Week 1
8/9/22

How internet works?
Computer Networking

Network: A computer connected to other computers

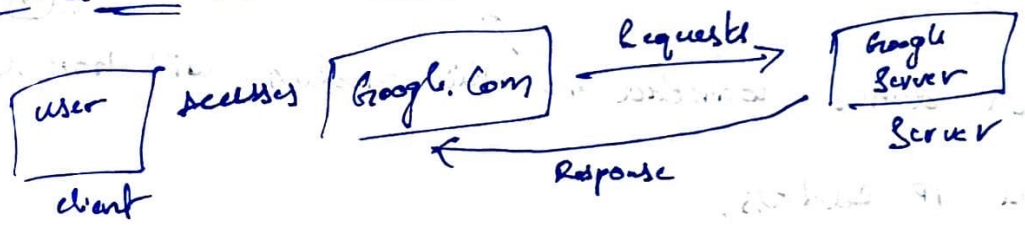
Internet: Collection of such computers connected on a global scale.

* Rules to make sure networking is performed well is called protocols.

→ Having set of rules/regulations is important cz it enables standardisation & universal communication.

→ Internet Society usually makes these changes or rules RFC are submitted, reviewed & implemented if req.

* Server & client:



• A user can also be server (if local host)

* TCP (Transmission - Control Protocol):
Idea behind it is, data will be transmitted to its destination without any loss of packets / corruption.

* UDP (user datagram Protocol):
when it is not mandatory that 100% of data is being transfered (like video calling etc).

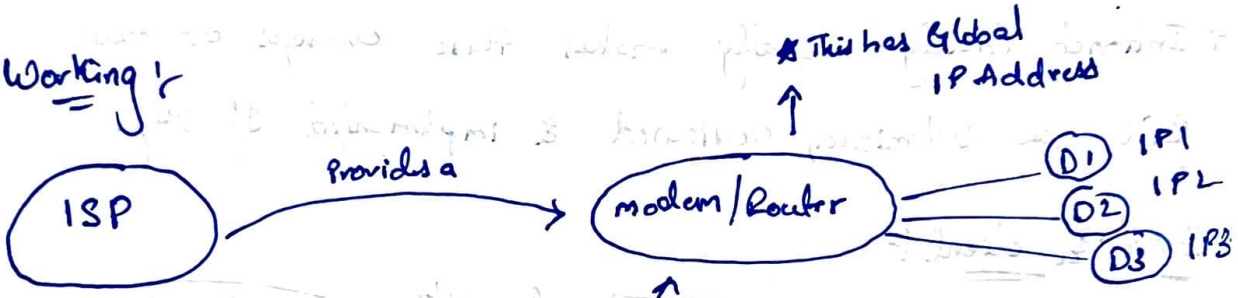
* HTTP (Hyper text transfer): Defines format of data being transfered b/w web clients: (used on www).

- Everything in computers is 0 & 1, So files or large files can be sent in smaller chunks. called "PACKETS"
- Every device on internet has a unique address called IP Address. Eg: #.#.#.#
  ↓
  Range from 0 to 255

To check your own IP Address :-

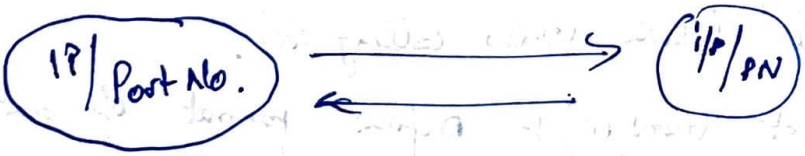
```
$ curl ifconfig.me -S   //command
```

Working :-



So all devices connected to this modem will have the same IP Address.

- IP1, IP2, IP3 etc are local IP Addresses given to each device connected to modem by modem.
- The modem assigns these IP Addresses using DHCP
- IP Address is used to decide which device the data/response should be given. But, which application in device gets it is decided by PORTS



- Ports are 16 bits nos

- HTTP functions at port 80.
- mongoDB port = 27017

Ports from 0 – 1023 are reserved.

- 1024 - 49152 are reserved for applications.
- Remaining ones we can use for personal apps / custom.

* How communication happens ?

2 ways : Guided & unguided ways.

Guided : path is guided and physical / defined

unguided : No single path but network is working like wifi

- Submarine cables are run under water in oceans from all countries. (optical fiber cable)

Control chain of cables :

Country entity → Smaller entities → ISP → users etc.

- These cables are heavily guarded

Physically : optical fibre cable, coaxial cable

Wireless : Bluetooth, Wi Fi, 5G/LTE etc

- We use these over satellites as these are much faster.

LAN (Local Area Network) :-

Small Area / Office via ethernet, adapters etc

MAN (Metropolitan " ") :-

Across cities

WAN (Wide Area network) :-

Across countries using optical fiber cables.

- Internet is collection of all these.

(i) SONET (Synchronous optical networking) :-

carries data over large dist. using optical fibers.

(ii) Frame Relay :-

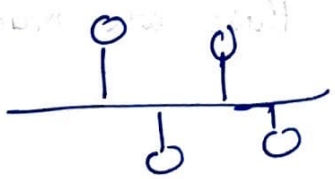
A way to connect LAN to WAN.

- modem / Router :- converts digital to analog signals & vice versa.

* How computers connected ?

Topologies :-

(i) Bus topology :- connected to one backbone cable

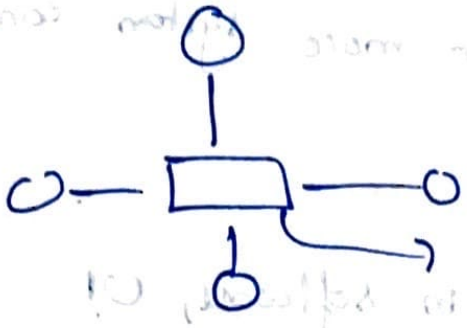


1 brake = stop transfer.

(ii) Ring :- connected in a ring manner. Each sys. communities with each other.

call A to B, But all will be called.

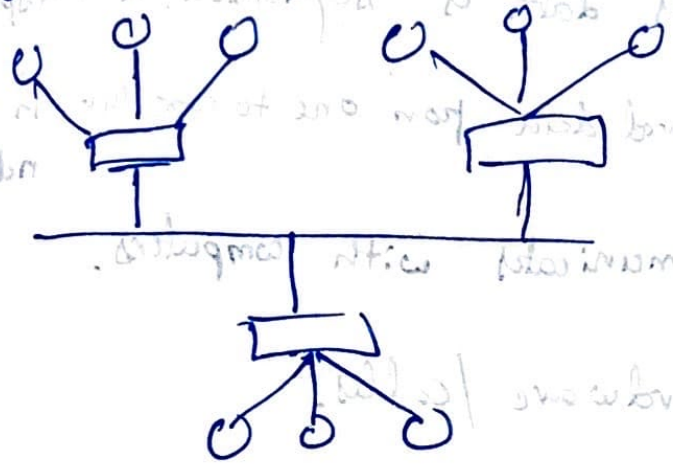
(iii) Star :- All computers are connected to one controlling and centralised systems.



If fails, all fails.

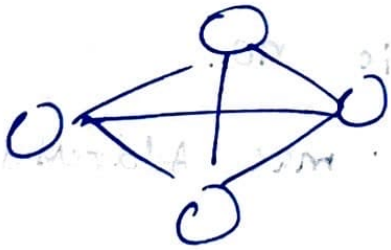
(iv) Tree (Bus-star) :-

Comb. of Bus & star network. Every bus can have many star networks.



(v) mesh topology :- All networks / computers is connected to every other computer.

- Expensive, less scalable.



* Structure of network:

OSI Model (Open systems interconnection)

- To standardise how 2 or more system communicate.
- 7 Layers:

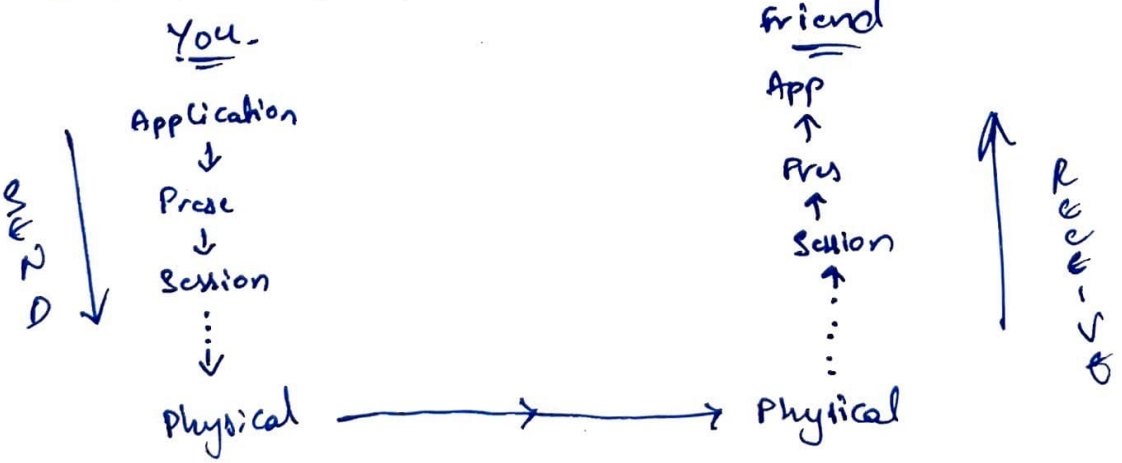
Application ⇒ implemented in software, UI
↓
Presentation ⇒ Converts data to machine readable Binary lang. & encrypts the data. & Abstraction.
↓
Session ⇒ helps managing/setting up connection/termination.
↓
Transport ⇒ using protocols, data is segmented & transported
↓
Network ⇒ Trans. of received data from one to another in dif. networks.
↓ Routing
Data Link ⇒ Directly communicates with computers.
↓
Physical ⇒ Contains hardware/cables.

Router lives here.

* MAC Address: Physical Addresses which are 12-digit Alphanumeric no.

one computer can have multiple mac Addresses.

Eg of working: Communication b/w two pcopc.

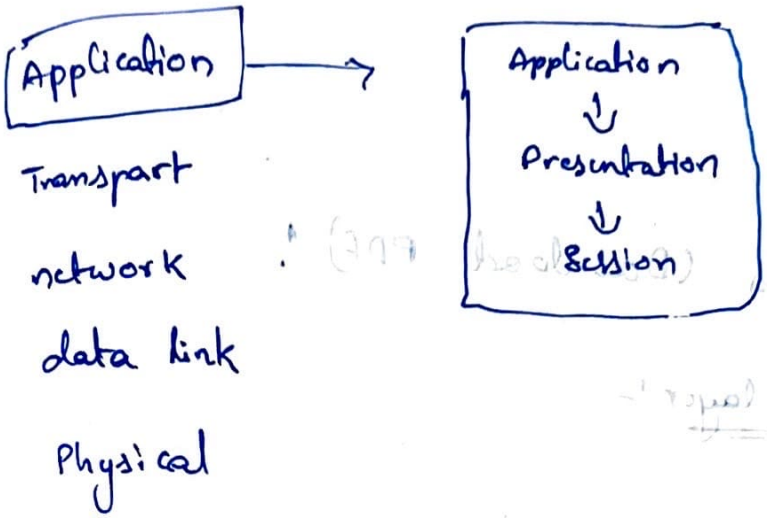


* Another model :-

. TCP/IP model :- mostly similar.

Internet protocol suite.

. It has only 5 layers.



. This is used more practically. OSI is more conceptual.

---

* Application Layer :-

. users interact with this (eg whatsApp, Youtube).

. lies on the devices.

. Protocols involved

. client-server Architecture ①

* Server is a system that controls your web/app.

. Ping time is the round trip time for messages sent from origin to receiver & back.

Ping time is already ideal and cannot be reduced further.

⑧ Peer-to-Peer (P2P) Architecture :-

- Apps on various devices are connected directly with each other. without any server.
- Every system will be server/client.
- easily scalable.
- Decentralised.

* Networking Devices (Download PDF) :

* Protocols on App. layer :-

- web protocols :

VNC / HTTP | DHCP | FTP | SMTP | POP3 & IMAC | SSH

TCP/IP

→ TELNET : Port 23 (low-level protocol).

→ UDP : stateless connection. Data can be lost.

One program can have many instances, running an instance is called processes.

lighter version of a process => thread (one instance)

A process can have multithreading running.

* Sockets :- It's a software interface/process b/w process & internet to send/receive data/message

* Ports :- used to determine which app should get the data. But, if a app has multiple processes/instances

Ephemeral Ports are used.

↳ internally multiple ports are randomly assigned.

They can exist on client side but on server side, you have to know the code number.

---

* HTTP :- A client-server protocol at App. layer.

. uses TCP → Transport layer.

. It's stateless

(GET, POST, PUSH, DELETE)

Methods :- something that tells server what to do.

(i) GET :- get data

(ii) POST :- client giving data like ID Pass to server.

(iii) PUT :- Puts data at a specific location

(iv) Delete :- delete data from server.

* Status Code :- when a req. is made, we need to know if req success, error etc, so status codes are used.

Eg: 200 → Successful
400 → Bad req.
404 → Not found
500 → internal Server error.

lies in 100's range → Informational category.
200's → success
300's → Redirecting
400's → client error
500's → Server error.

* Cookies :- A unique string stored in clients browser.

∵ HTTP is stateless, new reqs. are added on with cookies, so server will know who is the user is etc.

- It has a expiration date.

→ Third-Party Cookies :- set for URL's you don't visit.

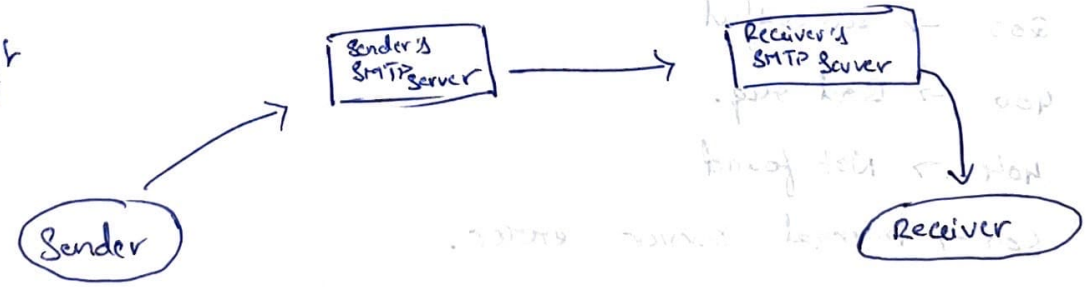
* HTTP is stateless, so server wont know it's you irrespective of how many times you req. so cookies are useful.

---

* How Gmail Works?

- Email uses SMTP at application layer (Simple mail transfer)
  ↳ to send email.
  POP3 protocol to receive emails.
- TCP at transport layer.

Eg:-



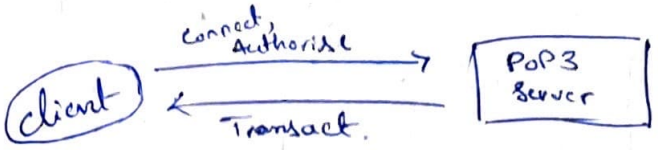
Each client for mail has its own server.

- Error handling takes place at transport layer.

Command for IP & name of servers (SMTP) :-

~ nslookup -type=mx gmail.com

- To download/Receive mails, POP3 (Post office Protocol) is used. Port 110.



* IMAP (Internet message Access Protocol):

used to access mails on multiple devices

So mails will be on server. (unless you delete them).

---

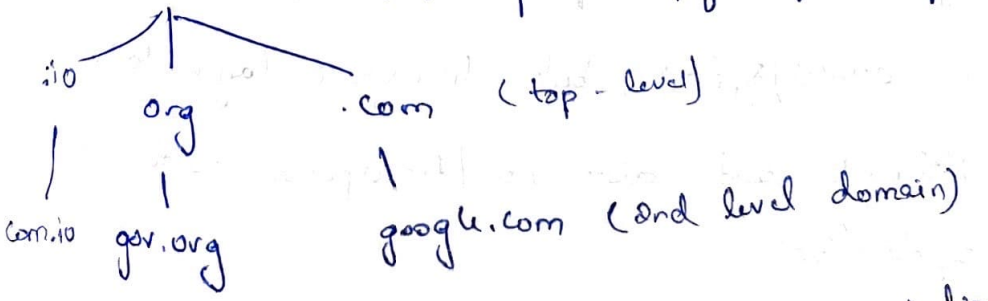
* DNS (Domain Name System): Most popular.

- Domain names are mapped to IP addresses, we use a service to look up in the database.
- It is directory service for IP Address.
- Databases are categorized into classes of domains.

Eg: mail.google.com → Top level
↓ → 2nd level domain
Sub-domain

- multiple databases for all 3 categories instead of one.

* Root DNS Server: top level / first point of contact.

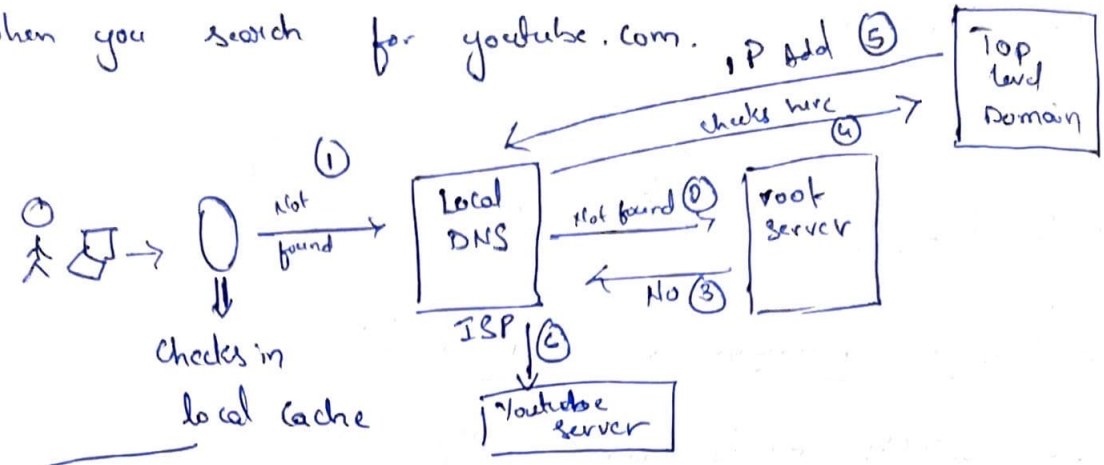


- Top-level domains are basically organization-type specific.

like .uk, .com (or) .org etc.
↓ ↓
country Govt

- ICANN is responsible for registering & managing TLDs.

How DNS works?



when you already a site for 1st time, they are usually stored locally on device or so in form of local cache.

- You cannot ~~buy~~ buy, only rent Domain name.

~ dig server_name. (to check messages received from server etc)

★ Transport Layer (on phones, laptops etc)

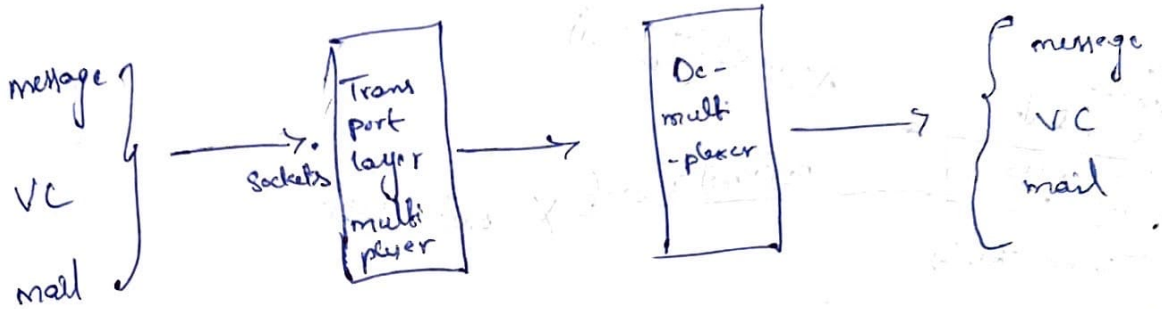
- This lies inside the devices or computer (end systems)
- Take data from the "Network" or network layer and to the applications.
- Network layer takes care of actual transport of data from/between devices/systems, but transport layer takes data from Network and distributes/transports within the device to applications.
- Protocols are TCP & UDP.

* How transport layer works?

Suppose,

You are texting, sending files/mails and video calling all at once.

- multiplexing is used, It allows to send all the above in a single medium.
- Transport layer has multiplexer and demultiplexer.
- Demultiplexer is opposite multiplexer.



- Data travels in packets & transport layer will attach these socket port nos
- Transport layer also takes care of congestion control.
- Congestion control algorithms are built into TCP.

* How data is transported without corruption or loss of data or improper sequence?

- Transport layer protocols take care of this using "CheckSum".
- Checksum is a value derived from data being transmitted. The checksum value is added on to data and sent.
- If the receiver gets the same data with checksum

as sender, the data is correct without any corruptions etc.

(checksum of data matches on both sides ⇒ ✓)

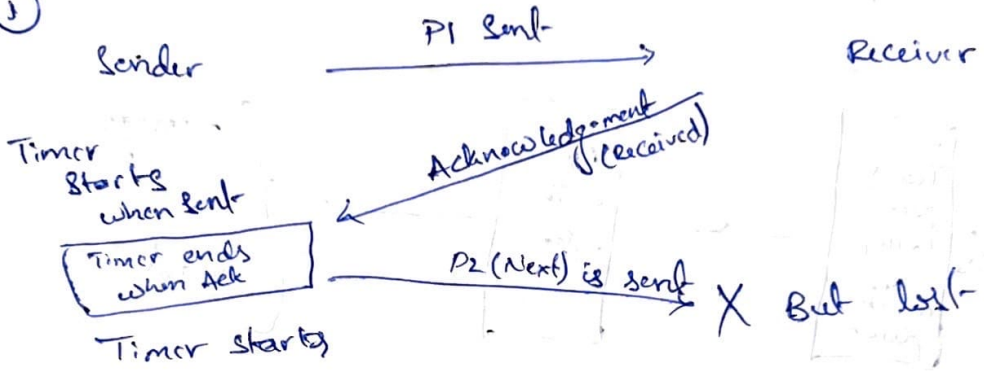
or

data lost or corruption.

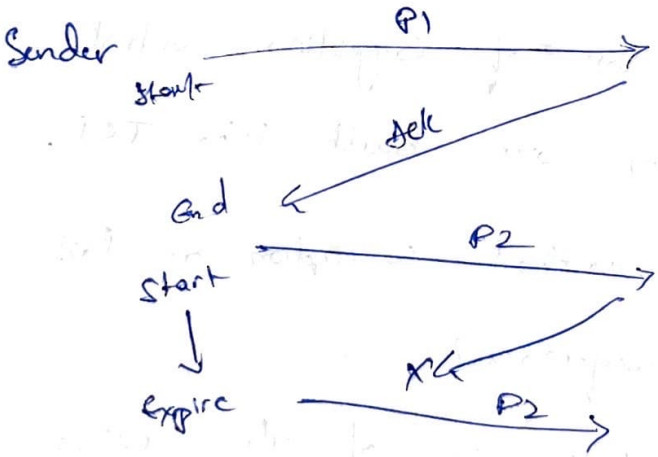
* what if packets/data are lost?

Timers :- Retransmission timer.

(1)



Expires after a while, shows that packet was lost.



Has duplicate packets

this is solved by sequence no. (Duplicates will be ident

* User - Datagram Protocol ; (UDP)

- Data may/maynot be delivered.
- " " change
- " " not in order.

- It's a connectionless data
- UDP uses checksums but won't do anything in case of errors

UDP packet is 2^{16} size

- It contains source / dest. port no., checksum, length of datagram and data itself.
- UDP is faster, so used. ∴
- Video Conferencing, Gaming and DNS use UDP

tcpdump -c no-packets // command to see packets

TCP (Transmission Control Protocol):

- Transport layer protocol.
- App. layer sends lot of raw data, TCP divides it or segments into chunks, add headers, checksum etc

→ Congestion Control.

- when data doesn't deliver / arrive
- maintains order of the data.

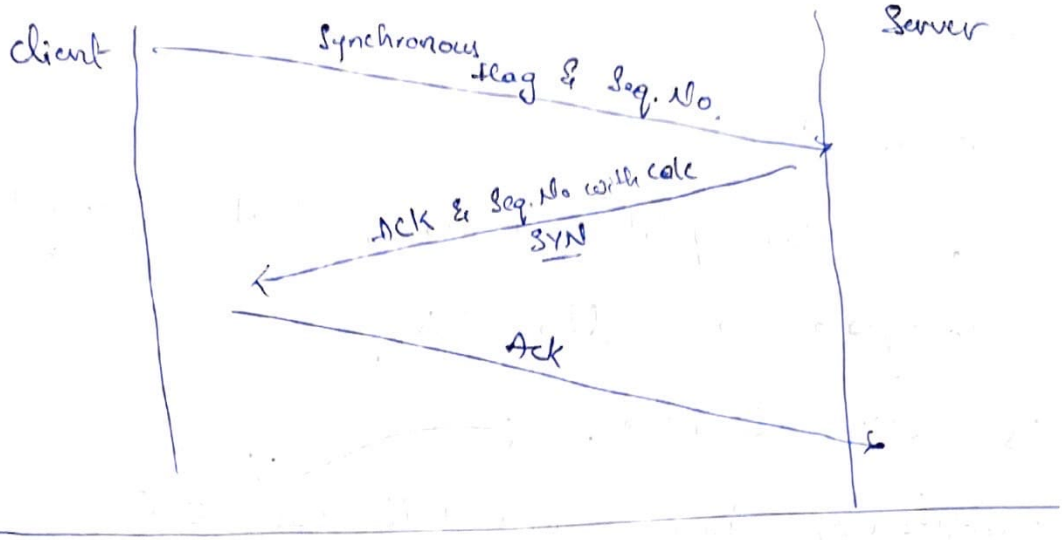
→ This is connection-oriented

→ Error control

→ Full duplex (sending files back forth simultaneously)

- One TCP between 2 comps only.

✱ 3-way Handshake :



✱ Network Layer : It has data as packets.

- Here coc work with routers.
- Every router has a network address.
- Each router has routing (which has forwarding) table for distination address.

↳ This is called Hop-by-Hop Hoping from one to another router.

✱ Control plane in network layer is used to build these routing tables.

2 types of routing are used to create these tables :

(i) Static Routing : manually add (time consuming)

(ii) Dynamic " :- It evolves with changes.

* Internet Protocol (IP) is in network layer.

- We used IPV4 till now → 32 bit, 4 words
- IPV6 is future → 128 bits

192.68.52.3
↳ device Address (Host ID)
Network Address (Subnet ID)

- ISPs are given blocks of IP Addresses, this called Subnetting.

Classes of IP Addresses :-

A - 0.0.0.0 to 127.255.255.255
B - 128.0.0.0 to 191.255.255.255
C - 192.0.0.0 to 223.255.255.255
D - 224.0.0.0 to 239.255.255.255
E - 240.0.0.0 to 255.255.255.255

* Subnet masking :- It masks the network part of IP Address.

* Variable length Subnet :- Set your own length of subnet network.

Eg: 192.0.1.0 /24

* IETF assigns IPs to ISPs based on region.

* Packets:-

- 20 bytes is size of header excluding data.
- IPv, length, ID no., flag, checksum, Address, TTL etc

- Time-to-live (TTL):- Sometimes if packets keep flopping in loop, then the packet is dropped.

* IPv6:- It's 128 bytes (4* IPv4)

- Not backward compatible.
- Lot of efforts, shift / Hardware work.

Representation:-

a : a : a : a : a : a : a : a

↓

Hexadecimal (16 bit)

Eg:- ABFE : F001 : 3210 : 9182 : 0 : 0 : 1 : 3

- Subnetting can be done

* middleboxes:- Extra devices that also interact with IPs

- Lies on network / transport layer.

① Firewall (A middlebox):- Global / your trusted network

- It can filter IP packets based on various rules
  - → Address
  - → flag
  - → Protocols
  - → Port nos
  - → modify packets

Types : — Stateless (no state maintained)
— Stateful (stores in cache, efficient)

⑤ Network Address Translation : (NAT)

- A method to modify/map an IP Address into another making the network ID's from network private.
- Can have private IP Address.

★ Data-Link layer : • It transfers in frames.

- Responsible to send data/packets over a physical links. or b/w connected devices.
- DHCP (Dynamic Host Configuration) server has pool of IP Address, so when new device connects, it assigns a new Address.
- Data-link layer Address. (Can be manually allocated or dynamically).
- Frames contain DLLA & Dest. IP Address.