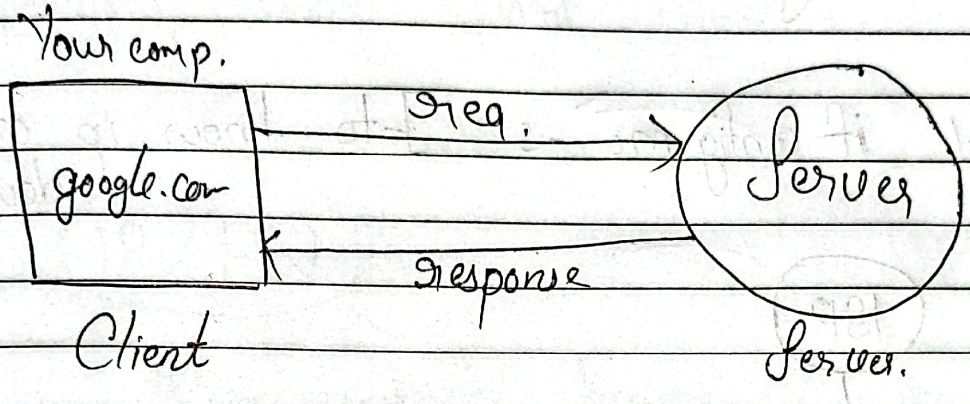
Computer Networking.

TCP → Transmission Control Protocol.



Protocols - They are just rules and regulations that are set by the Internet Society.

(*) TCP - It will ensure that the data will reach its destination without being corrupted on the way.

(User datagram)

(*) UDP → No need to send 100% of the data.

(*) HTTP → Hypes Text transfer protocols.

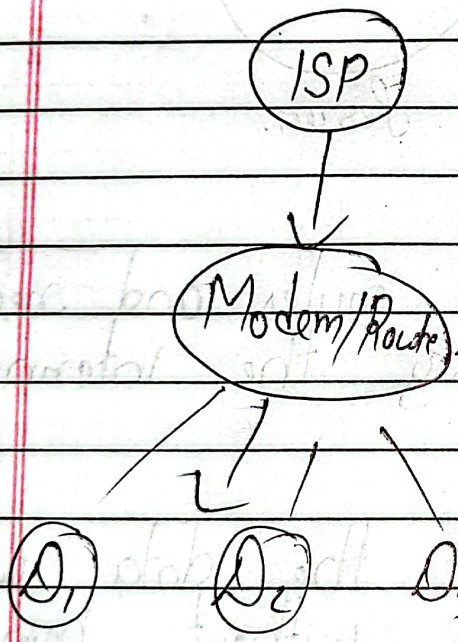
→ used by web browsers.

→ format of data being sent by webclients

IP address => X.X.X.X (format)

every X can have value (0-255)

curl if config.me -s] -to know ip of your device



→ Every device connected to this have same IP address. (Global)

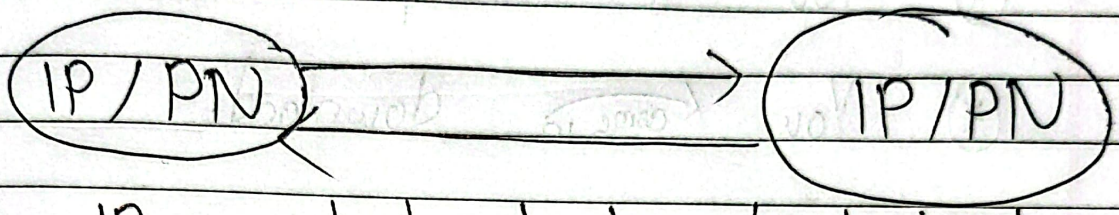
↳ They have different local IP address given by modem using DHCP Protocol.

Modem / Routes decides who was one one device sending requests using NAT (Network Access Translator)

16 bits → 16 cells → each cell 0 or 1

Total port number = 2^{16}

In a particular device, after getting the response from modem/router, the device decides from which application, it was requesting using port number.



IP ⇒ which location / which device.

PN ⇒ which application.

Ports are 16 bits number.

SQL = 1433

(All websites)

HTTP will happen on port - 80.

PN - (0 - 1023) ⇒ reserved ports.

(1024 - 49152) ⇒ registered for applications.

remainings ⇒ you can use.

1 mbps ⇒ 1 mega. bits per second

10^6 bit of data can be sent per second.

* You go out → upload
* You come in → download.

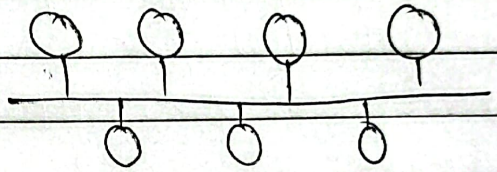
Internet
- LAN → Local Area Network. (network adapter)
  via ethernet cables, wifi
- MAN → Metropolitan Area Network (City)
- WAN → Wide Area Network (Countries)
  via optical fibre cables,
  - ① SONET ⇒ Synchronus Optical Networking
    It basically carries the data using optical fibre network. Hence, it can ~~carries~~ covers the longer distances.
  - ② Frome relay ⇒ Way to connect your LAN to wider network.

Modem → converts digital signal into analog and vice versa.

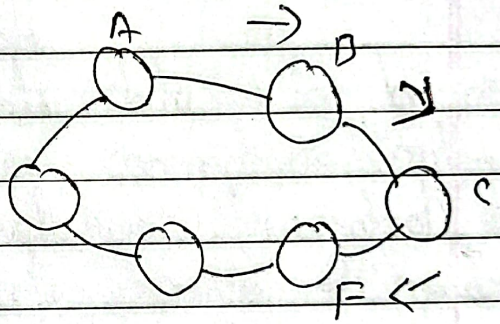
Router → device that routes the data package based on their IP address.

Topologies.

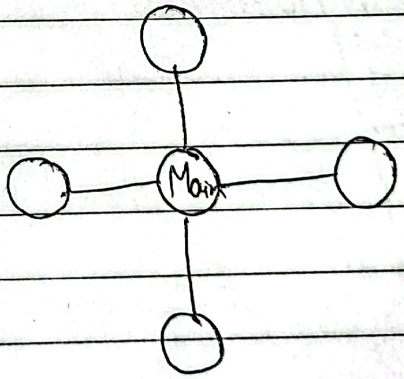
① Bus Topologies.



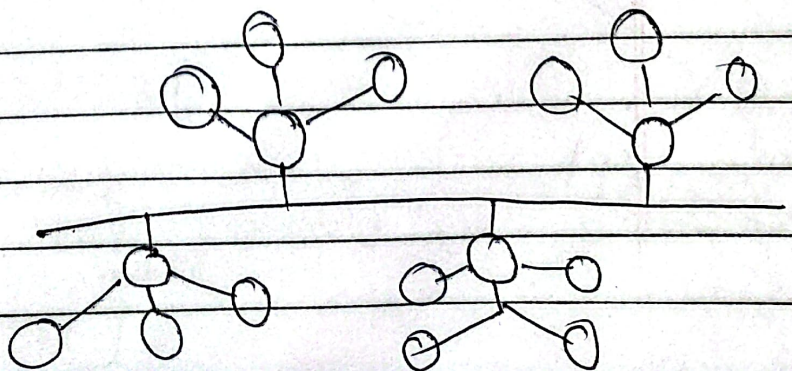
② Ring Topologies.



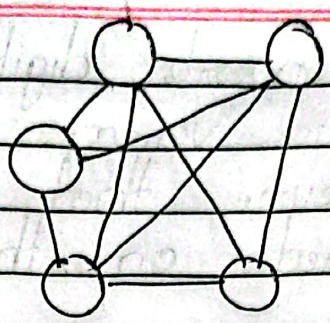
③ Star Topologies.



④ Tree Topologies.
(Bus - Star)



5. Mesh =>



- Exp.
- Scalability issue

Structure of the Network.

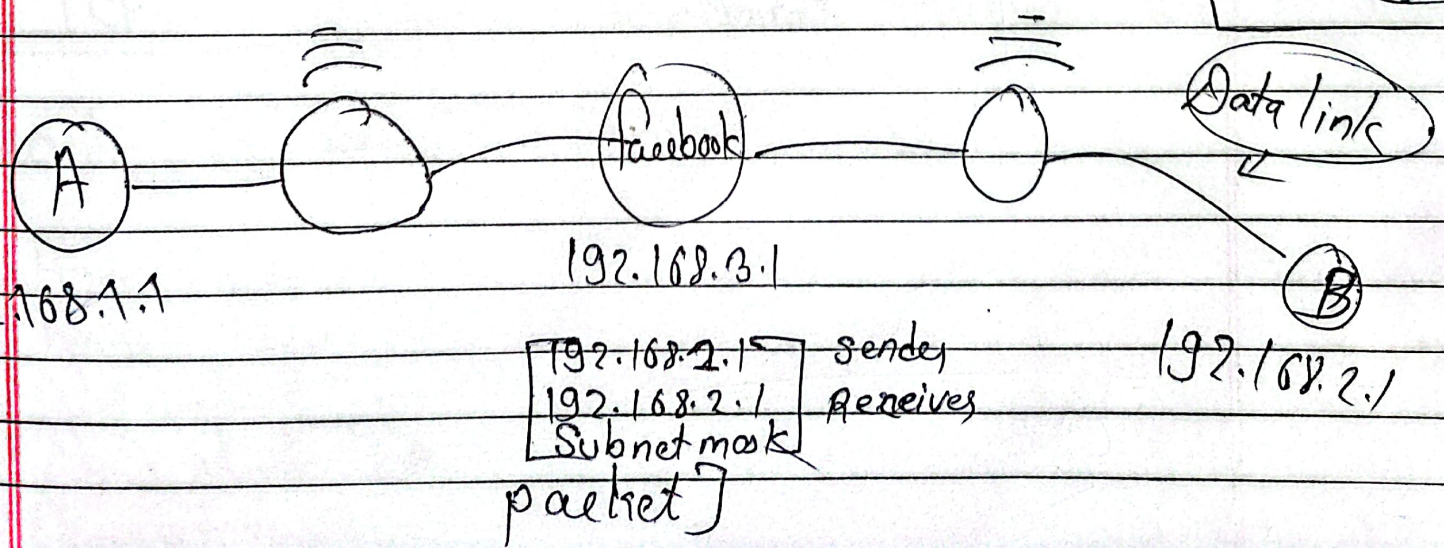
Application layer.

Imp

OSI Model.

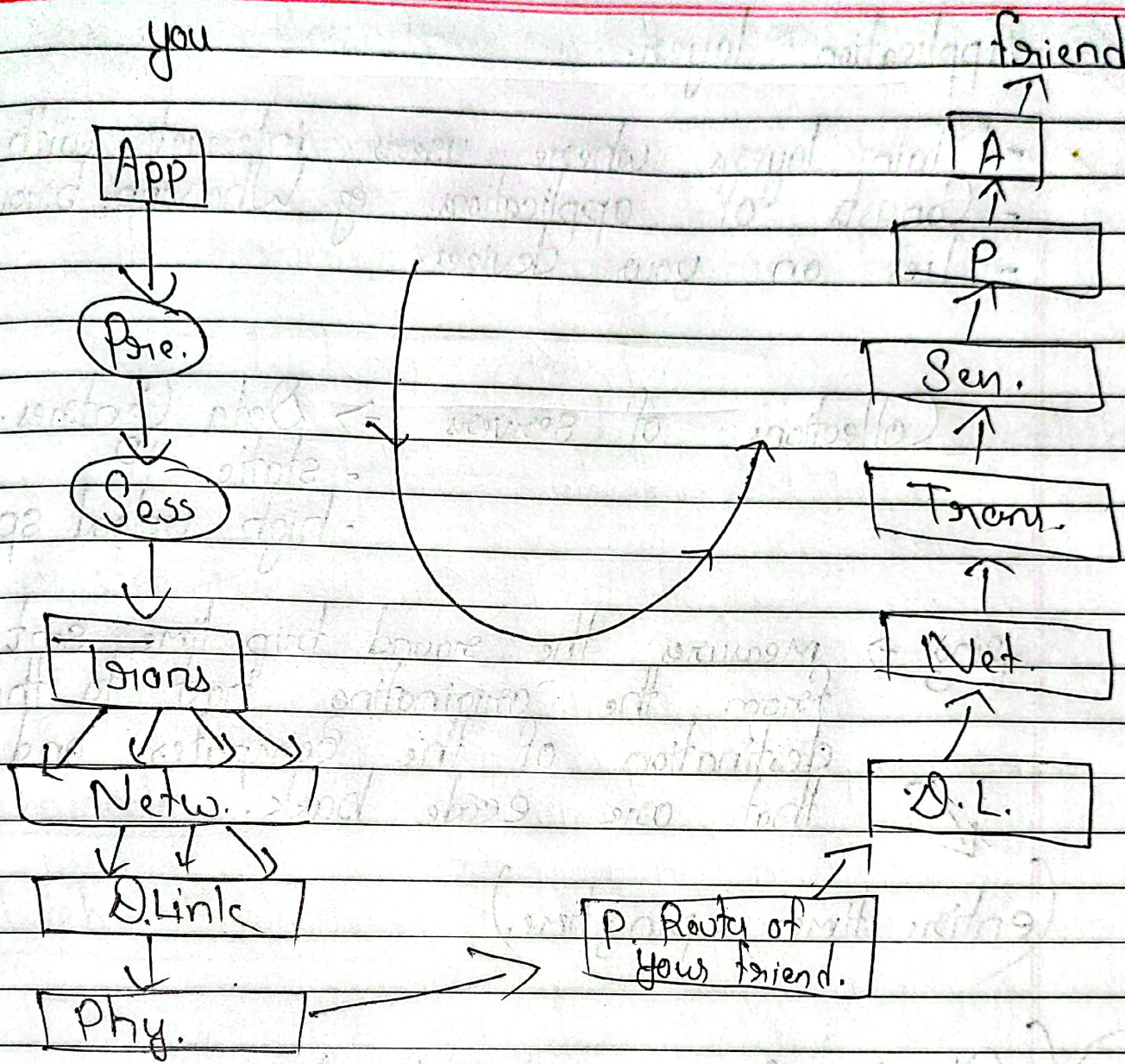
- Open System Inter-connection Model.
- Layers of OSI Model.

1. Application → software.
2. (SSL) Presentation → converts data into binary.
3. Session → enables connections, authentication
4. (checksum) Transport → segmentation, flow control, error cont.
5. Network → connects receiver & sender, logical add
6. Data Link → allows dir. communicate with hosts, log. add. load balancing
7. Physical



Mac address is a 12 digit alpha-numeric number of the network interface of the computer.

Computer doesn't have only one Mac address. Its bluetooth may have one and its wifi may have another.



TCP/IP Model. (Internet Protocal suit).

only 5 layes

- App. layes
- Trans. layes
- Network
- Data Link
- Physical

TTL = Time to live.

1 (*) Application layer.

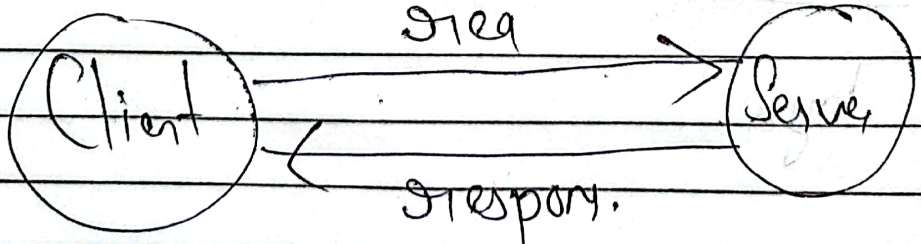
- Main layers where users interact with.
- Consists of applications. eg. Whatsapp, browsers
- lies on your devices.

Collections of servers => Data Centres.
- static IP
- high upload speed.

ping => Measures the round trip time sent from the originating host to the destination of the computer and that are ecode back.

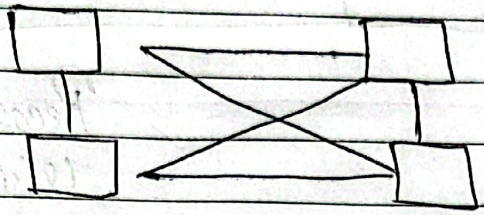
(entire time = ping time.)

(#) (Client - Server Architecture)



(#) (Peer to Peer Architecture) (P2P)

→ Various application that are being communicated, they get connected with each other. No servers, datacentres in between.



(easily scallable)

→ decentralized
→ everyone is client and server itself.

(#) Protocols.

Web Protocols.

TCP/IP:

- (*) HTTP
- (*) DHCP.
- (*) FTP → Files Transfer.
- (*) SMTP → Simple Mail Transfer.
- (*) POP3 → To receive mails
  IMap

SSH → Secure Shell.
VNC → Virtual Network Comp.
Telnet → Terminal emulation that enables users to connect to remote host that uses telent

PN→23 Outside:

(*) UDP → stateless connection.

Q. • How do these applications talks to each other?

Program: Whatsapp.

Process → Sending message, Recording video

Thread is a lighter version of a process.

Process → Recording a video.
Thread → Open cam + record + save.

Addresses → IP / Ports

Sockets : interface between process & internet

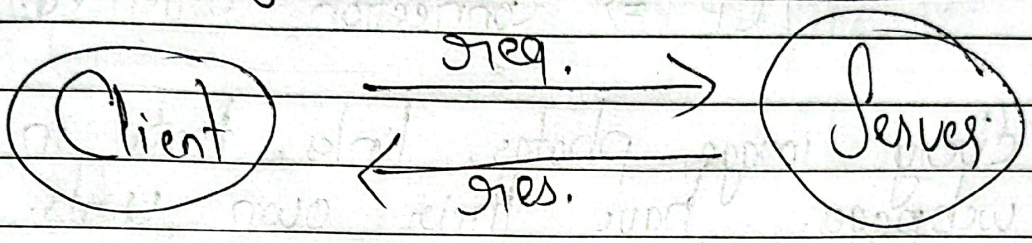
Ports : IP → device
Ports → app.

But inside one app also there may be diff. places. Like in google chrome, you want to send to which tab. How to determine.

For this we have, ephemeral ports.

Ephemeral ports exists on the client side but servers must have that number.

HTTP (Hypes Text Transfer Protocol)



This happens using http protocols.

⇒ HTTP is a client server protocol which tells us how you requests the data ~~the~~ from the servers and also how the server will sent back data to the ~~server~~ client.

HTTP req. → Client
HTTP response → Server.

Get something → get req.
post something → post req.

HTTP uses TCP (no data lon).
↳ stateless protocol (Server doesn't store by default)

(No data store of client)

If the same client is making a same req. again and again, the server will not think it as a same client.

- HTTP is in the application layer and it uses TCP in the transport layer.

TCP ⇒ Connection oriented.

Every image, photos, links, texts in the webpages have their own urls.

https://www.google.com/images/google ....

Protocol — Url — link to the resource.

After ? ⇒ arguments

HTTP Methods.

Method is something that tells the server basically what to do.

1. Get: → req.
2. Post: → Giving something
3. Put: → puts data at specific loc.
4. Delete:

Status code.

When you make a req. to the server, you need something to know whether the code executed successfully or failed. For that there exist status code.

E.g. ⇒ 200 → successfull
404 → failed
400 → bad req.
500 → Internal server error.

Class: 1xx → Informational category
2xx → success
3xx → redirecting
4xx → Client error
5xx → Server error.

=> (file)

* Cookies:

Cookie is a unique string which is stored in the client browser

When you visit a website for the first time, it will setup a cookie. And after that whenever you revisit again, in that request's header a cookie will be sent. Then the server will know and check the database and will know who is requesting.

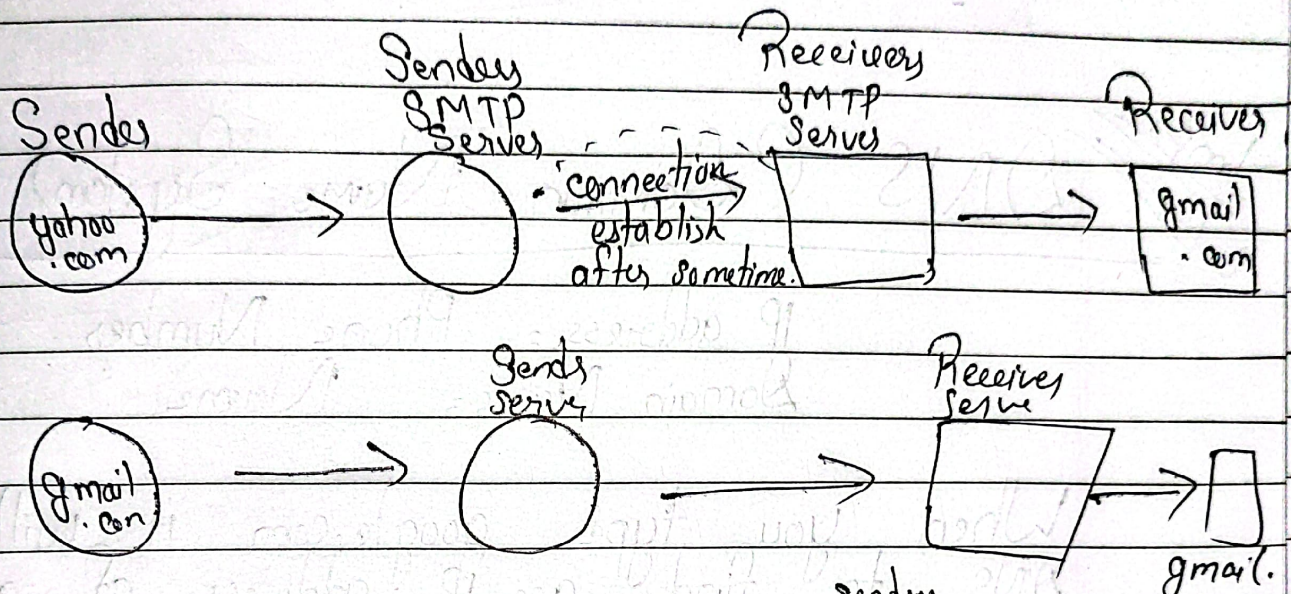
Cookies are misused and you can be tracked.

* Third party cookie.

They are set for the URLs that you donot visit.

How Email Works. → TCP for transport.

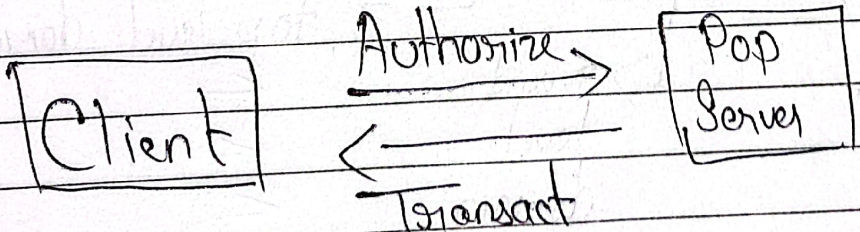
SMTP → for sending
POP3 → for receiving.



⊖ When receiver is offline, the server will keep trying for few days before it gives up.

-type=mx gmail.com [to know SMTP servers of gmail]

POP → Post office Protocol. (Port 110)



Drafts, sent items are not synced here.

Ⓐ IMAP → Internet Message Access Protocol.

- allows you to view your emails on multiple devices.

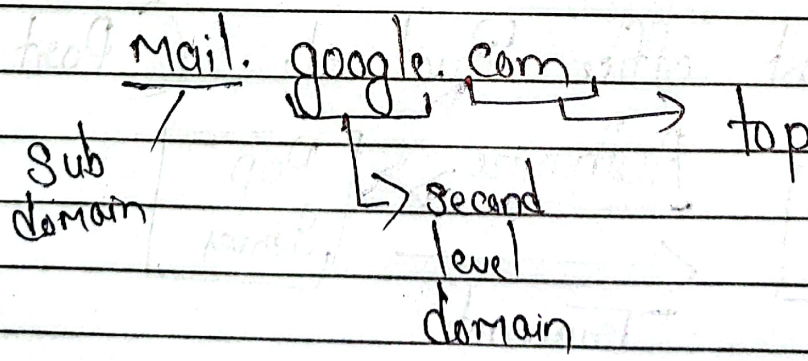
DNS (Domain Name System).

IP address = Phone Numbes
Domain Name = Name

When you type google.com it will use DNS to find ~~geo~~ IP address of google's server.

DNS is like a database which have all ip address & domain name.

DNS is ~~dat~~ classified into classes



(Top level)

@ Root DNS Servers - First point of contact after local ~~server~~ system for query and stuffs when you write a domain name.

(b) => .io , .org , .com ↳ google.com.
student.io
commclassroom.org

Some companies have their own top level domains.

ICANN (~~registers~~) manages these domains.

(*) What happens when you type google.com.

(Your comp.)

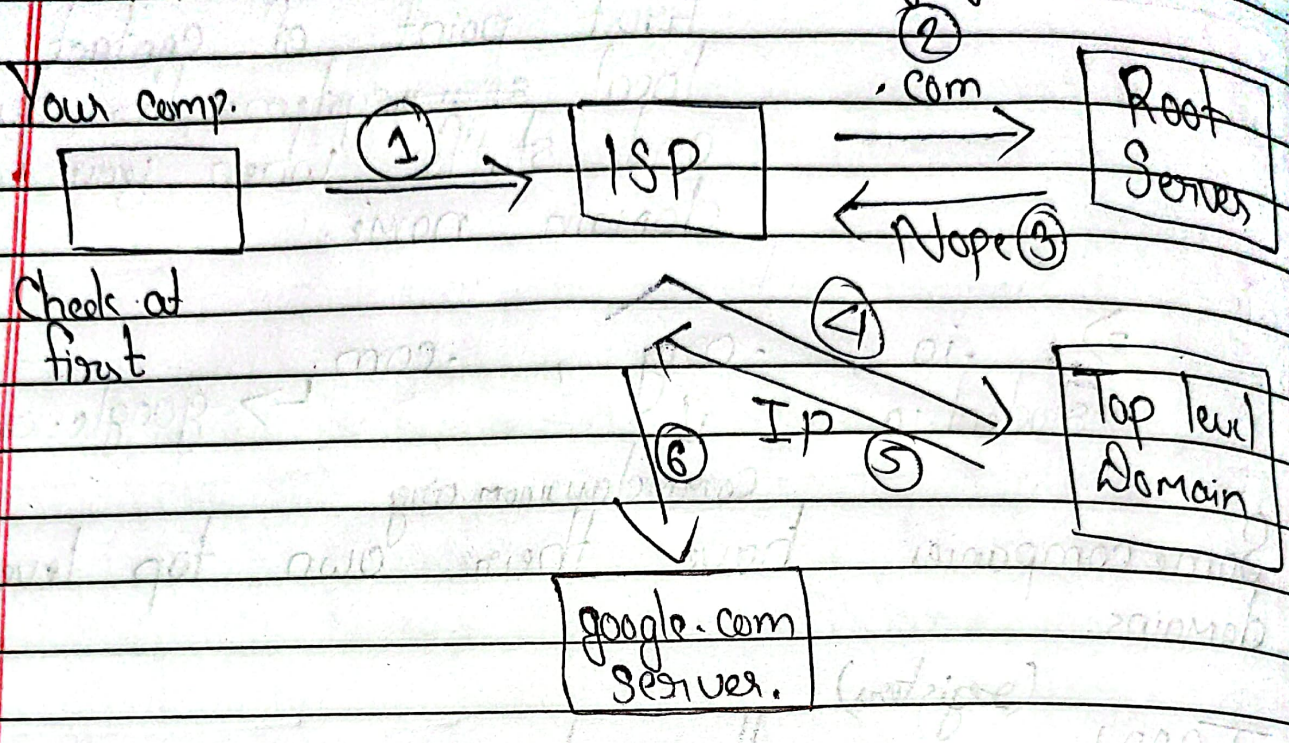
- Check in your own comp. whether you have logged in before or not. It is easier if you have logged in earlier coz it ~~s~~stores cache.

If not - Check in local DNS Servers. (like ISP)

If not - Check .com in the root ~~DNS~~ server.

If not - Check in the top level domain. If found it will give you the IP address. to ISP.

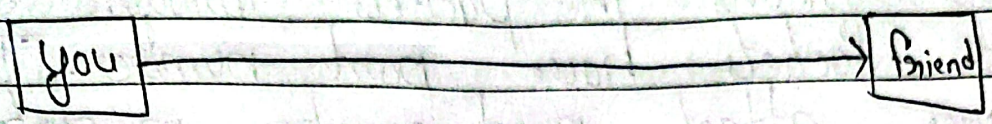
- ISP will connect with google.com



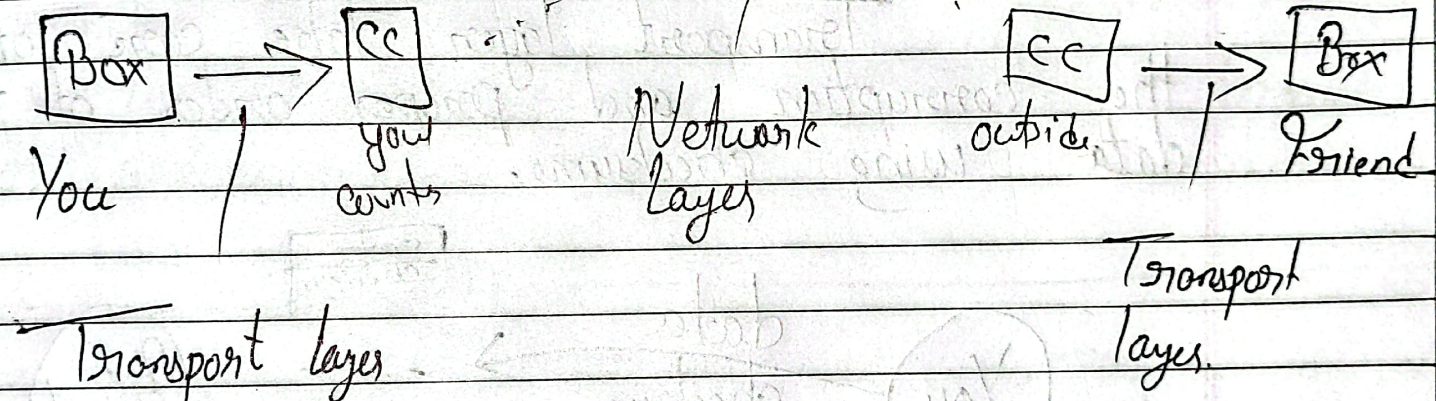
Can't buy domain name, only rent.

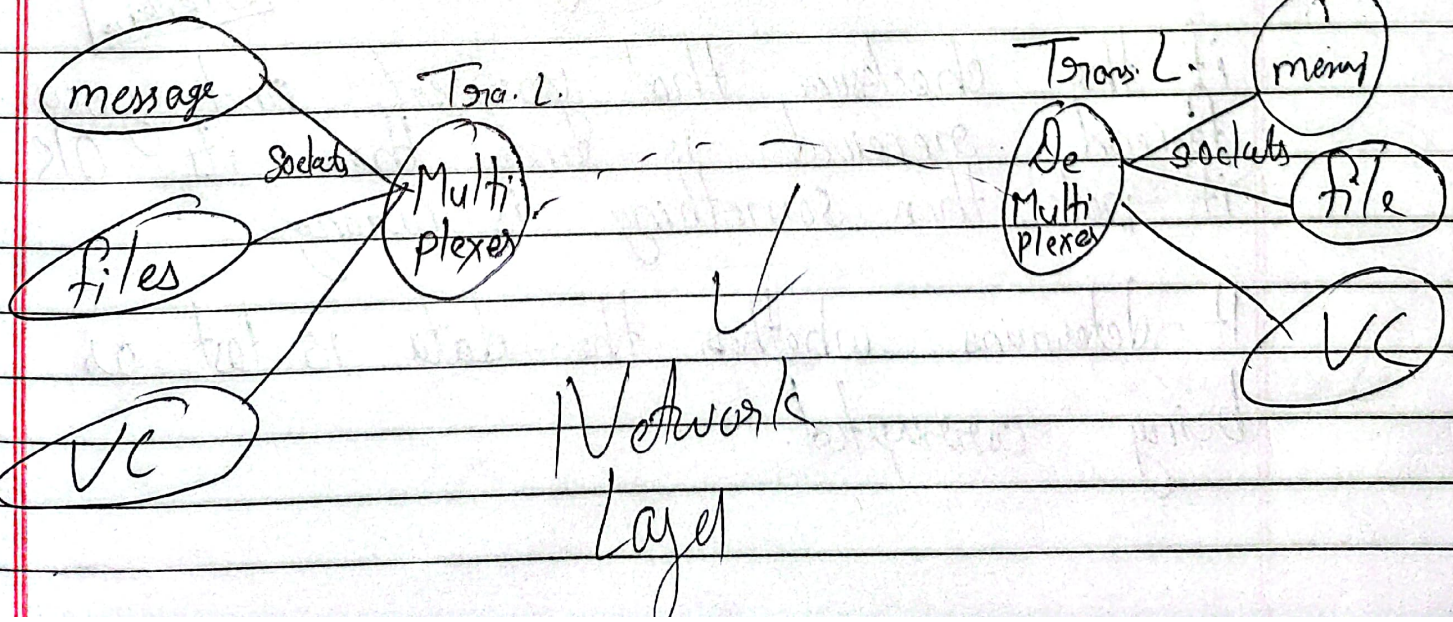
~ dig google.com ⇒ to see messages received by the DNS servers.

Transport layer.



The transport of message from you to your friend is done by the network layer. Then what is the work of transport layer. Transport layer lies inside your device and its work is to take messages from network and send it to the application and vice versa.





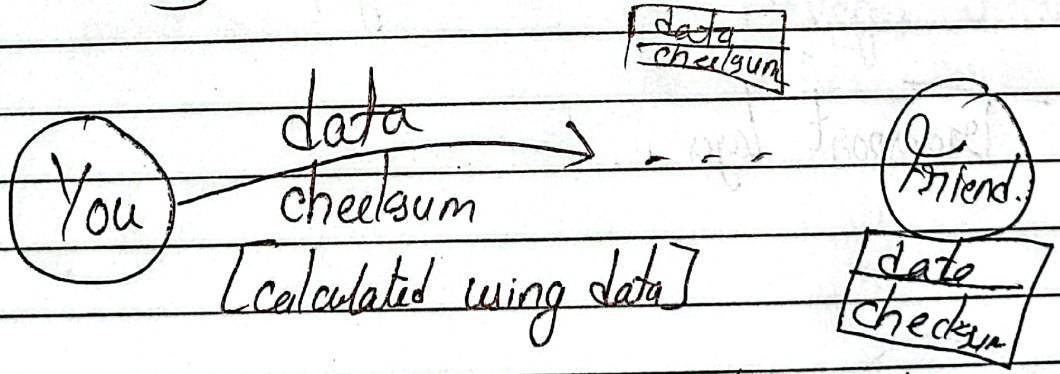
Data travels in packets through sockets and Sockets have port number. Transport layer attach port numbes in data such that it reaches destination.

Transport layer also takes care of confestion control (Traffic)

Conjections controls algorithms built in TCP.

Checksums:

Transport layer take care of the corruption and proper order of the data using checksums.

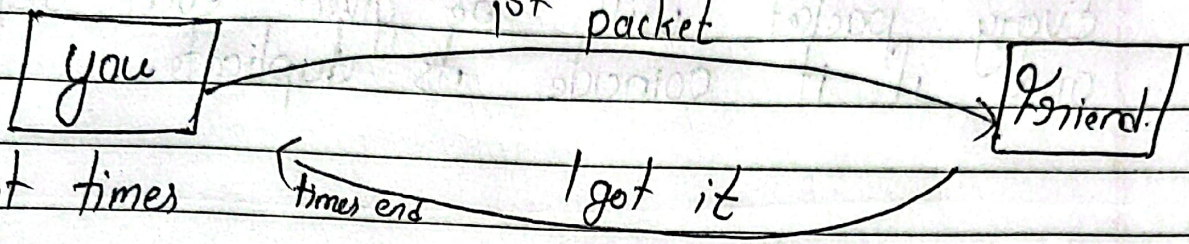


if the checksum that you sent and your friend received is same then it's OK if not then something is wrong.

It determins whether the data is lost or being corrupted.

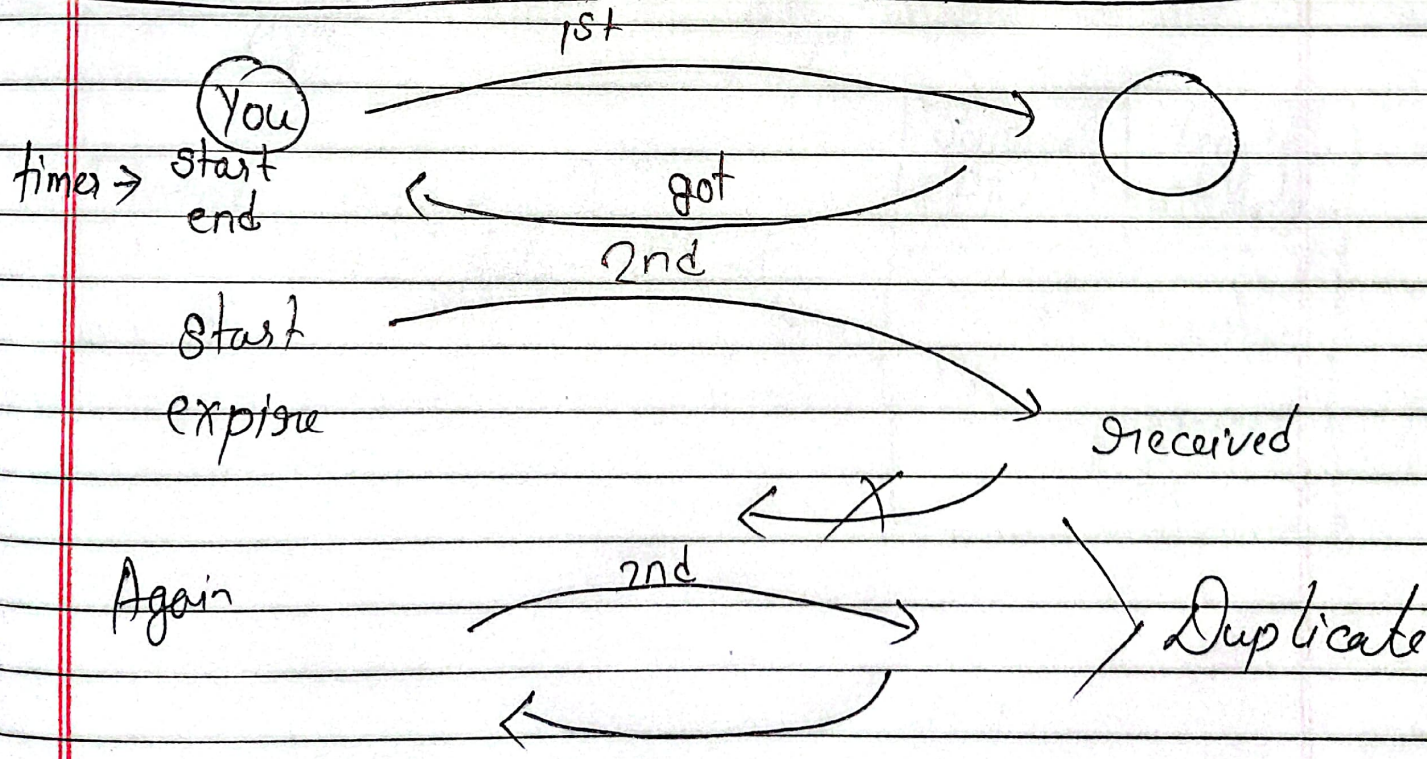
(*) How would I know that the packets that I sent is received by my friend.

⇒ Here come into place Timers.



timer start — another packet — Not received

times will expire and you will know data is not sent. This is known as retransmission times.



Now friend have 2 packet no. 2. Duplicates.

We solve this problem using sequence number Every packet will be given each seq. no and if it coincide its duplicate.

Protocols.

UDP → User Datagram Protocol.

★ Data may or maynot be delievered or changed on the way, may not be in orderh.

- Connection less protocol.

UDP uses checksums. You will know data is corrupted or not. But UDP leaves as it is.

UDP Packets.

Packkets have Source PN, Destination PN, length of datagram, checksum, data.

2byte 2byte 2byte 2byte

Total size = 2^{16}

" size of data = $2^{16} - 8$.

Source PN	length of Datagm
Destination PN	Checksum
Data	

sudo tcpdump

UDP is faster than TCP. So we use it.

Use Cases of UDP ⇒ fast
- Video Conferencing
- DNS.
- Gaming.

TCP (Transmission Control Protocols)

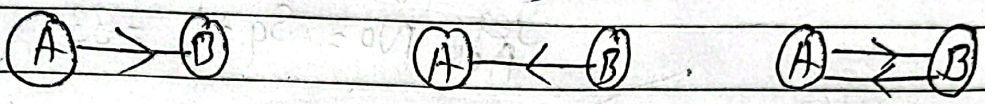
- App. layers sends lots of raw data. TCP segments this data, add headers, checksums and many more. It may also collect the data from network layers. At the receiver end, those data which were segmented from the network layers.

- Conjection Control.

Takes cares of ⇒ When data doesn't arrive
⇒ maintains the order.

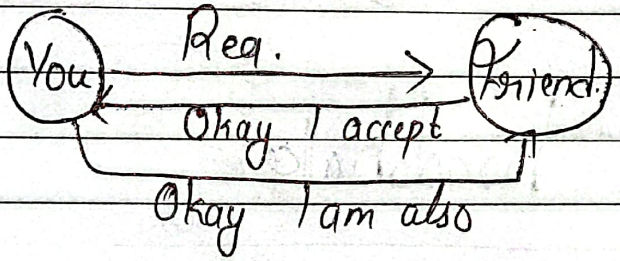
Features.

- Connection oriented. [First make connection, only data send].
- Error control.
- Conjection control.
- Bidirection (Full Duplex).

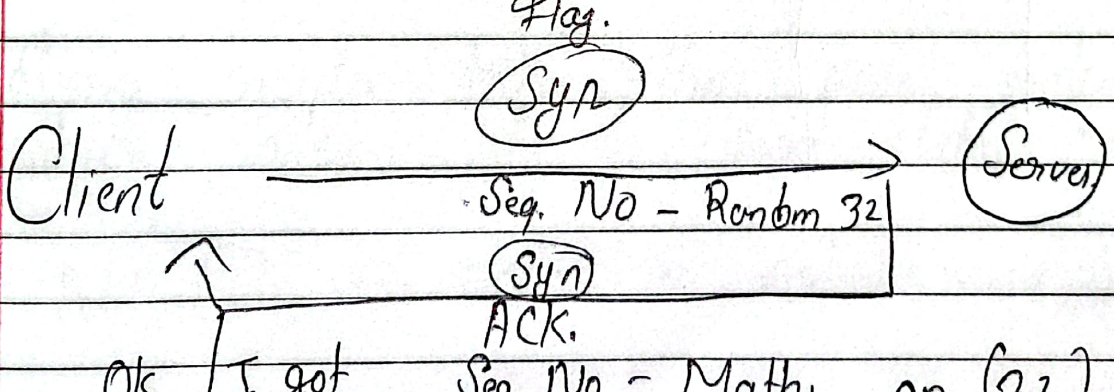


One TCP connection, only between two computers.

Seq. numbers + Acknowledgement numbers.



3-way handshake.



Seq. No are random for sec. purposes

Client

Syn Flag
Random Seq. No 32

Server

Syn & Acknowlegement
No - (Maths on Rano) 32 = 56
Seq. Ack No = Sq + 1 = 33

Ack. Seq. No
32+1 = 33
Ack No = 56+1
= 57

3 - way handshake.

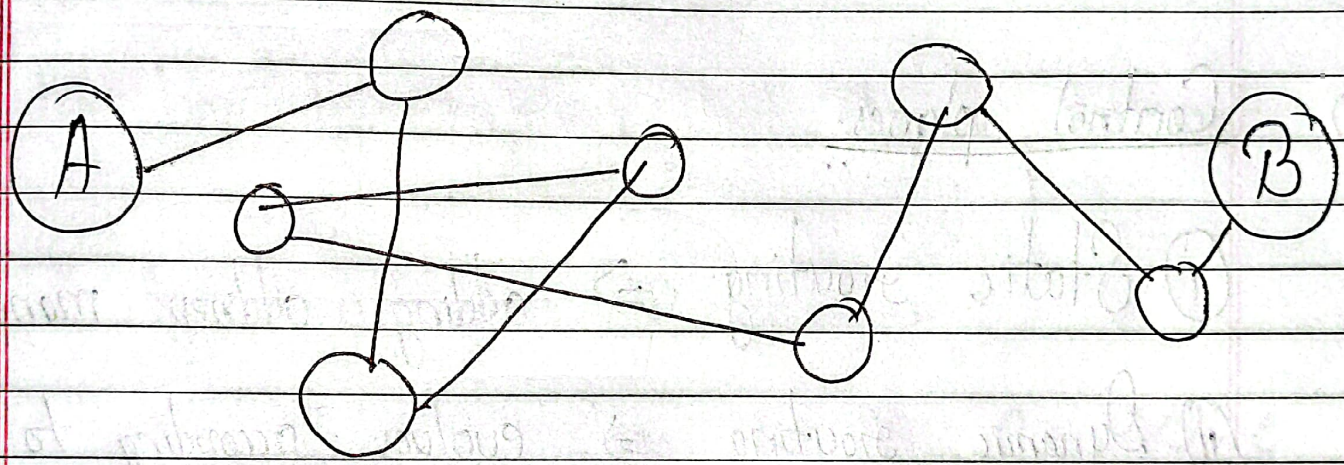
Network Layer.

Transport → Segments

Network → Packets

Data Link → Frames.

Here we work with Router. This is where data travels.



Many routers connected to each other.

Every routes have their own network address. which allows us to send data packets

NP-① → check in ~~networ~~ routing table

Forwarding table. (inside router)

Hop by Hop Forwarding.

192.168.2.30

Network address (Sub-net id)

device address. (host id)

Who creates these types of tables

⇒ Control planes.

(i) Static routing ⇒ adding address manually

(ii) Dynamic routing ⇒ evolves according to changes.

Imp

Ver - 4
Ver - 6

Classmate
Date
Page

Internet Protocol (IP addresses).

IPV4 → 32 bit numbers with 4 words.

IPV6 → (future) 128 bits

5.6.9.14 (8 bit number) — 8×4 = 32

00000101 = 5 (8 bits)

In hop by hop → first know subnet then only host.

(5) 3 types of class IP address.

- Class A, B, C, D, E

0.0.0.0

A ⇒ 127.255.255.255

B ⇒ 128.0.0.0 ⇒ 191.255.255.255

C ⇒ 192.0.0.0 ⇒ 223.255.255.255

D ⇒ 224.0.0.0

E ⇒ 240.0.0.0

Subnet masking.

It means that the subnet mask is going to mask the network part of the IP address and it will leave for us to use the host part

Variable length Subnet.

12.0.0.0 / 31 (31 parts are subnets)

192.168.0.0 /24 [24 parts are subnet]

=> 192.168.0 will not change (24 bits)
only last 8 can be change.

IETF assign IP address based on regions.

Reserved IP addres => 127.0.0.0 / 8

Eg => local host => 127.0.0.1

Loopback addresses.

Packets.

Header is of 20 bytes.

- IP version
- total length
- Identification no.
- flags, checksums
- addresses, etc.
- TTL ⇒ Time to live.

During hoping, if the packets doesn't reached after certain numbes of hops, it will drop. No. of hops after it will drop = TTL.

DHCP ⇒

IPV4 ⇒ 32 bits = 2^{32} uniq IP address

IPV6 ⇒ 128 bit = 2^{128} uniq. IP addrs

Cons IPV6 : not backward compatible.
: ISP would have to shift. Lots of hardware works.

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

16 bits (Hexa decimal)

(16 x 8 = 128)

fixed.

IPV6 Eg: 12A : 1CF : 2 : 0 : 1 : AE1 : ABC : 3AE

IPV4 Eg: 127 : 1 . 2 : 4 /8 (prefix)

1 : 0 : 0 : 0 : 9 = 1 : 000 : 000 : ... : 9

⇒ 1 : : 9

Middle Boxes.

One of the thing that interacts with the network packets that come from the network.

Extra devices that also reacts with the IP Packets. in NL also in TL.

① Firewall : → Global network

→ Your trusted network

Provides filters

Stateless firewall vs Stateful firewall
↳ see the packets and (store in cache) maintains the state
→ more effivent

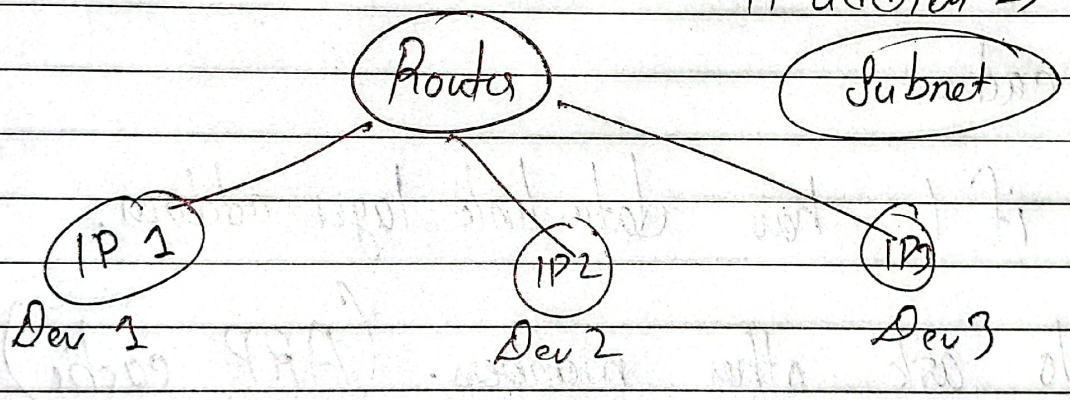
This is in network. [hosted by host network.

NAT

(*) TCP (DATA LINK LAYER).

IP address → ISP.

DHCP



DHCP → Dynamic Host Configuration Protocol.

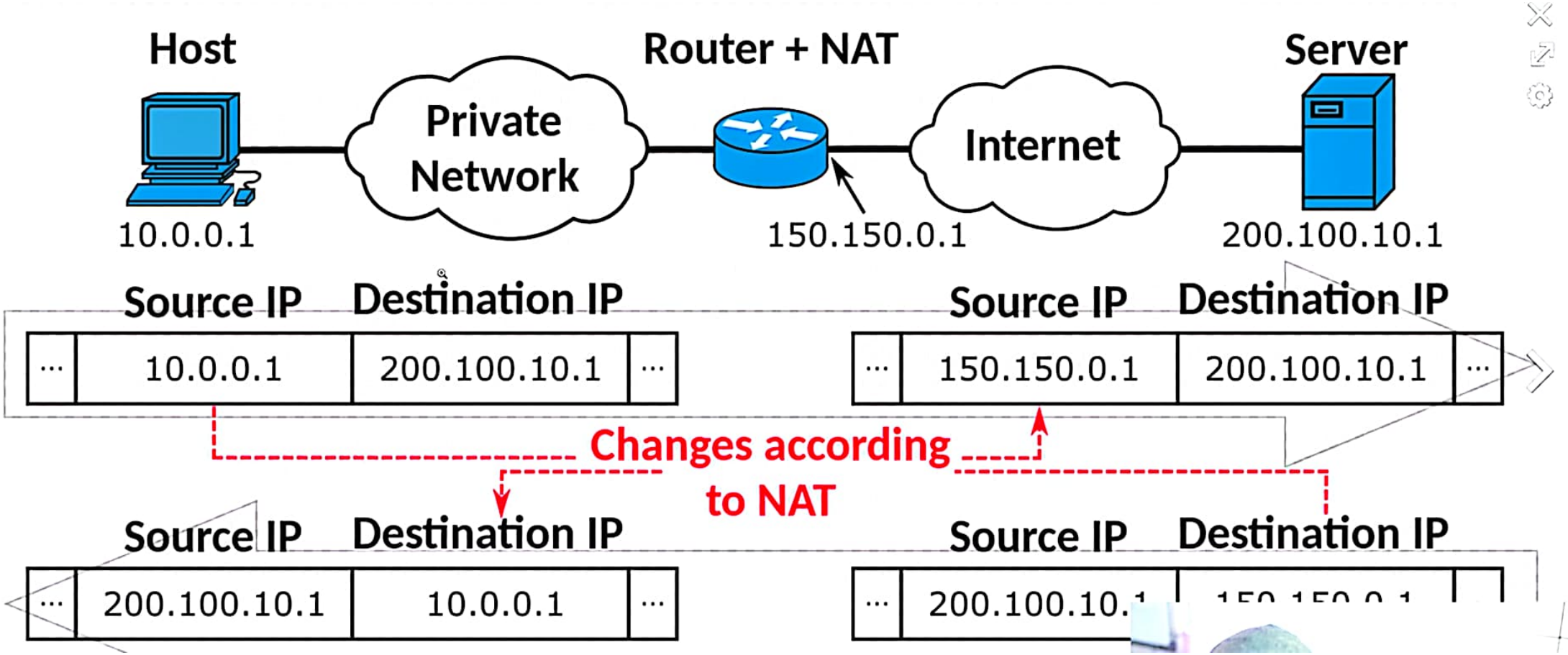
When new device got added to Router,

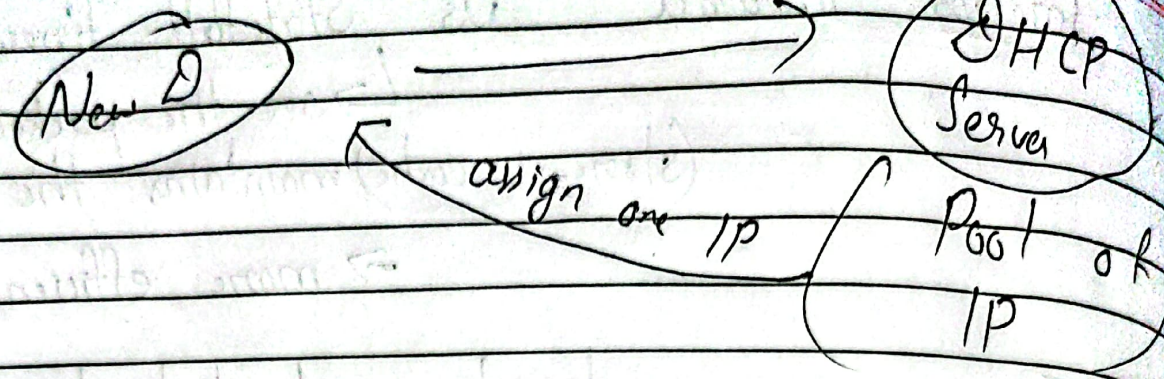
Network address translation

From Wikipedia, the free encyclopedia

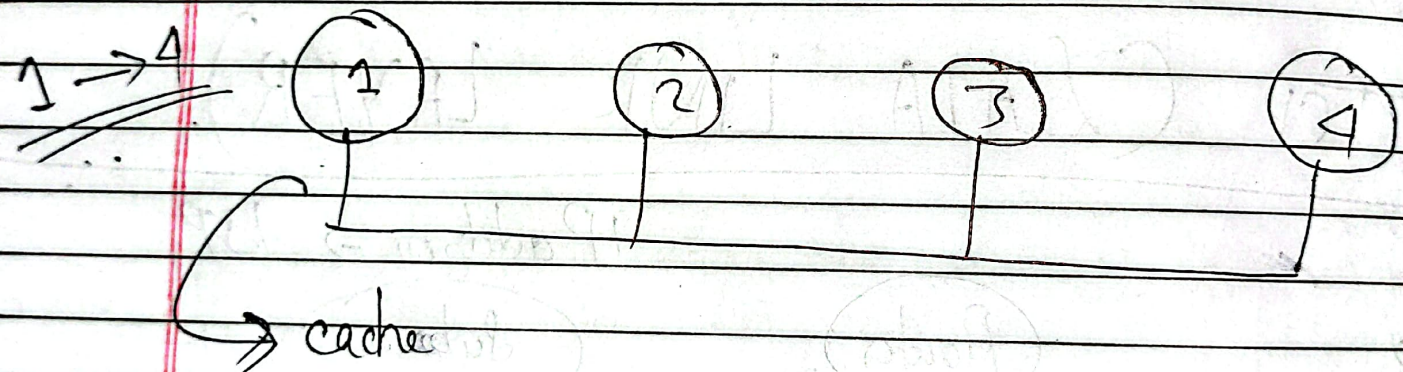
Network address translation (NAT) is a method of mapping an IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device.[1] The technique was originally used to avoid the need to assign a new address to every host when a network was moved, or when the upstream Internet service provider was replaced, but could not route the network's address space. It has become a popular and essential tool in conserving global address space in the face of IPv4 address exhaustion. One Internet-routable IP address of a NAT gateway can be used for an entire private network.[2]

As network address translation modifies the IP address information in packets, NAT implementations may vary in their specific behavior in various addressing cases and their effect on network traffic. The specifics of NAT behavior are not commonly documented by vendors of equipment containing NAT implementations.[2]


Host
Router + NAT
Server
Private Network
Internet
10.0.0.1
150.150.0.1
200.100.10.1
Source IP
Destination IP
10.0.0.1
200.100.10.1
Source IP
Destination IP
150.150.0.1
200.100.10.1
Changes according to NAT
Source IP
Destination IP
200.100.10.1
10.0.0.1
Source IP
Destination IP
200.100.10.1

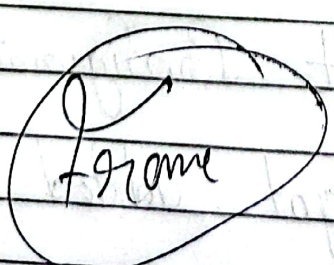


(*) Even though 2 computers are connected in same ISP or whatever, but at the data link layer they communicate each other using data link layer address.



Ask if I have data link layer address.

If No ask other members. (ARP cache)



ARP => Address Resolution Protocol

Frame contains DLL Address Sender and IP address of destination.

DLL Address = Mac address

Mac address is not of device. Its of components.

(DLL) very close to Physical Layer.

Physical Layer falls into Hardware Part

- Mac address (Router will assign)