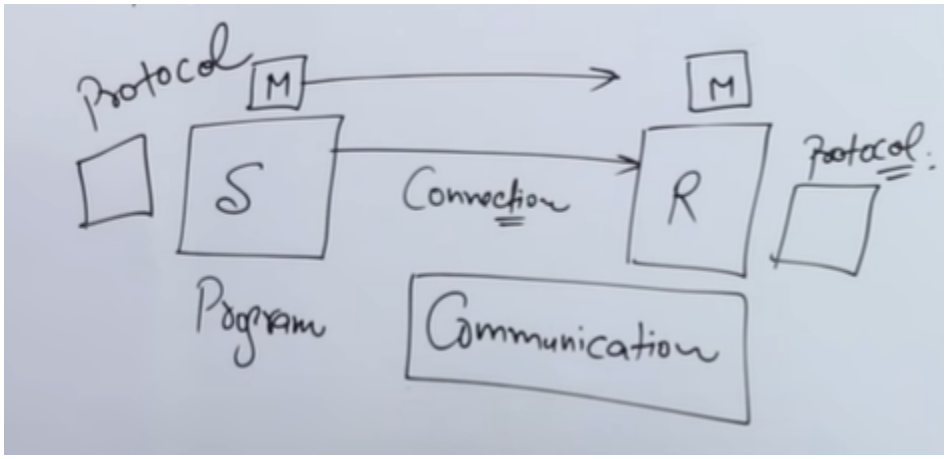
By : Hrithik Maddirala

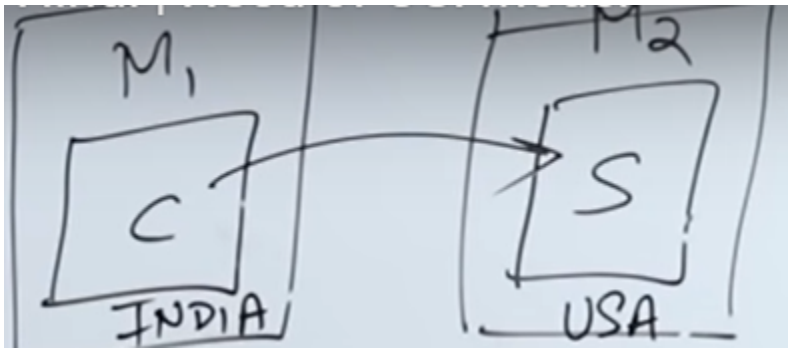
COMPUTER NETWORK NOTES

Computer Network is defined as an interconnection between 2 or more computers which helps facilitate data exchange and effective communication.
It can be seen as a combination of nodes and links, with the nodes being computers, mobile phones, and network devices like router, modem, switches, etc. The links can be optical fiber cables,coaxial cables, etc and free space in case of wireless networks.

Protocols are a set of rules and regulations that govern the data transfer. Protocols help to make sure that the data is being sent by the sender in such a way/format that the receiver is able to understand it.



If the sender & receiver are in the same machine, then it is known as interprocess communication which the OS of the system will handle.
Computer Networks is dealing with scenarios where the sender and the receiver are on different machines. An environment should be created such that the systems don't feel that they are separate.



For example, when sending a request from your host computer to access a webpage, a request is made from the host computer to the server and then the server sends a reply back within a

fraction of seconds. For us as users, we feel that the information is present on our computer itself.

Protocols provide many functionalities :

Mandatory functionalities

_____

1) Error Control : When data is sent from the sender to receiver, we need to make sure that the data is being received as it is. If message M is sent by the sender, message M itself should be received by the receiver.
2) Flow Control: Helps to maintain the rate of flow of packets through the network. If a large amount of data is sent at a time, then problems like congestion within the network can occur. Also helps to make sure that the sender is not sending data at a faster rate than what the receiver can receive at.
3) Multiplexing/Demultiplexing
and so on.....

Optional functionalities

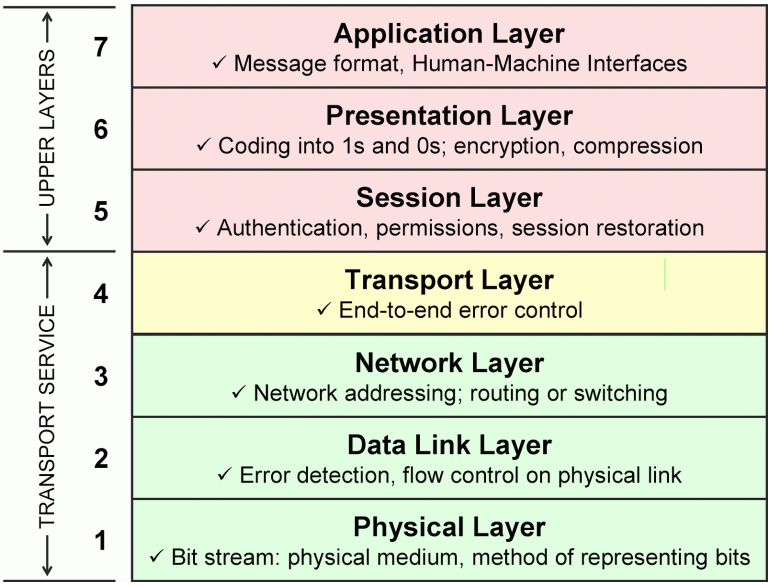
_____

1) Encryption/Decryption : Data could be encrypted before being sent across the network to protect against intruders, hacking, eavesdropping,etc.
2) Checkpoints : For example, while downloading a file of 500MB, if the download fails at 300MB, download should resume at 300MB rather than starting from the beginning. For applications like Whatsapp, checkpoints are not necessary, since the messages are itself very small in size.

OSI MODEL

_____

Stands for Open Systems Interconnection.

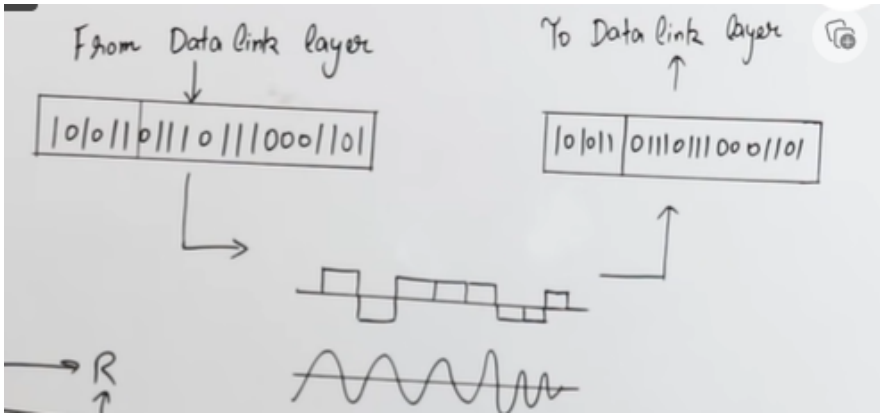


https://takeuforward.org/computer-network/osi-model/

PHYSICAL LAYER

It is the lowermost layer of the OSI Model. On the sender side, the bits from the data link layer will be sent to the physical layer, which is then responsible for converting the bits into signals. These signals are then received by the physical layer on the receiver side and converted back into the bits.

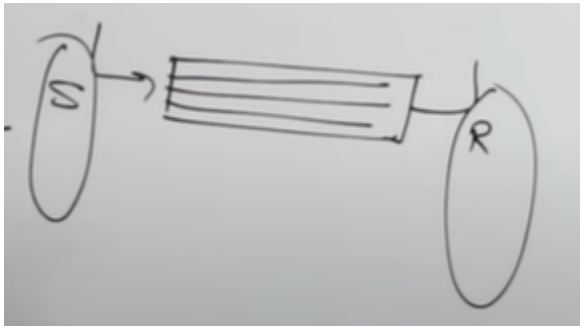
HARDWARE DEVICES BELONGING TO THIS LAYER : Repeaters, and Hubs



Functionalities :

1) Determines the type of cable & connectors to use (i.e optical fiber cables, coaxial cables)
2) Determines the network topology
3) Determines the type of transmission mode (i.e simplex, half duplex, full duplex)
4) Multiplexing/Demultiplexing : Normally a sender will send the signal through the transmission media and then it will be received on the receiver end. If we have multiple senders & multiple receivers communicating at the same time, we don't actually need to

have a separate channel b/w each sender-receiver pair. We can perform multiplexing where we combine all the signals generated by all the senders into a single signal, transmit it across the transmission medium, and then perform demultiplexing on the receiver end so that the desired signal is being sent to the actual receiver.

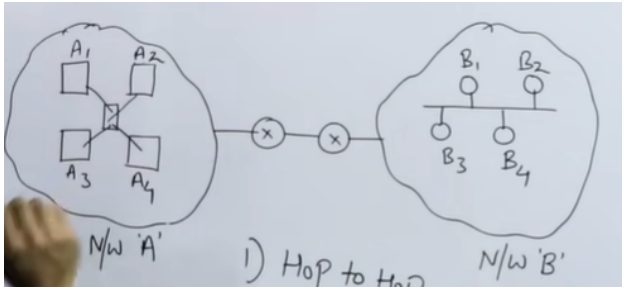


5) Encoding : Should the bits be converted into digital or analog signals

DATA LINK LAYER

It's the second layer of the OSI model, b/w the Physical layer and Network layer. The main functionality of the data link layer is that it helps to facilitate communication between 2 nodes present within the same network through the concept of MAC addresses. Whenever we want communication between 2 nodes in the same network, we can directly do it with the data link layer itself without the help of the network or transport layers. If we want to perform communication between 2 nodes present in different networks, then we have to use IP addresses(i.e MAC addresses cannot be used).

DEVICES : Bridges, switches



Functionalities (i.e Taking example of transporting data from A4 to B1):

1) Hop to Hop delivery : Data link layer is only responsible for the transport of the packet from one hop to another hop. For example if A4 wants to send a message to B1, data link layer will first focus on transporting the data from A4 to R1, then from R1 to R2, …. and finally from R2 to B1.

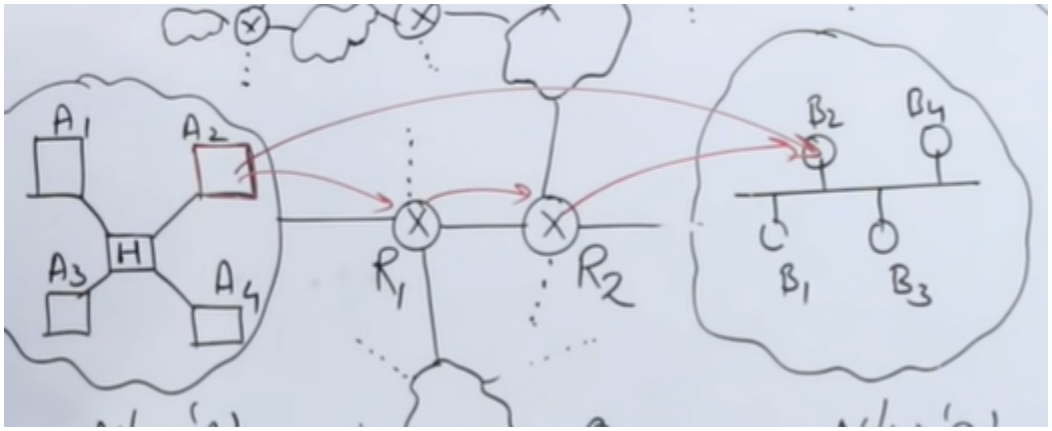
2) Flow control : Helps to make sure that the amount of data being sent is not more than what can be handled. Again flow control is done between one hop to another hop. Data link layer will first make sure that the data isn't being sent at a faster rate by A4 than R1 can handle, and so on…
3) Error control : Done at hop to hop level. Error control is more efficient than the error control done by the transport layer, since the transport layer does error control between the source and the destination but the data link layer is at hop to hop. As soon as an error is detected from one hop to another, we can retransmit the data. It is done by Cyclic Redundancy check at the data link layer. In case of transport layer, we use checksum.
4) Access control : If we take Network B as an example, we see that all the devices are connected through a common media channel. If 2 or more devices send messages at the same time, the data will collide with each other causing issues. Algorithms like CSMA/CD or Aloha are used to perform access control.
5) MAC Addresses : Data link layer adds the MAC addresses to the data that it receives from the network layer on the sender side.
6) Framing : Data link layer will divide the packets arrived from the network layer into fixed sized frames. To the collection of frames, a header as well as a trailer is added by the data link layer.

NETWORK LAYER

The network layer accepts data from the transport layer and provides data to the data link layer. The data within the network layer are known as packets.

DEVICES : Routers & Switches

The network layer is basically responsible for the source to destination delivery of a message (i.e from a host on one network to another host on another network). We know that for transporting messages within a network MAC addresses are sufficient, but to transfer a message between networks we require IP addresses.



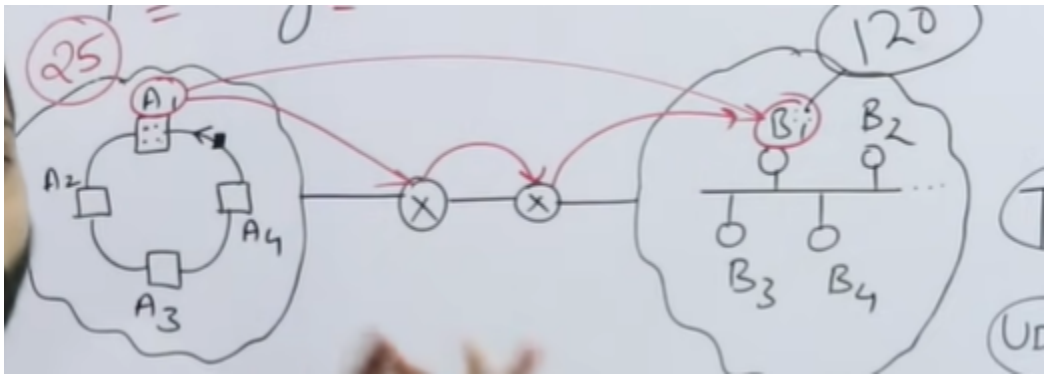
Direct communication b/w A2 and B2 : Network layer
Communication b/w A2 to R1 then R1 to R2 and R2 to B2 : Data link layer

Functionalities :

1) Host to Host delivery : Helps make sure that the message is being sent from source host to destination source belonging to different networks
2) Flow control : Helps to make sure that the amount of data being sent is not more than what can be handled. FLOW CONTROL IS ON HOST TO HOST LEVEL
3) It helps provide the IP addresses. An IP address will have 2 parts : Network part & Host part. The network part specifies which network the packet has to be sent to and the host part specifies to which host within the network. 2 machines within the same network will have the same network part but different host parts.
4) Routing : We know that whenever we are transferring a message from one network to another, the message will be transmitted through intermediate devices like routers. Each router will have its own dynamic routing table based on which the router will decide on which path the packet should be sent to. Routing of packets is one of the responsibilities of the network layer. Usually the shortest path will be preferred.
5) Fragmentation : The data that the network layer receives from the transport layer can be divided into fragments. Sometimes when the data is being sent from one host to another host, the intermediate device may say that the size of the data is larger than what it can accept. For this reason fragmentation of the data is done.

TRANSPORT LAYER

It is the 4th layer from the bottom of the OSI Model. Protocols such as TCP(i.e Transmission Control Protocol) and UDP(i.e User Datagram Protocol) are used in this layer.



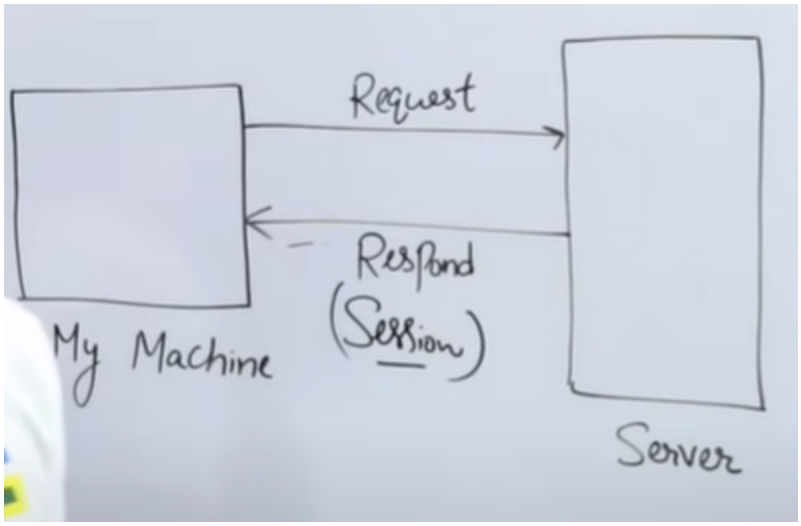
Functionalities :

1) Segmentation : Data that the transport layer receives from the application layer is a continuous stream of bits. We cannot directly send this stream of bits to the network layer and hence the stream is divided into segments(i.e header is added to the segment) after which the data is sent to the network layer.
2) End to End delivery (Port to Port delivery) : We know that the network layer helps perform host to host delivery of the packets. But once the data is received by the receiver, we need to send the data to the appropriate application/process running on that device. For sending the data from the process running on the source machine to the process running on the destination machine, we require the transport layer.
   For example, if we are sending mail using Gmail on the source machine, data will be sent from Port 25 using the SMTP protocol on the source machine to the desired port on the destination.
   NOTE : Port number is a 16 bit number, and there are 3 ranges : Reserved ports (0-1023), Registered ports(1024-49151), and Ephemeral ports(49152-65535).

3) Reliability : When we use the IP protocol in the network layer for transferring data from source to destination, there is no reliability that the data is received correctly. If we use the TCP protocol, it is made sure that the packet is received correctly with the help of acknowledgements.
4) Error control : Error control is formed at the End to End level. In the data link layer we used to use CRC to perform error control but in the transport layer, checksum is done. The process on the destination will compute the checksum and if the checksum is matching with the checksum sent by the source, then the packet is said to be free of errors.
5) Congestion & Flow control : Same things but at the end to end level
6) Multiplexing & Demultiplexing : Multiple applications can be sending data at the same time. All the data will be combined into a single stream(i.e multiplexing) on the sender side and then demultiplexing occurs at the receiver side.

SESSION LAYER

It is the 5th layer from the bottom in the OSI Model. It uses the facilities provided by the presentation layer and provides functionalities to the transport layer.



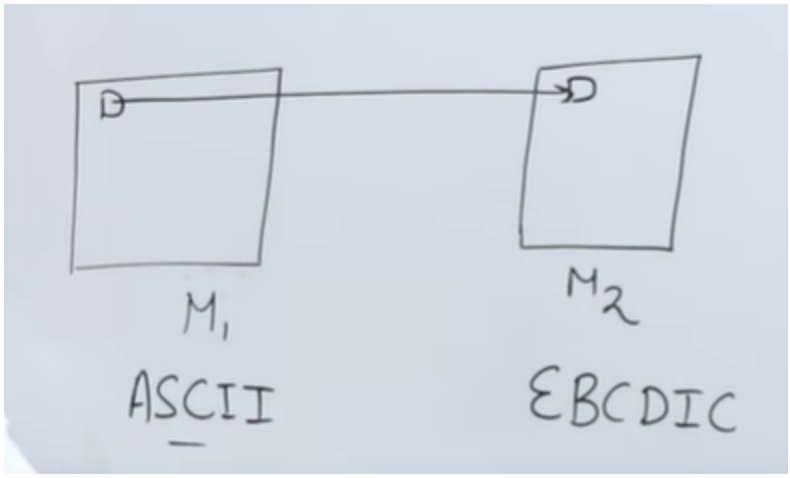
Functionalities :

synchronization(has some checkpoints to ensure the data transfer continues from where it had stopped recently), dialogue control(simplex, duplex)

1) Creates sessions : Whenever the machine wants to access a particular webpage, the host machine will send a request to the server having data related to that webpage. The server will then respond to this request by sending the related files to the host machine, ultimately setting up a session.
2) Authentication & Authorization : Once a session is set up, the session layer helps to provide authentication and authorization. Authentication is the process of verifying the user themselves by asking their username & password, etc. Authorization are the set of actions that the user is authorized to perform basically.
   For example if you're trying to login into the SBI webpage, authentication is done by asking for your account details, and authorization are the set of actions that the user is authorized to do, such as transfer money, etc.
3) Session restoration : If due to any circumstances if the current session encounters an issue, the session will be restored back. For example, if a session was active and then your laptop switched off, as soon as you turn the laptop back on, you can see an option for you to restore the previous session.

PRESENTATION LAYER

The presentation layer is the 6th layer from the bottom of the OSI Model.



Functionalities : encryption/decryption, translation, Compression

1) Code Conversion : From the above diagram, let's saying that an application of machine M1 which works on ASCII code is communicating with an application of machine M2 that is working on EBCDIC code. Now, if the application on machine M1 directly sends the data, the application on machine M2 will not be able to understand it. Hence, what happens is that the presentation layer present on machine M2 will convert the data from ASCII into EBCDIC format.
2) Encryption/Decryption : Packets that are being sent from one machine to another across the network will be vulnerable to attacks, threats, hacking, etc. Source machine can perform encryption and then the destination will perform decryption.
3) Compaction : Used to reduce the size of the data being transmitted b/w source and destination. The compression can either be lossy or lossless.

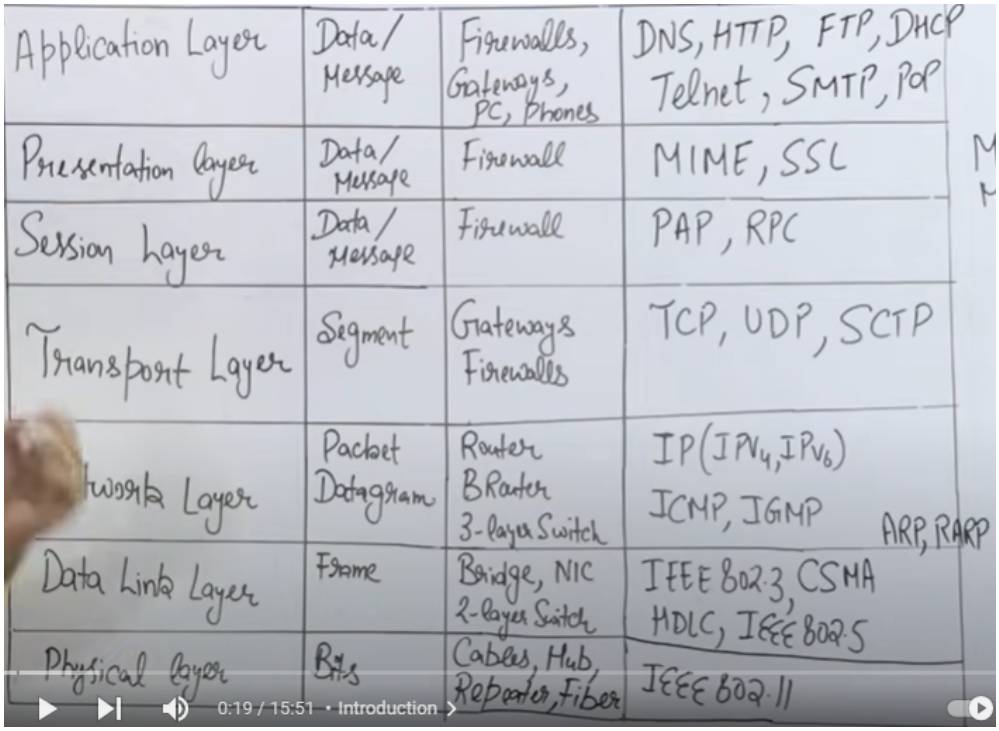
APPLICATION LAYER

This is the topmost layer of the OSI Model.
Basically, the users will be interacting with this application layer itself, through applications like mobile apps, web browsers, etc.

APPLICATION PROTOCOL	PORT NUMBER	TRANSPORT PROTOCOL
HTTP	80	TCP
HTTPS	443	TCP
FTP	20/21	TCP

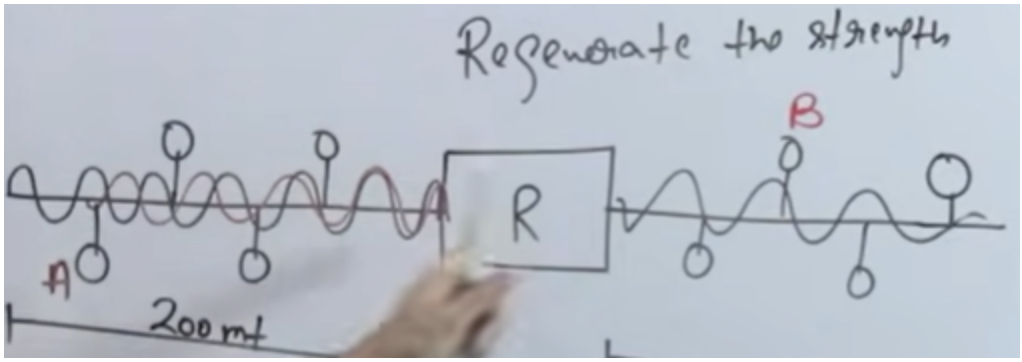
DNS	53	UDP
DHCP	67/68	UDP

OVERALL SUMMARY



NETWORKING DEVICES

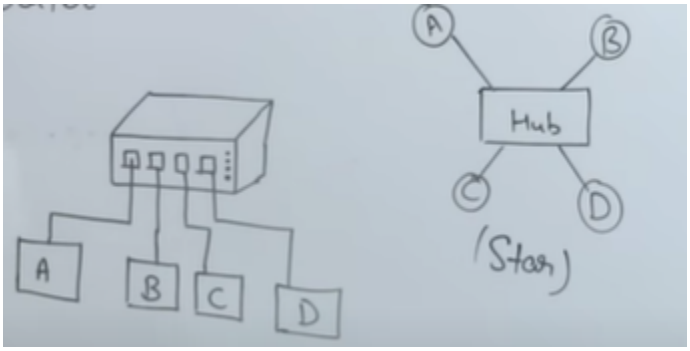
1) Repeater (i.e Belongs to the PHYSICAL LAYER)



- The role of a repeater is to regenerate the signal that it receives enabling us to send the signal over a longer distance across the network.
- As we can see in the diagram, a repeater is a 2-port device.
- The repeater does not perform filtering. Filtering means preventing the unnecessary data from traveling through the network.
  Usually what happens is that if a signal is being sent from A to B(i.e as seen in the diagram), the signal will go from A to the repeater, the signal gets regenerated, and then the signal goes to B.
  Now let's say A and B are on the same side of the repeater R. The signal will go from A to B, and after reaching B, the signal will reach the repeater, and the repeater still sends the signal to the other port even though there is no use of doing so.
- Collision Domain of repeaters is high since multiple devices can be sending signals to the repeater at the same time, and the repeater will directly send the signals to the other port without checking if the destination port is free or not. This occurs since the repeater does not have any buffer within it to hold the signals that it receives. If it had a buffer, the repeater could have sent the signals one by one to prevent collision
- NOTE : Difference between an amplifier and repeater : Let us consider a signal having intensity 'x1'. Now the signal travels for some distance and now the intensity becomes 'x2'. An amplifier can amplify the signal to an intensity greater than the original signal intensity itself (i.e > x1). However a repeater will always regenerate the signal and make the intensity go back to the original signal intensity itself. (i.e x2 back to x1 again).

2) Hub (i.e Belongs to the PHYSICAL LAYER)

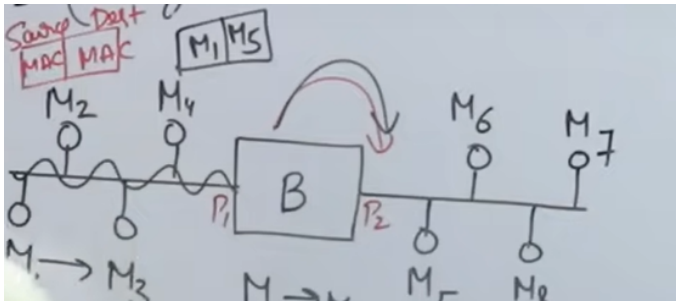
Similar to repeaters itself.



- Hubs are multiport repeaters. Multiple devices can be connected to the hub at the same time unlike the repeater which had only 2 ports.

- No Filtering : Take the above picture as an example. If device A wants to send data to device B, not only will the hub send the data from device A to device B but at the same time it will send it to device C and D too.
- Collision domain is high just like a repeater. If multiple devices send signals at the same time to the hub, again the signals will collide with each other.

3) Bridges (i.e Belongs to the DATA LINK LAYER)



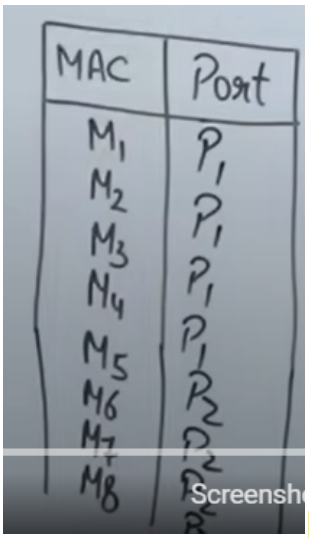
(i.e Diagram 1)

A bridge is a networking device which is used to connect 2 LANs(i.e Local Area Networks) together.
Let's assume that M1,M2,M3,M4 are connected to the bridge through port P1, and M5,M6,M7,M8 are connected through port P2.

- Bridges operate at the data link layer and hence they can check the MAC addresses of the devices that are connected.
- Forwarding : Bridges can forward the network traffic in an intelligent manner. Let us assume that device M1 is sending a packet to device M6. Within this packet, there will be 2 fields : Source MAC address and destination MAC address. When the packet reaches the bridge 'B' from M1, the bridge will examine the destination MAC address,

and will decide if it has to send the packet forward or not. In this case, since M6 is connected to port P2, it would forward the packet.
If the destination device was connected to P1 itself, then forwarding would not have been done.
For the bridge to decide if forwarding has to be done or not, it has to have a mechanism through which it knows which device is connected to which port. There are 2 types of bridges based on how it performs this task :

- Static bridges : Static bridges maintain a static table which maps the device to a port number. Whenever a packet arrives at the static bridge, the bridge will check the destination MAC address present in the packet and find this in the static table. From the static table, the bridge will get to know whether forwarding has to be done or not.



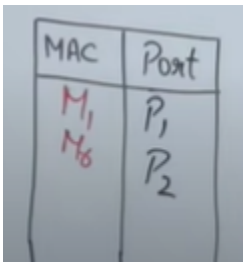
(i.e Static table for Diagram 1)

The problem with static bridges is that whenever a change occurs in the network, the static table has to be changed manually.

- Dynamic bridges : In case of dynamic bridges, the table is first empty. As packets start to arrive at the bridge, the bridge will start to build up the table.

Let's say first a packet is sent from M1 to M6. When the packet arrives at the bridge, the bridge sees the source MAC address present in the packet, and

assigns M1 to P1 in the table(i.e since the packet itself arrived from P1 side). At this stage, the bridge is still unsure whether M6 is on P1 or P2, so it will broadcast the packet.
After M6 receives the packet, it will send an acknowledgement back to M1 telling that it received the packet. The acknowledgement packet would now have the source MAC address as M6 and destination M1. This packet would now reach the bridge from P2. It's at this moment that the bridge will confirm that M6 is connected to P2, and will update the table again.
In this manner the table will dynamically be changing as packets are sent in the network.



- Filtering : Bridges, unlike repeaters and hubs, are capable of performing filtering. Using the table that bridges have, the bridge can decide whether the packet has to be sent further in the network or not.
- Collision Domain : We know that both repeaters and hubs have a high collision domain since they don't have any internal buffering. In case of bridges, they have a low collision domain since they are based on the "Store & Forward" mechanism. The bridge has internal buffering, using which the packets would be stored in the bridge, processed, and then sent out of the bridge.

4) Switches (i.e Belongs to the DATA LINK LAYER)

A switch is a networking device that **connects multiple devices within a local network**. (i.e Bridges are used to connect devices across 2 LAN's)

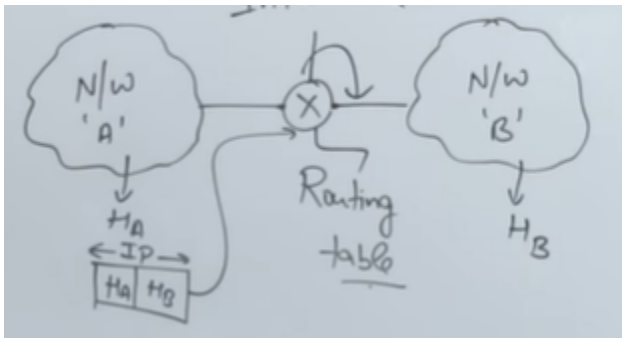
A switch receives incoming data packets or frames and forwards them to their intended destination within the network. Unlike a hub, which simply broadcasts incoming data to all connected devices, a switch intelligently examines the destination MAC (Media Access Control) address of each data frame and selectively forwards it to the appropriate destination device. (i.e Very similar to how the bridges operate)
The switch keeps a table, known as a MAC address table or forwarding table, which associates MAC addresses with specific switch ports. This table is dynamically updated as devices join or leave the network.

Hence, switches are known as "**Multiport Bridges**".

5) Routers (i.e Belongs to the NETWORK LAYER)

In the case of bridges, we know that a bridge connects 2 LANs.
In the case of a router, the router will connect 2 or more WAN's.

Routers belong to the network layer, and hence packets are basically routed based on the source and destination IP addresses. Whenever packets are being transmitted from one WAN to another WAN, we need to use IP addresses itself.



where X is the router.

Functionalities of a router :

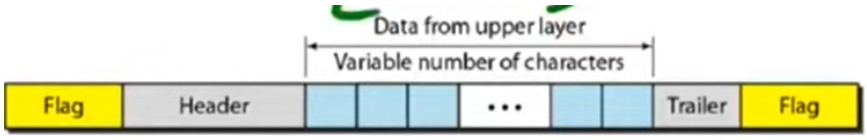
1) Forwarding : Based on the table that the router maintains, it will decide on which outgoing line should the packet be forwarded on(i.e based on the destination IP address).
2) Filtering : Based on the table maintained by the router, the router can prevent the packet from unnecessarily going to unwanted parts of the network.
3) Routing : Routing of the packets is done by the routers.
4) Flooding : If the router is not able to decide on which outgoing line the packet should be sent to, the router will send the packet on all the outgoing lines. This is known as flooding.
5) Collision : Routers follow the "Store & Forward" mechanism. They consist of an internal buffer which allows them to store the packet received, process it, and then send the packet on the desired outgoing line. Hence, the collision domain is very low.

DATA LINK LAYER FUNCTIONALITIES

1) **Framing :** We know that the data link layer performs framing to the data that it receives from the network layer.
The data link layer will divide the data into frames, so that each frame is distinguishable from each other.
While performing the framing, the data link layer can perform either "byte stuffing" or "bit stuffing" according to the need.

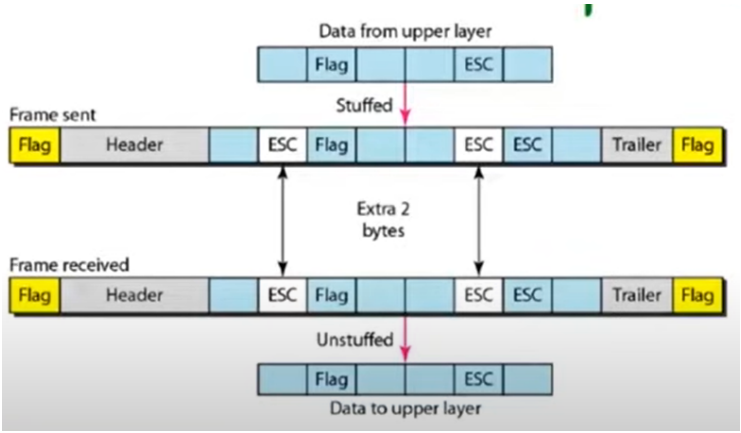
To mark the beginning and the end of the frame, the data link layer will add a flag at the beginning and ending. Whenever the frame is received on the receiver end, the receiver can easily understand that the information present b/w the 2 flags is the actual data.

However if the flag that is used to mark the beginning and ending of the frame is present within the data itself, then the receiver can make the mistake of thinking that the frame ends somewhere it actually doesn't. Hence we perform stuffing.



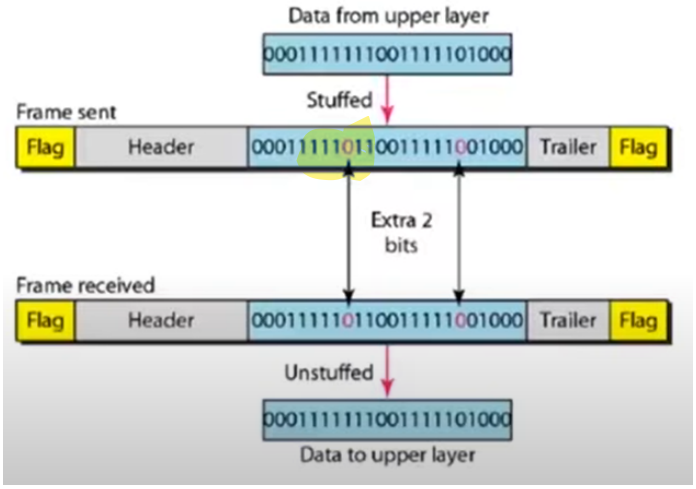
Byte Stuffing :

- Every instance of the "flag" in the data is stuffed with an ESC.
- Every instance of "ESC" in the data is stuffed with another ESC.

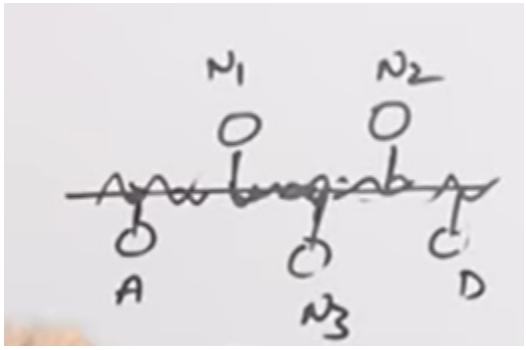


Bit Stuffing :

- Similar to byte stuffing, but only 1 bit is used for stuffing. In case of byte stuffing, we're stuffing an additional 8 bits.
- The flag usually used to denote the starting and ending of the frame is 01111110.. So hence, we stuff a "0" bit whenever we see the occurrence of 011111(i.e zero followed by five ones) in the data.



2) **Media Access Control :** One of the major responsibilities of the data link layer is Access Control. Whenever multiple devices are connected to a common carrier, access control plays an important role to determine which device is allowed to send data on the common carrier at a particular time.



The data link layer uses these algorithms to perform media access control :

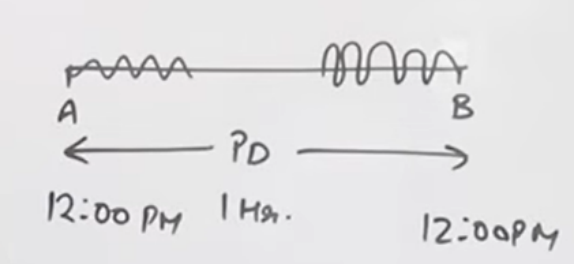
- **CSMA (i.e Carrier Sense Multiple Access) :** The device will first sense the carrier to see if the carrier is free or not after which the data is sent.
  There are 3 types :
    - **1-persistent :** The device will **continuously** sense the carrier. As soon as the device senses that the carrier is free, it will **immediately** send the data on the carrier.
      Problem with 1-persistent : **The chance of collision is very high.**
      For example , let's say that a signal is being transmitted from A to D. At this time, nodes N1,N2, and N3 will sense that the carrier is busy so they will not send their data. As soon as the signal has been transmitted to D, the nodes N1,N2, and N3 will sense that

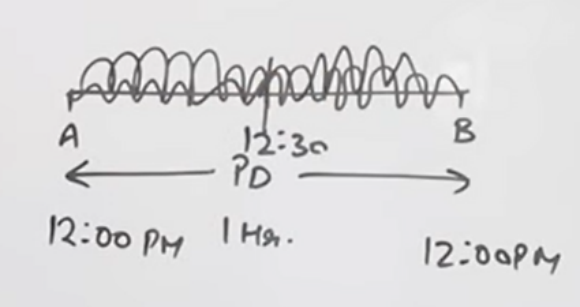
the carrier is free again. All three of them will send the signal at the same time **causing a collision**.

- **0-persistent :** If the carrier is free, then the node will immediately send the data on the carrier. If the node senses that the carrier is not free, then the node will **wait for a random amount of time** after which it will sense again. (i.e in 1-persistent, it was repeatedly sensing).
  The probability of collision occurring is lesser compared to 1-persistent.

- **p-persistent :** If the carrier is free, then the node will immediately send the data on the carrier. If the node senses that the carrier is not free, then **depending on the p-value**, the data will be sent. Basically 'p' is the probability value which determines if the data is sent or not by the node when the carrier is free.
  P-persistent is a combination of both 0-persistent and 1-persistent.

- **CSMA/CD (i.e Carrier Sense Multiple Access with Collision Detection) :** CSMA/CD is a modification of CSMA. Using CSMA/CD, we can get to know if the signal sent by a particular node has collided with another signal or not.
  Why can't we use acknowledgement for this : Collision is already taking place in the network. If we use acknowledgement messages too, then the network will get even more congested.
  - When 2 signals sent by 2 different nodes collide with each other, the resultant signal will have a frequency that doesn't match with any node(i.e known as the collision signal).
    - If the collision signal arrives at the source node while the source node is sending data, then the source node will know that it's signal has gotten collided.
    - If the collision signal arrives after the source node finished sending data, then the source node will not know if its own signal got collided or the signal of some other station got collided.





- Take the following example : Let's say A senses the carrier at 12:00 PM. It sees that the carrier is free so it starts sending the signal towards B. Even B senses the carrier at 12:00 PM, and since a node can only sense the carrier wherever it's connected, it will think the carrier is free and will start to send its signal towards A.
  At 12:30 PM, both the signals collide with each other, forming a collision signal of some random frequency. This collision signal will then propagate towards both A as well as B.
  At 1:00 PM, the collision signal will arrive at A & B. If A & B are still sending data at 1:00 PM, then A & B will understand that their signal got collided.
  If they stopped sending data before 1:00 PM, then they won't be sure if their signal actually got collided or not.

  Hence, **Transmission Time >= Propagation Delay**.
  In worst case, Transmission Time >= 2 * Propagation Delay

- **CSMA/CA (i.e Carrier Sense Multiple Access with Collision Avoidance) :**

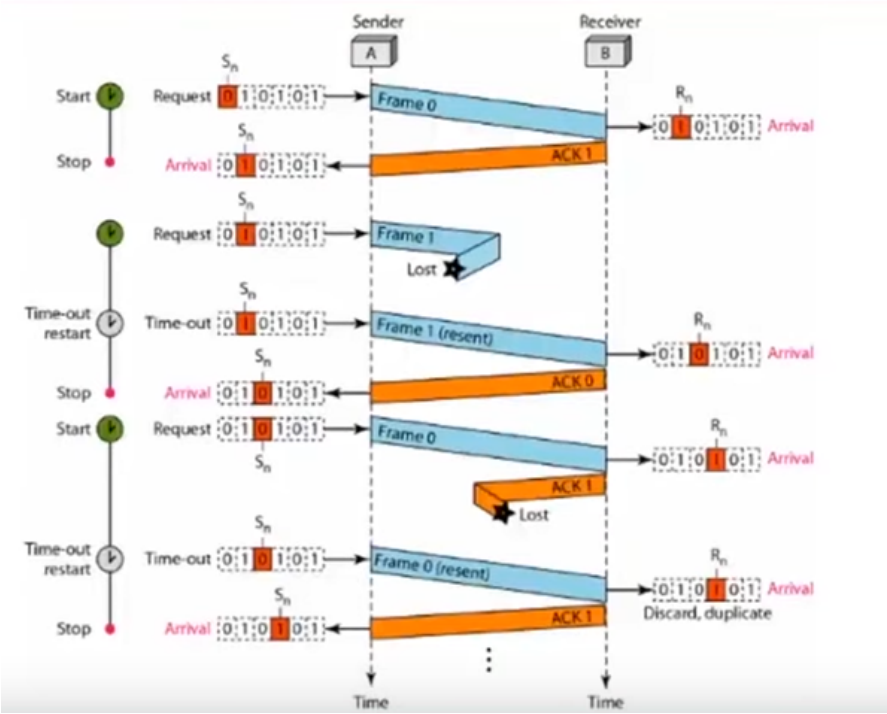
3) **Error Control :**

- **Stop and Wait ARQ(i.e Automatic Repeat Request) :** It's a simple error control protocol used in the data link layer.
  In Stop and Wait ARQ, the window size of both the sender & receiver is 1.

Hence, we use the bits 0 and 1 to indicate which frame that we are sending or which frame is expected to be received next(i.e in case of ACK).

The sender will send a particular frame, after which the sender will wait for a fixed period of time for an ack from the receiver. If the sender does not receive the ack within a period of time, then it will assume that the frame it sent got lost and will retransmit the frame again.

The protocol is known as "Stop and Wait" since the sender will stop sending frames and will wait to receive an ack, only after which the next frame will be sent.



First sender A will send Frame 0. Receiver B will receive Frame 0, change its bit to 1, and then will send an ACK message of 1 to the sender(i.e this indicates to the sender that Frame 1 has to be sent).
Now suppose the sender sends Frame 1 and it gets lost. After waiting for a certain period of time called the turnaround time, it still wouldn't have received an ack from the receiver, indicating to the sender that its frame got lost. Hence, the sender would again retransmit the frame.

Now let's suppose that a frame was received by the sender but the ack got lost. In this case, since the sender didn't get the ack from the receiver, it will retransmit the same frame again. However, the sender would have changed its bit & it knows that it had received the same frame previously. Hence, the sender will discard the frame.
(i.e Frame 0 was sent to the receiver and the receiver changed its bit to 1. The ACK 1 got lost, and the sender again sent the same frame 0. Frame 0 however got discarded by the receiver since it was expecting Frame 1(i.e the bit that the receiver has indicates what frame it's expecting).

- Go-Back-N ARQ : https://www.youtube.com/watch?v=QD3oCelHJ20
- Selective Repeat ARQ : https://www.youtube.com/watch?v=WfIhQ3o2xow

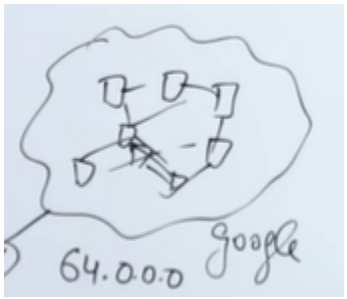
NETWORK LAYER FUNCTIONALITIES

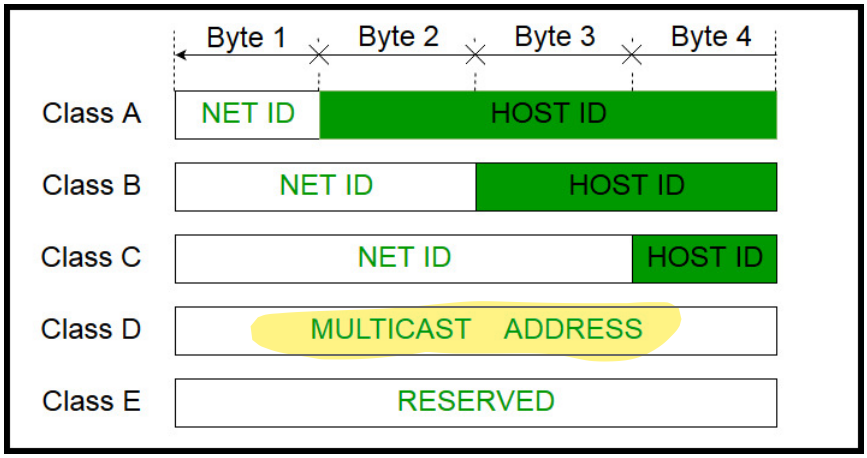
We know that the network layer is associated with IP addresses. The network layer is responsible for the source to destination delivery of the data, and for that we require IP addresses(i.e in case the source and destination are in 2 different networks).

Classful Addressing

CLASS	LEADING BITS	NET ID BITS	HOST ID BITS	NO. OF NETWORKS	ADDRESSES PER NETWORK	START ADDRESS	END ADDRESS
CLASS A	0	8	24	2^7 (128)	2^{24} (16,777,216)	0.0.0.0	127.255.255.255
CLASS B	10	16	16	2^{14} (16,384)	2^{16} (65,536)	128.0.0.0	191.255.255.255
CLASS C	110	24	8	2^{21} (2,097,152)	2^8 (256)	192.0.0.0	223.255.255.255
CLASS D	1110	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	224.0.0.0	239.255.255.255
CLASS E	1111	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	240.0.0.0	255.255.255.255

By looking at the leading bits of the ip address, we can get to know which class the IP address belongs to.
In case of class A : We can never use the address “network_id.0.0.0” for a particular host. The “network_id.0.0.0” ip address is used to represent the entire network itself. We can allocate the addresses from “network_id.0.0.1” to “network_id.255.255.254”. Again, the last address “network_id.255.255.255” cannot be used for a host, since it represents the **broadcast address**. This similar concept is also applicable for class B and class C too.



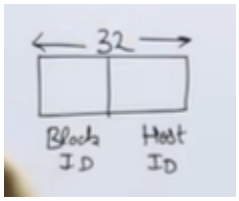


The problem with classful addressing is that there is an **extensive wastage of IP addresses**. For example, let's say you have a network with 1024 hosts. Now we need to decide which class of IP address should be used for this network. A class B address would support upto 65,535 hosts(i.e wasting 65535 - 1024 addresses) and class C address supports only 256 hosts. Hence, a large amount of wastage of IP addresses occurs.

Classless Addressing

To combat the problem with classful address, classless addressing came into existence.

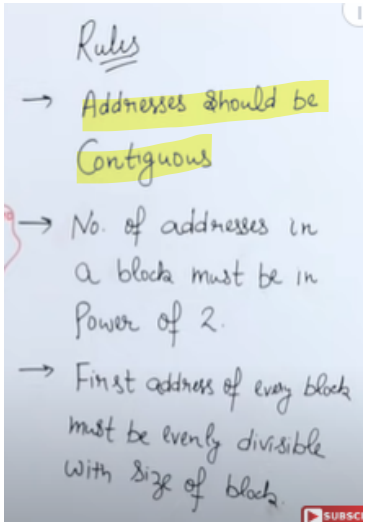
In classless addressing, every IP address is composed of 2 parts : Block id & Host id



Unlike classful addressing where we know exactly the number of bits that are used for representing the network id and the number of bits for host id, we actually can't tell directly in classless addressing.
Hence, every IP address is represented as,

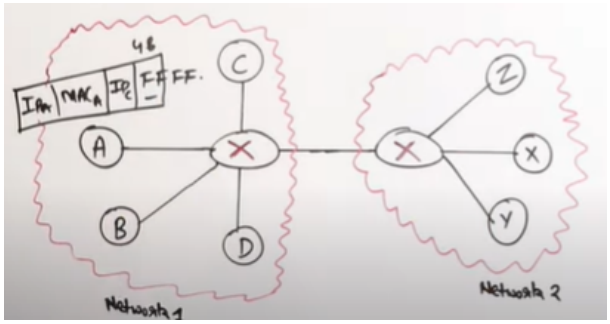
w.x.y.z / n

where 'n' denotes the number of bits used to represent the block/network id.



PROTOCOLS USED IN ~~DATA LINK LAYER~~
NETWORK LAYER ITSELF

1) **ARP(i.e Address Resolution Protocol) :** This protocol belongs to the ~~data link layer~~ and is responsible for finding the MAC address of a device given the IP address. (i.e converts IP address into MAC address)



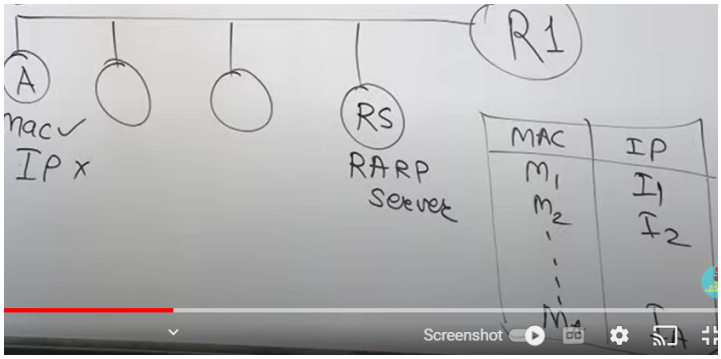
Case 1(i.e Intranetwork communication) : Let's say Node A wants to communicate with Node C. Now Node A will have access to its own IP and MAC address, but it will know only the IP address of Node C(i.e since IP addresses are public and MAC are usually private). The network layer is responsible for the source to destination delivery, however this delivery occurs through multiple hop to hop deliveries done by the data link layer. Hence, for the hop to hop deliveries we require the MAC address of the devices.
For obtaining the MAC address of Node C, Node A will create a packet(i.e consisting of IP address of A, MAC address of A, IP address of C, and the field for MAC address of C will be filled with **FFFF** indicating broadcast address). Since we provided a broadcast address, the **packet will be broadcasted** from A to all the other nodes. When the packet arrives at C, C will see that the destination IP address is its own, and hence it will replace the broadcast address

with its MAC address, and give a **unicast** reply to Node A. After this, Node A can now easily transfer packets to Node C since now Node A has the MAC address of Node C.

Case 2(i.e Internetwork communication) : Let's say Node A want's to communicate with Node Z. First Node A has to determine the MAC address of the first router. Then after it finds the MAC address of the first router, it needs to find the MAC address of the second router, and ultimately the MAC address of Z.
Basically at every hop, you'll send the packet with a broadcast address, then you'll receive a unicast reply indicating the MAC address, and then the packet will be sent to the next hop.

2) **RARP(i.e Reverse Address Resolution Protocol) :** This protocol belongs to the data link layer and is responsible for finding the IP address of a device given the MAC address. (i.e converts MAC address into IP address)

   Use case of RARP : Let's say that a device has been newly introduced into the network. At this point of time, the device knows its MAC address, but does not know its IP address. To figure out the IP address, it uses the RARP protocol.
   The newly added device will **broadcast a packet** consisting of its MAC address to all the other nodes in the network. A special node in the network called the **RARP Server**, will have a static table that maps the MAC address of all the devices connected in the network to their respective IP addresses. When the packet arrives at RARP Server, the server will check this table to find the IP address of the newly added device, and will send the **IP address as a unicast reply** back to the device.

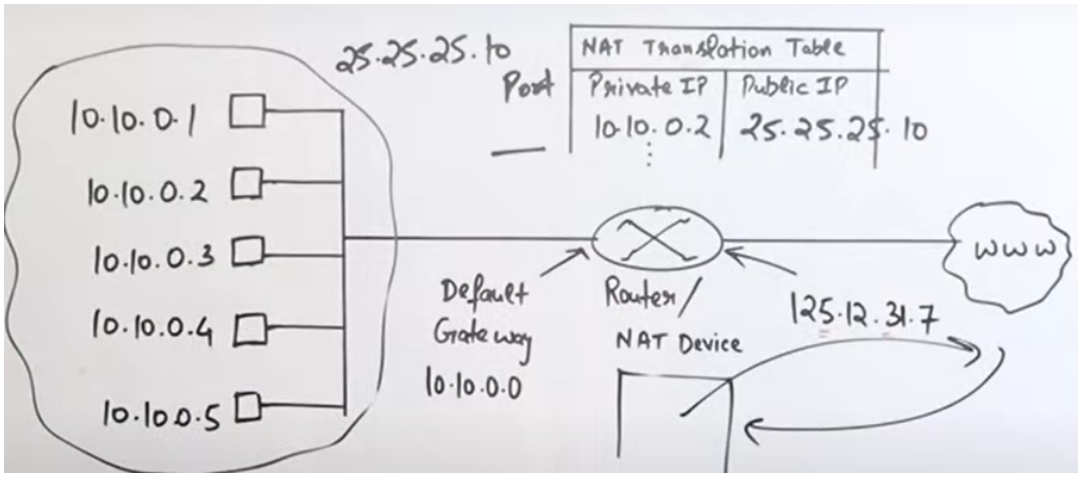


PROTOCOLS USED IN NETWORK LAYER

There are various protocols in the network layer including IP protocol, ICMP, & the various routing protocols(i.e RIP,OSPF,BGP).

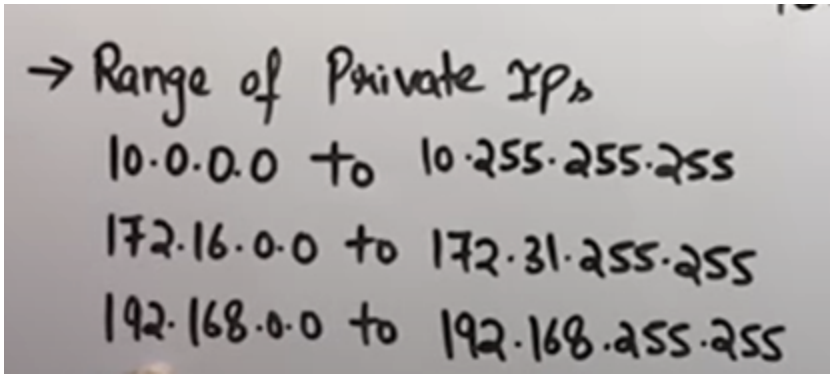
1) **NAT(i.e Network Address Translation) :** The job of NAT is to map private IP addresses to public IP addresses and vice versa.
   As time is passing by, the number of devices connected to the internet are growing rapidly and it's not possible to give a public IP address for each device. Hence, what happens is that within organizations, the organization makes a private network where all the **hosts within that network will have private addresses.**

This network to the outside world will be known by a single public IP address. By doing this, we will be able to handle more devices.



Let us consider an organization that has formed a private network. All the hosts have been given private addresses(i.e 10.10.0.1, 10.10.0.2 all the way up to 10.10.0.5). This private network to the outside world is known by 125.12.31.7.
Let's assume that host 10.10.0.2 is sending a request to the server having IP address 25.25.25.10. In the packet being sent by 10.10.0.2 the source address would be 10.10.0.2 and destination address is 25.25.25.10. Now the outside world doesn't know what 10.10.0.2 means.
Hence the NAT will replace the source address from 10.10.0.2 to 125.12.31.7(i.e public address that the private network is known by), and then the packet is sent to the destination. While doing this, the NAT will maintain a table mapping the private IP address of the host to the public destination address.
When the server sends back a reply, that packet would have the source address as the public address of the server, and the destination address as the public address of the private network. However, we need to send the packet to the desired host within the private network. Hence, the NAT will then replace the destination address of the packet from the public address of the private network with the private address of the host within the private network by using the NAT table.

Ranges of private IP addresses are :



3) **ICMP(i.e Internet Control Message Protocol) :** This protocol belongs to the network layer and works in companion with the IP protocol.
The IP protocol does not provide any error reporting and error correcting mechanisms. The ICMP helps provide these to the IP protocol.

Functionalities of ICMP :

- Error reporting : Whenever data is being transmitted between source and destination and some error occurs, the ICMP protocol will report the error back to the source device. Some examples of errors are **Destination Unreachable, Time Exceeded, Source Quench**, etc.
- N/w diagnostics : Using ICMP we will be able to get various N/w diagnostics like the **traceroute**(i.e routing path b/w 2 devices) as well as **ping**(i.e the speed of the connection b/w 2 devices). responsiveness and existence of the destination device..

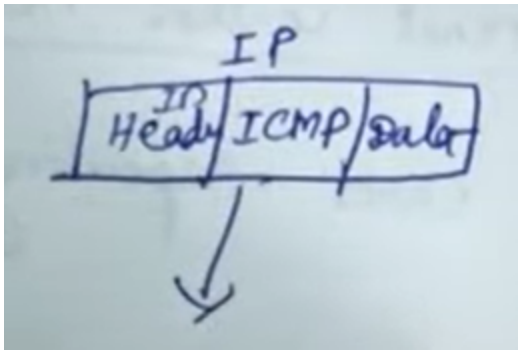
ICMP can send 2 types of messages :

- Error reporting messages : Reports problems that the host encounters while they are transferring the packets
- Query messages : Helps a host/router to get specific information about another host

We know that ICMP works in companion with IP protocol. So in the network layer, the data is first encapsulated by the ICMP after which the IP header is added. This entire packet is then passed down to the data link layer.

4) IP PROTOCOL:

IP uses the receiver's IP address to determine the best path for the proper delivery of packets to the destination. When a packet is too large to send over a network medium, the sender host's IP splits it up into smaller fragments. The fragments are reassembled into the original packet on the receiving host. IP is unreliable since it does not ensure delivery or check for errors.



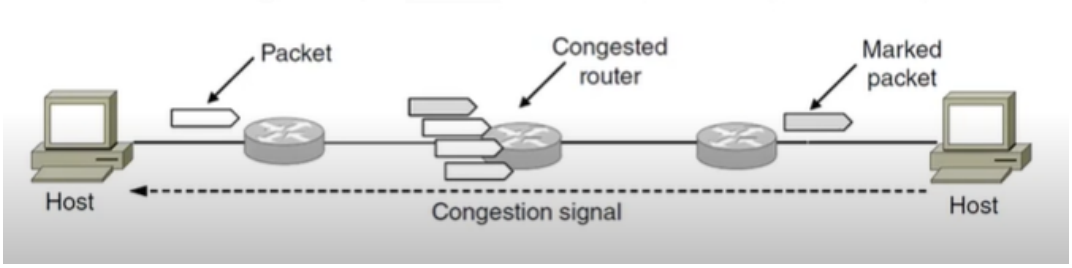
CONGESTION CONTROL IN NETWORK LAYER

1) **RED(i.e Random Early Detection) :** RED stands for Random Early Detection, which is an active queue management (AQM) algorithm used in computer networks. It is typically implemented at network routers to help control network congestion by managing the queue length of incoming packets. The main goal of RED is to provide early indications of network congestion before the packet queue becomes excessively full. It uses a probabilistic dropping mechanism to selectively drop or mark packets based on the length of the queue, allowing the sender to reduce its transmission rate in response to congestion indications.
   Basically, in short, if the length of the queue becomes greater than the threshold length of the queue at a router, the router will drop the excess packets, and an alert is sent to the sender to decrease the rate at which they are sending the packets across the network.

2) **ECN(i.e Explicit Congestion Notification) :**

   In the case of RED, there was a direct packet drop that occurred when the queue length of the router exceeded the threshold.
   In the case of ECN, there isn't actually a packet drop occurring. When the packet reaches a router which is on the onset of congestion, the packet will get marked(i.e a bit will get set in the IP header) by that particular router, and will ultimately reach the destination node.
   When the packet reaches the destination node, it will see that the packet is marked, and will send a congestion signal back to the source node, telling the source node to reduce the rate at which the packets are being sent on the network.

   The advantage of ECN is that :
   - Congestion is detected early, and no packet loss occurs.
   - Routers are able to deliver clear signals to hosts telling them to slow down.



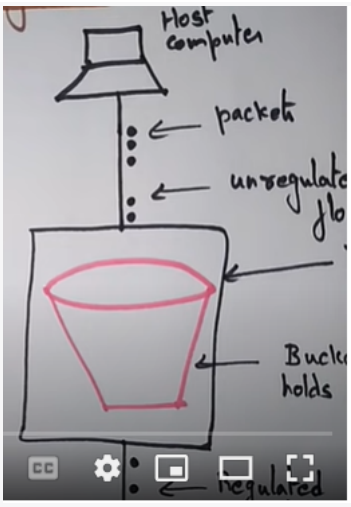
3) **Traffic Shaping :**
Traffic shaping regulates the flow of network traffic by controlling the rate at which packets are transmitted. It smooths out bursty traffic by buffering packets and releasing them at a controlled rate, preventing congestion caused by sudden spikes in data transmission.

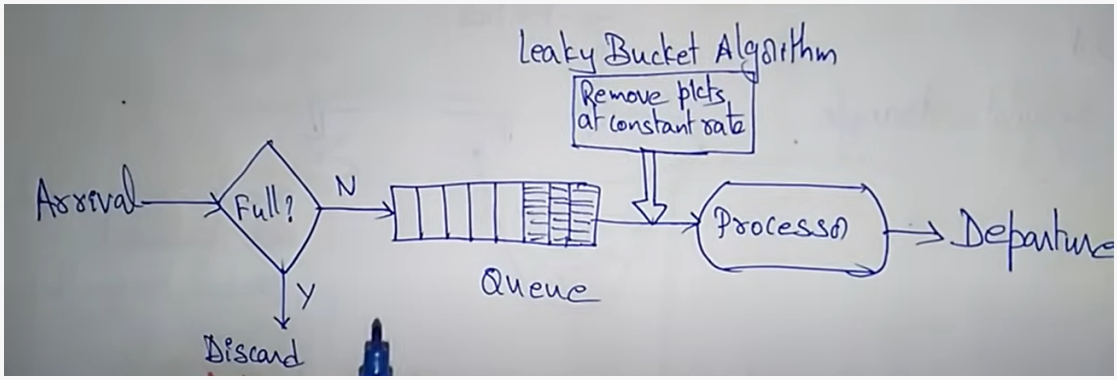
There are 2 methods :

- **Leaky Bucket :** Imagine a bucket with a small hole at the bottom. No matter at what rate the water enters into the bucket, the rate of outflow of water will be constant when there is water in the bucket, and zero when the bucket is empty. Also when the bucket is full, any additional water entering into the bucket will spill over the sides and will be lost.

  Now we can use the concept of this leaky bucket in computer networks.
  Every network interface will consist of this leaky bucket. The rate at which the data packets arrive at the leaky bucket can be variable, however there is a constant rate of outflow of packets into the network.

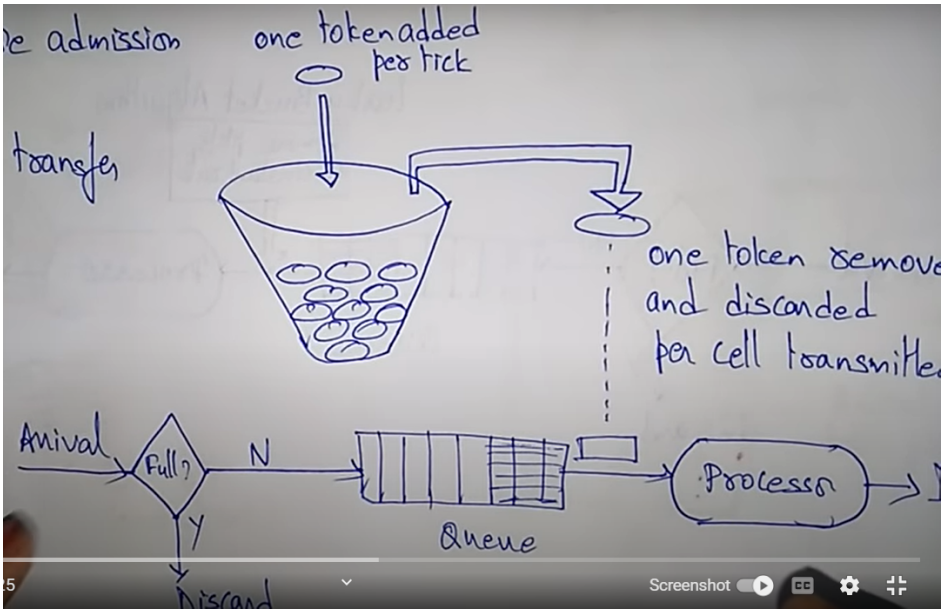
  Since we're able to maintain a fixed & constant rate of outflow of packets, the problem of congestion can be solved.



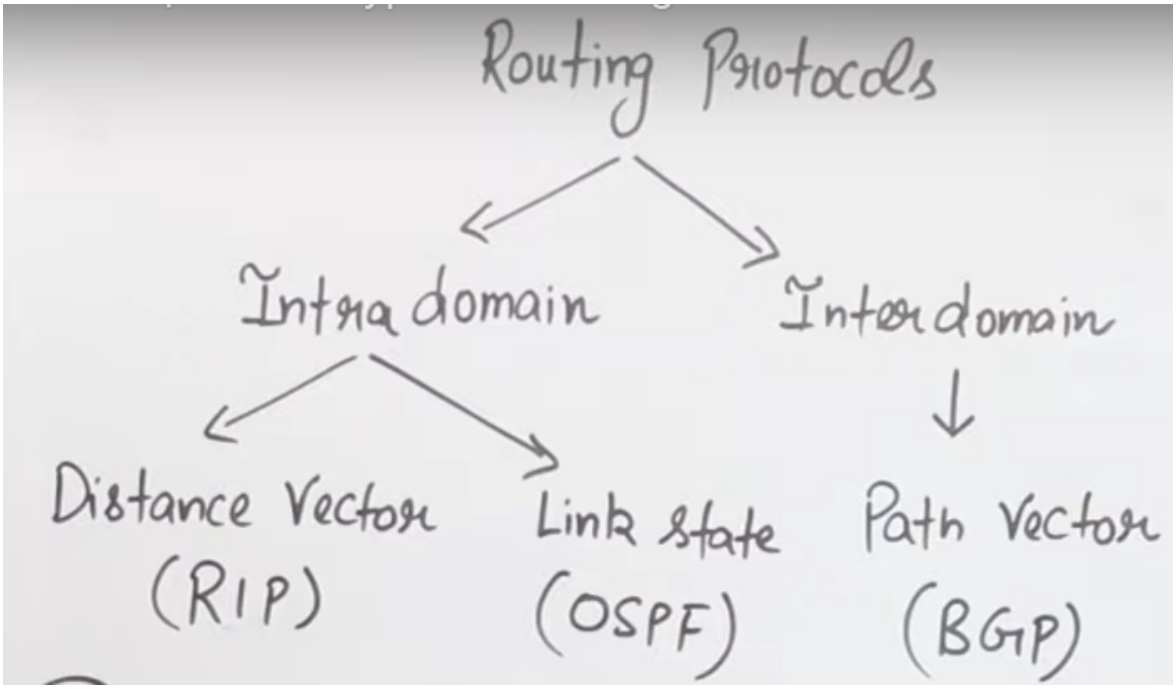


- **Token Bucket :** Kinda similar to leaky bucket, but here within the bucket, there are tokens. For a particular packet to be transmitted in the network, a **token** has to be available & taken from the bucket. Tokens will be added to the bucket at regular intervals of time.
  If there are no tokens available in the bucket, then the packet cannot be transmitted at that moment of time.

  The major difference b/w Leaky & Token Bucket : In case of Leaky Bucket, the outflow rate is constant, whereas in Token Bucket we can change the outflow rate given the availability of tokens in the bucket. Hence, by using Token Bucket, we'll be able to send larger bursts at a faster rate.

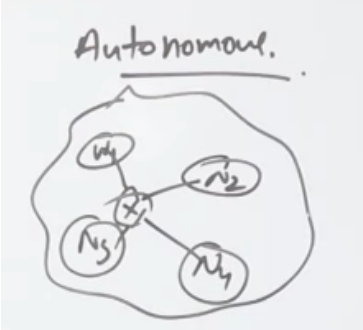


ROUTING PROTOCOLS IN NETWORK LAYER



The internet itself is divided into several Autonomous Systems.

An Autonomous System (AS) is a collection of connected IP networks that operate under a single administrative domain, with a common routing policy.
For example, all the small networks in Delhi will together form a single autonomous system.



Intradomain routing is the routing done within an autonomous system, and interdomain is the routing done between 2 autonomous systems.

Routing can also be classified into 2 categories :

- **Static Routing :** Each router will have a static/fixed routing table based on which the data packets will be forwarded. It's the responsibility of the network administrator to update the routing table manually whenever a change has to be made.
- **Dynamic Routing :** Each router will have a dynamic routing table. Routers will share their routing tables with each other, and will update their own routing tables as it's needed.

The advantage of dynamic routing is that whenever a new device is connected to the network, or in other situations, the tables get updated automatically.

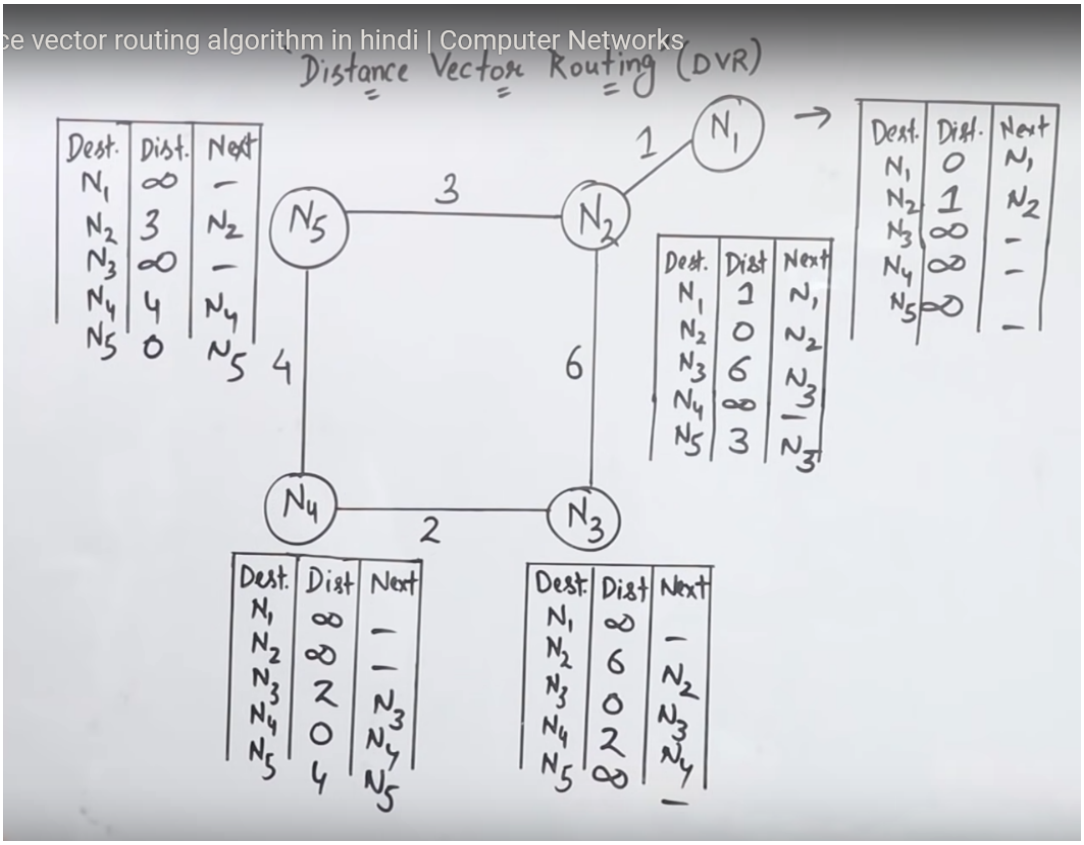
Distance Vector Routing :

The distance vector routing algorithm is a dynamic routing algorithm, and works using the RIP protocol (i.e Routing Information Protocol).

Every router in the network knows how many total number of routers are present within the network. Each router will understand who its neighbors are by sending HELLO messages.

Every router will maintain its own routing table, using which the router will decide where the packet has to be routed.

Step 1 : Every router will create its local routing table. At this point of time, each router only knows the cost to go to its neighbors. They don't know how to go to another router which is not a neighbor, hence it assumes the cost to go to that router to be INFINITE.



Step 2 : Now the routers will start to share their routing tables with their neighbors.

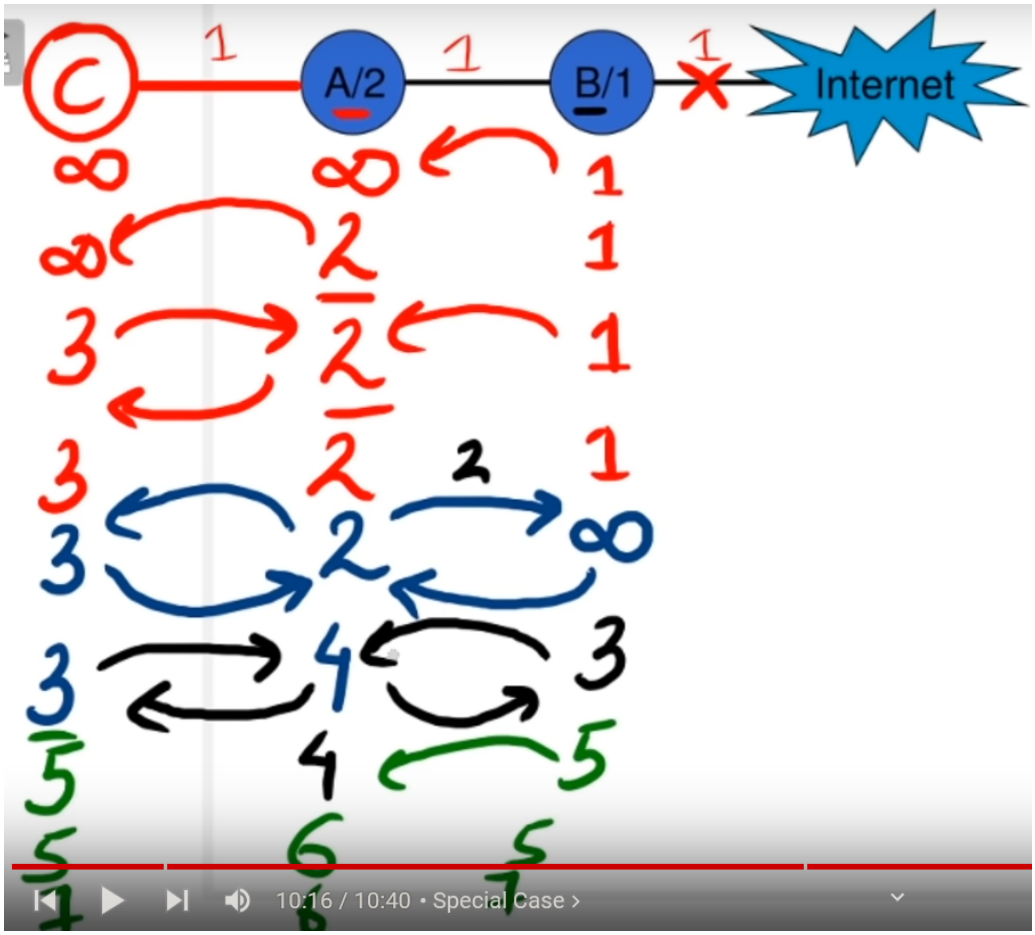
- The entire routing table isn't shared with the neighbors. ONLY THE DISTANCE VECTOR WILL BE SHARED.

Example : N3 will share its distance vector with N2 and N4, and N3 will receive distance vectors from N2 and N4.

Step 3 : These routing tables will be shared with each other after every few seconds. Even if we've updated all the entries of all the routing tables, the routing tables will still be shared. This is because we never know when a link may break within the network, etc and to be able to send information of such situations to the other devices, it's required we share it at regular intervals.

Problem with Distance Vector Routing :

Count to Infinity Problem



NORMAL CASE :
Initially every router will only know about its neighbors, so initially B knows it can go to the internet with cost 1 but B and C don't know.

Once B shares its table with A, A will realize it can go to the internet with a cost of 2(i.e 1+1), and in the next iteration C will get to know (2 + 1 = 3).

SPECIAL CASE :
Now assume that the link between B and the internet broke. Now B immediately knows that it cannot go to the internet anymore, so it will update the distance to INFINITE.
Parallely A will share its table to B and C, and similarly B & C. So B will update as 2 + 1 = 3(i.e through A), and C will not update since 2 + 1 = 3(i.e through A). Then A will update itself as 3 + 1 = 4(i.e through C), since now distance to reach each of the neighbors from A is 1 + distance to go from neighbor to internet is 3. (i.e 1 + 3 = 4).

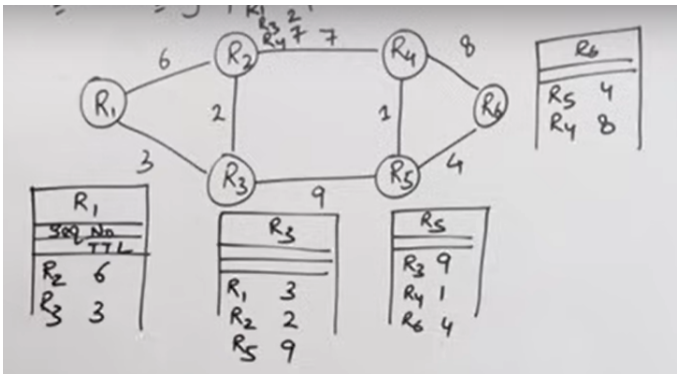
The entries will keep getting updated and ultimately all the values will become INFINITE.

The reason why this problem occurs is because in Distance Vector Routing, every router only shares the distance vector, and does not specify details regarding through which router are we able to reach the destination.
If this additional information was provided, then the count to infinity problem wouldn't have occurred.

Link State Routing :
The link state routing algorithm is a dynamic routing algorithm, and works using the OSPF protocol (i.e Open Shortest Path First).

Every router in the network knows how many total number of routers are present within the network. Each router will understand who its neighbors are by sending HELLO messages.

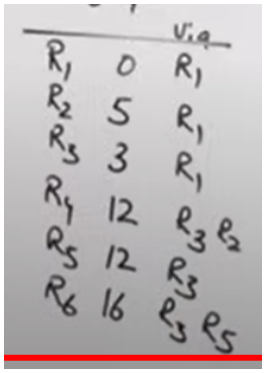
Step 1 : Every router will first create its own link state table. The link state table will consist of the distances between that router and its neighbors, along with additional information, such as Sequence Number & TTL.



Step 2 : The link state table is now flooded on the network. This means that the link state table of one router is shared to all the other routers on the network(i.e in case of distance vector routing, the distance vector was shared only to the neighbors).

Step 3 : Since the link state tables are being flooded, every router will have complete information about the network. Now, every router will use the Djikstra's algorithm to find the shortest path between itself and all the other routers.

Basically, Djikstra's algorithm is applied once for every router in the network.



Routing table for router R1

PROTOCOLS USED IN TRANSPORT LAYER

The transport layer is responsible for end-to-end delivery(i.e port-to-port delivery). There are 2 ways in which this can be achieved : Connection-oriented manner OR Connection-less manner.

TCP(i.e Transmission Control Protocol) :

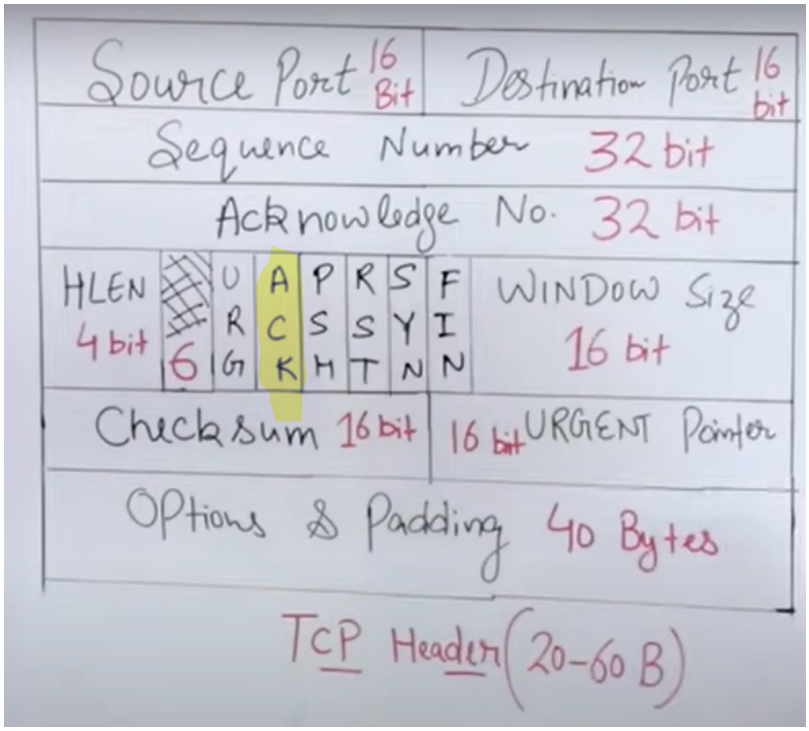
- The TCP is a connection oriented protocol. Before the data can be transmitted from source to destination, a connection first has to be established b/w them.
- **Highly reliable :** TCP has 2 features which allows the transfer of data to be reliable.
    - Acknowledgement : Once the destination receives the data, the destination sends an acknowledgement back to the source telling that it received the data.
    - Automatic retransmission : In case the data was not able to be transferred properly to the destination, the TCP will make the source to retransmit the packet once more.
- **Byte Streaming Protocol** : The data that the transport layer receives from the higher layers is a continuous stream of bits. The transport layer first groups the bits into bytes, and then performs segmentation. **Every segment can be defined as a group of bytes, and the segment will have either the TCP/UDP header.**
- **Full Duplex :** Once the connection b/w source and destination has been established, the source can send data to the destination and vice versa.
- **Piggybacking :** Piggybacking is the concept where along with the data, the acknowledgement will also be sent in the same packet. (i.e instead of sending a separate packet for the data and a separate one for the acknowledgement). This helps reduce the load on the network.
- **Flow Control**
- **Error Control :** In case of the data link layer, we're using CRC for performing error control but in case of TCP, checksum is used.

- **Congestion Control :** The network layer also performs congestion control. However, when we talk about congestion control in the transport layer, we consider **both the capacity of both the destination as well as the network links**.(i.e network layer will only consider the capacity of the destination).

TCP: Retransmits specific segments upon request or in response to a timeout, ensuring reliable delivery.

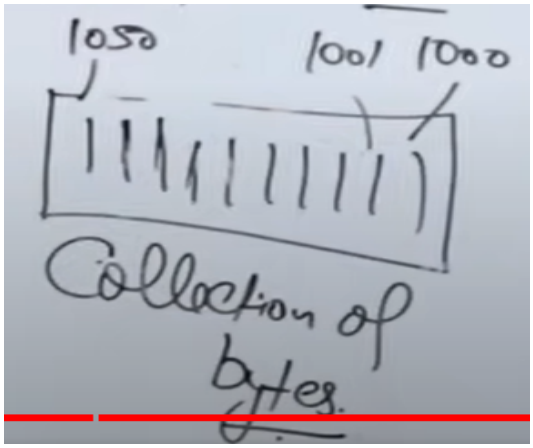
TCP Header :

The TCP Header can be anywhere b/w 20 to 60 bytes in size.



TCP helps in end-to-end delivery so that's why port numbers are mentioned in the TCP Header.

1) Source port : The port number of the source machine.
2) Destination port : The port number of the destination machine.
3) Sequence number : Every byte within a sequence will be numbered. The starting byte number within the sequence being sent will be the sequence number in the header.



4) Acknowledgement number : The acknowledgement number usually specifies from what sequence number the data should be sent from next. For example, if machine A sent a packet with sequence number 70 to machine B, then machine B will send an acknowledgement packet having the acknowledgment number as 71.
5) HLEN : Represents the length of the header. (i.e Multiply HLEN value into 4 to get the size of the header in bytes). (i.e HLEN is 4 bits which makes the range from 0-15, and we know that the header size can be anywhere between 20 to 60 bytes).
6) URG : Urgent Flag
7) ACK : Whenever we send a packet as acknowledgement, we make the ACK = 1.
8) PSH : Usually, what TCP does is that, a certain amount of data will be accumulated at the source after which the data will be transmitted to the destination. If we want some data to be sent immediately to the destination, we make PSH = 1.
9) RST : For resetting the connection b/w source and destination
10) SYN : When we are establishing a connection using TCP, the SYN flag would be set(i.e SYN = 1). It stands for Synchronize flag.
11) FIN : To terminate a connection, we make FIN = 1.
12) Window Size : Refers to the maximum amount of data that the destination can receive. Whenever a TCP connection is established, the source & destination reserve resources such as buffer space, etc. Based on the amount of buffer space reserved, the window size will vary.
13) Checksum : Used so that error control can be done.
14) Urgent Pointer : In case there is some urgent data & URG = 1, then the urgent pointer will be pointing to the location where the urgent data actually is.

15) Options & Padding : Contains many other stuff but the most important is MSS.

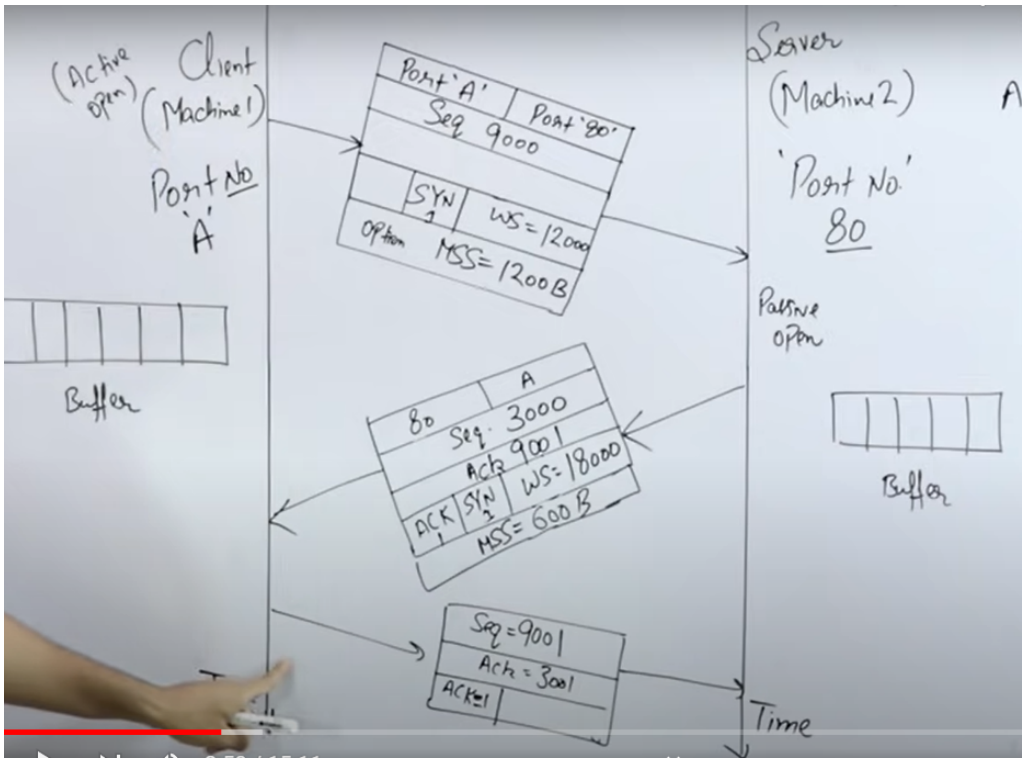
- MSS(i.e Maximum Segment Size) : The maximum size a segment can be is determined by MSS. Usually networks won't be able to transport segments greater than a particular size. For example, if ethernet has been used in the data link layer, then the MSS is 1200 bytes.
  Difference b/w Window Size & MSS :
  Let's say 2 machines A & B established a connection, and B has told that its window size is 10000 bytes, and MSS is 1000 bytes. This means that at max, A can send 10

segments of 1000 bytes size to B. (i.e A can't directly send one segment of size 10000 bytes).

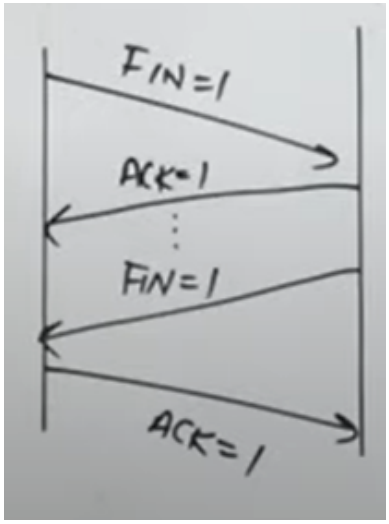
Connection Establishment through 3 way handshake :

In general,

Example :
Assuming that : Port A is the port number of Machine 1, and Port 80 is the port number of Machine 2.
Initially Machine 1 will randomly generate a sequence number, which is 9000 in this case, and Machine 2 does the same, which is 3000 in this case.
WS stands for Window Size, and MSS stands for Maximum Segment Size.



Connection termination :



(i.e 4 way handshake)

As we know, when a TCP connection is set up, both the source and destination machines will reserve a set of resources including buffer space, etc.

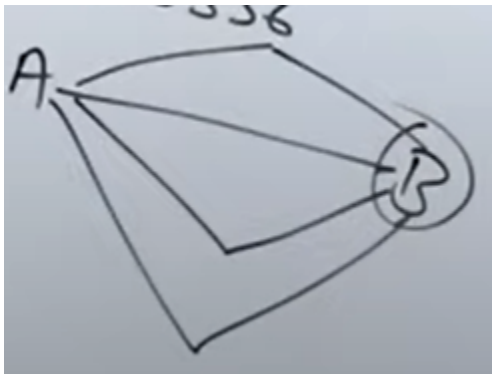
When machine A sends a FIN segment to machine B, that indicates that machine A has no more data to be sent to machine B. Hence, machine B will **release** all the resources that it had reserved during the TCP connection establishment. **Now machine A can no longer send data to machine B, however it is still allowed to send acknowledgement to machine B.**

Connection termination can also be done with 3-way handshake by combining the ACK and FIN sent by machine B. However this can only be done if machine B also decides to immediately close the connection after receiving a FIN from machine A.
If machine B still has data to be sent after machine A sent the FIN, then 4 way handshake itself would occur.

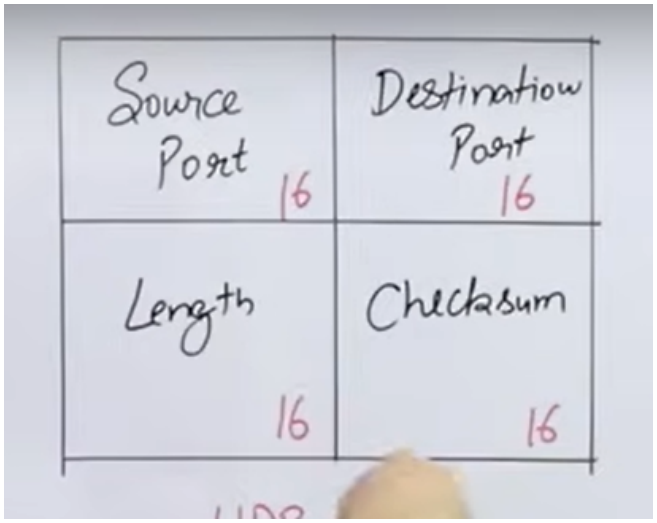
(i.e 3 way handshake)

UDP(i.e User Datagram Protocol)

- Provides a connectionless service. The data will be transferred b/w the source & destination without forming a connection first.
- Less reliable compared to TCP, due to 2 reasons :
    - There is no concept of acknowledgments in the case of UDP.
    - There is no automatic retransmission of packets in case some packets get lost,etc.
- In the case of TCP, the concept of Sequence No's was present so that the destination knows in which order the data should be read. In case of UDP, the segments can arrive at the destination in any order, and UDP won't know the proper ordering of them.
- UDP is used whenever we require high speeds of data transfer. This is due to the fact that the size of the UDP header is very small compared to the TCP header (i.e so lesser payload). Another reason is that since UDP doesn't form a connection with the destination first, the segments will be sent on multiple available paths.

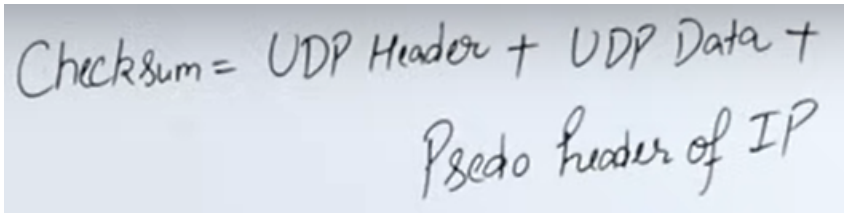


UDP Header :



1) Source Port : The port number of the source machine.

2) Destination Port : The port number of the destination machine.
3) Length : The total length, including the UDP Header + DATA. (i.e In case of TCP, the HLEN parameter gives only the length of the TCP Header).
The length is a 16 bit number, which means the UDP Header + Data <= 65535 bytes.
4) Checksum : Used for error control.



Note : THE CHECKSUM FIELD IN THE UDP HEADER IS OPTIONAL.

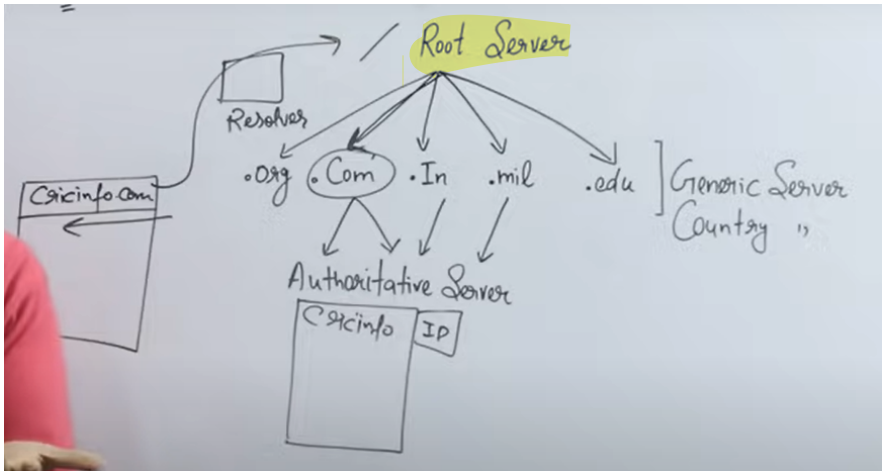
Applications of UDP : DNS, Voice over IP, Online Games, Video Streaming. Even when we want to send a multicast or broadcast a message, we use UDP. Otherwise we'll have to separately create a connection with each and every destination node.

Application of TCP : HTTP, FTP, SMTP

PROTOCOLS USED IN APPLICATION LAYER

1) **DNS(i.e Domain Name System) :** The DNS helps in mapping domain names to IP addresses.
   For example, when we want to access www.cricinfo.com, the DNS helps in mapping www.cricinfo.com to the IP address of the server which holds the resources of www.cricinfo.com.

   The reason why DNS is important :
   - It is easier for us to remember the domain names rather than the IP address.
   - IP addresses of domains are dynamic in nature. The IP address of a particular domain can change at any moment in time. If we don't have DNS, then we'll have to manually make sure that the IP address of each domain is still valid or not.

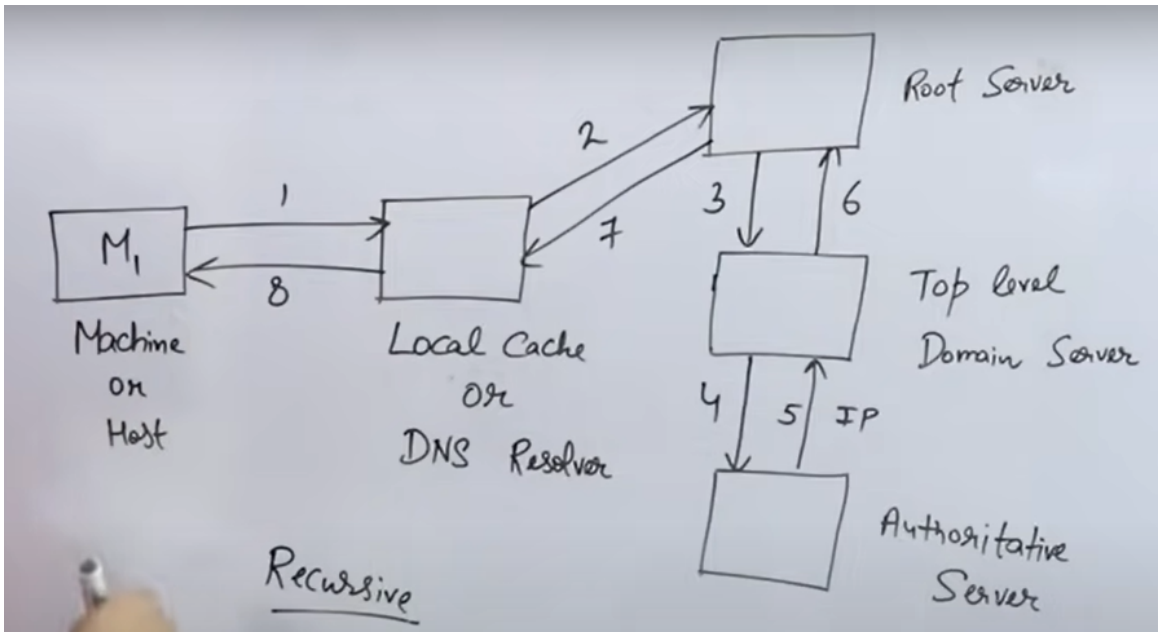


How DNS works :

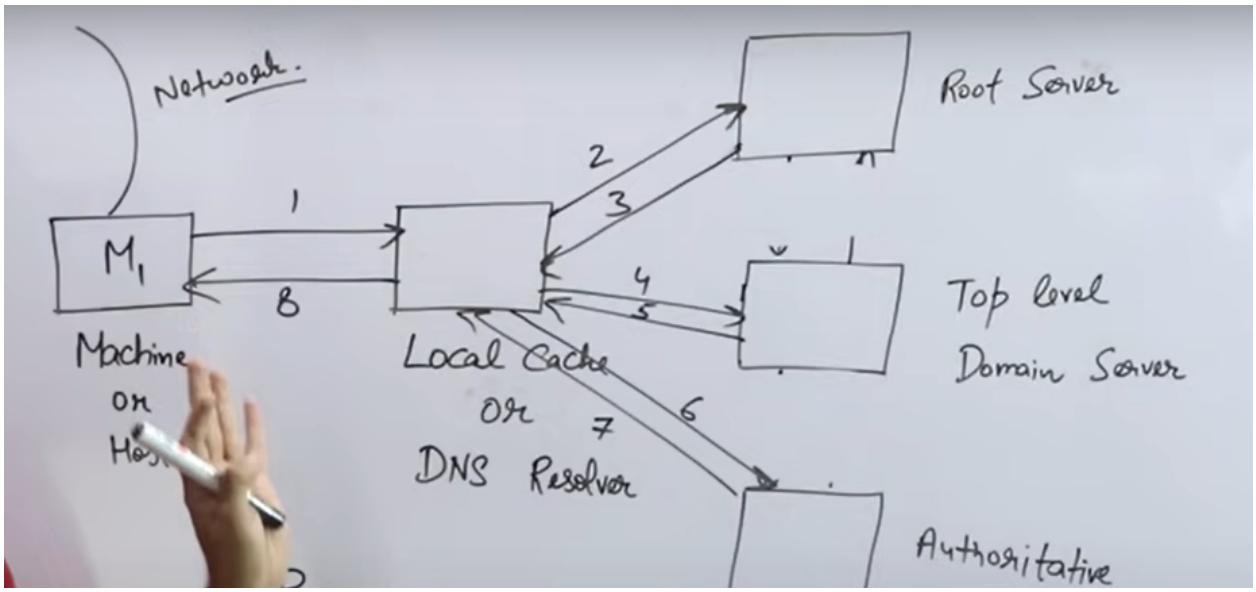
- Step 1 : Once we type the domain name on our machine, we check if the IP address of that domain is present in our **local cache(i.e usually the ISP caches the mapping of popular domain names)**.
  If the IP address is present in the local cache, then directly we can send a request to that IP address for obtaining the resources related to the domain. In this case, we don't need to perform the further steps.
- Step 2 : If the mapping of the domain is not present in the local cache, then the request for mapping will be sent to the **Resolver**, and then to the **Root Server**. Across the world there are **only 13 Root Servers** available.
- Step 3 : Using the Root Server, we'll be able to find the IP address of the appropriate **Generic Server**.
- Step 4 : Using the Generic Server, we'll be able to find the IP address of the appropriate **Authoritative Server**.
- Step 5 : The Authoritative Server is the server which **actually holds a table having a mapping between the domain name and the IP address.**
- Step 6 : The IP address of the domain is now given back to the web browser, and the web browser can send a request to that web server for obtaining the files & resources of the requested domain.
- Step 7 : The web server will send a response consisting of the requested files back to the web browser.

The DNS process can either be recursive or iterative in nature.

DNS (i.e Recursive) :



DNS(i.e Iterative) :



Resolving process takes place in the right to left direction. If we type “www.cricinfo.com” for the first time, the request will be sent to the root server through the resolver. Since there is a “.com”, the root server will then return the IP address of the “.com” generic server to the resolver.
The resolver makes another request to the “.com” generic server, and the generic server will return the IP address of the Authoritative server. Finally, the resolver will make a last request to

that particular Authoritative server, and the Authoritative server will send back the IP address of the domain.

DNS uses UDP, so that the entire process is very fast.

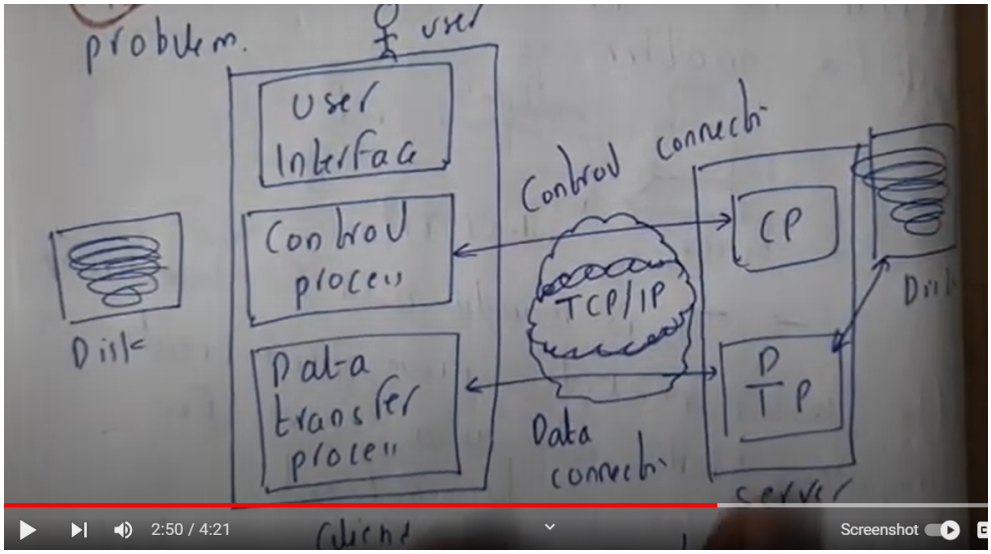
Types of records in DNS :

- **A Record :** Has a mapping of domain name to IPV4 address
- **AAAA Record (i.e Quad-A Record) :** Has a mapping of domain name to IPV6 address
- **CNAME Record :** Has a mapping of a subdomain to another domain. The subdomain is basically an alias to the other domain. For example, let's say you have a website with multiple webpages. The root domain can have an A record, and the subdomains can have CNAME records pointing to the root domain. (i.e Rather than creating separate A Records for each subdomain). The advantage of using CNAME records is that in the future if the IP address changes, we need to change only the A record of the root domain.
- **MX Record (i.e Mail Exchanger Record) :** Has a mapping of the domain name to the server to which the mail has to be forwarded. There is a field called "priority". For a particular domain, the mail will be forwarded to the server having the least priority value(i.e least value indicates more priority is given). Whenever you send an email, your MTA (i.e Mail Transfer Agent) will query for the MX Record to see the appropriate mail server.
- **PTR Record :** Has a mapping between IP addresses to the domain name. PTR records are the opposite of A or Quad A records. PTR records are used in association with email to prevent email spam. When an email server receives an email, it will determine whether the email is spam or not by checking whether the domain of that email is legitimate or not(i.e use PTR here for converting the IP address to domain name).

2) **FTP(i.e File Transfer Protocol) :** The FTP protocol is used whenever we want to transfer files from one host to another host. It is mainly used for transferring web files from the web servers to the other hosts. It is also used to download/share files across the internet.
Although transferring files from one system to another is straightforward, there could be many problems that can occur such as the hosts b/w which the file transfer is occurring having different data representations.
FTP protocol helps to solve these problems.
Port numbers 20 and 21 are used for FTP, and FTP protocol internally uses TCP for the transmission of the files.

The FTP connection consists of 2 connections : **Control connection & Data connection**

- Through the control connection, commands are transferred between the hosts.
- Through the data connection, the actual data is transferred between the hosts.

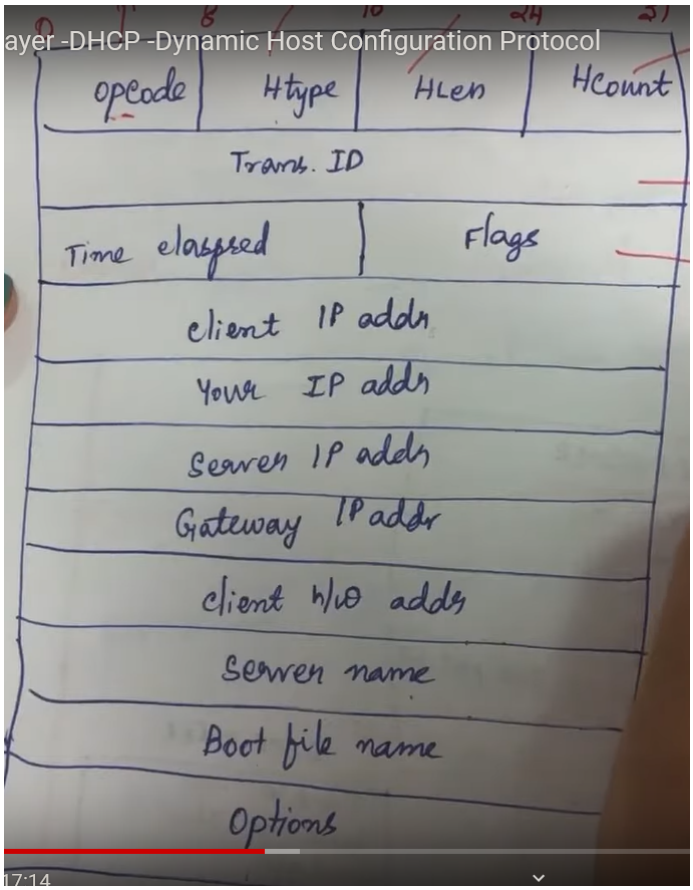


3) **DHCP(i.e Dynamic Host Configuration Protocol) :** The ICANN organization(i.e International Corporation for Assigned Names & Numbers) usually allocates a block of IP addresses for an organization. The ISP(i.e Internet Service Provider) provides these IP addresses to the organization, and usually it's the responsibility of the Network Administrator to actually allocate each IP address with each device.

Instead of the Network Administrator manually performing this task, we can use DHCP which will automatically assign all the IP addresses to the devices of the organization.

DHCP is an application layer protocol, and it's also a plug-n-play protocol(i.e we can download it whenever we want). It helps the TCP/IP at the network layer.

DHCP Header :



Opcode : Specifies if it's a request or a reply message

Htype : Specifies the hardware type. Whether the devices are connected by ethernet, etc.

Hlen : Specifies the length of the hardware address

Hcount : Maximum number of hops that the packet can travel

Trans ID : Specifies the transaction ID

Time elapsed : Specifies the number of seconds it has been since the client has started to boot

Flags : 0 for unicast message & 1 for multicast message. 16 bits have been reserved for flags but 15 of them are not used.

Client IP address : IP address of the client. Initially when the protocol starts, the client IP address will be set to 0.0.0.0 since the client doesn't know it's IP address.

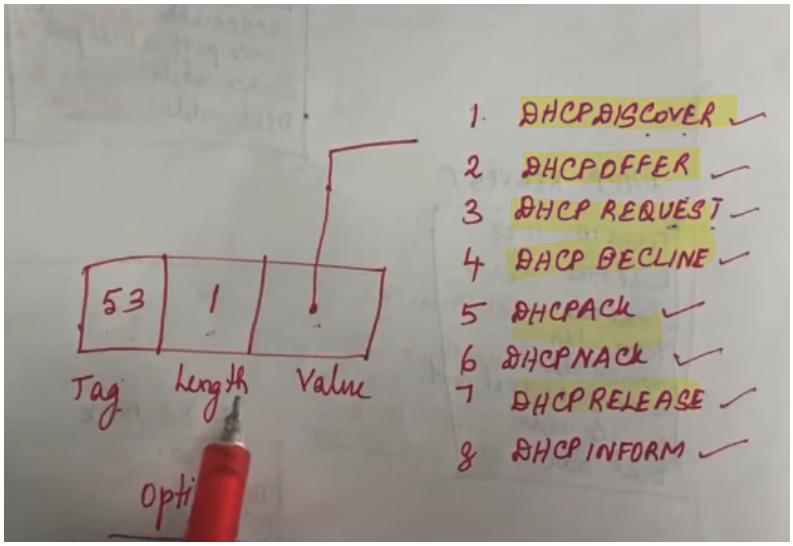
Your IP address : IP address that has been assigned by the server to the client

Server IP address : IP address of the server

Gateway IP address : IP address of the default router

Server name : Name of the server

Options :



The tag & length will be fixed to 53 and 1 respectively. Only the value will change depending on the type of message being sent between the client and server.

DHCP operation :

The DHCP operation takes place by sending 4 different messages :

Initially the client does not have any IP address assigned to it.

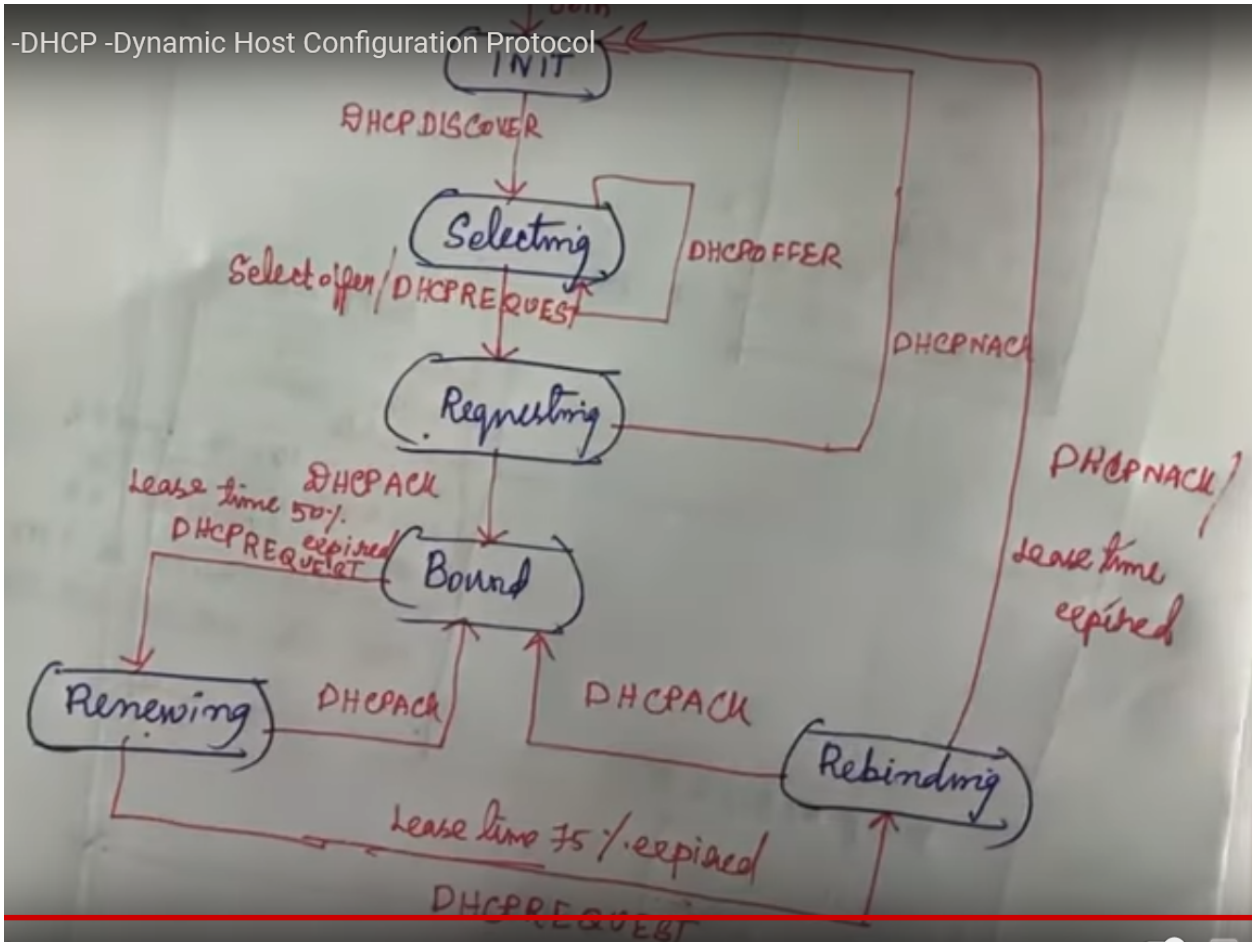
1.) **DHCP DISCOVER :** The client will first send a broadcast DHCP Discover message to all the nearby servers.

2.) **DHCP OFFER :** Whichever servers have IP addresses available & which are willing to provide one to the client will send a DHCP OFFER message back to the client.

3.) **DHCP REQUEST :** The client would have received multiple DHCP OFFER packets arriving from all the different servers which are willing to provide an IP address. Out of all of these, the client will decide which is the best, and will send a DHCP REQUEST packet to that server.

4.) **DHCP ACK :** The server will send an acknowledgement message back to the client.

Now the client has been assigned with an IP address.



Lease time : Refers to the amount of time for which the server has decided to provide that IP address to the client for.
Once 50% of the lease time is completed, the server will send a DHCP REQUEST once again so that it can renew the IP address. If a DHCP ACK is received from the server, then no issues. If the server did not send a DHCP ACK, the client will wait till 75% of the lease time is completed and will send another DHCP REQUEST.

4) **HTTP & HTTPS (i.e HyperText Transfer Protocol & HyperText Transfer Protocol Secure) :** The HTTP protocol is used for viewing web pages. HTTP is an application layer protocol, uses TCP, and has port number 80. Basically when a user types a url on the web browser, dns occurs to map the domain name to the IP address of the web server that has to be contacted. A TCP connection will be established between the host computer and the web server. Once the connection is formed, using HTTP, the data is transferred between the computer and the server.

When HTTP is used, all the information is sent in the form of clear text, making it vulnerable to attacks and unauthorized access.

Hence HTTPS came into existence. HTTPS is basically an advanced version of HTTP which provides security features. HTTPS uses TCP in the same way as HTTP does and has a

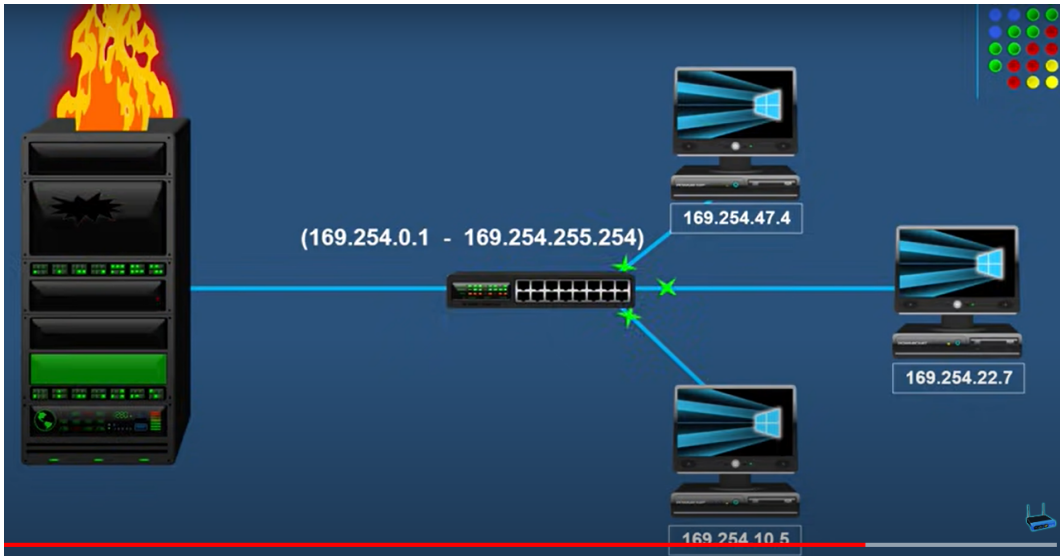
port number of 443. The data is encrypted on the sender side, and then gets decrypted on the receiver end. Encryption is defined as the process of converting the data into an unreadable format, and this helps protect the data.
HTTPS provides this security using 2 different protocols :

- SSL (i.e Secure Socket Layer) : It is a protocol used to provide security on the internet. It uses public key encryption to secure the data. First, the computer's web browser will ask the website to identify itself. Then, the web server will send a copy of its SSL certificate to the web browser to show its identity. The web browser will validate the certificate. If the certificate is valid, then the computer will send a message to the web server telling us that we can start communication to which the web server will send back an acknowledgement.
  After this normal encrypted data transfer takes place.
- TLS (i.e Transport Layer Security) : TLS is basically the successor of SSL. In the same way as SSL, it authenticates the client & server, and encrypts the data.

APIPA (Automatic Private IP Addressing)
APIPA is related to the DHCP.
Usually when a device is connected to the network, an IP address will be assigned to that device which will allow it to communicate with any other device on the network. This IP address is either allocated by the network administrator or through DHCP.
Now when the DHCP server itself faces an issue and isn't able to allocate an IP address to the device, that's when APIPA is performed.

Scenario :
Let us consider 3 nodes connected to a switch, and that switch is connected to the DHCP server. Now when the nodes make a request, they realize that the DHCP server has gone down due to some reason. The nodes are then allocated a private IP address in the range **169.254.0.1 to 169.254.255.254 (i.e subnet mask is 255.255.0.0)**, with each node having a different private IP address.

Using this private IP address, a node will only be able to communicate with other nodes in the same network. They will not be able to communicate with the internet as well as nodes belonging to other networks.
From time to time, it is checked to see if the DHCP server has come back, and if it has, then the private IP address that has been allocated to the device will now be given a normal public IP address by the DHCP server.



General Information :

MAC Addresses :

- The MAC address which stands for Media Access Control address is an identifier which uniquely identifies every node present on the network.
- The MAC address is present on the NIC (i.e Network Interface Card).
- It is **48 bit in length**, with the 3 bytes to the left (i.e MSB bytes) representing the **manufacturer of the device itself**, and the next 3 bytes is a unique number which helps in **identifying the device** itself.
- The major difference between MAC addresses & IP addresses is :
    - **MAC addresses help in identifying the device, whereas IP addresses tell where a device is located (i.e like in which network it's located in and so on).**
    - For example : IP addresses can be visualized as the address of a house, whereas the MAC address is the name of the person living in that house.
    - Whenever one device wants to send a message to another device, it first examines the IP address of the destination device to understand whether the destination is on the same network or different network.
      Once that's done, using the ARP protocol, hop to hop delivery will be performed till the message is sent to the destination (i.e Just refer to the previous ARP explanation).

NIC (Network Interface Card) :

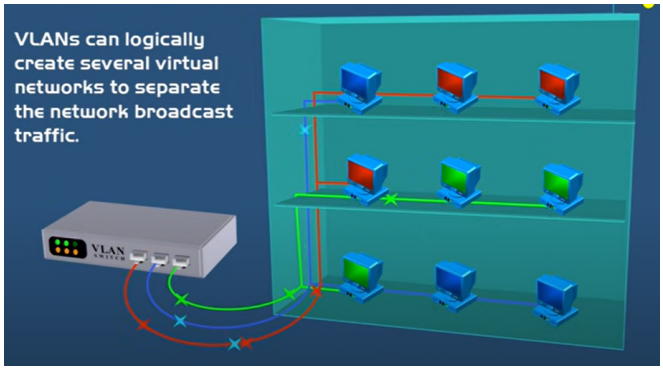
- **The NIC is a hardware device without which a computer cannot be connected to the internet.**
- The **MAC/physical address** is present on the NIC itself, which helps in uniquely identifying every device on the network.
- There are 2 types of NIC's :

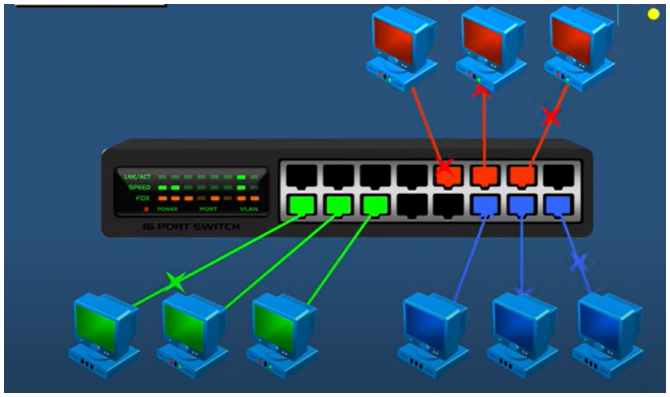
    - **Internal NIC :** There is a slot provided in the motherboard itself, where the NIC can be inserted.
    - **External NIC**
- **The NIC is a networking device which belongs to the Data Link layer of the OSI Model.**

VLAN (Virtual Local Area Network) :

- Through the concept of VLAN, we will be able to logically connect the various computers, servers, and other network devices irrespective of their physical location.
- Few advantages of using VLANs are :
    - Improved network security
    - Traffic management
- Scenario :
    - Let's suppose there is an office building with 3 floors, with each floor having a set of computers. These computers can belong to 3 different departments : Either accounting, shipping, or support, and each floor can have computers belonging to different departments.
    - All the computers are connected to a single switch, and hence it can be considered as a LAN. Currently, the computers of one department not only can access the traffic of its own department, but also the traffic of the other departments. (i.e All the network traffic is mixed in with other departments.)
    - Suppose we wanted to separate the network traffic between these departments :
        - Solution 1 (CONCEPT OF SUBNETS) : We can move all the computers belonging to the same department to a single floor, and then we can make them physically connected to a switch. So we can maintain 3 subnets. These switches can then be connected by using routers. However, it may not be possible to physically move the devices, and this approach requires us to use additional hardware such as the routers, cables, etc.
        - Solution 2 (CONCEPT OF VLANS) : We use the concept of VLANs. We try to create several virtual networks to separate the network traffic between the departments. On a VLAN-supported switch, we create these virtual networks by using the concept of ports. Every port on the switch can be configured to belong to a particular VLAN.
          We assign a few ports to each virtual network. Now computers belonging to the shipping department will be connected to the ports allocated for shipping, and so on. In this way, irrespective of where the devices are actually present, we've been able to segregate the network traffic effectively.

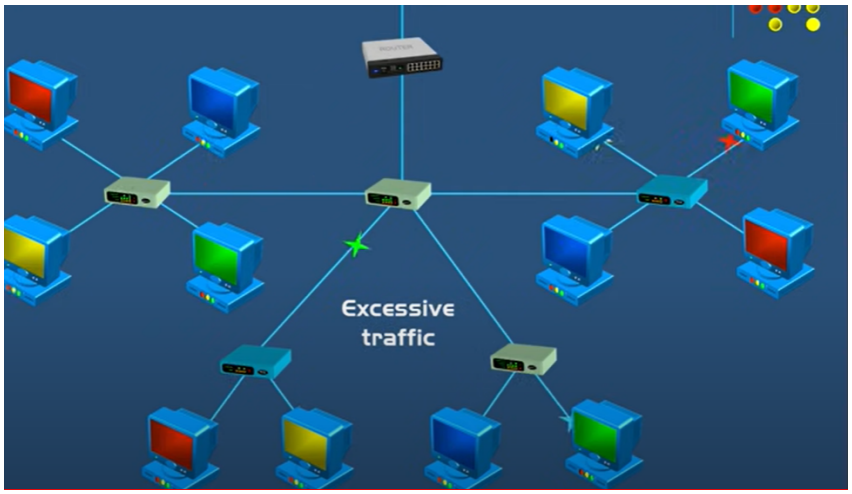
          The approach of VLANs is preferred over Subnets, especially for medium to large sized networks.



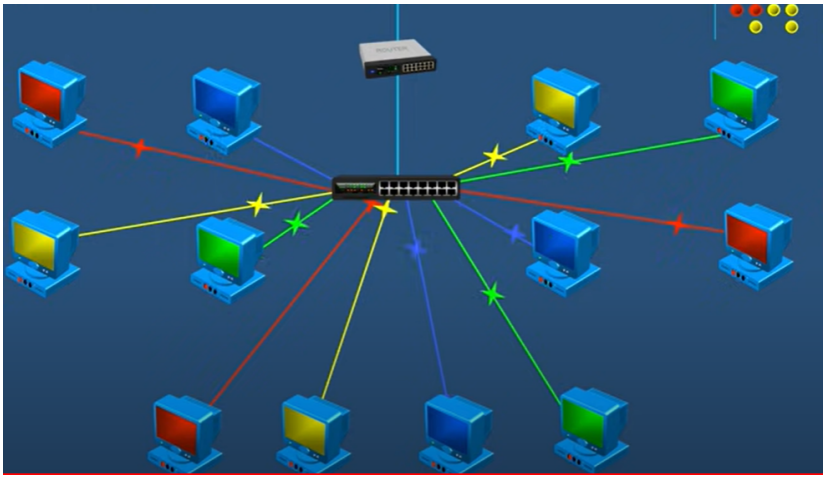


VLANs also help in solving the issue of congestion. If many nodes are connected to the network and all the nodes are sending a lot of data, congestion can occur on the lines. Now by creating virtual networks, each logical group of nodes will have their own lines and hence the problem of congestion can also be solved.

BEFORE VLANs



AFTER VLANs



Ping Command :

- The ping command is used to test the network connectivity between 2 devices.
- **Syntax :** ping ip_address/domain_name
- **Example :** ping yahoo.com / ping 127.0.0.1
- When we type ping, **4 data packets** are sent to the destination. Now based on how many packets are received back, how many are lost, etc, we can easily determine the issue.
    - **Case 1 :** If all 4 data packets are received back, this means that there is network connectivity between the 2 devices & there are no issues.
    - **Case 2 :** If few data packets are lost, then this could be caused either due to **network congestion** or either **due to faulty hardware**(i.e could be the NIC, router, modem, etc).
        - To test if the NIC is working properly, use the “loopback address”(i.e 127.0.0.1). Type “ping 127.0.0.1” and see if the packets are being sent and received properly.
    - **Case 3 :** If all the data packets are lost, then either :
        - There is no network connectivity between the 2 devices.
        - Network connectivity is available, but the destination is turned off, due to which no packets are sent back.
- Using ping, we’ll be able to test DNS as well. When we type a command like “ping yahoo.com”, and packets are being sent, then that means that the DNS is able to map yahoo.com into the IP address, indicating no issues with DNS. If it’s not able to, then there is a problem with DNS.

```
C:\Users\Admin> ping 192.168.1.5

 Reply from 192.168.1.5:  bytes=32  time=1ms  TTL=47
 Reply from 192.168.1.5:  bytes=32  time=1ms  TTL=47
 Reply from 192.168.1.5:  bytes=32  time=1ms  TTL=47
 Reply from 192.168.1.5:  bytes=32  time=1ms  TTL=47

 Ping statistics for 192.168.1.5 :
    Packets:  Sent = 4,  Received = 4,  Lost = 0
```

Traceroute command :

- The traceroute command is used to show the route that is taken by the data packet from the source to the destination.
- Syntax : tracert ip_address/domain_name (i.e Windows) / traceroute (i.e LINUX)
- Example : tracert yahoo.com
- **Traceroute tells us more information compared to ping. Traceroute will ping every router on its way to the destination.**
- **Every router is pinged by sending 3 packets.**
- Output :

```
C:\Users\Admin> tracert  google.com

1   <1 ms    <1 ms    <1 ms  192.168.0.1
2    8 ms     7 ms     8 ms  96.120.36.133
3    8 ms     8 ms     9 ms  96.110.110.209
4    9 ms     9 ms     9 ms  fl.pompano.comcast. [16.2.151.122.2]
5   11 ms    12 ms    10 ms  68.86.90.205
6   12 ms    14 ms    14 ms  miami.fl.ibone. [68.86.8.7]
7   15 ms    17 ms    16 ms  108.170.249.17
8   20 ms    21 ms    22 ms  mia07s56-in-fl [143.250.64.206]
```

The first column is basically the number of hops that the packet takes to reach the destination. For every hop/router, traceroute performs ping by sending 3 packets, so the next 3 columns are the RTT times for each respective hop. The last column gives us the IP address & other information related to that hop itself.

- By using traceroute, we'll be able to determine where bottlenecks are present within the network.

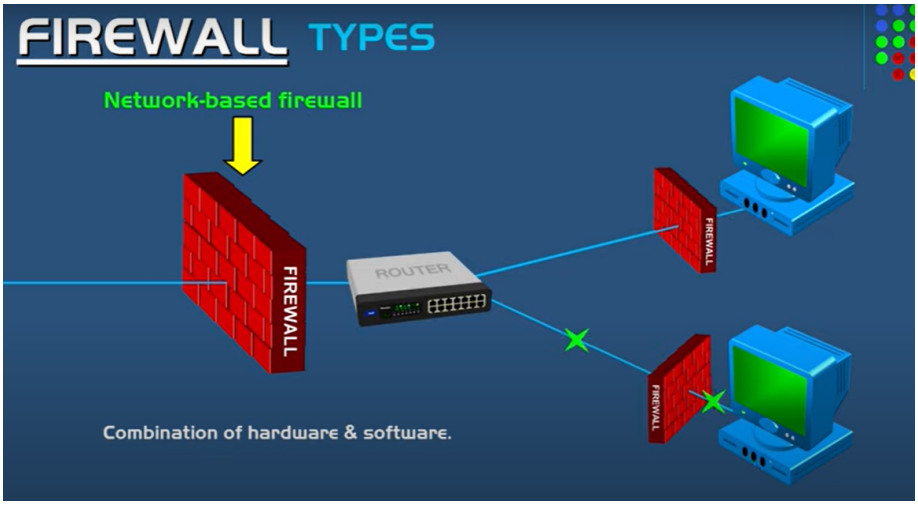
```
C:\Users\Admin> tracert example.com

1   <1 ms    <1 ms    <1 ms  192.168.0.1
2    8 ms     7 ms     8 ms  96.120.36.133
3    8 ms     8 ms     9 ms  96.110.110.209
4  215 ms   210 ms   205 ms  68.86.90.205
5  220 ms   215 ms   222 ms  miami.fl.ibone. [68.86.8.7]
6  225 ms   230 ms   225 ms  miami.fl.ibone. [12.68.220.1]
```

In this output, we see that there is a major increase in the RTT between hops 3 & 4. This could indicate that hop 4 is the bottleneck, however it doesn't guarantee that. It

could merely also mean that the distance between hop 3 & 4 is very large, so it takes more time for the data packet to travel.

Firewalls :

- **A firewall is a system which helps to protect a private network by preventing unauthorized access to it.**
- The firewall will prevent unwanted traffic from entering into the network, and allows wanted traffic to enter.
- The firewall basically acts as a barrier between the public internet and the private network.
- The firewall basically **filters out the traffic based on certain rules** that are defined on various parameters, including the **ip address & port numbe**r of the source, as well as the **protocols** being used(i.e tcp, udp, etc).
- There are 2 types of firewalls :
    - **Network-level firewalls :** Acts as a barrier between the private network and the outside world
    - **Host-level firewalls :** Help to protect an individual computer. If by any chance any unwanted traffic does go through the network-level firewall, the host-level firewall will still protect the computer from unauthorized access.



IP Spoofing :

IP spoofing is a technique used to manipulate the source address in an IP (Internet Protocol) packet. In this technique, an attacker disguises their identity by forging the source IP address of a packet to make it appear as if it originates from a different source than it actually does. The primary goal of IP spoofing is to deceive the recipient or intermediate devices (such as routers and firewalls) and gain unauthorized access to the network.

IP Spoofing is used in :

- DDoS Attacks : In Distributed Denial of Service (DDoS) attacks, attackers often use IP spoofing to flood a target server or network with a massive volume of traffic. By spoofing the source IP addresses, it makes it difficult for the victim to identify the actual sources of the attack, making mitigation more challenging.

VPN (i.e Virtual Private Network) :

- VPNs are used to establish a secure connection over an unsecure network like the Internet.
- They help in hiding the identity of the user and protect the user activity.
- Usually what happens is that the ISP provides you internet facilities, and the ISP server can monitor and see all the activity that you perform. There is a possibility that the ISP sells this activity data to other organizations and the government, making your activity not private.
- To protect your identity, we use VPNs, where instead of the request going through the ISP server, it goes through the VPN server. The other devices do not know information about the source since the VPN server replaces the source address with its own address making the access anonymous.
- In addition to this, VPN also makes sure that the data is secure by using the concept of tunneling and encryption. The VPN divides the data into smaller parts and then encapsulates each part to protect it from unauthorized access, kinda creating a private tunnel between the source & destination.

Important Terms with Definitions :

- Jitter : Defined as the variation in the time taken for the packet to travel from the source to the destination. This can be caused due to network congestion, etc.
- Latency : Defined as the time taken for the packet to travel from source to destination. High latency can result in delays in data transmission.
- Anycast IP Addressing : Anycast allows multiple servers or nodes to share the same IP address, and when a client sends a request to that address, the network's routing infrastructure directs the request to the nearest (or most appropriate) server that advertises that IP address. In other words, anycast enables multiple devices to announce the same IP address, but the network routes traffic to the "closest" or "best" instance of that address based on various criteria.
- Tunneling : Tunneling is defined as the process of placing data packets from one network protocol inside the data packets of another protocol to enable the transmission of data across networks that may not natively support the inner protocol.
  The concept of tunneling is used even in VPNs as mentioned above. (Refer to the VPN section).

Interview Questions :

1.) What is ISO?
Ans : In computer networking, ISO usually refers to the International Organization for Standardization's work in developing networking standards. The ISO has been instrumental in creating the Open Systems Interconnection (OSI) model.

Just explain the OSI model after that.

2.) What is the best example that one can use to explain the concept of networking to a layman?

Ans : A great example to explain the concept of networking to a layman is to compare it to a postal system or mail delivery service.

Imagine a postal system where you have different houses (computers/devices) in a neighborhood (local area network or LAN). Each house has a unique street address (IP address) that allows the mail carrier (data packets) to know where to deliver the letters (information). The mail carrier (data packets) picks up letters from one house and delivers them to the correct destination house based on the addresses.

In this analogy:

1. **Houses (Computers/Devices):** Represent individual computers, smartphones, or other devices that are part of the network.
2. **Street Address (IP Address):** Like each house having its unique address, each device in the network has its unique IP address that helps identify and locate it on the network.
3. **Mail Carrier (Data Packets):** The mail carrier collects letters and delivers them to their correct destinations. Similarly, data packets carry information between devices on the network, ensuring it reaches the right destination.
4. **Neighborhood (Local Area Network - LAN):** The neighborhood is like a small local area network where devices are close to each other and can communicate directly.

3.) What is a gateway, its functions, and difference b/w gateway and router?

https://takeuforward.org/computer-network/define-gateway-the-difference-between-gateway-and-router/

While routers and gateways are both network devices that connect multiple LANs (Local Area Networks) together, they serve different functions and operate at different layers of the OSI model. Routers and gateways both facilitate connectivity between multiple LANs, but routers are specifically designed to route data between networks at the Network Layer using IP addresses, while gateways can operate at different layers and often involve protocol translation and other functions to allow communication between networks with different characteristics

Router : Network Layer
Gateway : Application Layer

Functionalities of Gateway :

1. **Interconnects Different Networks:** Gateways are used to interconnect networks with different architectures or protocols, such as connecting a local area network (LAN) to the internet or linking networks that operate on different communication standards.
2. **Protocol Translation:** One of the primary functions of a gateway is to perform protocol translation. It converts data packets from one network's format into a format suitable for the destination network.
3. **Address Translation:** Gateways can also perform address translation, mapping the addresses used in one network to the corresponding addresses in the other network. This is commonly seen in network address translation (NAT) used in home routers to translate private IP addresses to a single public IP address for internet communication.
4. **Security Enforcement:** Gateways can act as a security checkpoint, implementing firewall rules to regulate and control data flow between networks. They help protect internal networks from unauthorized access from external networks.
5. **Data Routing:** Gateways use routing protocols to determine the best path for data to travel between the source and destination networks. They maintain routing tables to make informed decisions about forwarding packets.

Basically the router does only the 5th point, but gateway provides more functionality.

Gateway protocols are just Application layer protocols.

1) SMTP : Gateways help to send email messages between email servers from one network to another network.
2) FTP : Gateways help to send files between different networks using FTP protocol.
3) DHCP : The DHCP gateways help in assigning the IP addresses to the clients.
4) DNS : DNS gateways resolve domain names into IP addresses, by acting as DNS servers.
5) BGP(i.e Border Gateway Protocol)

IPV4 vs IPV6

- 32 bit to 128 bit
- Decimal Format to Hexadecimal Format
- Require NAT so that we can support more number of devices => Don't require NAT since we can support many devices without use of private IP addresses
- IPV4 addresses given manually or by using DHCP protocol => IPV6 addresses can be given by using DHCPv6 protocol or SLAAC.
- IPv4 header is 20-60 bytes, whereas IPV6 header is fixed at 40 bytes.
- Encryption & Authentication is not provided in IPV4, whereas they are provided in IPV6.
- IPv4 uses ARP(Address Resolution Protocol) to map to MAC addresses. Whereas, IPv6 uses NDP(Neighbor Discovery Protocol) to map to the MAC address.