internet. All the 1-to-1 connection is again done using DLL. The setup is called WLAN as they are all connected in Wireless Local Area Network. This network might have a collision.

The functions of the Data Link layer are :

1. **Framing:** 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. This can be accomplished by attaching special bit patterns to the beginning and end of the frame.
2. **Physical addressing:** After creating frames, Data link layer adds physical addresses (MAC address) of sender and/or receiver in the header of each frame.
3. **Error Detection:** Data link layer provides the mechanism of error control in which it detects and retransmits damaged or lost frames.
4. **Error and Flow Control:** The data rate must be constant on both sides else the data may get corrupted thus, flow control coordinates that amount of data that can be sent before receiving acknowledgement.
5. **Access control:** When a single communication channel is shared by multiple devices, MAC sub-layer of data link layer helps to determine which device has control over the channel at a given time.

** Packet in Data Link layer is referred as **Frame**.*
*** Data Link layer is handled by the NIC (Network Interface Card) and device drivers of host machines.*
**** Switch & Bridge are Data Link Layer devices.*

– Transmission and Propagation Delay

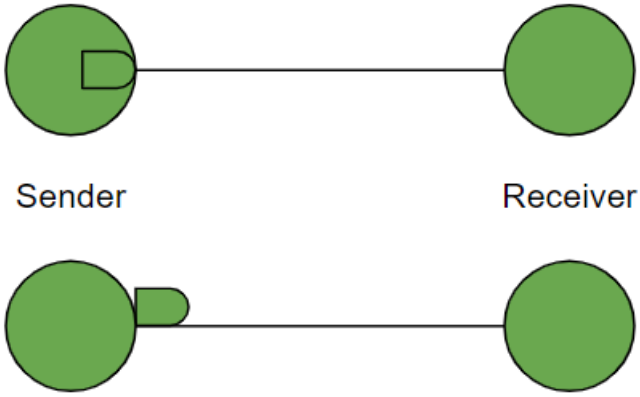
Delays in Packet switching :

1. Transmission Delay
2. Propagation Delay
3. Queuing Delay
4. Processing Delay

Transmission Delay :

Time taken to put a packet onto link. In other words, it is simply time required to put data bits on the wire/communication medium. It depends on length of packet and bandwidth of network.

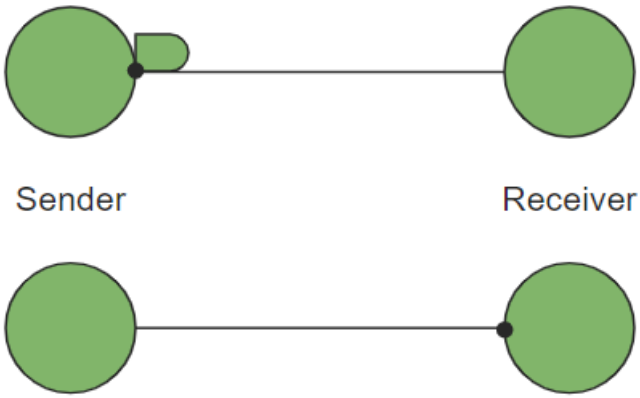
Transmission Delay = Data size / bandwidth = (L/B) second



For example,
Let the link bandwidth of a network be 100 bits/second and the packet length be 1000 bits. Therefore the tarnsmission delay for the following network will be 1000/100 = 10 seconds.

Propagation delay : Time is taken by the first bit to travel from sender to receiver end of the link. In other words, it is simply the time required for bits to reach the destination from the start point. Factors on which Propagation delay depends are Distance and propagation speed.

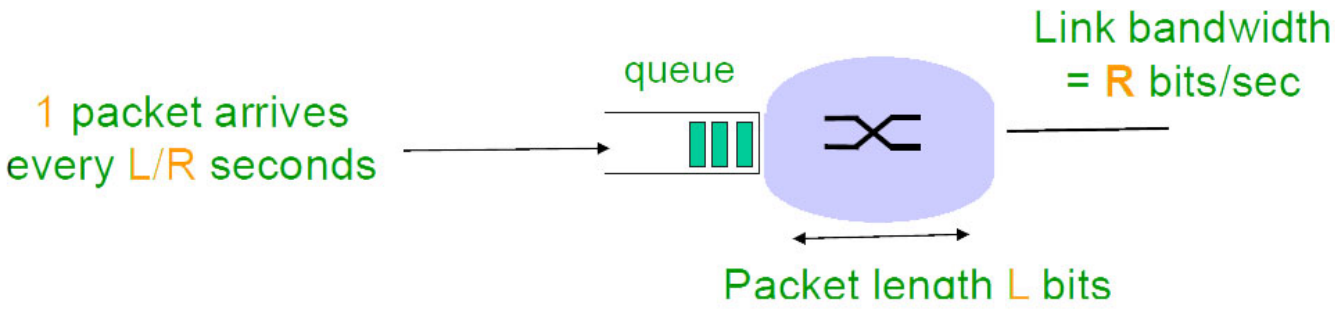
```
Propagation delay = distance/transmission speed = d/s
```



Lets take another example with the distance as 30 km, velocity being 3 x 10^8 meter/second.
Therefore, the propagation delay = distance/velocity = (30 * 10^3)/(3 * 10^8) = 0.1 milli-second.

Queuing Delay : Queuing delay is the time a job waits in a queue until it can be executed. It depends on congestion. It is the time difference between when the packet arrived Destination and when the packet data was processed or executed. It may be caused by mainly three reasons i.e. originating switches, intermediate switches or call receiver servicing switches.

Queueing delay



Arrival rate: a = 1/(L/R) = R/L (packet/second)

Traffic intensity = aL/R = (R/L) (L/R) = 1

Average queueing delay = 0
(queue is initially empty)

Average Queuing delay = (N-1)L/(2*R)

where N = no. of packets

L=size of packet

R=bandwidth

Processing Delay : Processing delay is the time it takes routers to process the packet header. Processing of packets helps in detecting bit-level errors that occur during transmission of a packet to the destination. Processing delays in high-speed routers are typically on the order of microseconds or less.
In simple words, it is just the time taken to process packets.

Total time *or* **End-to-End time**
= Transmission delay + Propagation delay+ Queuing delay
+ Processing delay

For M hops and N packets -

Total delay

= M*(Transmission delay + propagation delay)+

(M-1)*(Processing delay + Queuing delay) +

(N-1)*(Transmission delay)

- Stop and Wait ARQ

Characteristics

- Used in Connection-oriented communication.
- It offers error and flow control
- It is used in Data Link and Transport Layers
- Stop and Wait ARQ mainly implements Sliding Window Protocol concept with Window Size 1

Useful Terms:

- **Propagation Delay:** Amount of time taken by a packet to make a physical journey from one router to another router.

  Propagation Delay = (Distance between routers) / (Velocity of propagation)

- RoundTripTime (**RTT**) = 2* Propagation Delay
- TimeOut (**TO**) = 2* RTT
- Time To Live (**TTL**) = 2* TimeOut. (Maximum TTL is 180 seconds)

Simple Stop and Wait

Sender:

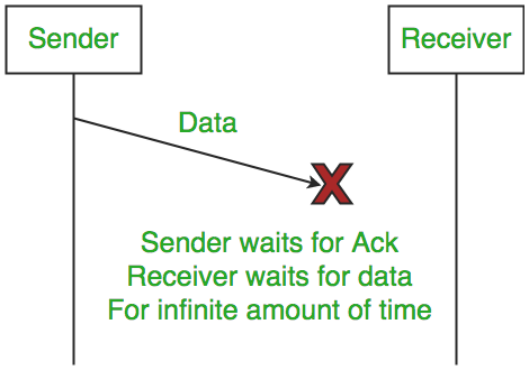
Rule 1) Send one data packet at a time.
Rule 2) Send next packet only after receiving acknowledgement for previous.

Receiver:

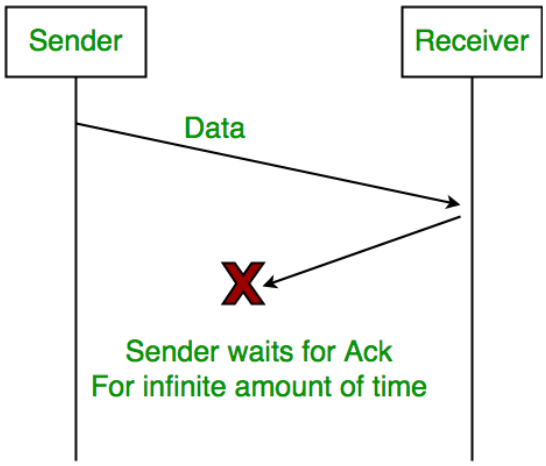
Rule 1) Send acknowledgement after receiving and consuming of the data packet.
Rule 2) After consuming packet acknowledgement need to be sent (Flow Control)

Problems :

1. Lost Data



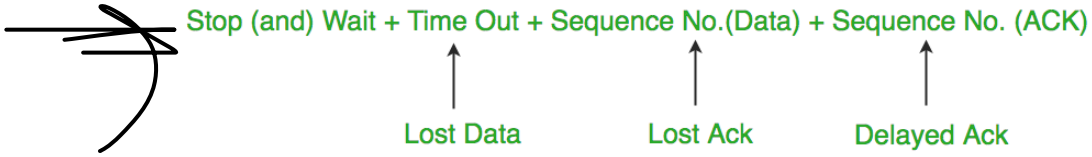
2. Lost Acknowledgement:



3. Delayed Acknowledgement/Data: After timeout on sender side, a long delayed acknowledgement might be wrongly considered as acknowledgement of some other recent packet.

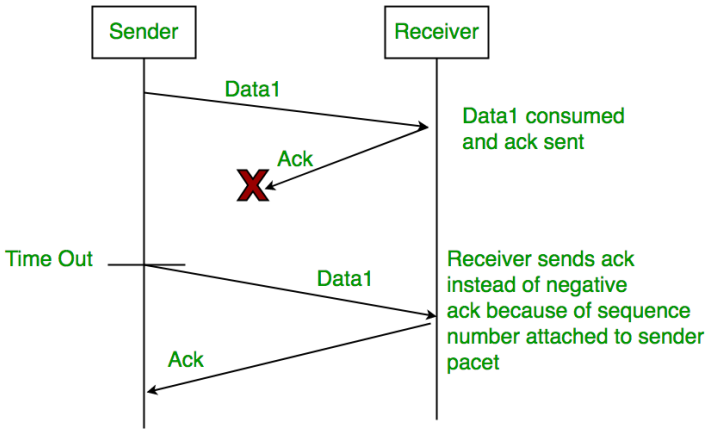
Stop and Wait ARQ (Automatic Repeat Request)

Above 3 problems are resolved by Stop and Wait ARQ (Automatic Repeat Request) that does both error control and flow control.



1. Time Out:

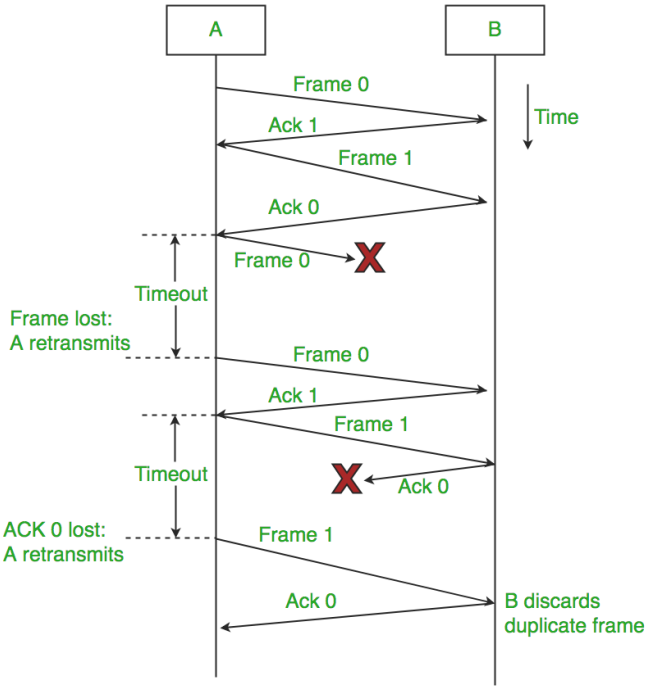
2. Sequence Number (Data)



3. Delayed Acknowledgement: This is resolved by introducing sequence number for acknowledgement also.

Working of Stop and Wait ARQ:

1) Sender A sends a data frame or packet with sequence number 0.
2) Receiver B, after receiving data frame, sends an acknowledgement with sequence number 1 (the sequence number of next expected data frame or packet)
There is only a one-bit sequence number that implies that both sender and receiver have a buffer for one frame or packet only.



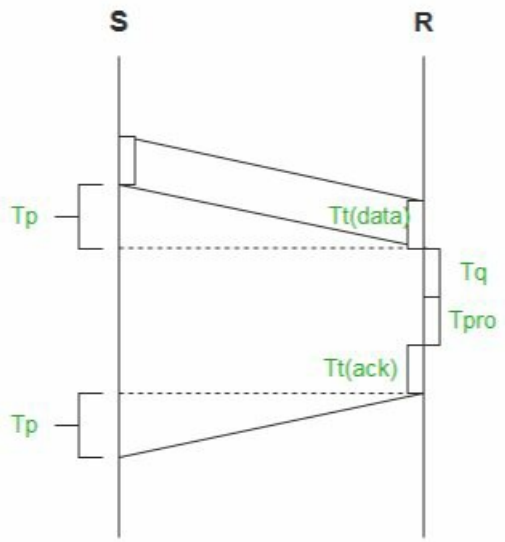
Characteristics of Stop and Wait ARQ:

- It uses the link between sender and receiver as half-duplex link
- Throughput = 1 Data packet/frame per RTT
- If Bandwidth*Delay product is very high, then stop and wait protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
- It is an example for "**Closed Loop OR connection-oriented** " protocols
- It is a special category of SWP where its window size is 1
- Irrespective of number of packets sender is having stop and wait for protocol requires only 2 sequence numbers 0 and 1

The Stop and Wait ARQ solves main three problems but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country through a high-speed connection). To solve this problem, we can send more than one packet at a time with a larger sequence number.

So Stop and Wait ARQ may work fine where propagation delay is very less, for example, LAN connections, but performs badly for distant connections like satellite connection.

Efficiency: Stop and Wait is a flow control protocol. In which the sender sends one packet and waits for the receiver to acknowledge and then it will send the next packet. In case if the acknowledgement is not received, the sender will retransmit the packet. This is the simplest one and easy to implement. but the main disadvantage is the efficiency is very low.



Total time taken to send one packet,

= Tt(data) + Tp(data) + Tq + Tpro + Tt(ack) + Tp(ack)

Since,

Tp(ack) = Tp(data)

And,

Tt(ack) << Tt(data).

So we can neglect Tt(ack)

Tq = 0 and Tpro = 0

Hence,

Total time = Tt(data) + 2 * Tp

Where,

Tt(data) : Transmission delay for Data packet

Tp(data) : propagation delay for Data packet

Tq: Queuing delay

Tpro: Processing delay

Tt(ack): Transmission delay for acknowledgment

Tp(ack) : Propagation delay for acknowledgment

We know that the **Efficiency (η)**,

= Useful time / Total cycle time.

= Tt / (Tt + 2*Tp)

= 1 / (1+2*(Tp/Tt))

= 1 / (1+2*a)

where,

a = Tp / Tt

Throughput: Number of bits send per second, which is also known as Effective Bandwidth or Bandwidth utilization.

Throughput,

= L/(Tt + 2*Tp)

= ((L/BW)*BW)/(Tt + 2*Tp)

= Tt/(Tt + 2*Tp) * BW

= 1/(1 + 2a) * BW

Hence, Throughput

= η * BW

where,

BW : BandWidth

L : Size of Data packet

Factors affecting Efficiency:

n = 1/(1 + 2*(Tp/Tt)

= 1/(1 + 2*(d/v)*(BW/L))

where,

d = distance between source and receiver

v = velocity

Let's see an example.

Example: Given,

Tt = 1ms

Tp = 2ms

Bandwidth = 6 Mbps

Efficiency(η)

= 1/(1 + a)

= 1/(1 + (2/1))

= 1/3

= 33.33 %

Throughput

= η * BW

= (1/3) * 6

= 2 Mbps

Note: As we can observe from the above given formula of Efficiency that:

1. On increasing the distance between source and receiver the Efficiency will decrease. Hence, Stop and Wait is only suitable for small area network like LAN. It is not suitable for MAN or WAN, as the efficiency will be very low.
2. If we increase the size of the Data packet, the efficiency is going to increase. Hence, it is suitable not for small packets. Big data packets can be send by Stop and Wait efficiently.

- Go Back N

Sliding Window Protocol is actually a theoretical concept in which we have only talked about what should be the sender window size (1+2a) in order to increase the efficiency of stop and wait arq. Now we will talk about the practical implementations in which we take care of what should be the size of receiver window. Practically it is implemented in two protocols namely :

1. Go Back N (GBN)
2. Selective Repeat (SR)

In this article, we will explain you about the first protocol which is GBN in terms of three main characteristic features and in the

last part we will be discussing SR as well as comparison of both these protocols

Sender Window Size (WS) It is N itself. If we say the protocol is GB10, then Ws = 10. N should be always greater than 1 in order to implement pipelining. For N = 1, it reduces to Stop and Wait protocol.

Efficiency Of GBN = N/(1+2a) Where a = Tp/Tt

If B is the bandwidth of the channel, then Effective Bandwidth or Throughput = Efficiency * Bandwidth = (N/(1+2a)) * B.

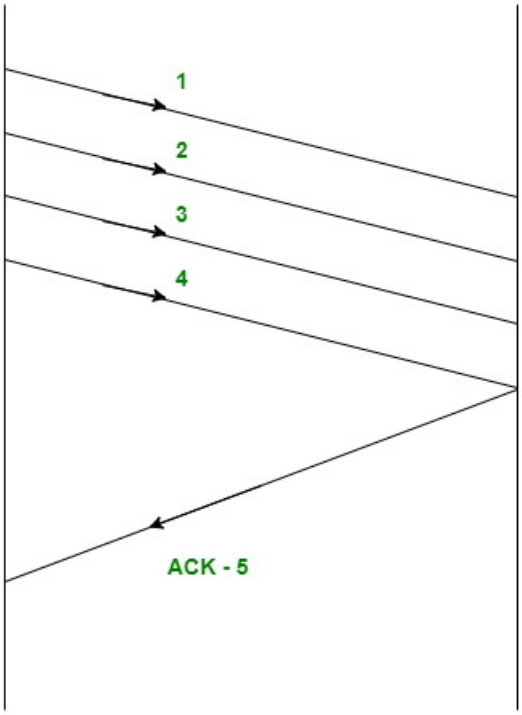
Receiver Window Size (WR) WR is always 1 in GBN.

Now what exactly happens in GBN, we will explain with the help of an example. Consider the diagram given below. We have a sender window size of 4. Assume that we have lots of sequence numbers just for the sake of explanation. Now the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, the receiver is now expecting packet 2 and sender window has also slide to further transmit the packets 4 and 5. Now suppose the packet 2 is lost in the network, the Receiver will discard all the packets which sender has transmitted after packet 2 as it is expecting sequence number of 2. On the sender side for every packet send there is a time out timer which will expire for packet number 2. Now from the last transmitted packet 5 senders will go back to the packet number 2 in the current window and transmit all the packets till packet number 5. That's why it is called Go Back N. Go back means the sender has to go back N places from the last transmitted packet in the unacknowledged window and not from the point where the packet is lost.

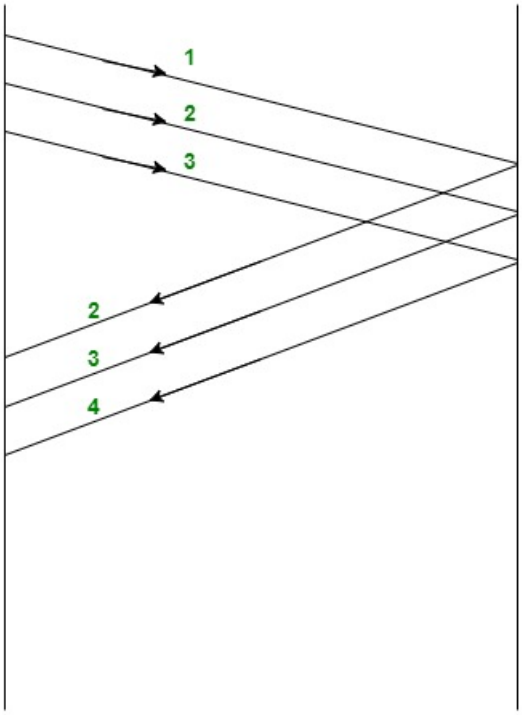
Acknowledgements

There are 2 kinds of acknowledgements namely :

- **Cumulative Ack** - One acknowledgement is used for many packets. Main advantage is traffic is less. Disadvantage is less reliability as if one ack is loss that would mean that all the packets sent are lost.
- **Independent Ack** - If every packet is going to get acknowledgement independently. Reliability is high here but disadvantage is that traffic is also high since for every packet we are receiving independent ack.



Cummulative



INDEPENDENT

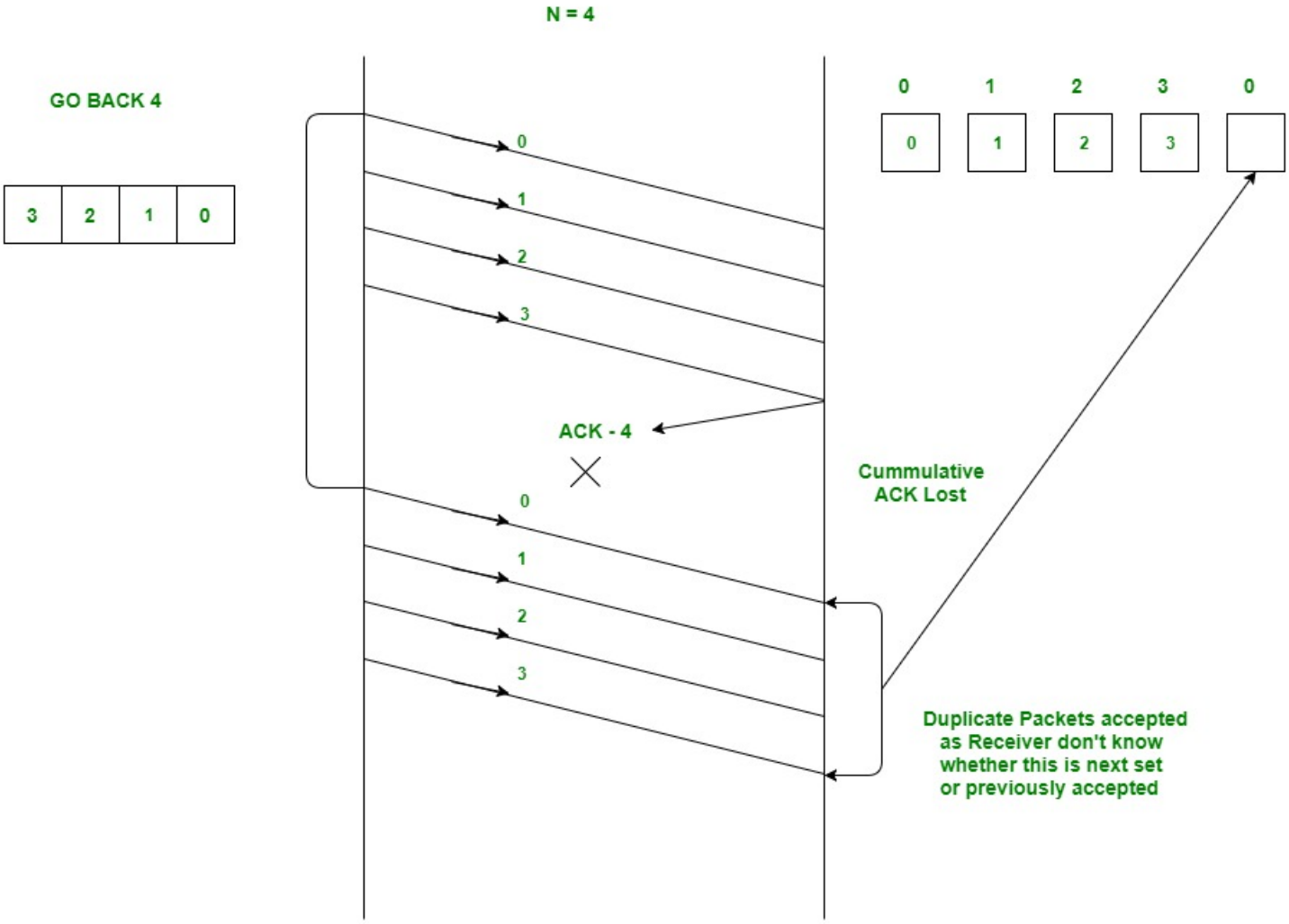
GBN uses Cumulative Acknowledgement. At the receiver side, it starts an acknowledgement timer whenever receiver receives any packet which is fixed and when it expires, it is going to send a cumulative Ack for the number of packets received in that interval of timer. If the receiver has received N packets, then the Acknowledgement number will be N+1. An important point is Acknowledgement timer will not start after the expiry of first-timer but after the receiver has received a packet.
Time out timer at the sender side should be greater than Acknowledgement timer.

Relationship Between Window Sizes and Sequence Numbers We already know that sequence numbers required should always be equal to the size of the window in any sliding window protocol.

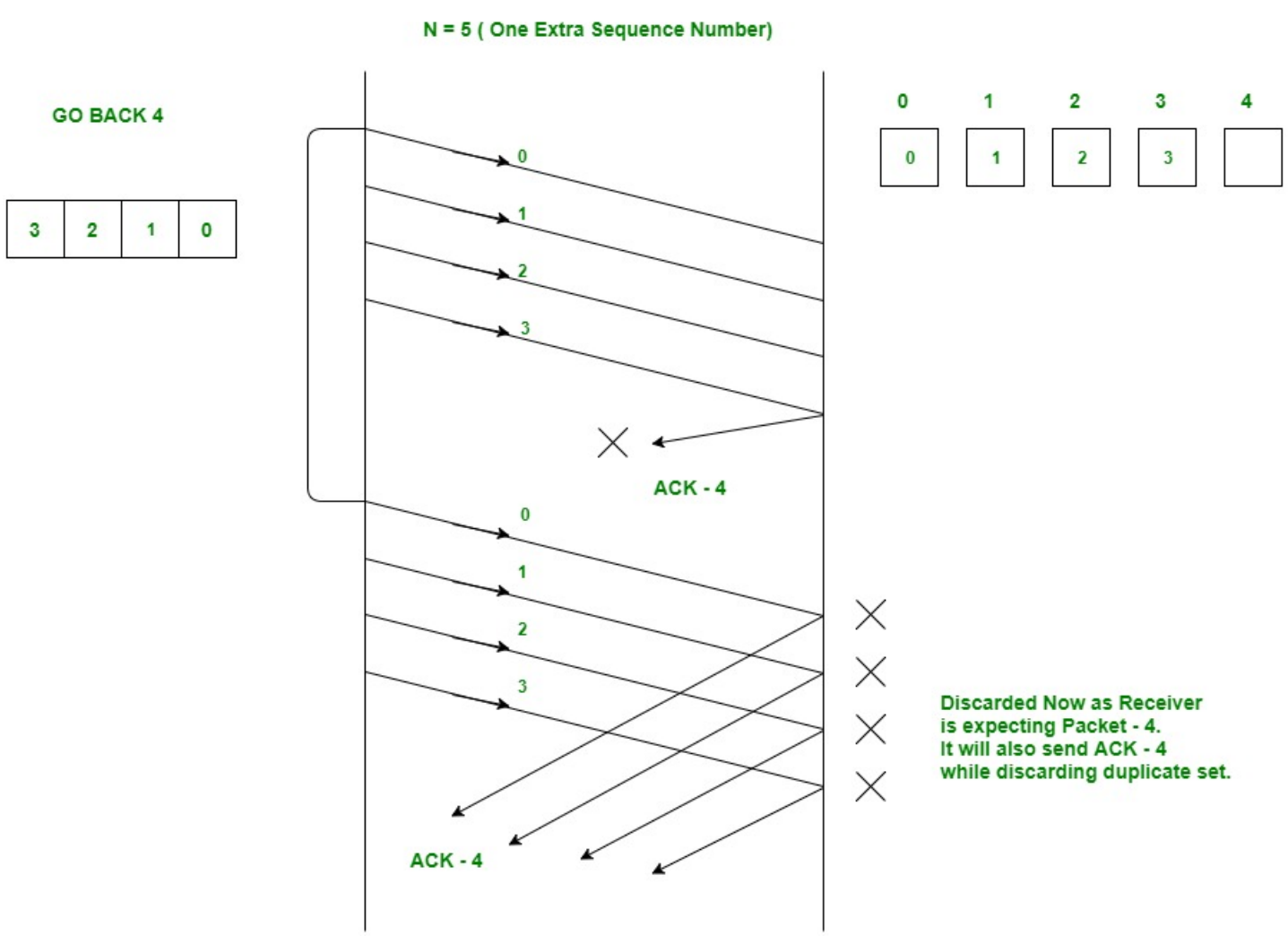
Minimum sequence numbers required in GBN is N+1.
Bits Required will be ceil(log2(N+1))
The extra 1 is required in order to avoid the problem of duplicate packets as described below.

Consider an example of GB4. Sender window size is 4, therefore, we require a minimum of 4 sequence numbers to label each packet in the window. Now suppose receiver has received all the packets(0, 1, 2 and 3 sent by sender) and hence is now waiting for packet number 0 again(We can not use 4 here as we have only 4 sequence numbers available since N = 4). Now suppose the cumulative ack for the above 4 packets is lost in the network. On the sender side, there will be a timeout for packet 0 and hence all the 4 packets will be transmitted again. The problem now is receiver is waiting for a new set of packets which should have started from 0 but now it will receive the duplicate copies of the previously accepted packets. In order to avoid this, we need one extra sequence number. Now the receive could easily reject all the duplicate packets which were starting from 0 because now it will be waiting for packet number 4(We have added an extra sequence number now).

Trying with Sequence numbers 4.



Now Trying with one extra Sequence Number.



In the next article, we will explain Selective repeat and comparison between the 2 protocols.

– Sliding Window Background

In sliding window protocol, the sender sends more than one frames to the receiver side and re-transmit the frame which are/is damaged or suspected. Efficiency of sliding window protocol is more than Stop-and-Wait Protocol. Sender window size of sliding window protocol is N. Receiver window size of sliding window protocol may 1 or N. In sliding window protocol, sorting may be or may not be necessary. The efficiency of the sliding window protocol is N/(1+2*a).

The Stop and Wait ARQ offers error and flow control, but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country though a high speed connection), you can't use this full speed due to limitations of stop and wait.

Sliding Window protocol handles this efficiency issue by sending more than one packet at a time with a larger sequence numbers. The idea is same as pipelining in architectures.

Few Terminologies :

Transmission Delay (Tt) - Time to transmit the packet from host to the outgoing link. If B is the Bandwidth of the link and D is the Data Size to transmit

```
Tt = D/B
```

Propagation Delay (Tp) - It is the time taken by the first bit transferred by the host onto the outgoing link to reach the destination. It depends on the distance d and the wave propagation speed s (depends on the characteristics of the medium).

```
Tp = d/s
```

Efficiency - It is defined as the ratio of total useful time to the total cycle time of a packet. For stop and wait protocol,

```
Total cycle time = Tt(data) + Tp(data) +
                   Tt(acknowledgement) + Tp(acknowledgement)
                 =  Tt(data) + Tp(data) + Tp(acknowledgement)
               =   Tt + 2*Tp
```

Since acknowledgements are very less in size, their transmission delay can be neglected.

```
Efficiency = Useful Time / Total Cycle Time
           = Tt/(Tt + 2*Tp) (For Stop and Wait)
           = 1/(1+2a)  [ Using a = Tp/Tt ]
```

Effective Bandwidth(EB) or Throughput - Number of bits sent per second.

```
EB = Data Size(L) / Total Cycle time(Tt + 2*Tp)
Multiplying and dividing by Bandwidth (B)
      =  (1/(1+2a)) * B   [ Using a = Tp/Tt ]
      =  Efficiency * Bandwidth
```

Capacity of link - If a channel is Full Duplex, then bits can be transferred in both the directions and without any collisions. Number of bits a channel/Link can hold at maximum is its capacity.

```
Capacity = Bandwidth(B) * Propagation(Tp)

For Full Duplex channels,
Capacity = 2*Bandwidth(B) * Propagation(Tp)
```

Concept Of Pipelining

In Stop and Wait protocol, only 1 packet is transmitted onto the link and then sender waits for acknowledgement from the receiver. The problem in this setup is that efficiency is very less as we are not filling the channel with more packets after 1st packet has been put onto the link. Within the total cycle time of Tt + 2*Tp units, we will now calculate the maximum number of packets that sender can transmit on the link before getting an acknowledgement.

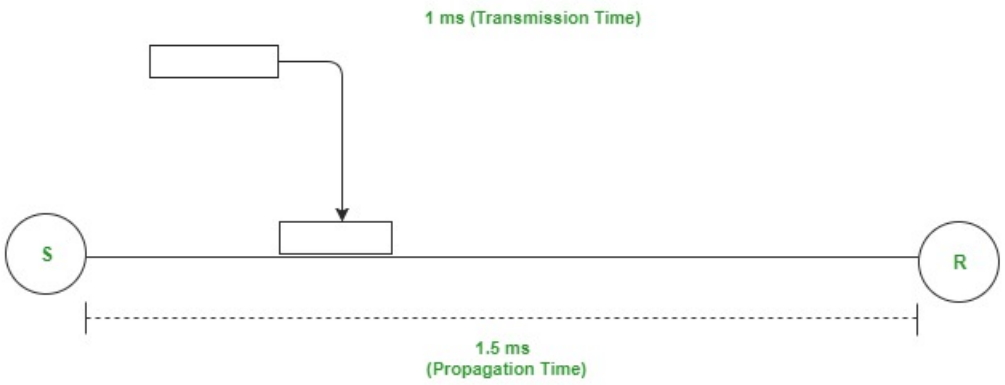
```
In Tt units ----> 1 packet is Transmitted.
In 1 units  ----> 1/Tt packet can be Transmitted.
In  Tt + 2*Tp units ----->  (Tt + 2*Tp)/Tt
                            packets can be Transmitted
                ------>  1 + 2a  [Using a = Tp/Tt]
```

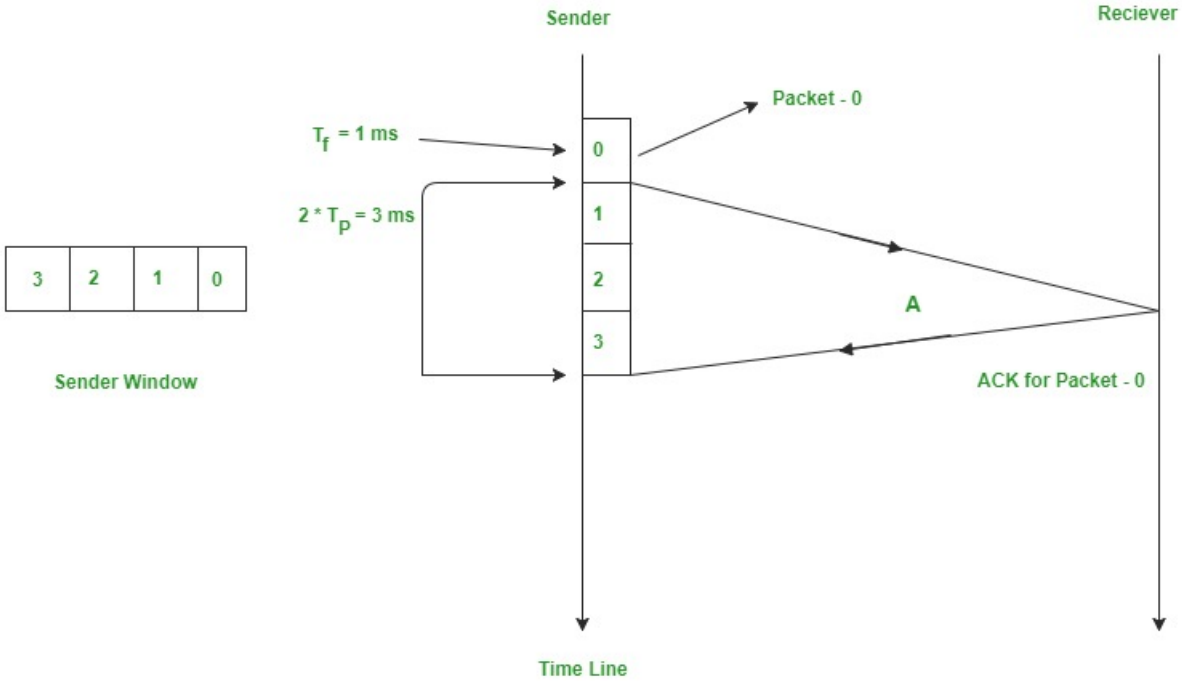
Maximum packets That can be Transmitted in total cycle time = 1+2*a

Let me explain now with the help of an example.

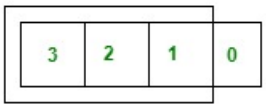
Consider Tt = 1ms, Tp = 1.5ms.

In the picture given below, after sender has transmitted packet 0, it will immediately transmit packets 1, 2, 3. Acknowledgement for 0 will arrive after 2*1.5 = 3ms. In Stop and Wait, in time 1 + 2*1.5 = 4ms, we were transferring one packet only. Here we keep a **window of packets which we have transmitted but not yet acknowledged**

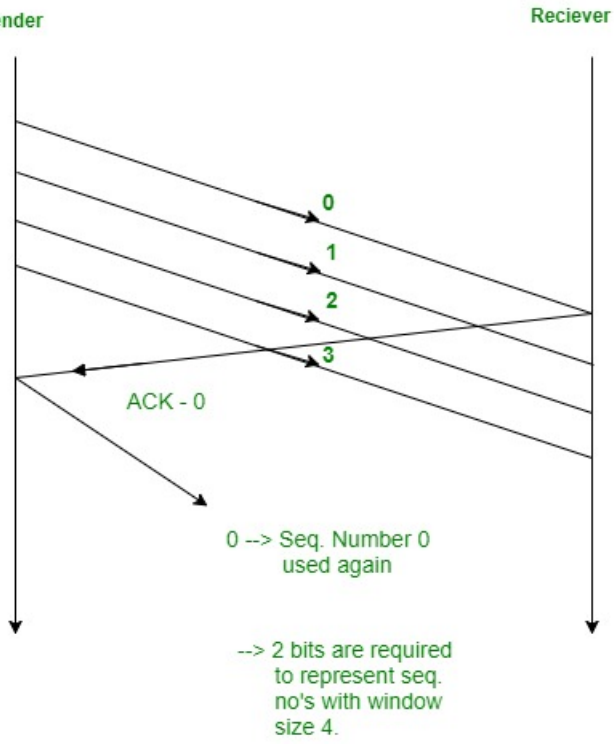




After we have received the Ack for packet 0, window slides and the next packet can be assigned sequence number 0. We reuse the sequence numbers which we have acknowledged so that header size can be kept minimum as shown in the diagram given below.



Window Slided On Receiving Ack of Packet - 0



- Selective Repeat Protocol

Why Selective Repeat Protocol? The go-back-n protocol works well if errors are less, but if the line is poor it wastes a lot of bandwidth on retransmitted frames. An alternative strategy, the selective repeat protocol, is to allow the receiver to accept and buffer the frames following a damaged or lost one.

Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors) :

- Receiver must be able to accept packets out of order.
- Since receiver must release packets to higher layer in order, the receiver must be able to buffer some packets.

Retransmission requests :

- **Implicit -** The receiver acknowledges every good packet, packets that are not ACKed before a time-out are assumed lost or in error.Notice that this approach must be used to be sure that every packet is eventually received.
- **Explicit -** An explicit NAK (selective reject) can request retransmission of just one packet. This approach can expedite the retransmission but is not strictly needed.

- One or both approaches are used in practice.

Selective Repeat Protocol (SRP) : This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintains a window of size. SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. SRP also requires a full-duplex link. Backward acknowledgements are also in progress.

- Sender's Windows (Ws) = Receiver's Windows (Wr).
- Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the size of the window is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.
- Sender can transmit new packets as long as their number is with W of all unpacked packets.
- Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.
- Receiver ACKs all correct packets.
- Receiver stores correct packets until they can be delivered in order to the higher layer.

- In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Figure - the sender only retransmits frames, for which a NAK is received

Efficiency of Selective Repeat Protocol (SRP) is same as GO-Back-N's efficiency :

Efficiency = N/(1+2a)
Where a = Propagation delay / Transmission delay
Buffers = N + N
Sequence number = N(sender side) + N (Receiver Side)

Difference between Stop and Wait, GoBackN and Selective Repeat:

1. **Stop and Wait -** The sender sends the packet and waits for the ACK (acknowledgement) of the packet. Once the ACK reaches the sender, it transmits the next packet in row. If the ACK is not received, it re-transmits the previous packet again.

2. **Go Back N -** The sender sends N packets which is equal to the window size. Once the entire window is sent, the sender then waits for a cumulative ACK to send more packets. On the receiver end, it receives only in-order packets and discards out-of-order packets. As in case of packet loss, the entire window would be re-transmitted.

3. **Selective Repeat -** The sender sends packet of window size N and the receiver acknowledges all packet whether they were received in order or not. In this case, the receiver maintains a buffer to contain out-of-order packets and sorts them.The sender selectively re-transmits the lost packet and moves the window forward.

Differences:

Properties	Stop and Wait	Go Back N	Selective Repeat
Sender window size	1	N	N
Receiver Window size	1	1	N
Minimum Sequence number	2	N+1	2N
Efficiency	1/(1+2*a)	N/(1+2*a)	N/(1+2*a)
Type of Acknowledgement	Individual	Cumulative	Individual
Supported order at Receiving end	-	In-order delivery only	Out-of-order delivery as well
Number of retransmissions in case of packet drop	1	N	1

Where,

- a = Ratio of Propagation delay and Transmission delay,
- At N=1, Go Back N is effectively reduced to Stop and Wait,/li>
- As Go Back N acknowledges the packed cumulatively, it rejects out-of-order packets,

- As Selective Repeat supports receiving out-of-order packets (it sorts the window after receiving the packets), it uses Independent Acknowledgement to acknowledge the packets.

– Network Layer

Network layer works for the transmission of data from one host to the other located in different networks. It also takes care of packet routing i.e. selection of the shortest path to transmit the packet, from the number of routes available. The sender & receiver's IP address are placed in the header by the network layer.
The functions of the Network layer are:

1. **Routing:** The network layer protocols determine which route is suitable from source to destination. This function of network layer is known as routing.
2. **Logical Addressing:** In order to identify each device on internetwork uniquely, network layer defines an addressing scheme. The sender & receiver's IP address are placed in the header by network layer. Such an address distinguishes each device uniquely and universally.

** Segment* in Network layer is referred as **Packet**.

** Network layer is implemented by networking devices such as routers.
Let's look at some primary needs of the Network layer and why it is so important to implement:

1. **Internetworking:** Made possible using Routers. This can be across various types like 802.11, 3G, Ethernet etc

2. **Addressing:** This involves processing of IP Adresses

3. **Routing and Forwarding:** A routing table is maintained by the routers to decide how a packet must be transmitted globally to its specific IP addresses. This process does the global connection is called routing. Forwarding is more of a local concept instead of global.

4. **Scalability (Using hierarchy in Networks):** This refers to the hierarchial organisation of packets.

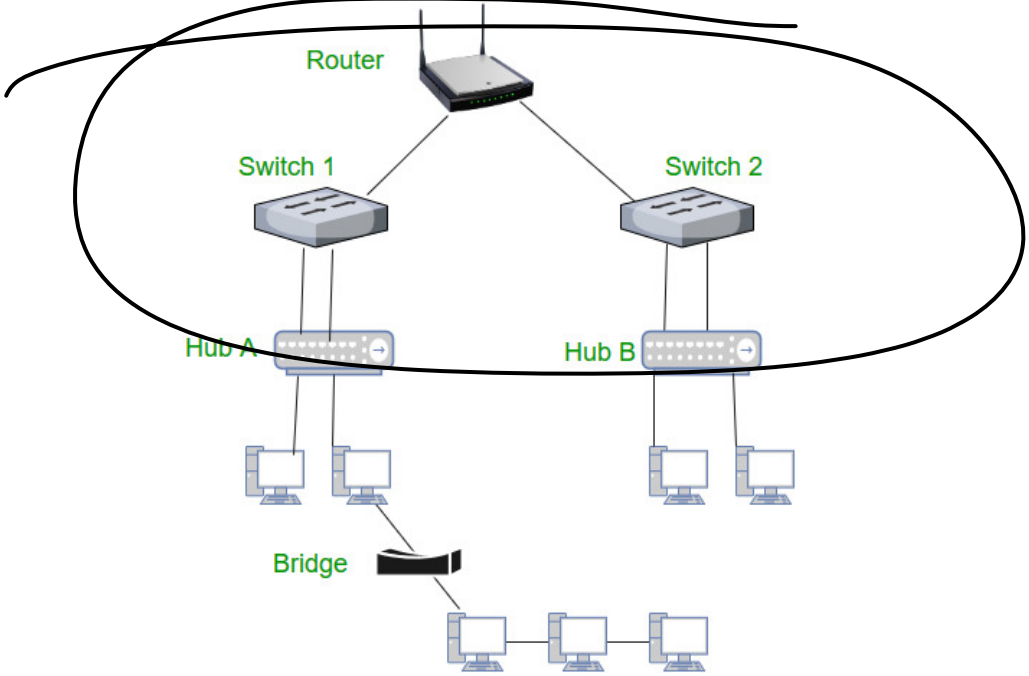
5. **Bandwidth Control:** There must be a good utilisation of Bandwidth.

6. **Fragmentation and Re-assembly:** Division of bigger packets into multiple small packets and rearranging them to get the original packet is called Fragmentation and Re-assembly respectively.

Before understanding the working at the Networking layer, let's get familiar with a few technical devices that has a great role to play in this system:

1. **Switch** - A switch is a multiport bridge with a buffer and a design that can boost its efficiency(a large number of ports imply less traffic) and performance. The switch is a data link layer device. The switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only. In other words, switch divides collision domain of hosts, but broadcast domain remains the same.

2. **Routers** - A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets. Router divide broadcast domains of hosts connected through it.



3. **Brouter** - It is also known as the bridging router is a device which combines features of both bridge and router. It can work either at the data link layer or a network layer. Working as a router, it is capable of routing packets across networks and working as a bridge, it is capable of filtering local area network traffic.

4. **Repeater** - A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network. An important point to be noted about repeaters is that they do not amplify the signal. When the signal becomes weak, they copy the signal bit by bit and regenerate it at the original strength. It is a 2 port device.

5. **Hub** - A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, the collision domain of all hosts connected through Hub remains one. Also, they do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.

   Types of Hub
   - **Active Hub:-** These are the hubs which have their own power supply and can clean, boost and relay the signal along with the network. It serves both as a repeater as well as wiring centre. These are used to extend the maximum distance between nodes.

   - **Passive Hub:-** These are the hubs which collect wiring from nodes and power supply from active hub. These hubs relay signals onto the network without cleaning and boosting them and can't be used to extend the distance between nodes.

6. **Bridge** - A bridge operates at data link layer. A bridge is a repeater, with add on the functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port, thus making it a 2 port device.

   Types of Bridges
   - **Transparent Bridges:-** These are the bridge in which the stations are completely unaware of the bridge's existence i.e. whether or not a bridge is added or deleted from the network, reconfiguration of the stations is unnecessary. These bridges make use of two processes i.e. bridge forwarding and bridge learning.

   - **Source Routing Bridges:-** In these bridges, routing operation is performed by source station and the frame specifies which route to follow. The hot can discover frame by sending a special frame called discovery frame, which spreads through the entire network using all possible paths to destination.

7. **Gateway** - A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically works as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer.

Gateways are generally more complex than switch or router.

Functions of Network Layer: 1) It helps in the delivery of data in the form of packets.

2) It helps in the delivery of packets from source host to the destination host.

3) The network layer is basically used when we want to send data over a different network.

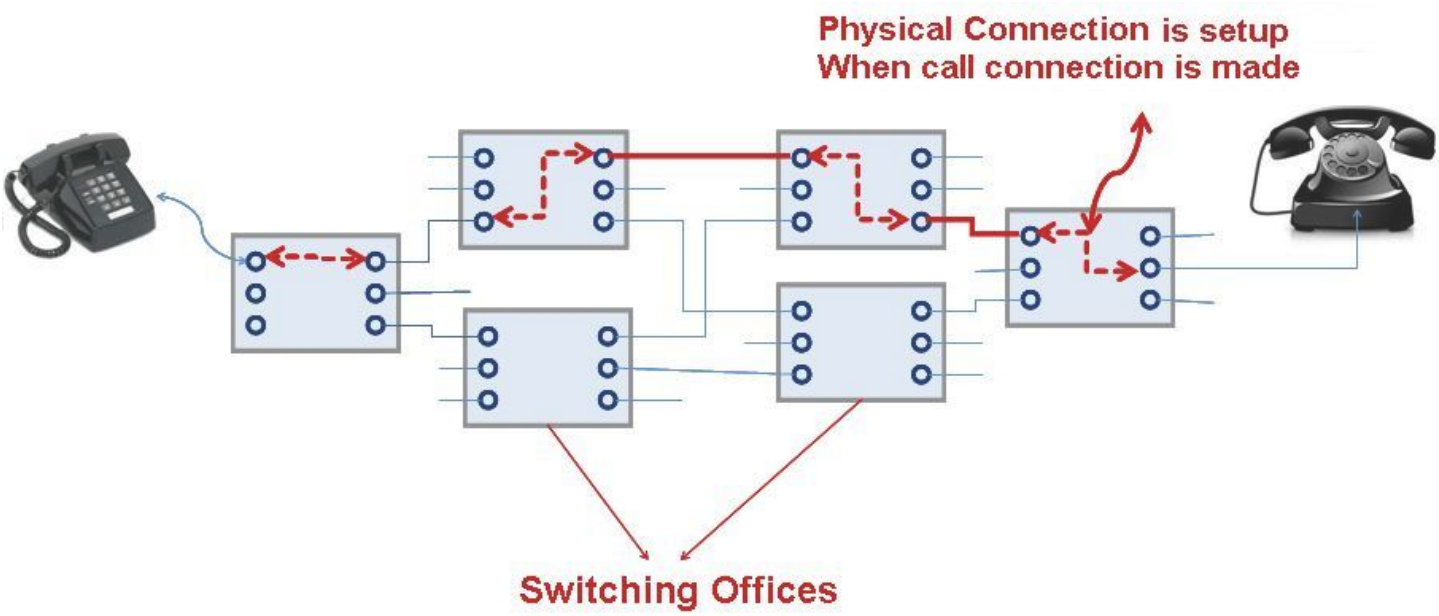
4) In this logical addressing is used ie. when data is to be sent in the same network we need an only physical address but if we wish to send data outside network we need a logical address.

5) It helps in routing ie. routers and switches are connected at this layer to route the packets to its final destination.

Circuit Switching v/s Packet Switching

Circuit-Switching and Packet-Switching are 2 standard ways of transmitting packets between two end-devices over a network. Circuit-Switching is a historically used scheme which has been currently replaced by Packet-Switching.

Circuit Switching In circuit-switching, network resources are dedicated to establish a connection between the end-devices. Thus, a dedicated fixed path is established where data is transmitted without delays (as there is no concept of network congestion). A telephone system works under this scheme.



Key points of circuit switching:

- suitable for continuous transmission (dedicated line)
- guaranteed data-rate
- Inefficient (no transmission even if line is free)
- Under-utilization of resources in most cases

Packet-Switching Packet switching is a method of transferring the data to a network in the form of packets. In order to transfer the file fast and efficient manner over the network and minimize the transmission latency, the data is broken into small pieces of variable length, called **Packet**. At the destination, all these small-parts (packets) has to be reassembled, belonging to the same file. A packet composes of payload and various control information. No pre-setup or reservation of resources is needed. Packet Switching uses **Store and Forward** technique while switching the packets; while forwarding the packet each hop first store that packet than forward. This technique is very beneficial because packets may get discarded at any hop due to some reason. More than one path is possible between a pair of source and destination. Each packet contains Source and destination address using which they independently travel through the network. In other words, packets belonging to the same file may or may not travel through the same path. If there is congestion at some path, packets are allowed to choose different path possible over an existing network

In packet-switching, data is broken down into several packets which are transmitted and routed over the network (according to protocols). No dedicated line is established. i.e. some packets may follow some other path, causing delay and out-of-order reception of data.

Some of the key points of packet switching are:

- Efficient utilisation of network resources
- out-of-order reception of packets
- Transmission delay (variable data-rate)

Packet-Switched networks were designed to overcome the *weaknesses* of Circuit-Switched networks since circuit-switched networks were not very effective for small messages.

Advantage of Packet Switching over Circuit Switching :

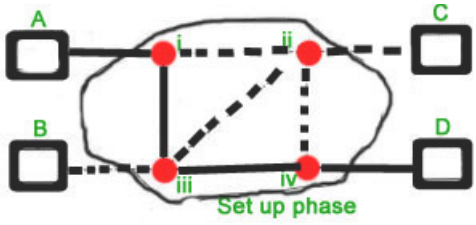
- More efficient in terms of bandwidth, since the concept of reserving circuit is not there.
- Minimal transmission latency.
- More reliable as destination can detect the missing packet.
- More fault tolerant because packets may follow different path in case any link is down, Unlike Circuit Switching.
- Cost effective and comparatively cheaper to implement.

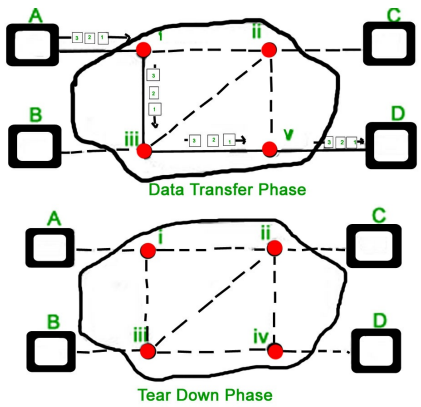
Disadvantage of Packet Switching over Circuit Switching :

- Packet Switching don't give packets in order, whereas Circuit Switching provides ordered delivery of packets because all the packets follow the same path.
- Since the packets are unordered, we need to provide sequence numbers to each packet.
- Complexity is more at each node because of the facility to follow multiple path.
- Transmission delay is more because of rerouting.
- Packet Switching is beneficial only for small messages, but for bursty data (large messages) Circuit Switching is better.

Modes of Packet Switching :

1. **Connection-oriented Packet Switching (Virtual Circuit) :-** Before starting the transmission, it establishes a logical path or virtual connection using signalling protocol, between sender and receiver and all packets belongs to this flow will follow this predefined route. Virtual Circuit ID is provided by switches/routers to uniquely identify this virtual connection. Data is divided into small units and all these small units are appended with help of sequence number. Overall, three phases takes place here- Setup, data transfer and tear down phase.





Phases in virtual circuit packet switching

All address information is only transferred during setup phase. Once the route to destination is discovered, entry is added to switching table of each intermediate node. During data transfer, packet header (local header) may contain information such as length, timestamp, sequence number etc.
Connection-oriented switching is very useful in switched WAN. Some popular protocols which use Virtual Circuit Switching approach are X.25, Frame-Relay, ATM and MPLS(Multi-Protocol Label Switching).

2. **Connectionless Packet Switching (Datagram) :-** Unlike Connection-oriented packet switching, In Connectionless Packet Switching each packet contains all necessary addressing information such as source address, destination address and port numbers etc. In Datagram Packet Switching, each packet is treated independently. Packets belonging to one flow may take different routes because routing decisions are made dynamically, so the packets arrived at destination might be out of order. It has no connection setup and teardown phase, like Virtual Circuits.
Packet delivery is not guaranteed in connectionless packet switching, so the reliable delivery must be provided by end systems using additional protocols.

A---R1---R2---B

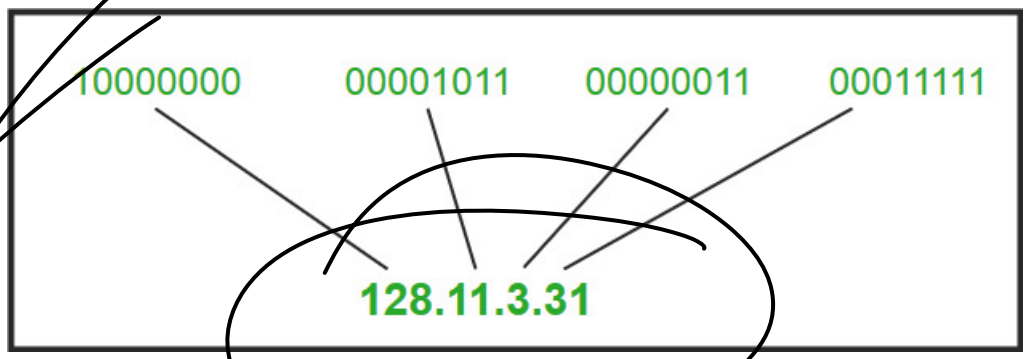
A is the sender (start)
R1, R2 are two routers that store and forward data
B is receiver(destination)

To send a packet from A to B there are delays since this is a Store and Forward network.

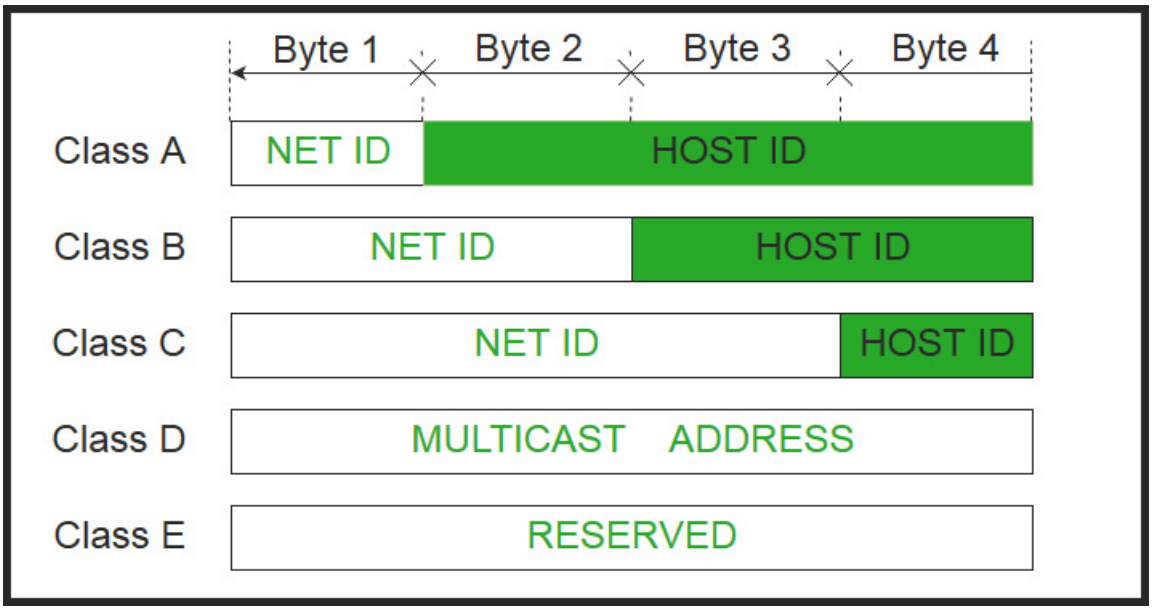
IP and Classful Addressing

To uniquely identify a device in a network, we use logical addresses known as IP addresses. It is a **32-bit** value generating a total of 2^{32} addresses. Most of the current systems are still using these 32-bit addresses and are termed as IPv4. But, due to the large no. of existing devices and more growing day-by-day, even this large no. won't suffice in uniquely identifying each device. So, **IPv6 (having 128-bit addresses)** has been introduced. We shall discuss more differences between IPv4 and IPv6 in the upcoming section.

IPv4 addresses are generally represented in dotted decimal representation:

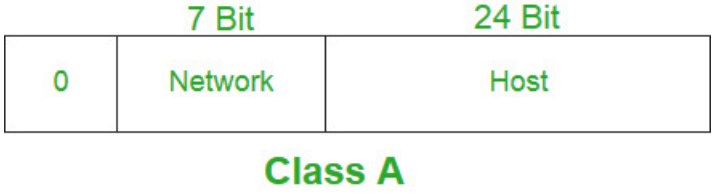


The whole 32-bit address space is divided into 5 classes of varying capacity (no. of unique addresses available) depending on the size and needs of an organization. This form of division is known as **Classful Addressing**. Each of these classes have a valid range of IP addresses. Classes *A, B, C* are used for *unicast* addressing for different organization size. *Class D* is reserved for *multicasting* and *E* for *experimental & military applications*. In this scheme, the address bits are divided into:
Network ID - Same for all the addresses in the same class. Uniquely identifies the network as a whole.
Host ID - Identifies each host within a network uniquely.



Let us delve into each of the classes in detail:

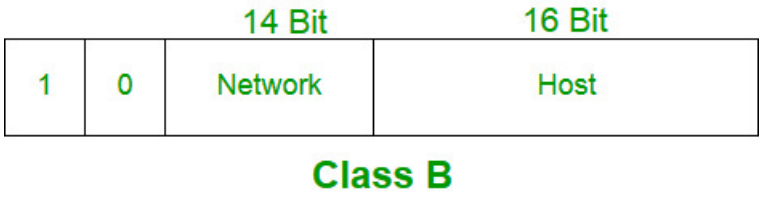
Class A The Network ID is of 8-bits, leaving the host part with 24-bits. The 1st bit of Network part is always set to 0.
Subnet Mask - 255.0.0.0.
Size of Network - $(2^{24} - 2)$ = 16,777,214 Host IDs (2 is subtracted because of x.0.0.0 is reserved for Network ID and x.255.255.255 is used for limited-broadcasting)
No. of unique networks - $(2^7 - 2)$ = 127 Networks (2 is subtracted because 0.0.0.0 and 127.x.y.z are reserved for special purposes)



Class A

IP addresses belonging to class A ranges from 1.x.x.x – 126.x.x.x.

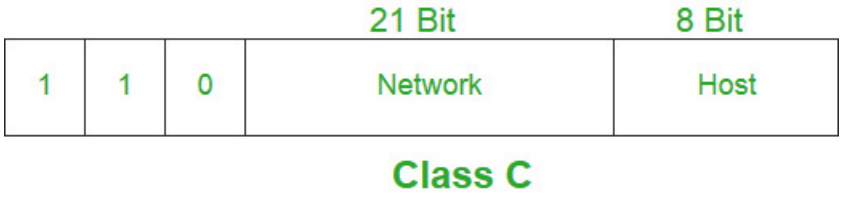
Class B The Network ID is of 16-bits, leaving the host part with 16-bits. However, the higher-order bits of Network part is always set to 10.
Subnet Mask - 255.255.0.0.
Size of Network - $(2^{16} - 2)$ = 65534 Host IDs (2 is subtracted because of x.y.0.0 is reserved for Network ID and x.y.255.255 is used for limited-broadcasting)
No. of unique networks - 2^{14} = 16384 Networks



Class B

IP addresses belonging to class B ranges from 128.0.x.x – 191.255.x.x.

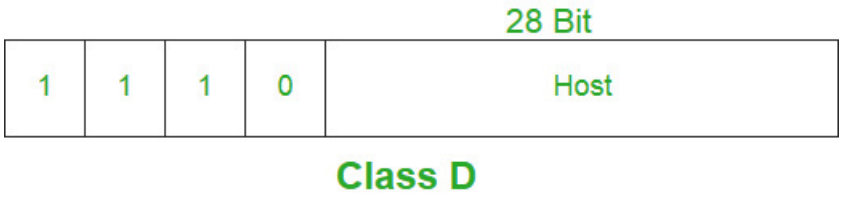
Class C The Network ID is of 24-bits, leaving the host part with 8-bits. However, the higher-order bits of Network part is always set to 110.
Subnet Mask - 255.255.0.0.
Size of Network - (2^8 - 2) = 254 Host IDs (2 is subtracted because of x.y.z.0 is reserved for Network ID and x.y.z.255 is used for limited-broadcasting)
No. of unique networks - 2^{21} = 2097152 Networks



Class C

IP addresses belonging to class C ranges from 192.0.0.x - 223.255.255.x.

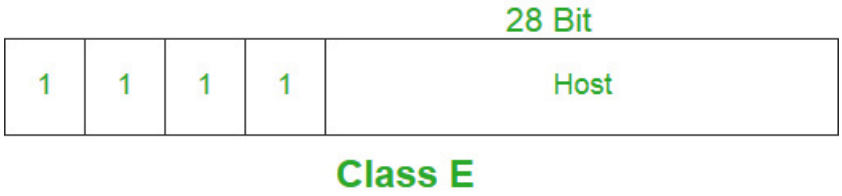
Class D Class-D addresses are reserved for multi-casting. The higher-order bits are set to 1110. Remaining bits are reserved for interested hosts. Class-D networks don't have any subnet mask, as there is no concept of a subnet for this class of IPs.



Class D

IP addresses belonging to class-D ranges from 224.0.0.0 – 239.255.255.255.

Class E IP addresses belonging to class E are reserved for experimental, research & military applications. IP ranges from 240.0.0.0 – 255.255.255.254. This class too doesn’t have any subnet mask. The higher order bits of first octet of class E are always set to 1111.



Class E

- Flow-control Protocols

Flow Control is required in Computer Networks because, most of the time the sender has no idea about the capacity of buffer at the receiving end, and thus may transmit packets exceeding the current capacity causing them to get dropped at the receiver end. Thus, the flow control mechanism is required for re-transmission in case packets get lost. Some of the popular schemes are as follows:

Stop & Wait (ARQ) In this scheme, the sender waits for ACK (acknowledgment) from the receiver before transmitting the next packet. If it doesn't receive ACK for a certain packet within a pre-defined timeout (ARQ variant ~ Automatic Repeat Request), it re-transmits said packet (assuming it got dropped).

In this scheme, packets are sent one-by-one (inefficient).

Go-back-N In this scheme, the sender sends all the packets equating to the receiver window size (say n) all at once. The receiver then sends ACK_{n+1} (requesting the next packet ~ $(n+1)^{th}$). GBN uses *cumulative acknowledgment*. If any of the transmitted packets get lost, all the subsequent packets are dropped at the receiver end. Instead, a NACK (negative-acknowledgment indicating lost packet no.) is transmitted. Thereafter, all packets starting from the lost packet is re-transmitted.

As can be seen, if packet #1 gets lost, the whole window will be re-transmitted by going back n places, hence the name. It is also not much useful as unnecessarily we are repeating transmission of the whole window. We can do better as in Selective Repeat.

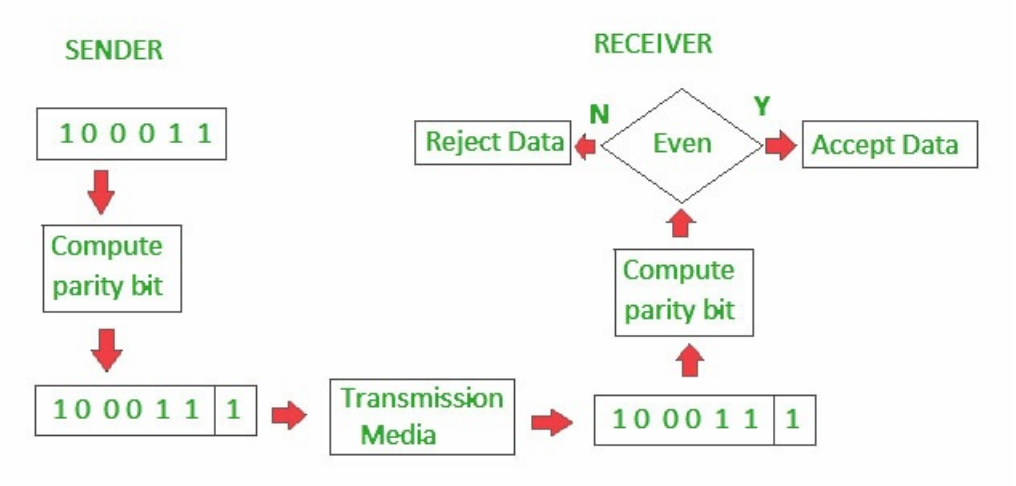
Selective Repeat In this scheme, when a packet gets lost, the receiver sends a NACK, however, unlike GBN, it still receives subsequent packets (GBN drops them as shown in the diagram above). Upon the reception of NACK, only that particular packet is re-transmitted.

- Error Detection

Due to noise in network and signal interference, bit values may get changed during transmission leading to so called errors. They need to be detected at the Data-Link layer, and upon detection re-transmission is requested or correction is done (as in Hamming Code). Some of the common error-detection schemes are given below:

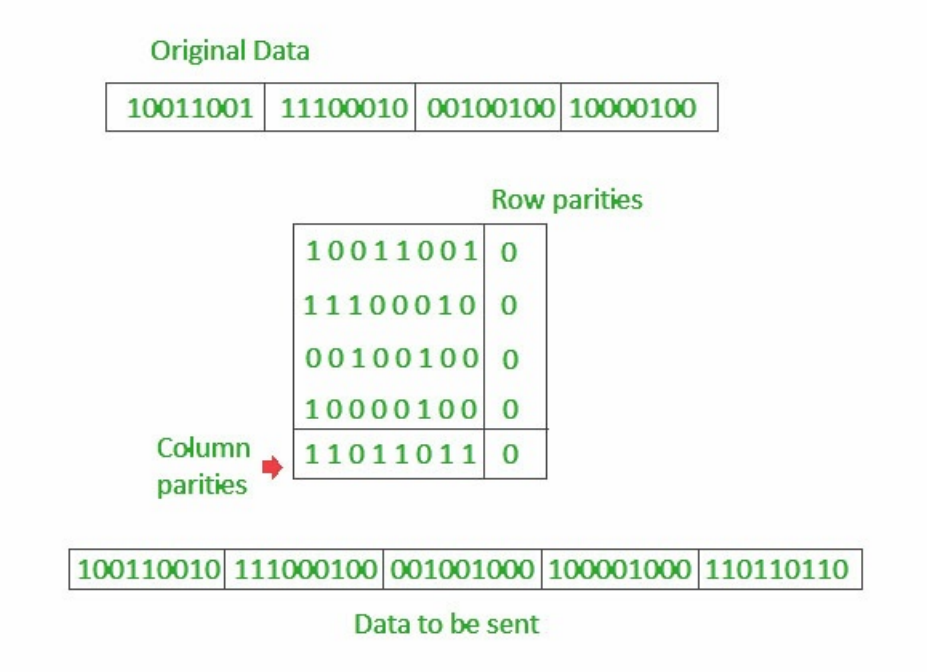
Parity Check

Parity check works by counting the no. of 1s in the bit-representation and then appending 1 in case there exists odd no. of ones, or 0 in case of even no. of 1s. Thus, the total no. of 1s become even. Hence, this scheme is also called even-parity check. Thus, if due to error any bit changes, the total no. of 1s will become odd.



2D 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.



Checksum

The procedure for usage of checksum is as follows:

- Data is divided into k segments each of m-bits.
- At the sender, 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, all received segments are added using 1's complement arithmetic. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

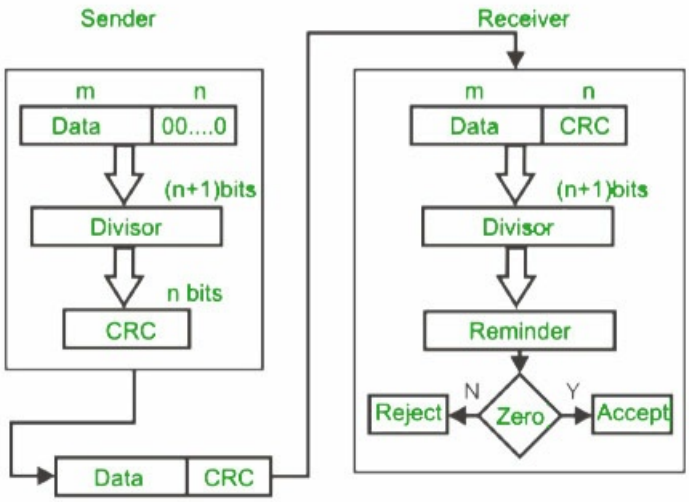
CRC (Cyclic Redundancy Check)

CRC is based on binary division, and it works as:

- 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.

The whole procedure can be better understood with an example:



– Network Address Translation or NAT

Before proceeding to NAT, let's understand the various types of IP addresses. There are basically two types of IP addresses.

1. **Public IP Address:** A public IP address is an IP address that can be accessed over the Internet. Like postal address used to deliver a postal mail to your home, a public IP address is the globally unique IP address assigned to a computing device. Your public IP address can be found at What is my IP Address page.

2. **Private IP Address:** Private IP address, on the other hand, is used to assign computers within your private space without letting them directly exposed to the Internet. For example, if you have multiple computers within your home you may want to use private IP addresses to address each computer within your home. In this scenario, your router gets the public IP address, and each of the computers, tablets, and smartphones connected to your router (via wired or wifi) gets a private IP address from your router via DHCP protocol.

   Internet Assigned Numbers Authority (IANA) is the organization responsible for registering IP address ranges to organizations and Internet Service Providers (ISPs). To allow organizations to freely assign private IP addresses, the Network Information Center (InterNIC) has reserved certain address blocks for private use. The following IP blocks are reserved for private IP addresses.

   FOR CLASS A:
   ADDRESS RANGE: 10.0.0.0-10.255.255.255

   FOR CLASS B:
   ADDRESS RANGE: 172.16.0.0-172.31.255.255

   FOR CLASS C:
   ADDRESS RANGE: 192.168.0.0-192.168.255.255

   APIPA(Automatic Private IP Addressing)-169.254.0.0-169.254.255.255

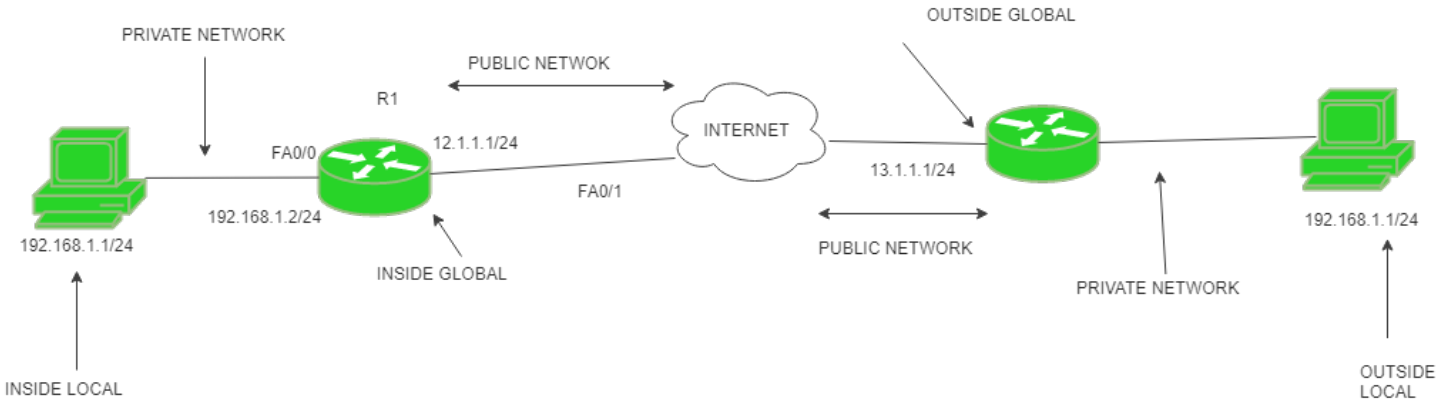
Network Address Translation (NAT): To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of private IP address to a public IP address is required. **Network Address Translation (NAT)** is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on router or firewall.

Network Address Translation (NAT) working - Generally, the border router is configured for NAT i.e the router which has one interface in local (inside) network and one interface in the global (outside) network. When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address. When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.

If NAT run out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

Why mask port numbers?
Suppose, in a network, two hosts A and B are connected. Now, both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time. If NAT does an only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network and sent to the destination. Destination will send replies on the public IP address of the router. Thus, on receiving reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A and B are same). Hence, to avoid such a problem, NAT masks the source port number as well and makes an entry in the NAT table.

NAT inside and outside addresses - Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organisation. These are the network Addresses in which the translation of the addresses will be done.



- **Inside local address -** An IP address that is assigned to a host on the Inside (local) network. The address is probably not a IP address assigned by the service provider i.e., these are private IP address. This is the inside host seen from the inside network.
- **Inside global address -** IP address that represents one or more inside local IP addresses to the outside world. This is the inside host as seen from the outside network.
- **Outside local address -** This is the actual IP address of the destination host in the local network after translation.
- **Outside global address -** This is the outside host as seen form the outside network. It is the IP address of the outside destination host before translation.

Network Address Translation (NAT) Types - There are 3 ways to configure NAT:

1. **Static NAT -** In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global address. This is generally used for Web hosting. These are not used in organisations as there are many devices who will need Internet access and to provide Internet access, the public IP address is needed.

   Suppose, if there are 3000 devices who need access to the Internet, the organisation have to buy 3000 public addresses that will be very costly.

2. **Dynamic NAT -** In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP address. If the IP address of pool is not free, then the packet will be dropped as an only a fixed number of private IP address can be translated to public addresses.

   Suppose, if there is a pool of 2 public IP addresses then only 2 private IP addresses can be translated at a given time. If 3rd private IP address wants to access Internet then the packet will be dropped therefore many private IP addresses are

mapped to a pool of public IP addresses. NAT is used when the number of users who wants to access the Internet is fixed. This is also very costly as the organisation have to buy many global IP addresses to make a pool.

3. **Port Address Translation (PAT) -** This is also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address. Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address. This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

Advantages of NAT -

- NAT conserves legally registered IP addresses .
- It provides privacy as the device IP address, sending and receiving the traffic, will be hidden.
- Eliminates address renumbering when a network evolves.

Disadvantage of NAT -

- Translation results in switching path delays.
- Certain applications will not function while NAT is enabled.
- Complicates tunneling protocols such as IPsec.
- Also, router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.

– Subnetting

When a bigger network is divided into smaller networks, in order to maintain security, then that is known as Subnetting. so, maintenance is easier for smaller networks.

Network Address and Mask

Network address - It identifies a network on internet. Using this, we can find range of addresses in the network and total possible number of hosts in the network.

Mask - It is a 32-bit binary number that gives the network address in the address block when AND operation is bitwise applied on the mask and any IP address of the block.

The default mask in different classes are :

Class A - 255.0.0.0

Class B - 255.255.0.0

Class C - 255.255.255.0

Example : Given IP address 132.6.17.85 and default class B mask, find the beginning address (network address).

Solution : The default mask is 255.255.0.0, which means that the only the first 2 bytes are preserved and the other 2 bytes are set to 0. Therefore, the network address is 132.6.0.0.

Some values calculated in subnetting :

1. Number of subnets : Given bits for mask - No. of bits in default mask

2. Subnet address : AND result of subnet mask and the given IP address

3. Broadcast address : By putting the host bits as 1 and retaining the network bits as in the IP address

4. Number of hosts per subnet : $2^{(32 - \text{Given bits for mask})} - 2$

5. First Host ID : Subnet address + 1 (adding one to the binary representation of the subnet address)

6. Last Host ID : Subnet address + Number of Hosts

Example : Given IP Address - 172.16.0.0/25, find the number of subnets and the number of hosts per subnet. Also, for the first subnet block, find the subnet address, first host ID, last host ID and broadcast address.

Solution : This is a class B address. So, no. of subnets = $2^{(25-16)} = 2^9 = 512$.

No. of hosts per subnet = $2^{(32-25)} - 2 = 2^7 - 2 = 128 - 2 = 126$

For the first subnet block, we have subnet address = 0.0, first host id = 0.1, last host id = 0.126 and broadcast address = 0.127

In order to find the Network ID (NID) of a Subnet, one must be fully acquainted with the Subnet mask Subnet Mask is used to find which IP address belongs to which Subnet. It is a 32-bit number, containing 0's and 1's. Here network id part and Subnet ID part is represented by all 1's and host ID part is represented by all 0's.

Example: If Network id of a entire network = 193.1.2.0 (it is class C IP). For more about class C IP see Classful Addressing.

Example-1:

```
If IP address = 193.1.2.129 (convert it into binary form)

          = 11000001.00000001.00000010.10000001

Subnet mask   = 11111111.11111111.11111111.11000000

Bit Wise AND  = 11000001.00000001.00000010.10000000

Therefore, Nid = 193.1.2.128
```

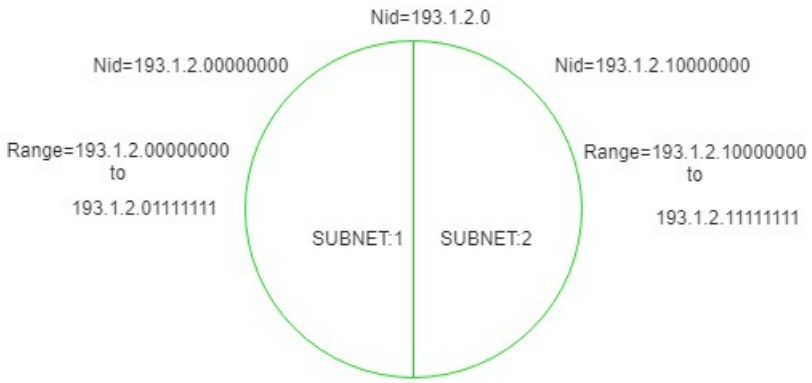
Hence, this IP address belongs to subnet:3 which has Nid = 193.1.2.128

Example-2:

```
If IP address = 193.1.2.67 (convert it into binary form)

          = 11000001.00000001.00000010.01000011

Subnet Mask   = 11111111.11111111.11111111.11000000

Bit Wise AND  = 11000001.00000001.00000010.01000000

Therefore, Nid = 193.1.2.64
```

Hence, this IP address belongs to subnet:2 which has Nid = 193.1.2.64

Dividing Networks Now, let's talk about dividing a network into two parts:
so to divide a network into two parts, you need to choose one bit for each Subnet from the host ID part.



In the above diagram, there are two Subnets.

Note: It is a class C IP so, there are 24 bits in the network id part and 8 bits in the host id part.

- **For Subnet-1:** The first bit which is chosen from the host id part is zero and the range will be from (193.1.2.00000000 till you get all 1's in the host ID part i.e, 193.1.2.01111111) except for the first bit which is chosen zero for subnet id part.

  Thus, the range of subnet-1:

  ```
  193.1.2.0 to 193.1.2.127
  ```

- **For Subnet-2:** The first bit chosen from the host id part is one and the range will be from (193.1.2.100000000 till you get all 1's in the host ID part i.e, 193.1.2.11111111).

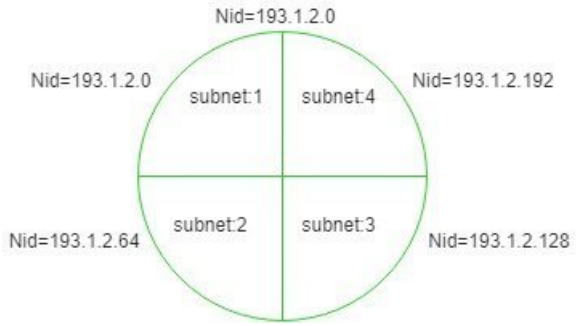
  Thus, the range of subnet-2:

  193.1.2.128 to 193.1.2.255

Note:

1. To divide a network into four (2^2) parts you need to choose two bits from host id part for each subnet i.e, (00, 01, 10, 11).
2. To divide a network into eight (2^3) parts you need to choose three bits from host id part for each subnet i.e, (000, 001, 010, 011, 100, 101, 110, 111) and so on.

Now, let's talk about dividing a network into four parts:



In the above diagram entire network is divided into four parts, which means there are four subnets each having two bits for Subnet ID part.

Subnet-1: 193.1.2.0 to 193.1.2.63

Subnet-2: 193.1.2.64 to 193.1.2.127

Subnet-3: 193.1.2.128 to 193.1.2.191

Subnet-4: 193.1.2.192 to 193.1.2.255

The above IP is class C, so it has 24 bits in network id part and 8 bits in host id part but you choose two bits for subnet id from host id part, so now there are two bits in subnet id part and six bits in host id part, i.e.,

24 bits in network id + 2 bits in subnet id = 26 (1's) and

6 bits in host id = 6 (0's)

Therefore,

Subnet Mask = 11111111.11111111.11111111.11000000

= 255.255.255.192

If any given IP address performs bitwise AND operation with the subnet mask, then you get the network id of the subnet to which the given IP belongs.

Subnetting is useful in many ways like:

1. It provides security to one network from another network. eg) In an Organisation, code of the Developer department must not be accessed by another department.
2. It may be possible that a particular subnet might need higher network priority than others. For example, a Sales department need to host webcasts or video conferences.

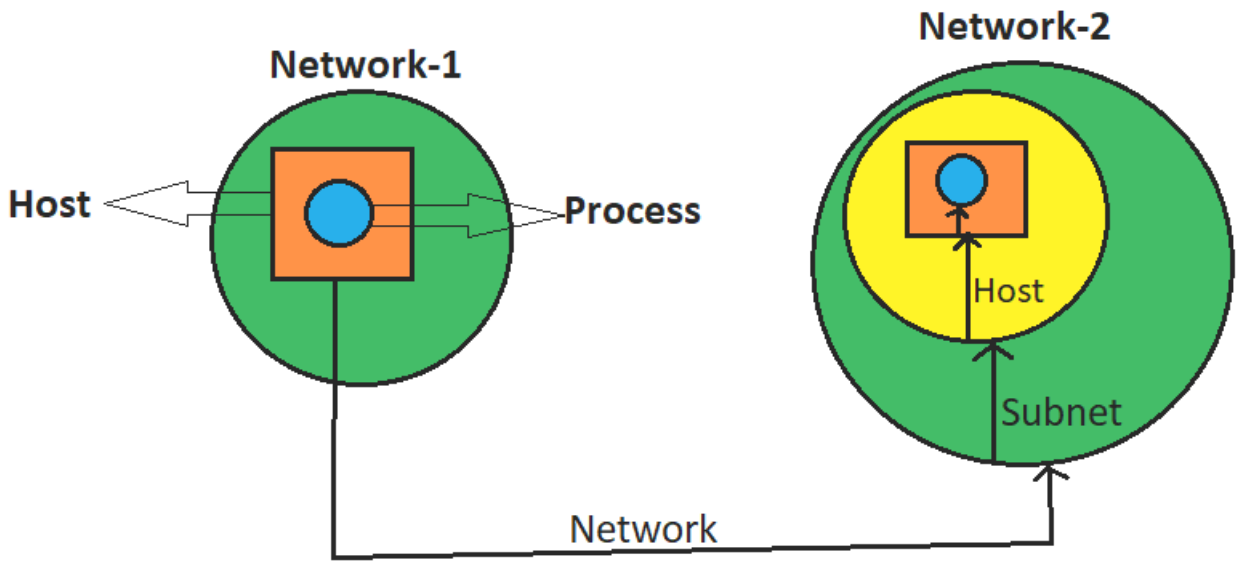
3. In the case of Small networks, maintenance is easy.

Along with these advantages, Subnetting also has some disadvantages:

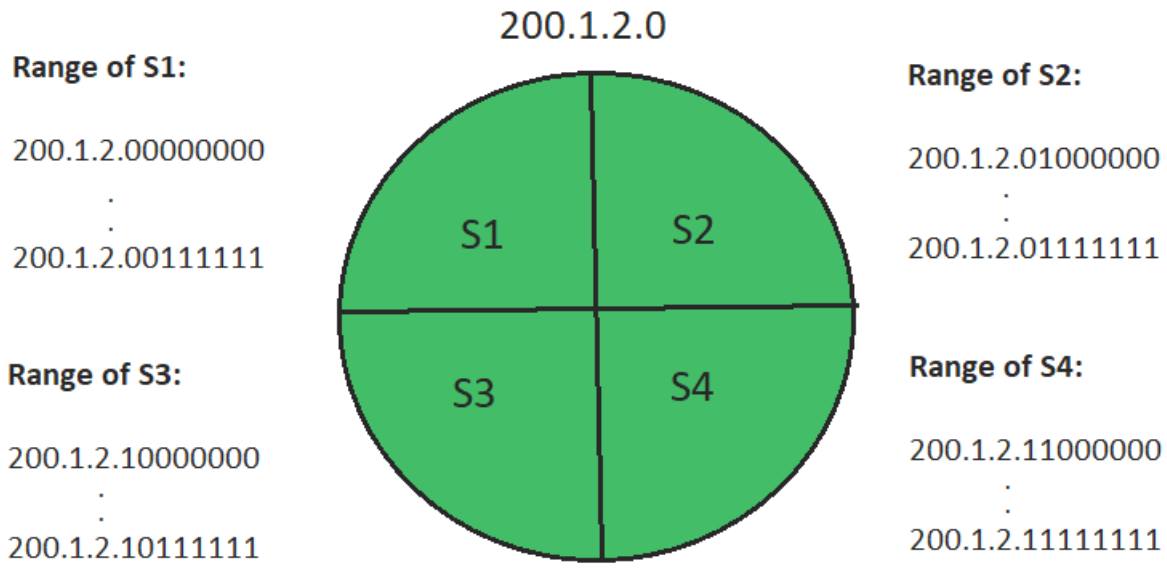
1. In case of the single network, only three steps are required in order to reach a Process i.e Source Host to Destination Network, Destination Network to Destination Host and then Destination Host to Process.

   But in the case of Subnetting four steps are required for Inter-Network Communication. i.e Source Host to Destination Network, Destination Network to proper Subnet, then Subnet to Host and finally Host to Process.

   Hence, it increases Time complexity. In the case of Subnet, more time is required for communication or data transfer.



2. In the case of Single Network only two IP addresses are wasted to represent Network Id and Broadcast address but in case of Subnetting two IP addresses are wasted for each Subnet.
   Example: If a Network has four Subnets, it means 8 IP addresses are going to waste.



Network Id for S1: 200.1.2.0

Broadcast address of S1: 200.1.2.63

Network Id for S2: 200.1.2.64

Broadcast address of S2: 200.1.2.127

Network Id for S3: 200.1.2.128

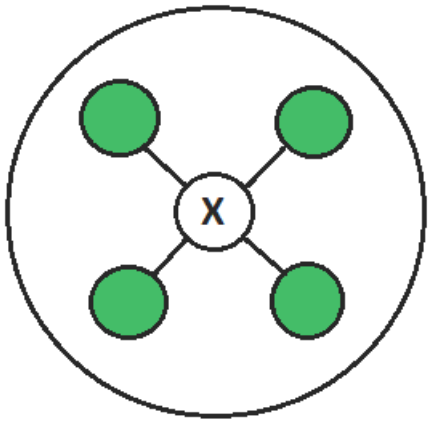
Broadcast address of S3: 200.1.2.191

Network Id for S4: 200.1.2.192

Direct Broadcast address of S4: 200.1.2.255

Hence, we can say that Network size will also decrease. We can't use our Network completely.

3. Cost of the overall Network also increases. Subnetting requires internal routers, Switches, Hubs, Bridges etc. which are very costly.



4. Subnetting and network management require an experienced network administrator. This adds to the overall cost as well.

Introduction of Variable Length Subnet Mask (VLSM): In this the subnet, the design uses more than one mask in the same network which means more than one mask is used for different subnets of a single class A, B, C or a network. It is used to increase the usability of subnets as they can be of variable size. It is also defined as the process of subnetting of a subnet.

– Classless Addressing (CIDR), Subnetting & Supernetting

As we have already learned about Classful Addressing, so in this article, we are going to learn about Classless Inter-Domain Routing. which is also known as Classless addressing. In the Classful addressing the no of Hosts within a network always remains the same depending upon the class of the Network.

Class A network contains 2^{24} Hosts,

Class B network contains 2^{16} Hosts,

Class C network contains 2^{8} Hosts

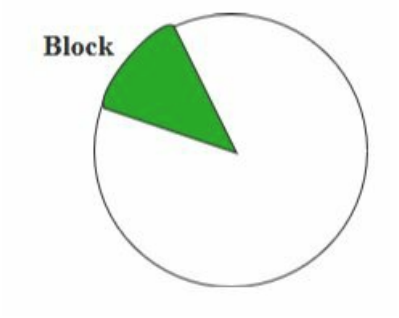
CIDR (Classless-Inter-Domain-Routing) was introduced as an improvement to the earlier Classful addressing scheme, as Classful networking scheme had several shortcomings as given below:

- Class A & B networks have an unfathomable amount of IP addresses, most of which gets wasted even in very large organizations.
- On the other hand, Class C networks have inadequate no. of addresses to cater to the needs of an organization.
- Class D is available as a single block (no further grouping possible), and Class E is reserved for experimental/research purposes.

CIDR scheme doesn't fix the network and host parts into categories. It instead allows us to subnet the whole address space to get a network from the whole address space according to our requirement (no wastage).

Now, let's suppose an Organization requires 2^{14} hosts, then it must have to purchase a Class B network. In this case, 49152 Hosts will be wasted. This is the major drawback of Classful Addressing.

In order to reduce the wastage of IP addresses a new concept of **Classless Inter-Domain Routing** is introduced. Now a days *IANA* is using this technique to provide the IP addresses. Whenever any user asks for IP addresses, IANA is going to assign that many IP addresses to the User.



Representation: It is as also a 32-bit address, which includes a special number which represents the number of bits that are present in the Block Id.

```
a . b . c . d / n
```

Where, n is number of bits that are present in Block Id / Network Id.
Example:

```
20.10.50.100/20
```

Rules for forming CIDR Blocks:

1. All IP addresses must be contiguous.
2. Block size must be the power of 2 (2^n).
   If the size of the block is the power of 2, then it will be easy to divide the Network. Finding out the Block Id is very easy if the block size is of the power of 2.
   Example: If the Block size is 2^5 then, Host Id will contain 5 bits and Network will contain 32 - 5 = 27 bits.

3. First IP address of the Block must be evenly divisible by the size of the block. in simple words, the least significant part should always start with zeroes in Host Id. Since all the least significant bits of Host Id is zero, then we can use it as Block Id part.

Example: Check whether 100.1.2.32 to 100.1.2.47 is a valid IP address block or not?

1. All the IP addresses are contiguous.
2. Total number of IP addresses in the Block = 16 = 2^4.

3. 1st IP address: 100.1.2.00100000
   Since, Host Id will contains last 4 bits and all the least significant 4 bits are zero. Hence, first IP address is evenly divisible by the size of the block.

All three rules are followed by this Block. Hence, it is a valid IP address block.

Subnetting & Subnet Mask Subnetting is the process of dividing the whole address-space into a block of contiguous IP addresses. The no. of host devices that can be accommodated and the Network ID can be found out using Subnet Mask. A subnet mask is a 32-bit binary value which upon taking bitwise-AND with the IP-address gives us the Network ID bits. We give the IP address and define the number of bits for mask along with it (usually followed by a '/' symbol), like, 192.168.1.1/28. Here, the subnet mask is found by putting the given number of bits out of 32 as 1, like, in the given address, we need to put 28 out of 32 bits as 1 and the rest as 0, and so, the subnet mask would be 255.255.255.240.
As another example:
Given IP Address – 172.16.21.23/25,
IP: 10101100.00010000.00010101.00010111
Subnet Mask: 11111111.11111111.11111111.10000000 (binary) ~ 255.255.255.128
Taking AND, we get Network ID: 172.16.21.0
No. of usable hosts: $(2^{32-25} - 2) = (2^7 - 2) = 126$ (excluding Network ID and broadcast address).

Supernetting Supernetting is the reverse operation of subnetting where we group two contiguous blocks of classless networks of same size to form a larger group. There are however some rules required to follow for supernetting networks:

- All the IP addresses should be contiguous.
- Size of all the small networks should be equal and must be in form of 2
- First IP address should be exactly divisible by whole size of supernet

As an example, consider 4 networks:

```
N1: 200.1.0.0/24
N2: 200.1.1.0/24
N3: 200.1.2.0/24
N4: 200.1.3.0/24
```

We see that all the addresses are contiguous. N1 ranges from 200.1.0.0 to 200.1.0.255. Adding 0.0.0.1 to the last address yields 200.1.1.0, which is the start address of N2. Similarly, for all the subsequent networks N2, N3 & N4.
The sizes of all the networks are also same ~ 2^8 (a power of 2 as well). The 1st IP address is also divisible by the total size. Here, 200.1.0.0 is the 1st IP address and the total size of supernet will be $4*2^8 \sim 2^{10}$, implying last 10 bits should be 0.
200.1.0.0 = 11001000.00000001.00000**00.00000000**.
Hence, we can group them together into a supernet as: **200.1.0.0/22**.

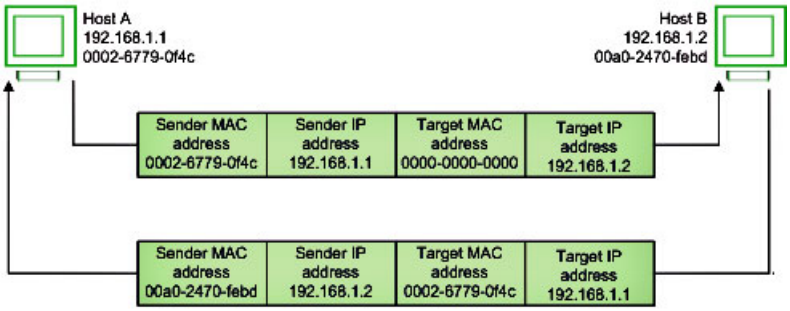
- ARP (Address Resolution Protocol) & Reverse-ARP

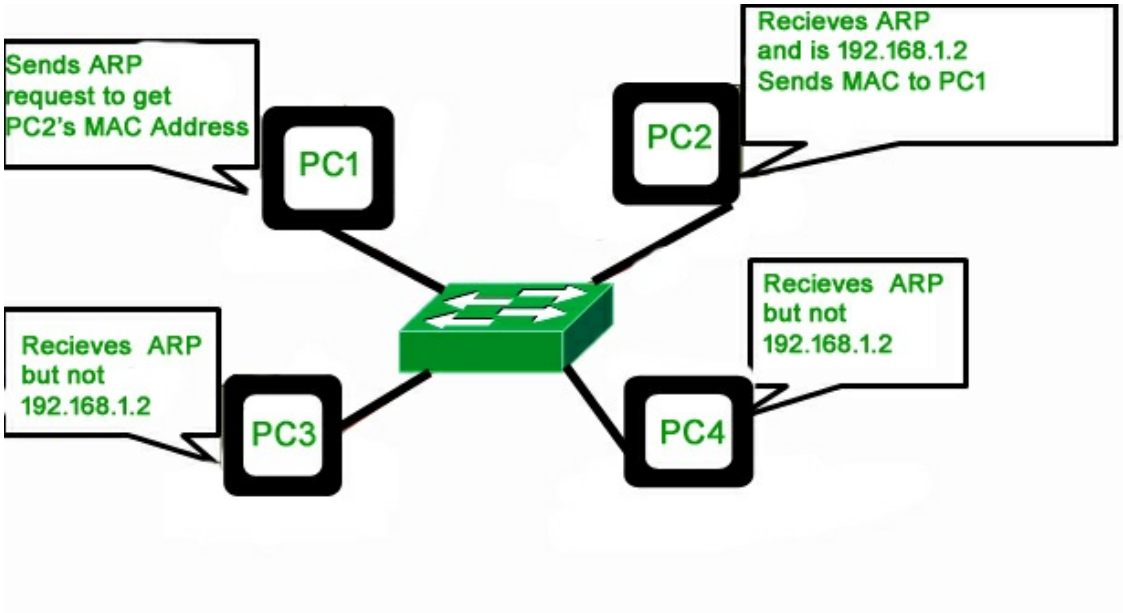
In a computer network, we have 2 addresses associated with a device (physical & logical). Physical Address is permanent and fixed (although can be changed, but shouldn't be done - MAC spoofing) for a device and doesn't change if the device changes network. Logical Addresses (IP) is transient and changes once device leaves current network and joins another. **To finally transmit data from one device to another, however physical/MAC address is required** (at Data-Link-layer). But, all a Network Layer knows is the logical address/IP of the next-hop-device. **ARP** (Address-Resolution-Protocol) is the de-facto method of acquiring the **physical address of next-hop from it's logical address** Similarly, Reverse-ARP is the process of getting the IP address from the device's physical address.

ARP To get the MAC address of the target machine, the sender broadcasts a special ARP-message over it's immediate neighbors, requesting the MAC address. The contents of this message are:

- Sender IP address
- Sender MAC address
- Destinaton MAC address (filled as all 0s initially)
- Destination IP addresss

Upon reception of this ARP message, the device associated with the destination IP, fills it's MAC address into the destination MAC space (filled with 0s), and unicasts it to the sender (Sender MAC is provided for this purpose). All other machines simply ignore the request. Finally, the sender recieves the reply and gets to know the destination MAC address.





Reverse ARP Reverse ARP is a networking protocol used by a client machine in a local area network to request its Internet Protocol address (IPv4) from the gateway-router's ARP table. The network administrator creates a table in gateway-router, which is used to map the MAC address to corresponding IP address.
When a new machine is setup or any machine which don't have memory to store IP address, needs an IP address for its own use. So the machine sends a RARP broadcast packet which contains its own MAC address in both sender and receiver hardware address field.

A special host configured inside the local area network, called as RARP-server is responsible to reply for these kind of broadcast packets. Now the RARP server attempt to find out the entry in IP to MAC address mapping table. If any entry matches in table, RARP server send the response packet to the requesting device along with IP address.

IPv4 vs IPv6

IPv4 and IPv6 are internet protocol version 4 and internet protocol version 6, IP version 6 is the new version of Internet Protocol, which is way better than IP version 4 in terms of complexity and efficiency.

Difference Between IPv4 and IPv6:

IPv4	IPv6
IPv4 has 32-bit address length	IPv6 has 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end connection integrity is Unachievable	In IPv6 end to end connection integrity is Achievable
It can generate 4.29x109 address space	Address space of IPv6 is quite large it can produce 3.4x1038 address space

IPv4	IPv6
Security feature is dependent on application	IPSEC is inbuilt security feature in the IPv6 protocol
Address representation of IPv4 in decimal	Address Representation of IPv6 is in hexadecimal
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation performed only by sender
In IPv4 Packet flow identification is not available	In IPv6 packetflow identification are Available and uses flow label field in the header
In IPv4 checksumfield is available	In IPv6 checksumfield is not available
It has broadcast Message Transmission Scheme	In IPv6 multicast and any cast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided

Transport Layer

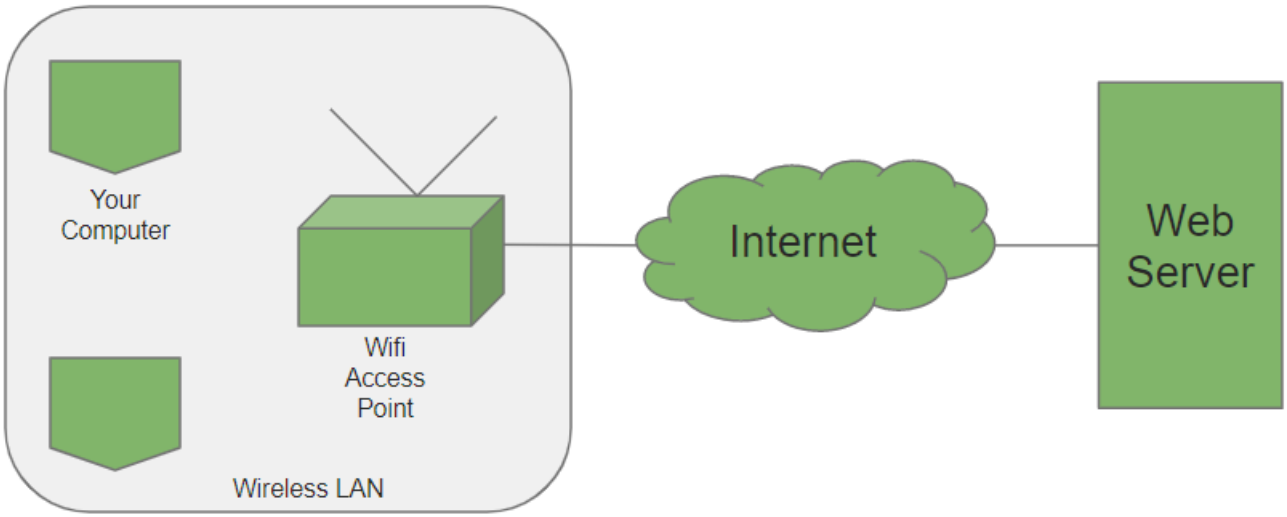
Transport Layer is the layer which lies just above the Network layer and is responsible for **end-to-end connectivity**. It is so-called because it provides point-to-point rather than hop-to-hop. The unit of transmission at the transport layer is called segmentation. **TCP** (Transmission Control Protocol), **UDP** (User Datagram Protocol) and **DCCP** (Datagram Congestion Control Protocol) are some of the protocols running in the transport layer. The transport layer also provides the acknowledgement of the successful data transmission and re-transmits the data if an error is found.

At sender's side: Transport layer receives the formatted data from the upper layers, performs **Segmentation** and also implements **Flow & Error control** to ensure proper data transmission. It also adds Source and Destination port number in its header and forwards the segmented data to the Network Layer.

Note: The sender need to know the port number associated with the receiver's application. Generally, this destination port number is configured, either by default or manually. For example, when a web application makes a request to a web server, it typically uses port number 80, because this is the default port assigned to web applications. Many applications have default port assigned.

At receiver's side: Transport Layer reads the port number from its header and forwards the Data which it has received to the respective application. It also performs sequencing and reassembling of the segmented data.

Let's look at the diagrammatic representation of the working at the transport layer:



Here is a list of few important port numbers and there uses:

PORT number	Use
80	HTTP
443	HTTPS
53	DNS
22	SSH
110	POP3
25	SMTP

Transport Layer has the following responsibilities:

- **Process to process delivery -** While Data Link Layer requires the MAC address (48 bits address contained inside the Network Interface Card of every host machine) of source-destination hosts to correctly deliver a frame and Network layer requires the IP address for appropriate routing of packets , in a similar way Transport Layer requires a Port number to correctly deliver the segments of data to the correct process amongst the multiple processes running on a particular host. A **port number** is a 16 bit address used to identify any client-server program uniquely.

- **End-to-end Connection between hosts -** The transport layer is also responsible for creating the end-to-end Connection between hosts for which it mainly uses TCP and UDP. TCP is a secure, connection- orientated protocol which uses a handshake protocol to establish a robust connection between two end- hosts. TCP ensures reliable delivery of messages and is used in various applications. UDP, on the other hand, is a stateless and unreliable protocol which ensures best-effort delivery. It is suitable for the applications which have little concern with flow or error control and requires to send the bulk of data like video conferencing. It is often used in multicasting protocols.

- **Multiplexing and Demultiplexing -** Multiplexing allows simultaneous use of different applications over a network which is running on a host. The transport layer provides this mechanism which enables us to send packet streams from various applications simultaneously over a network. The transport layer accepts these packets from different processes differentiated by their port numbers and passes them to the network layer after adding proper headers. Similarly, Demultiplexing is required at the receiver side to obtain the data coming from various processes. Transport receives the segments of data from the network layer and delivers it to the appropriate process running on the receiver's machine.

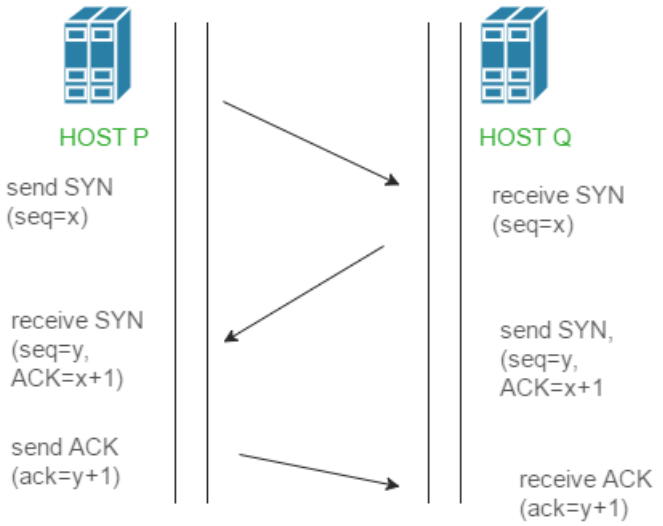
- **Congestion Control -** Congestion is a situation in which too many sources over a network attempt to send data and the router buffers start overflowing due to which loss of packets occur. As a result retransmission of packets from the sources increases the congestion further. In this situation, the Transport layer provides Congestion Control in different ways. It uses **open loop** congestion control to prevent the congestion and**closed loop** congestion control to remove the congestion in a network once it occurred. TCP provides AIMD- additive increase multiplicative decrease, leaky bucket technique for congestion control.

- **Data integrity and Error correction -** Transport layer checks for errors in the messages coming from application layer by using error detection codes, computing checksums, it checks whether the received data is not corrupted and uses the ACK and NACK services to inform the sender if the data has arrived or not and checks for the integrity of data.

- **Flow control -** The transport layer provides a flow control mechanism between the adjacent layers of the TCP/IP model. TCP also prevents data loss due to a fast sender and slow receiver by imposing some flow control techniques. It uses the method of sliding window protocol which is accomplished by the receiver by sending a window back to the sender informing the size of data it can receive.

– TCP & UDP (Differences)

TCP is a connection-oriented, stateful protocol which ensures security, reliability in data transfer. It is established as a **3-way handshake process** given below:



1. **SYN**: In the first step, sender sends a segment with SYN message (containing Synchronize Sequence Number) expressing it's wish to establish a connection. The Sequence No. determines what segment it wants to start the communication with.

2. **SYN + ACK**: Receiver responds by replying with a segment with *SYN-ACK* bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with.

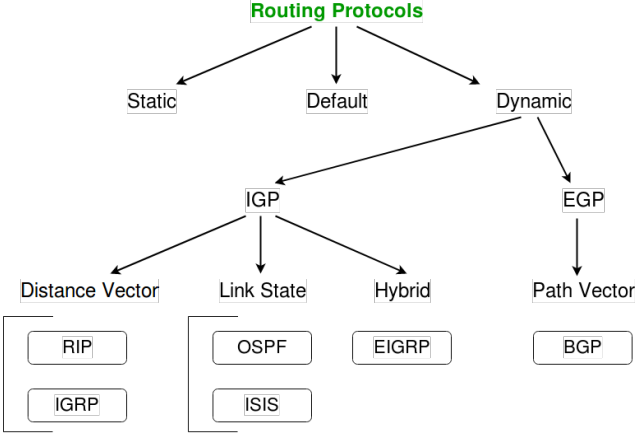
3. **ACK**: In the final part, sender acknowledges the response and they both establish a reliable connection with which actual data transfer can occur.

UDP on the other hand is a connection-less protocol, which doesn't care about reliability. Thus, if some packets get lost, they are skipped. It is thus used in applications where it doesn't matter if we lose out some data. e.g. Video Streaming/Call , VoIP (Voice-over-IP), Multiplayer Games. UDP's main focus is transmission speed (thus re-transmissions are not done in this protocol). More differences between TCP and UDP are given below:

Transmission control protocol (TCP)	User datagram protocol (UDP)
TCP is a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast type of network transmission.
TCP is reliable as it guarantees delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
TCP provides extensive error checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error checking mechanism using checksums.
Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in-order at the receiver.	There is no sequencing of data in UDP. If ordering is required, it has to be managed by the application layer.
TCP is comparatively slower than UDP.	UDP is faster, simpler and more efficient than TCP.
Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in User Datagram Protocol (UDP).
TCP header size is 20 bytes.	UDP Header size is 8 bytes.
TCP is heavy-weight.	UDP is lightweight.
TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP.

– Introduction to Routing Algorithms

Routing is the process of guiding a data packet over a network from source to destination by finding the optimal path (according to some algorithm which we will discuss). The network layer is responsible for routing, and it takes into account hop-count to decide the best possible route for a packet. Routing algorithms define the best path that is required to link the autonomous systems both from within and outside. Routing protocols are categorized as:



Now let's understand various types and key-terms related to Routing. There are 3 types of routing:

1. Static routing - Static routing is a process in which we have to manually add routes in the routing table.

Advantages -

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because the only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage -

- For a large network, it is a hectic task for an administrator to manually add each route for the network in the routing table on each router.
- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

Configuration -

R1 having IP address 172.16.10.6/30 on s0/0/1, 192.168.10.1/24 on fa0/0.
R2 having IP address 172.16.10.2/30 on s0/0/0, 192.168.20.1/24 on fa0/0.
R3 having IP address 172.16.10.5/30 on s0/1, 172.16.10.1/30 on s0/0, 10.10.10.1/24 on fa0/0.

Now configuring static routes for router R3:

```
R3(config)#ip route 192.168.10.0 255.255.255.0 172.16.10.2

R3(config)#ip route 192.168.20.0 255.255.255.0 172.16.10.6
```

Here, provided the route for 192.168.10.0 network where 192.168.10.0 is its network I'd and 172.16.10.2 and 172.16.10.6 are the next-hop address.

Now, configuring for R2:

```
R2(config)#ip route 192.168.20.0 255.255.255.0 172.16.10.1

R2(config)#ip route 10.10.10.0 255.255.255.0 172.16.10.1

R2(config)#ip route 172.16.10.4 255.255.255.0 172.16.10.1
```

Similarly for R1:

```
R1(config)#ip route 192.168.10.0 255.255.255.0 172.16.10.5

R1(config)#ip route 10.10.10.0 255.255.255.0 172.16.10.5

R1(config)#ip route 172.16.10.0 255.255.255.0 172.16.10.5
```

2. Default Routing - This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to router which is configured for default routing. It is generally used with stub routers. A stub router is a router which has only one route to reach all other networks.
Configuration - Using the same topology which we have used for the static routing before.

In this topology, R1 and R2 are stub routers so we can configure default routing for both these routers.
Configuring default routing for R1:

```
R1(config)#ip route 0.0.0.0 0.0.0.0  172.16.10.5
```

Now configuring default routing for R2:

```
R2(config)#ip route 0.0.0.0 0.0.0.0  172.16.10.1
```

3. Dynamic Routing - Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to reach it. RIP and OSPF are the best examples of dynamic routing protocol. Automatic adjustment will be made to reach the network destination if one route goes down.

A dynamic protocol have following features:

1. The routers should have the same dynamic protocol running in order to exchange routes.
2. When a router finds a change in the topology then router advertises it to all other routers.

Advantages -

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage -

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing.

After understanding the types, let's focus on 3 different classes of routing protocols:

1. Distance Vector Routing Protocol - These protocols select the best path in the basis of hop counts to reach a destination network in a particular direction. A dynamic protocol like RIP is an example of a distance vector routing protocol. Hop count is each router which occurs in between the source and the destination network. The path with the least hop count will be chosen as the best path.

Features -

- Updates of network are exchanged periodically.
- Updates (routing information) is always broadcast.
- Full routing tables are sent in updates.
- Routers always trust on routing information received from neighbor routers. This is also known as routing on rumors.

Disadvantages -

- As the routing information are exchanged periodically, unnecessary traffic is generated which consumes available bandwidth.
- As full routing tables are exchanged, therefore it has security issues. If an authorized person enters the network, then the whole topology will be very easy to understand.
- Also broadcasting of network periodically creates unnecessary traffic.

2. Link State Routing Protocol - These protocols know more about the Internetwork than any other distance vector routing protocol. These re also known as SPF (Shortest Path First) protocol. OSPF is an example of a link-state routing protocol.

Features -

- Hello messages, also known as keep-alive messages are used for neighbor discovery and recovery.
- Concept of triggered updates are used i.e updates are triggered only when there is a topology change .
- Only that much updates are exchanged which is requested by the neighbor router.

Link state routing protocol maintains three tables namely:

1. **Neighbor table-** the table which contains information about the neighbors of the router only, i.e, to which adjacency has been formed.
2. **Topology table-** This table contains information about the whole topology i.e contains both best and backup routes to particular advertised network.
3. **Routing table-** This table contains all the best routes to the advertised network.

Advantages -

- As it maintains separate tables for both best route and the backup routes (whole topology) therefore it has more knowledge of the internetwork than any other distance vector routing protocol.
- Concept of triggered updates are used therefore no more unnecessary bandwidth consumption is seen like in distance vector routing protocol.
- Partial updates are triggered when there is a topology change, not a full update like distance vector routing protocol where the whole routing table is exchanged.

3. Advanced Distance vector routing protocol - It is also known as hybrid routing protocol which uses the concept of both distance vector and link-state routing protocol. Enhanced Interior Gateway Routing Protocol (EIGRP) is an example of this class if routing protocol. EIGRP acts as a link-state routing protocol as it uses the concept of Hello protocol for neighbour discovery and forming an adjacency. Also, partial updates are triggered when a change occurs. EIGRP acts as a distance-vector routing protocol as it learned routes from directly connected neighbours.

Now let's understand a few major types of protocols.

- Interior Gateway Protocol (IGP)
- Exterior Gateway Protocol (EGP)
- Routing Information Protocols(RIP)
- Open Shortest Path First (OSPF)
- Enhanced interior gateway routing protocol (EIGRP)
- Border Gateway Protocol (BGP)
- Intermediate System-to-Intermediate System (IS-IS)
- Routing information protocols (RIP)

Interior Gateway protocol (IGRP)
It is distance vector IGP (Interior Gateway Protocol) make-believe by Cisco. The router used it to exchange routing data within an independent system. Interior gateway routing protocol created in part to defeat the confines of RIP (Routing Information Protocol) in large networks. It maintains multiple metrics for each route as well as reliability, MTU, delay load, and bandwidth. The maximum hop of EIGRP is 255 and routing updates are transmitting 90 seconds. It measured in the classful routing protocol, but it is less popular because of wasteful of IP address space. It consists of Distance Vector and Link State.

Exterior Gateway Protocol (EGP)
The absolute routing protocol for internet is exterior gateway protocol which is specified in 1982 by Eric C. EGP (Exterior Gateway Protocol) initially expressed in RFC827 and properly specified in RFC 904 in 1984. The Exterior Gateway Protocol (EGP) is unlike distance vector and path vector protocol. It is a topology just like a tree. It consists of the Path Vector like BGP.

RIP (Routing Information Protocol)
This is a forceful protocol type used in local area network and wide area network. RIP (Routing Information Protocol) type is categorized interior gateway protocol within the use of distance vector algorithm. Routing information protocols defined in 1988. It also has version 2 and nowadays both versions are in use. Technically it is outdated by more sophisticated techniques such as (OSPF) and the OSI protocol IS-IS.

Open shortest path first (OSPF)
Open Shortest Path First (OSPF) is an active routing protocol used in the internet protocol. Particularly it is a link-state routing protocol and includes into the group of interior gateway protocol. Open Shortest Path First (OSPF) operating inside a distinct autonomous system. The version 2 of Open Shortest Path First (OSPF) defined in 1998 for IPv4 then the OSPF version 3 in RFC 5340 in 2008. The Open Shortest Path First (OSPF) most widely used in the network of big business companies.

The Enhanced Interior Gateway Routing Protocol (EIGRP)
Enhanced Interior Gateway Routing Protocol (EIGRP) based on their original IGRP while it is a Cisco proprietary routing protocol. It is a distance-vector routing protocol in advance within the optimization to lessen both the routing unsteadiness incurred after topology alteration, plus the use of bandwidth and processing power in the router which support enhanced interior gateway routing protocol will automatically reallocate route information to IGRP (Enhanced Interior Gateway Routing Protocol) neighbours by exchanging the 32 bit EIGRP (Enhanced Interior Gateway Routing Protocol) metric to the 24 bit IGRP metric. Generally, optimization based on DUAL work from SRI which assured loop-free operation and offer a means for the speedy junction.

Border Gateway Protocol (BGP)
Border Gateway Protocol (BGP) is the core routing protocol of the internet and responsible to maintain a table of Internet protocol networks which authorize network reaching capability between AS. The Border Gateway Protocol (BGP) expressed as the path vector protocol. It doesn’t employ conventional IGP metrics but making a routing judgment based on path, network policies. It is created to replace the Exterior Gateway Protocol (EGP) routing protocol to permit completely decentralized routing in order to permit the removal of the NSF Net which consent to the internet to turn into a truly decentralized system. The fourth version of Border Gateway Protocol (BGP)has been in use since 1994 and the 4th version from 2006. The 4 version RFC 4271

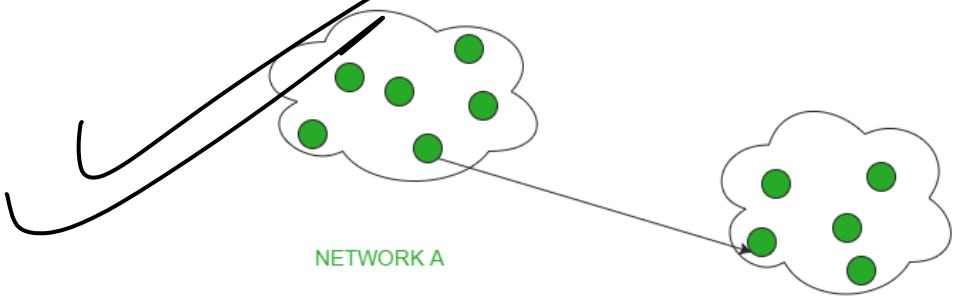
has many features such as it correct a lot of previous errors, illuminating vagueness and brought t the RFC much nearer to industry practice.

Intermediate System-to-Intermediate System (IS-IS)
Intermediate System-to-Intermediate System (IS-IS)is a great protocol used by network devices to determine the best way to promoted datagram from side to side a packet-switched network and this process is called routing. It was defined in ISO/IEC 10589 2002 within the OSI reference design. Intermediate system-to-intermediate system (IS-IS) differentiate among levels such as level 1and level 2. The routing protocol can be changed without contacting the intra-area routing protocol.

- Introduction to Routing Algorithms 2

1. Unicast -

This type of information transfer is useful when there is a participation of a single sender and a single recipient. So, in short, you can term it as a one-to-one transmission. For example, a device having IP address 10.1.2.0 in a network wants to send the traffic stream(data packets) to the device with IP address 20.12.4.2 in the other network, then unicast comes into the picture. This is the most common form of data transfer over the networks.

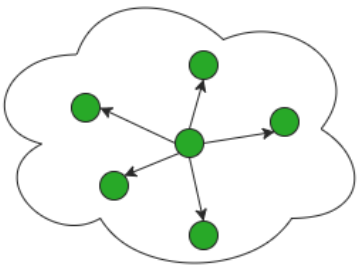


UNICAST EXAMPLE

2. Broadcast -

Broadcasting transfer (one-to-all) techniques can be classified into two types :

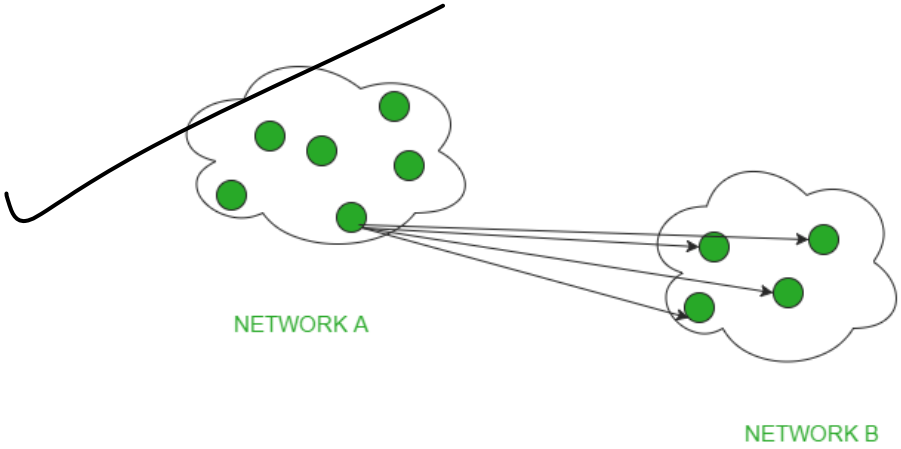
- **Limited Broadcasting -** Suppose you have to send a stream of packets to all the devices over the network that you reside, this broadcasting comes handy. For this to achieve,it will append 255.255.255.255 (all the 32 bits of IP address set to 1) called as **Limited Broadcast Address** in the destination address of the datagram (packet) header which is reserved for information transfer to all the recipients from a single client (sender) over the network.



NETWORK CLUSTER

- **Direct Broadcasting -** This is useful when a device in one network wants to transfer packet stream to all the devices

over the other network. This is achieved by translating all the Host ID part bits of the destination address to 1, referred as **Direct Broadcast Address** in the datagram header for information transfer.



This mode is mainly utilized by television networks for video and audio distribution.
One important protocol of this class in Computer Networks is Address Resolution Protocol (ARP) that is used for resolving IP address into physical address which is necessary for underlying communication.

3. Multicast -

In multicasting, one/more senders and one/more recipients participate in data transfer traffic. In this method traffic recline between the boundaries of unicast (one-to-one) and broadcast (one-to-all). Multicast lets server's direct single copies of data streams that are then simulated and routed to hosts that request it. IP multicast requires the support of some other protocols like **IGMP (Internet Group Management Protocol), Multicast routing** for its working. Also in Classful IP addressing **Class D** is reserved for multicast groups.

4. Anycast -

Anycast is communication between a single sender and the nearest of several receivers in a group. It is a traffic routing algorithm used for the speedy delivery of website content that advertises individual IP addresses on multiple nodes. User requests are directed to specific nodes based on such factors as the capacity and health of your server, as well as the distance between it and the website visitor. Anycast packet forwarding is a mechanism where multiple hosts can have the same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in the routing topology.

Since we know that there is a cost of every link that is created in routing algorithms, let's understand how could one decide this costs of links.

1. **Bandwidth:** This is the most used criteria as efficient use of bandwidth is the primary goal of any algorithms.
2. **Network Delay:** Consists of propagation and transmit delays.
3. **Hop Count:** This must be minimized for better efficiency.
4. **Path Costs:** Various paths result in various costs, sp one needs to choose wisely.
5. **Load:** Due to heavy congestions, the cost of load increases.
6. **Maximum Transmission unit:** The packet size that can be transferred over a link also affects the cost of link.

Now let's understand various Goals of Routing Algorithms:

1. Should correctly deliver
2. Efficient utilization of bandwidth

3. Should not starve a node

4. Should handle the changes well like there should be a fast convergence when any of the following situation arises
    - When the router goes down
    - When the link goes down
    - change in costs occur
    - addition of a new router

- Routing Protocols (DVR & LSR)

A **distance-vector routing (DVR)** protocol requires that a router inform its neighbours of topology changes periodically. Historically known as the old ARPANET routing algorithm (or known as Bellman-Ford algorithm).

Bellman Ford Basics : Each router maintains a Distance Vector table containing the distance between itself and ALL possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbours' distance vectors.

Information kept by DV router -
- Each router has an ID
- Associated with each link connected to a router, there is a link cost (static or dynamic).
- Intermediate hops

Distance Vector Table Initialization -
- Distance to itself = 0
- Distance to ALL other routers = infinity number.

Distance Vector Algorithm -

1. A router transmits its distance vector to each of its neighbors in a routing packet.
2. Each router receives and saves the most recently received distance vector from each of its neighbors.
3. A router recalculates its distance vector when:
    - It receives a distance vector from a neighbor containing different information than before.
    - It discovers that a link to a neighbor has gone down.

The DV calculation is based on minimizing the cost to each destination

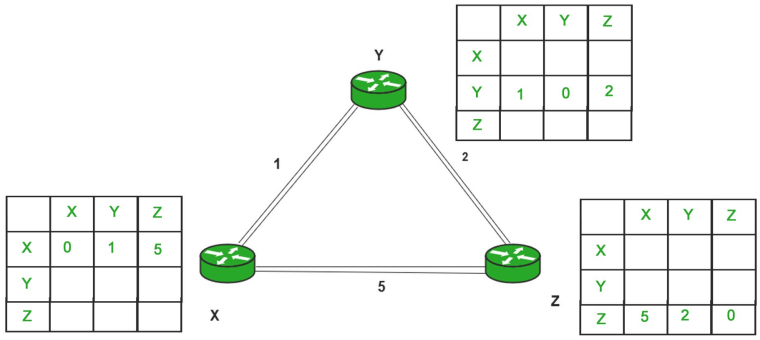
Dx(y) = Estimate of least cost from x to y
C(x,v) = Node x knows cost to each neighbor v
Dx = [Dx(y): y $\in$ N] = Node x maintains distance vector
Node x also maintains its neighbors' distance vectors
– For each neighbor v, x maintains Dv = [Dv(y): y $\in$ N]

Note -

- From time-to-time, each node sends its own distance vector estimate to neighbors.
- When a node x receives new DV estimate from any neighbor v, it saves v's distance vector and it updates its own DV using B-F equation:

  $Dx(y) = \min \{ C(x,v) + Dv(y)\}$ for each node $y \in N$

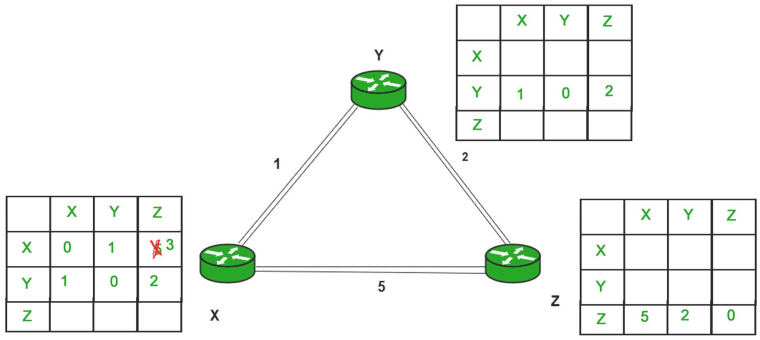
Example - Consider 3-routers X, Y and Z as shown in figure. Each router have their routing table. Every routing table will contain distance to the destination nodes.



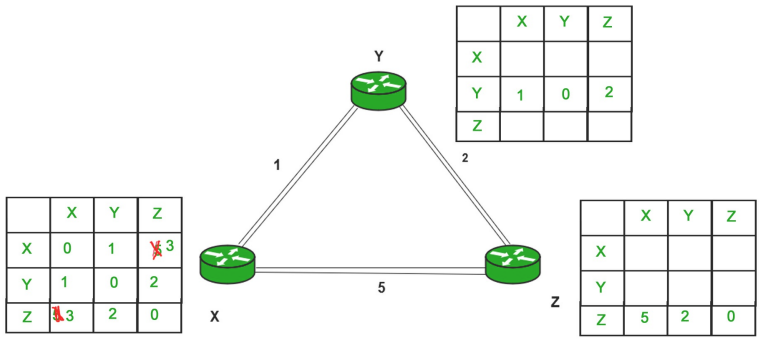
Consider router X , X will share it routing table to neighbors and neighbors will share it routing table to it to X and distance from node X to destination will be calculated using bellmen- ford equation.

$$D_x(y) = \min \{ C(x,v) + D_v(y)\} \text{ for each node } y \in N$$

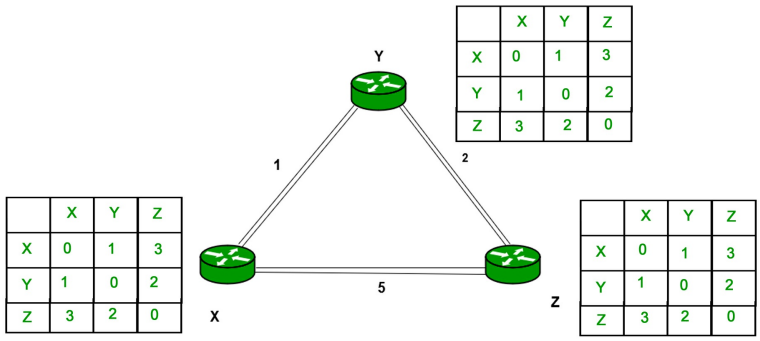
As we can see that distance will be less going from X to Z when Y is intermediate node(hop) so it will be update in routing table X.



Similarly for Z also -



Finally the routing table for all -



Advantages of Distance Vector routing -

- It is simpler to configure and maintain than link-state routing.

Disadvantages of Distance Vector routing -

- It is slower to converge than link state.
- It is at risk from the count-to-infinity problem.
- It creates more traffic than link-state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.
- For larger networks, distance vector routing results in larger routing tables than link-state since each router must know about all other routers. This can also lead to congestion on WAN links.

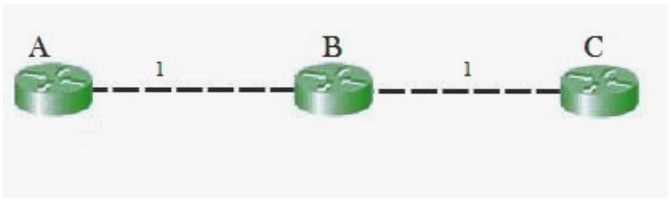
Note - Distance Vector routing uses UDP(User datagram protocol) for transportation.

The main issue with **Distance Vector Routing (DVR)** protocols is Routing Loops, since Bellman-Ford Algorithm cannot prevent loops. This routing loop in DVR network causes Count to Infinity Problem. Routing loops usually occur when an interface goes down or two routers send updates at the same time.

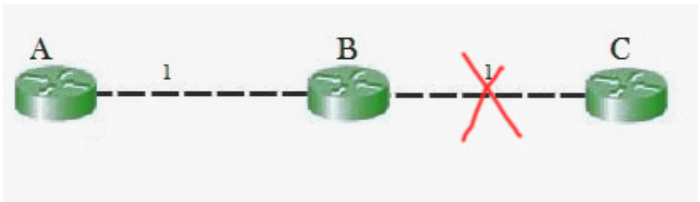
NOTE: RIP (Routing Information Protocol) is the practical implementation of this DVR algorithm. This RIP VI was first implemented in 1988. Few important points regarding this routing protocol are:

1. Hop count was used as a distance.
2. Maximum 15 distance. 16 was considered as infinite.
3. Used Poison reverse and Split Horizon for the count to infinite problem.
4. Return send distance vectors after 30 seconds.

Counting to infinity problem:



So in this example, the Bellman-Ford algorithm will converge for each router, they will have entries for each other. B will know that it can get to C at a cost of 1, and A will know that it can get to C via B at a cost of 2.



If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it

from its table. Before it can send any updates it's possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2. B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3. A will then receive updates from B later and update its cost to 4. They will then go on feeding each other bad information toward infinity which is called as **Count to Infinity problem**.

Let's look at this example and figure out what happens in each step:

Good news travel fast

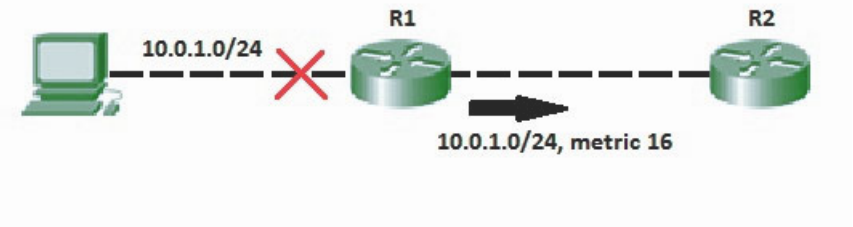
A	B	C	D	E
	.	.	.	.
	1	.	.	.
	1	2	.	.
	1	2	3	.
	1	2	3	4

Bad news travel slow

	A	B	C	D	E
Initial		1	2	3	4
After 1st Exchange		3	2	3	4
After 2nd Exchange		3	4	3	4
After 3rd Exchange		5	4	5	4
After 4th Exchange		5	6	5	6
After 5th Exchange		7	6	7	6
After 6th Exchange		7	8	7	8

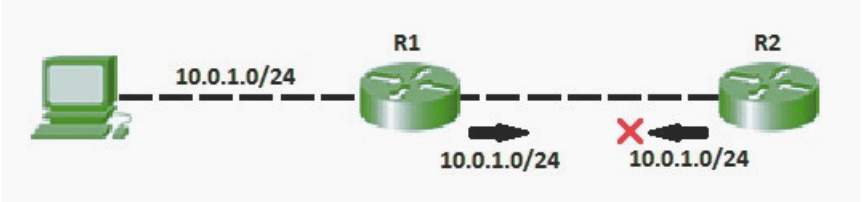
Solution for Count to Infinity problem:-

Route Poisoning: When a route fails, distance vector protocols spread the *bad news* about a route failure by poisoning the route. Route poisoning refers to the practice of advertising a route, but with a special metric value called Infinity. Routers consider routes advertised with an infinite metric to have failed. Each distance vector routing protocol uses the concept of an actual metric value that represents infinity. RIP defines infinity as 16. The main disadvantage of poison reverse is that it can significantly increase the size of routing announcements in certain fairly common network topologies.



Split horizon: If the link between B and C goes down, and B had received a route from A, B could end up using that route via A. A would send the packet right back to B, creating a loop. But according to Split horizon Rule, Node A does not advertise its route for C (namely A to B to C) back to B. On the surface, this seems redundant since B will never route via node A because the route costs more than the direct route from B to C.

Consider the following network topology showing Split horizon-



- In addition to these, we can also use split horizon with route poisoning where above both technique will be used combinedly to achieve efficiency and less increase the size of routing announcements.

- Split horizon with Poison reverse technique is used by Routing Information Protocol (RIP) to reduce routing loops. Additionally, **Holddown timers** can be used to avoid the formation of loops. Holddown timer immediately starts when the router is informed that the attached link is down. Till this time, router ignores all updates of down route unless it receives an update from the router of that downed link. During the timer, If the downlink is reachable again, the routing table can be updated.

Link State Routing Link-state algorithm is also a dynamic algorithm (updates according to topology changes like DVR), but the basic principle of it's working is the ***Djikstra***'s algorithm. The principle works by building the network graph at each router and then applying the shortest-path algorithm. LSR works in 2 phases:

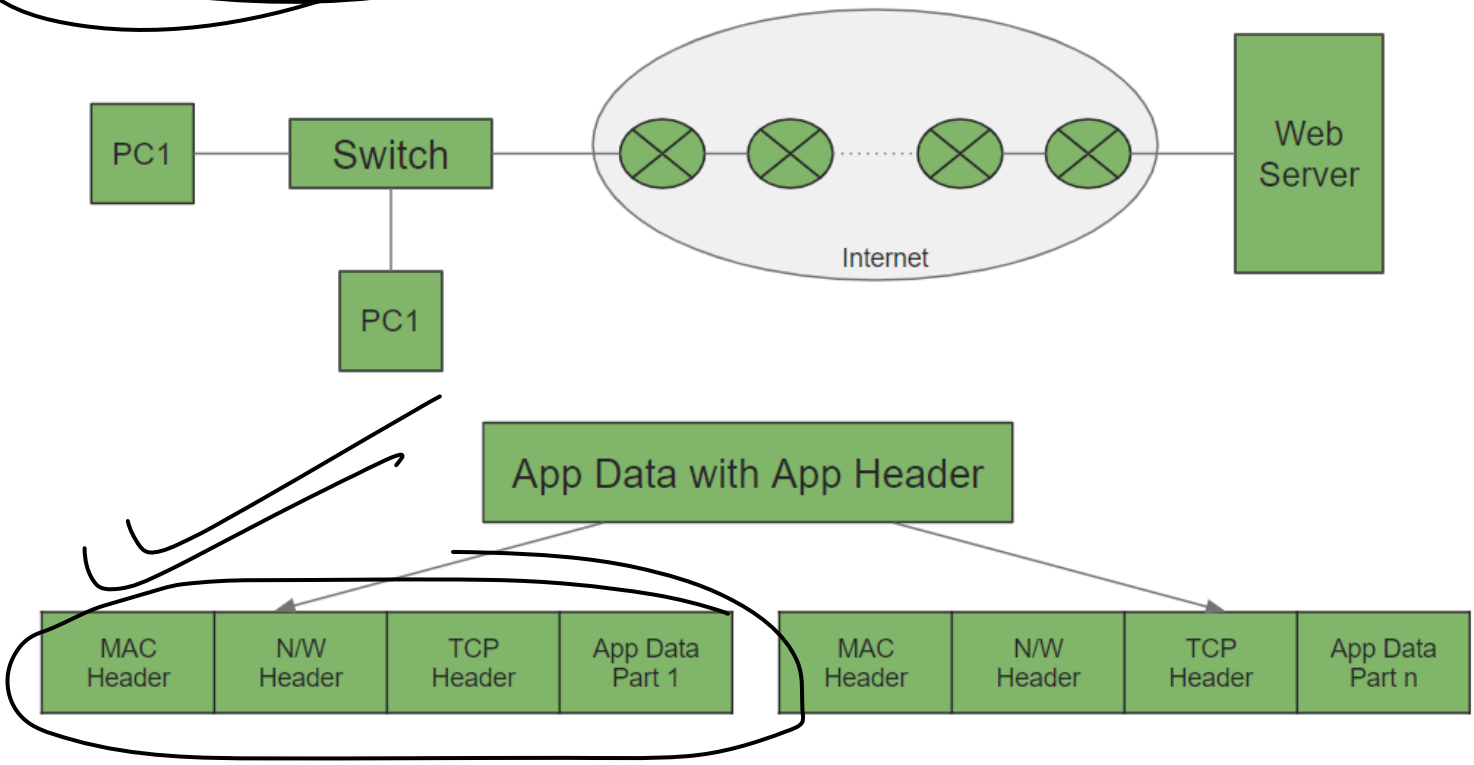
1. ***Reliable Flooding*** - Initially, each router knows the hop-cost to it's directly connected neighbors. This information should be relayed to all the routers (for building the graph). All the routers thus need to share this neighbor information will all the other neighbors. This is done by sending a short message called ***link-state-advertisement***. LSA consists of a
   - Sequence No.
   - Router ID
   - Neighbor Hop-cost

   Each router maintains an LSA table. Upon reception of a message, it compares sequence no. (from table) with the current no. recieved for a router. If the current one is a newer one, it means that topology has updated. Thus, it updates the entry for the router in it's table, and floods the packet to all it's direct neighbors.
2. ***Route Calculation*** - After flooding is completed, all the routers have the required information to build the network graph. Thus, each router then runs independently Djikstra to get the shortest-path and the corresponding next-hop for each destination router.

OSPF (Open-Shortest-Path-First) is the practical implementation of this routing algorithm.

Application Layer

The application layer is the topmost layer of the OSI Model. It is the layer in which user applications run. Various protocols run at this layer serving different requirements. Let's understand the working using the below diagram:



When someone browses the internet, then the browser generates Application Data with specific Header files. Following this, the transport layer breaks the Application data into various parts. To this TCP header is added to each part of the Application Data. Next comes the network layer, which adds the destination address in the network header. Then the link is made between the computer and the ISP routers. All the traffic are being sent to the routers. To send the data over the data link layer to any other routers, the computer uses MAC address where the routers MAC address is used to set the link. This MAC address is known using the ARP or Address Resolution Protocol. The switch reads the MAC header of the destination router. The routers laying on the Internet implements three layers namely network layer, DLL, and physical layer. Now finally when the data reached the webserver then using the TCP header, the webserver combines all the data.

We shall go over in brief over each of the protocols:

- **HTTP**: Stands for **H**yper **T**ext **T**ransfer **P**rotocol. It is a request-response protocol that is used to receive web-pages on a client-server architecture. The client requests for a resource (HTML page, javascript file, images or any other file) to the server. The server returns a response accordingly. It uses TCP as the underlying Transport Layer protocol. HTTP supports the following methods (modes) of requests:
  1. *GET* - Retrieve information from the server (GET Requests is intended only for data fetching and should not have any side-effect).
  2. *HEAD* - Retrieve only header (meta-information) and no response-body.
  3. *POST* - Post/Send data back to the server. e.g. User-data, form-data.
  4. *DELETE* - Delete the specified resource.
  5. *OPTIONS* - Returns the list of HTTP methods supported by the server.

  Port no. for HTTP: 80 (8080 occasionally)
- **HTTPS**: It is a secured version of HTTP made possible by encrypting the data transferred using TLS (Transport-Layer-Security). Earlier, its predecessor ~ SSL was used. The use of HTTPs over HTTP increases security by preventing eavesdropping, tampering and man-in-the-middle attacks. Port used in HTTPs is the same as that of HTTP.
- **TELNET**: Stands for **TEL**ecommunications **NET**work. It is used in terminal emulation, which one can use to access a remote system. It is used for file-access and for the initial setup of switches. One may draw it's similarities to SSH (Secure-Shell), but as the name suggests SSH is secure as it uses encryption in addition to normal terminal emulation. Telnet is thus no longer used thanks to SSH.
  Port no. for Telnet: 23
  Port no. for SSH: 22
- **FTP**: Stands for **F**ile **T**ransfer **P**rotocol. It provides reliable and efficient file-transfer between two remote machines.
  Port no. for FTP Data: 20
  Port no. for FTP Control: 21s
- **SMTP**: Stands for **S**imple **M**ail **T**ransfer **P**rotocol. Uses TCP under the hood. Using a process called “store and forward,” SMTP moves your email on and across networks. It works closely with something called the Mail Transfer Agent (MTA) to send your communication to the right computer and email inbox.
  Port no. for SMTP: 25.
- **DNS**: Stands for **D**omain **N**ame **S**ervice. DNS maps human-addressable english domain names to IP addresses.e.g. www.abc.com might translate to 198.105.232.4. We as humans are comfortable in dealing with named addresses of websites (facebook.com, google.com etc.). However, to uniquely identify the server hosting the application, numeric IP addresses are required by the machine. DNS servers contain this mapping of named addresses to IP addresses. Whenever we us a named address is used, client machine services a request to the DNS server to fetch the IP address.
  Port no. for DNS: 53
- **DHCP**: Stands for **D**ynamic **H**ost **C**onfiguration **P**rotocol. Used for dynamic addressing of devices in a network. DHCP server keeps a pool of available IP addresses. Whenever a new device joins the network, it provides it with an IP from the available pool with an expiration time. DHCP is required in place of static addresses because current requirements involve managing devices which are continuously leaving/joining a network. Thus, a pool of available addresses are required which can be leased to devices currently residing in the network.
  Port no. for DHCP: 67, 68

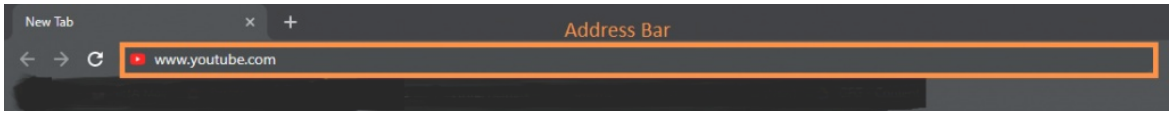
– Domain Name System

We are fortunate to have been living in the age of the internet, whose power and utility we often take for granted. We open our browser, type an URL or a search query of Google, and Boom!!, we are presented with thousands of results which we can browse, follow links within each page and visit multiple websites, the process which is commonly known as web-surfing. But,

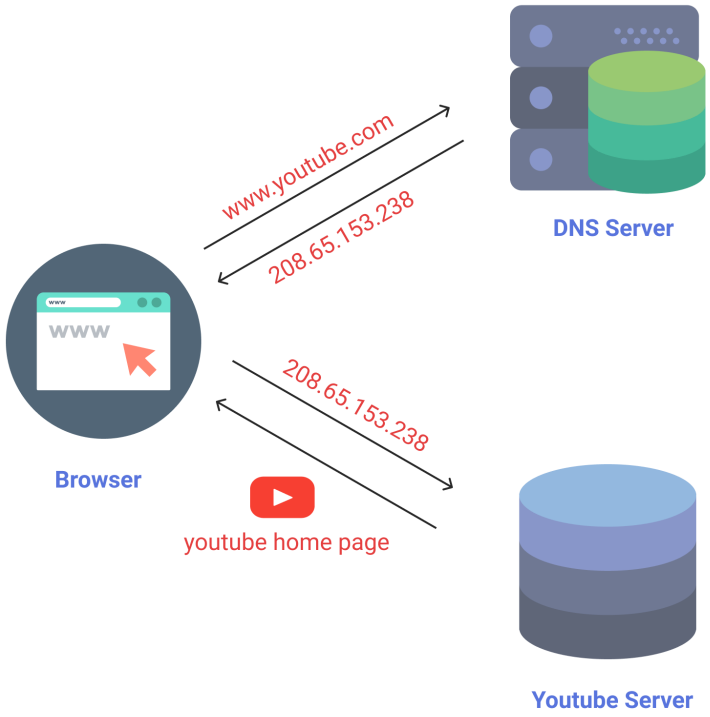
have you ever wondered what happens under the hood of this process which seems so simple and fast to the end-user. We shall cover the whole walkthrough of what happens step-by-step starting from entering an URL on the address bar of the browser till the loading of the actual webpage. So, let's hop on this exciting journey!!

Entering the URL

We use browsers to surf the internet (Chrome, Firefox, Edge, etc.). Each of them has an address-bar at the top where we provide the URL of the website we want to visit -

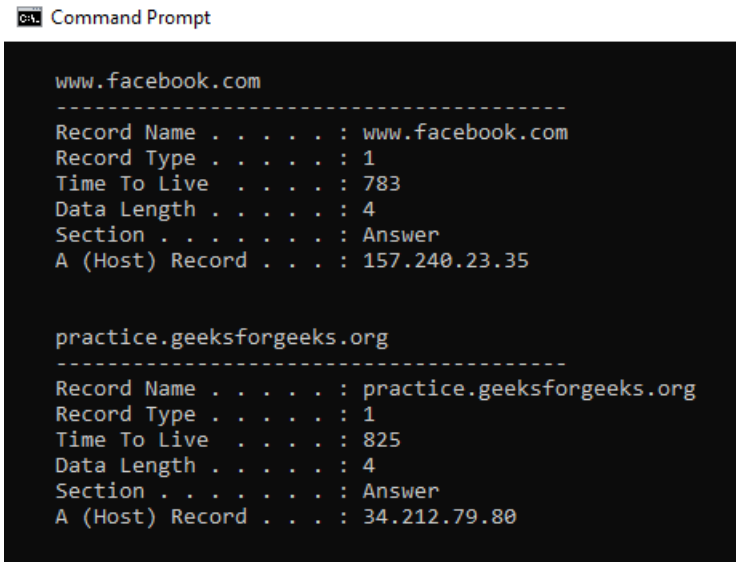


We enter some URL, say *www.google.com* or *youtube.com* as an example and press Enter. After waiting for a while, we are presented with the landing pages of the website. But, we all know Computer Systems can't understand human-language addresses. Also, human-language addresses have many issues such as (uppercase/lowercase, etc.). We have IP addresses (IPv4/IPv6) as the numerical equivalent for addressing in the realm of computer networks. An IP address is unique to a particular system at a time. i.e. A system running currently can't have multiple IP addresses ~ providing us the best scheme for addressing systems present in the network.
However, humans are not good with memorizing numbers (IPs) for websites, so the usage of English-language addresses can't be eliminated. Thus, we require some mechanism to map the English-language addresses to numeric IP addresses. Here, comes the role of DNS, which we cover in the next section. As of know, all we need to know is whenever we type an English-language address, the system calls the DNS server with the URL to get the corresponding IP address. Only, after the browser receives the IP, it can request the actual server for the webpage. The process looks as -



DNS Lookup

DNS calls are an extra overhead which serves us no good in loading the actual website. Thus, it would be very beneficial if we can cache DNS IP values for frequently visited websites in the user-system itself. Thus, comes the concept of DNS caching. Before making a call to the actual DNS server, the browser looks up the DNS cache of the system. The DNS cache looks as -



We can display the DNS cache information in Windows CMD as -

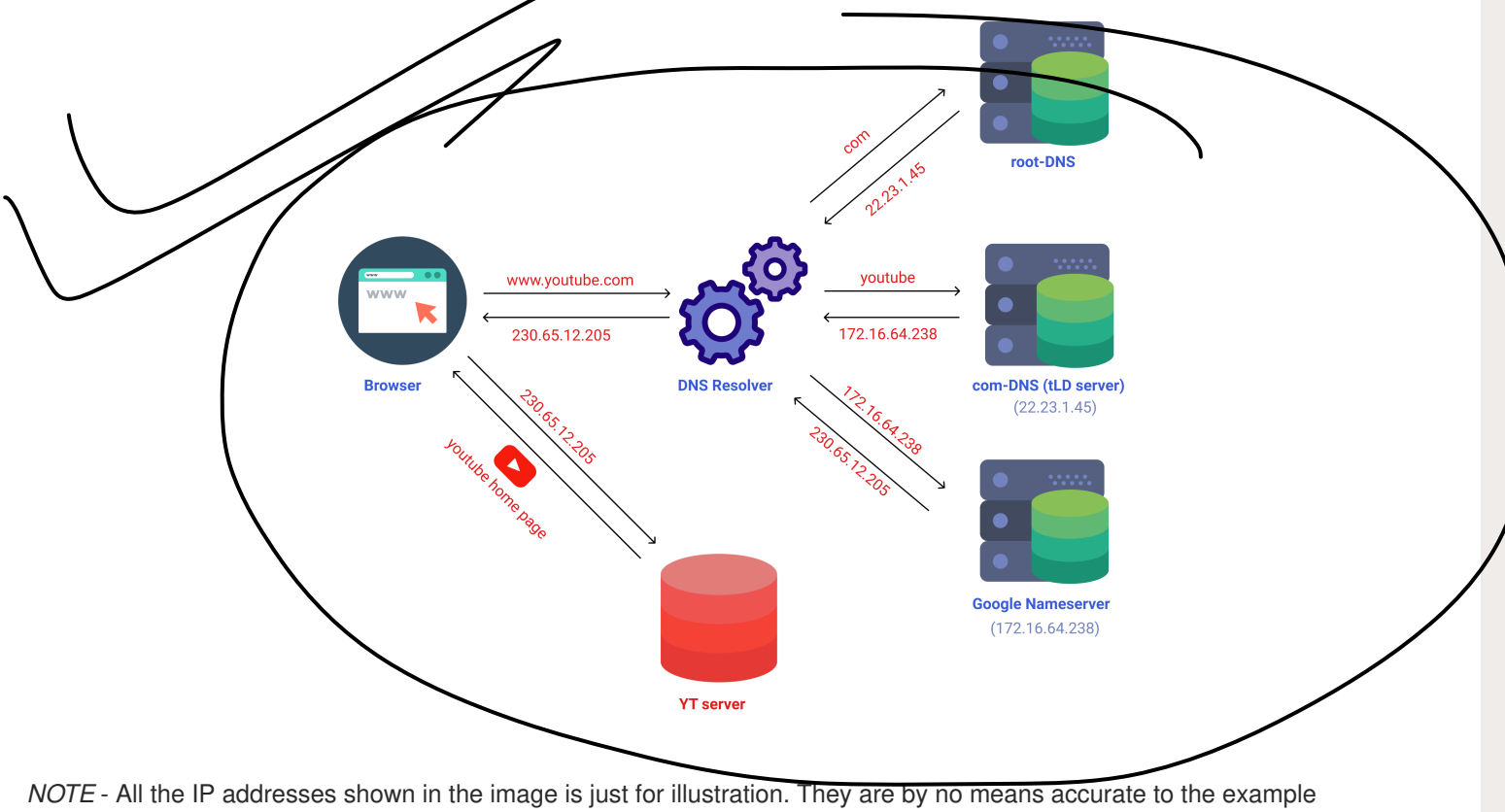
```
ipconfig /displaydns
```

Run

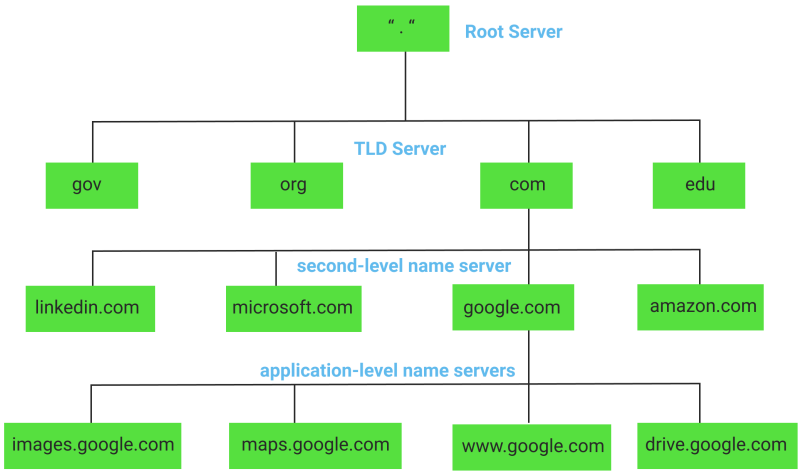
If no entry in Cache is found, then we call the DNS server. DNS server IP is either provided by the ISP, or there are public DNS servers provided by Google (8.8.8.8/8.8.4.4) or OpenDNS (208.67.222.222/208.67.220.220). These settings can be set/adjusted in Network Settings of the system. The system performs a DNS call with the address provided. The type of packets used is generally **UDP** because we require a lot of DNS calls and UDP packets are smaller in size (max. 512 bytes) as compared to TCP. Also, DNS requests are done on a separate port no. ~ 53.

DNS Resolution

We shall understand this with an example. Say we search ***www.youtube.com***. DNS resolution occurs from end to start of the address. i.e. for our query, ***com -> youtube -> www***. The DNS resolver first requests the**root-DNS** with *com* as the search query.
What is the root-DNS Server? Root-DNS is the topmost level DNS server which contains the addresses of tLD (top-level-domain) name-servers such as *com, org, gov*. We have queried *.com*, so it returns the address of *.com tLD server*.
We thereafter query the *.com-tLD* server with the request for *youtube*. This name server looks up into its database and other similar tLDs and returns back the list of name servers matching youtube. Youtube is owned by Google, so we are returned name server list ***ns1.google.com-ns4.google.com***.
We then query one of these Google name servers for youtube, and we get back the IP address for the website which is geographically closest to our location. (Nowadays, the same website is hosted in a distributed fashion over multiple regions).
Diagrammatically -



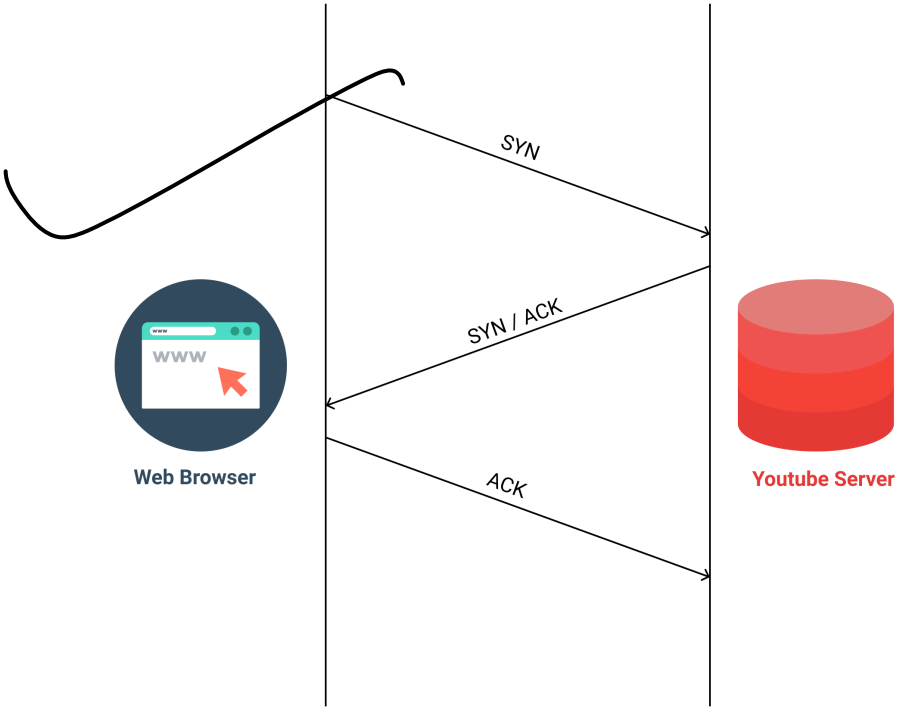
NOTE - All the IP addresses shown in the image is just for illustration. They are by no means accurate to the example discussed.



Above diagram shows the hierarchy of DNS servers and the various levels of resolution. After the browser receives the correct IP address for the requested website, it establishes a TCP connection which we describe below -

TCP Connection & HTTP Request

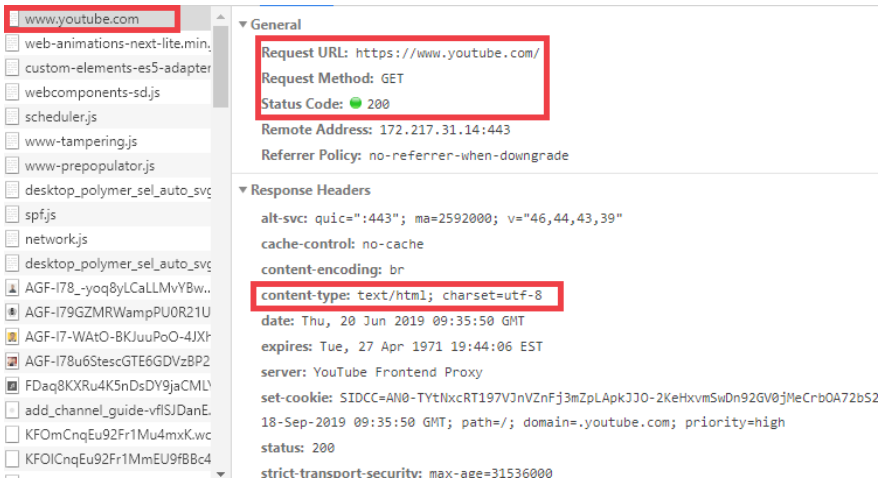
Establishing a TCP connection is a 3-way handshake process which is described in the figure shown below -



The client sends a **SYN** packet to the Youtube server to check whether it can accommodate any new connections or not. The server replies with a SYN/ACK (acknowledgment to the SYN request) back to the client. The client completes the 3-way handshake process by replying with its own acknowledgment (ACK).

Website & Resource Delivery

After client and server are successfully connected via a secure TCP connection, browser issues an HTTP Request to the server demanding it to serve the page requested by the user. The server responds with an HTTP Response (the HTML page containing images, links to videos and other relevant data ~ JSON data perhaps) back to the client. The browser application then renders the source files received onto the user screen as a webpage.



Sub-URL Resolution

What happens when we access, say *geeksforgeeks.org/data-strucure-and-algorithms/* or *geeksforgeeks.org/users/*. i.e. The issue we are trying to tackle here is how the part after the main URL gets resolved once we hit the main server for the website. This issue has been handled in 2 different ways, the 1st one of which has gone obsolete (the reason we discuss why) -

1. ***Seperate HTML files*** - In both the implementations, we are required to have one root file named**ndex.html**. We should have to keep the name as same, as this is the 1st webpage which is required to be served once HTTP request hits the server. It is the root/landing page in case a sub-URL is not specified. i.e. say we access *http://abc.com/*, then index.html present in the *abc server* will get served. Instead, if we access say http://abc.com/feed.html, then the server looks for feed.html file present in the directory and serves that to the client browser. Similarly, http://abc.com/profile.html asks the server to look for profile.html and serve it back. This kind of system relies on the presence of separate HTML files along with other resources (images, js files, CSS files, etc.). Each of them gets served based on the page requested. It is obvious that this kind of system won't scale. Imagine having a profile of millions of users. Then we would have to keep separate files like abc.com/profile/levi.html, abc.com/profile/eren.html, abc.com/profile/erwin.html

.

2. ***Single Bundle File with API end-points*** - Modern systems have a single JS bundle file which contains the basic HTML structure to load in-case each URL is requested. Whenever we request a website, the whole bundle is downloaded into the user system. Accordingly, when we browse to different web-pages, API calls are made and the response data is injected into the HTML template (served by the bundle). Understanding this requires good knowledge of modern front-end frameworks (such as React, Angular, etc.) and REST-API backend frameworks (Nodejs, Django, etc.). Hence, it is out of the scope of this article to explain the overall mechanism in detail.

- DHCP(Dynamic Host Configuration Protocol)

Dynamic Host Configuration Protocol(DHCP) is an application layer protocol which is used to provide:

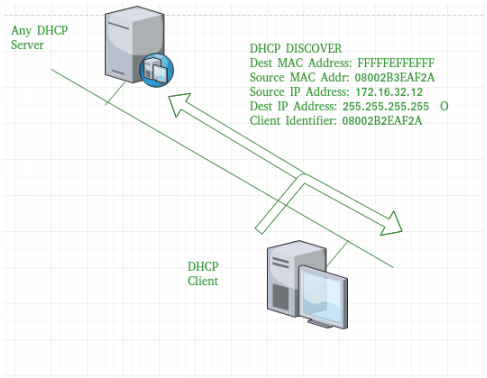
1. Subnet Mask (Option 1 - e.g., 255.255.255.0)
2. Router Address (Option 3 - e.g., 192.168.1.1)
3. DNS Address (Option 6 - e.g., 8.8.8.8)
4. Vendor Class Identifier (Option 43 - e.g., 'unifi' = 192.168.1.9 ##where unifi = controller)

DHCP is based on a client-server model and based on discovery, offer, request, and ACK.

DHCP **port number** for server is 67 and for the client is 68. It is a Client-server protocol which uses UDP services. IP address is assigned from a pool of addresses. In DHCP, the client and the server exchange mainly 4 DHCP messages in order to make a connection, also called DORA process, but there are 8 DHCP messages in the process.

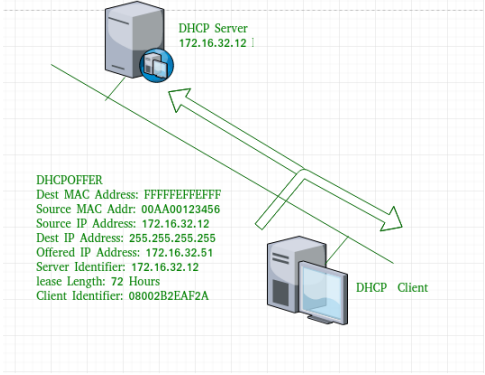
These messages are given as below:

1. **DHCP discover message -** This is a first message generated in the communication process between server and client. This message is generated by Client host in order to discover if there is any DHCP server/servers are present in a network or not. This message is broadcasted to all devices present in a network to find the DHCP server. This message is 342 or 576 bytes long



   As shown in the figure, source MAC address (client PC) is 08002B2EAF2A, destination MAC address(server) is FFFFFFFFFFFF, source IP address is 0.0.0.0(because PC has no IP address till now) and destination IP address is 255.255.255.255 (IP address used for broadcasting). As the discover message is broadcast to find out the DHCP server or servers in the network, therefore, broadcast IP address and MAC address is used.

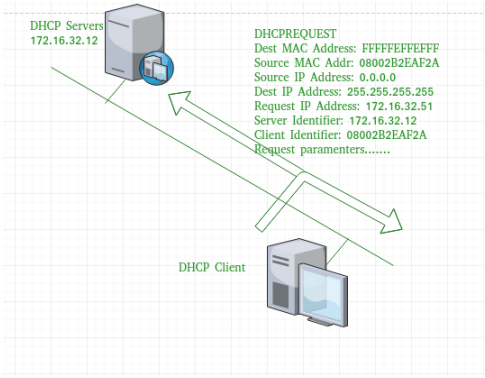
2. **DHCP offer message -** The server will respond to host in this message specifying the unleased IP address and other TCP configuration information. This message is broadcasted by the server. Size of the message is 342 bytes. If there are more than one DHCP servers present in the network then client host will accept the first DHCP OFFER message it receives. Also, a server ID is specified in the packet in order to identify the server.



Now, for the offer message, source IP address is 172.16.32.12 (server's IP address in the example), destination IP address is 255.255.255.255 (broadcast IP address) ,source MAC address is 00AA00123456, destination MAC address is FFFFFFFFFFFF. Here, the offer message is broadcast by the DHCP server, therefore, destination IP address is broadcast IP address and destination MAC address is FFFFFFFFFFFF and the source IP address is the server IP address and MAC address is server MAC address.

Also the server has provided the offered IP address 192.16.32.51 and lease time of 72 hours(after this time the entry of host will be erased from the server automatically) . Also the client identifier is PC MAC address (08002B2EAF2A) for all the messages.

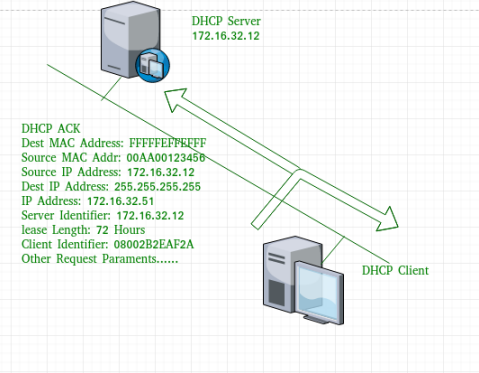
3. **DHCP request message -** When a client receives a offer message, it responds by broadcasting a DHCP request message. The client will produce a gratitutous ARP in order to find if there is any other host present in the network with same IP address. If there is no reply by other host, then there is no host with same TCP configuration in the network and the message is broadcasted to server showing the acceptance of IP address .A Client ID is also added in this message.



Now, the request message is broadcast by the client PC therefore source IP address is 0.0.0.0(as the client has no IP right now) and destination IP address is 255.255.255.255 (broadcast IP address) and source MAC address is 08002B2EAF2A (PC MAC address) and destination MAC address is FFFFFFFFFFFF.

Note - This message is broadcast after the ARP request broadcast by the PC to find out whether any other host is not using that offered IP. If there is no reply, then the client host broadcast the DHCP request message for the server showing the acceptance of IP address and Other TCP/IP Configuration.

4. **DHCP acknowledgement message -** In response to the request message received, the server will make an entry with specified client ID and bind the IP address offered with lease time. Now, the client will have the IP address provided by server.



Now the server will make an entry of the client host with the offered IP address and lease time. This IP address will not be provided by server to any other host. The destination MAC address is FFFFFFFFFFFF and the destination IP address is 255.255.255.255 and the source IP address is 172.16.32.12 and the source MAC address is 00AA00123456 (server MAC address).

5. **DHCP negative acknowledgement message -** Whenever a DHCP server receives a request for IP address that is invalid according to the scopes that is configured with, it send DHCP Nak message to client. Eg-when the server has no IP address unused or the pool is empty, then this message is sent by the server to client.

6. **DHCP decline -** If DHCP client determines the offered configuration parameters are different or invalid, it sends DHCP decline message to the server .When there is a reply to the gratuitous ARP by any host to the client, the client sends DHCP decline message to the server showing the offered IP address is already in use.

7. **DHCP release -** A DHCP client sends DHCP release packet to server to release IP address and cancel any remaining lease time.

8. **DHCP inform -** If a client address has obtained IP address manually then the client uses a DHCP inform to obtain other local configuration parameters, such as domain name. In reply to the dhcp inform message, DHCP server generates DHCP ack message with local configuration suitable for the client without allocating a new IP address. This DHCP ack message is unicast to the client.

Note - All the messages can be unicast also by dhcp relay agent if the server is present in different network.

Advantages - The advantages of using DHCP include:

- centralized management of IP addresses
- ease of adding new clients to a network
- reuse of IP addresses reducing the total number of IP addresses that are required
- simple reconfiguration of the IP address space on the DHCP server without needing to reconfigure each client

The DHCP protocol gives the network administrator a method to configure the network from a centralised area.
With the help of DHCP, easy handling of new users and reuse of IP address can be achieved.

Disadvantages - Disadvantage of using DHCP is:

- IP conflict can occur

GeeksforGeeks

room 5th Floor, A-118, Sector-136, Noida, Uttar Pradesh - 201305

email feedback@geeksforgeeks.org