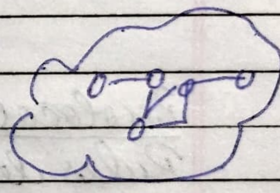
3) Computer Networking Full Course - OSI Model Deep Dive with Real Life Examples

Computer Networking Course

Computer ⇒ Commonly oriented Machine particularly used for training, education & Research

Network → computer connected (A) — (B)

Internet → network connected

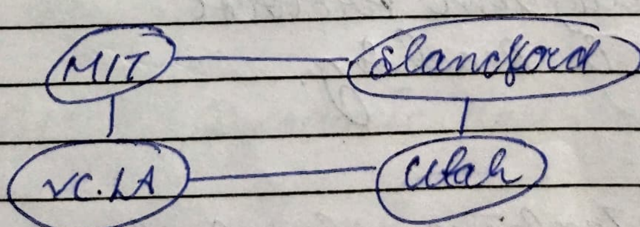


How did it start?

1957
→ submit
→ WON

US & Sovereign → Russia
Launching first satellite

ARPA
ARPAnet



(TCP) IP

Advanced
Research
Prop. Projects
Ajency

www → dim ~~vvvoc~~ Bwrnes
↳ hypermedia

info.cern.ch/hypertext/WWW/TheProject.html

Date
Page

<u>Protocols</u>
communication

Internet Society
RFC → Request for comments

<u>Client Server</u>

$\boxed{\text{google.com}}$ request → Server
client ← respone server

<u>Protocols</u>
Rules defined by Internet society

<u>TCP</u>
Transmission Control Protocol
→ Ensure that data will reach destination.

<u>UDP</u> → Userdatagram Protocol
Not all data is Reaching.
vedio cal

HTTP → Hyper Text Transfer Protocol
used by Web Browsers
Format of data between web clients and web servers. How? it will send

How data is Transferred?

Packets

IP Address

John → xxxxxxxxxx

Youtube

x.x.x.x → google.com

0-255

```
$ curl ifconfig.me -s
```

ISP

↓

Modem/Router → Global IP address

D1 IP1, D2 IP2, D3 IP3 } Local IP Address

DHCP → Dynamic Host ~~Control~~ Configuration Protocol

NAT → Network Access Translator

IP → device

Ports → Application
Port Number

IP/PN ⟷ IP/PN

Ports → 16 Bit Number
16 Cells → 0&1
$2^{16} \approx 65,000$

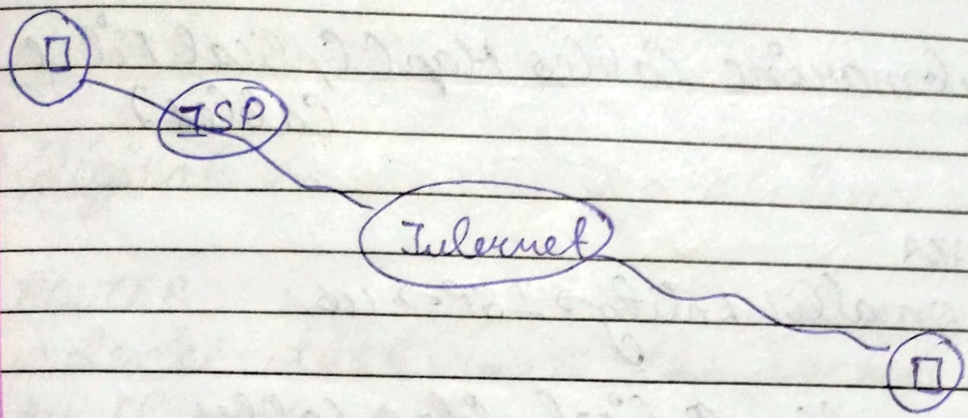
Web Pages → http → 80 Port

MongoDB = 27017
SQL = 1433

0 – 1023
Reserved ports

1024 – 49152
applications

Remaining one can use



1mbps = Megabits per second.
1000000 bits/s.

1gbps = 10^9 bits/s

1kbps = 1000 bits/s.

Upload → sending
Download → recieving

<u>Guided way</u>
Cables

Path already defined

<u>Unguided way</u>
Bluetooth,
wifi.

Communication is happening but no defined Path submarinecable.com

Submarine Cables Map (Optical Fibre Cables)

TATA
↳ smaller Entity → ISP → us.

Physically → Optical fibre, Cables
Coaxial Cables

Wireless → Radio Channels
Bluetooth
Wifi.
3G, 4G, LTE, 5G.

Faster → Cable (Then satellite)

LAN → Local Area Network
Small house/office
Ethernet Cable, wifi

Internet

MAN → Metropolitan Area Network
Across of City

WAN → Wide Area Network
Across Countries
Optical fibre Cables
① SONET
Synchronous Optical Networking
② Frame Relay
LAN to Internet

MODEM ROUTER

Digital to Analog signals & vice versa

ROUTER

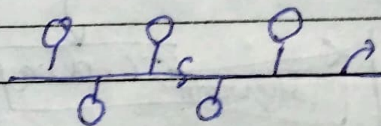
A device that routs the packets based on IP address

Tier 1 ISP → TATA

Tier 2 → Airtel, VI

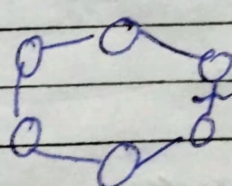
Topologies

BUS →



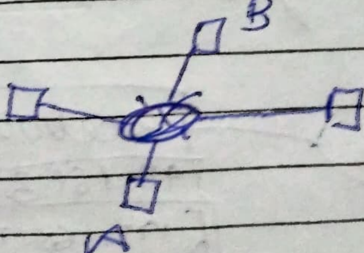
1 Person a time

RING

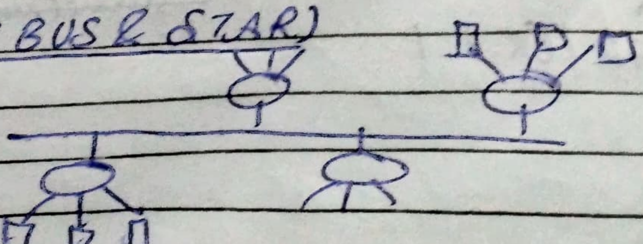


a lot of calls

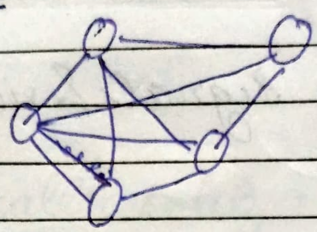
STAR



TREE (BUS & STAR)

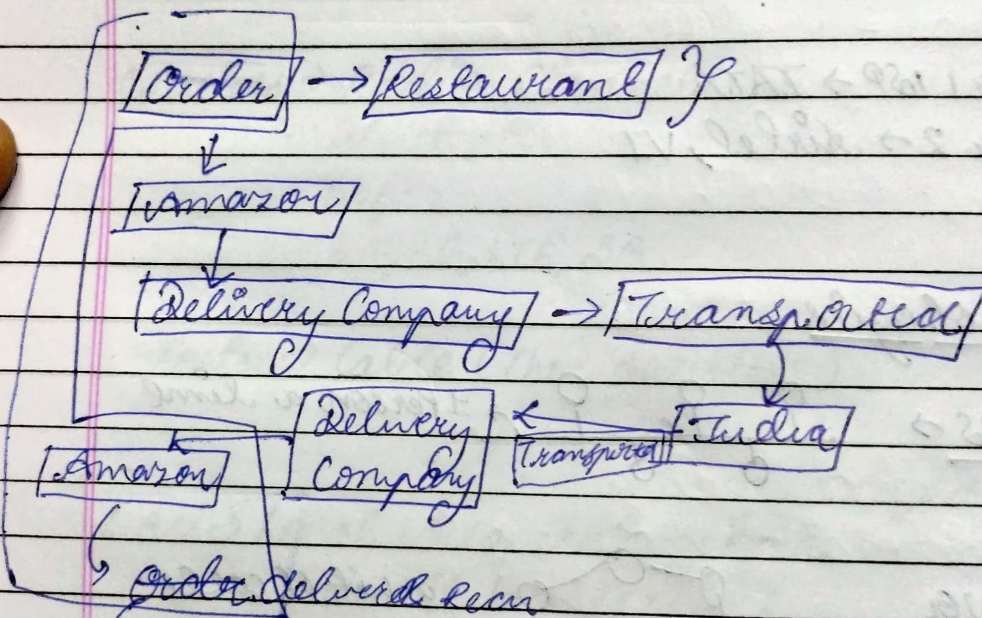


MESH

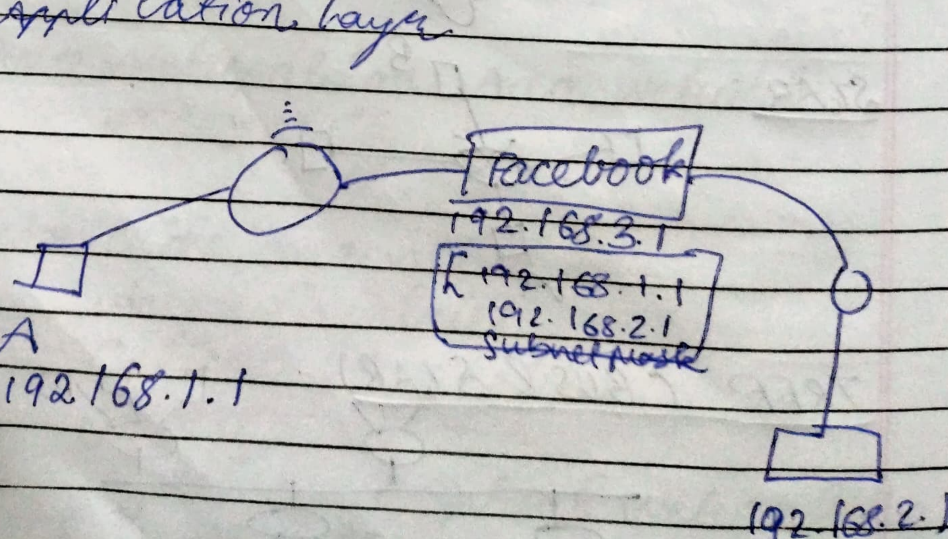


Expensive
Scability issues

Structure of Network



Application layer



You → Friend

OSI Model

Open system Interconnection Model

Application Layer → Software (Whatsapp, Mail)

Presentation → Machine Representation format
Encryption, Translation, Abstraction

Session → Authentication, Autherisation
(Username & PWD)

Transport → Segmentation (Protocols)
Segments in Ports, flow control, error control

Network → one to another computer in different network (Router)
(Logical addressing)

Data Link → Data Packet
directly communicate between computers

Physical
→ MAC Address

(TCP/IP Model)

Internet Protocol Suite (Developed by ARPA)

Application
↓
Transport
↓
Network
↓
Data Link
↓
Physical

Practically used

Aplication Layer

Users interact
Whatsapp, Browsers
where on: Devices
Protocols
Client-server Architecture.

Client $\xrightarrow{\text{request}}$ Server
Client $\xleftarrow{\text{response}}$ Server

ping google.com

~~Peer~~ P2P
Peer to Peer
Bittorrent.

Ping Time
Best Possible Time

Eg: college
scale it rapidly.
every single client & server

Protocols
Web Protocols:

- TCP/IP
  - HTTP
  - DHCP
  - FTP
- SMTP → send Mails
- POP3 & IMAC → Recieve Mails
- SSH
- VNC → Graphical Control
- Telnet : Port 23
- UDP : Stateless connection

Program: Whastapp

Send a MSG → Select the Page

Record of video → Open Camera

Process

Thread

Sockets → Messages from One system to Another
Interface between Internet & Process.

Ports:

Application

Ephemeral Ports

HTTP:

Hyper Text Transfer Protocol, Application Layer

How we will send
How we will recieve

TCP: Transport Layer
UDP
Stateless Protocol

HTTP URL Argument

Methods: Telling server what to do

GET : requesting
POST: Giving.
PUT : Puts data at Particular
DELETE: Delete

Error/ status codes

200 → successful
404 → couldn't Find
400 → Bad request
500 → Internal server Error

~~100~~ 1xx → Information
2xx → success
3xx → redirection
4xx → client error
5xx → server error

Cookies

Unique string
stored in my browser

set- cookie

Third Party cookies → cookies for URL
you do not visit.

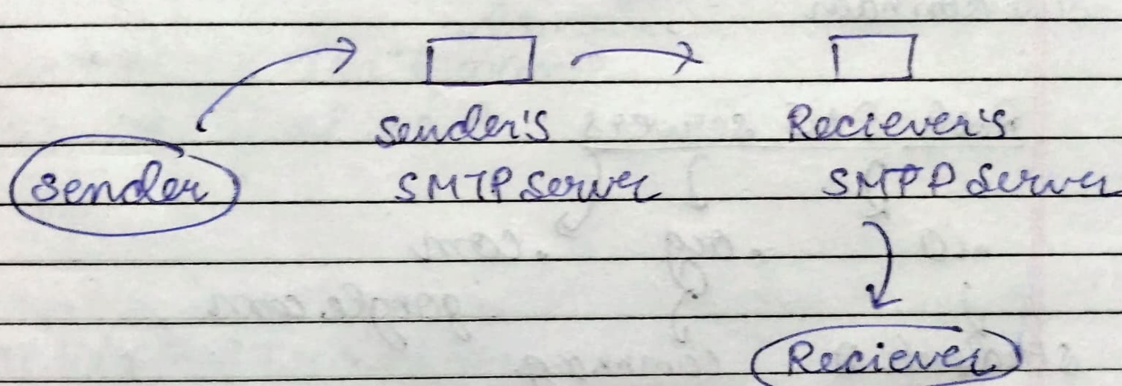
How Email works?

↱ Application layer

SMTP → Simple Mail Transfer Protocol

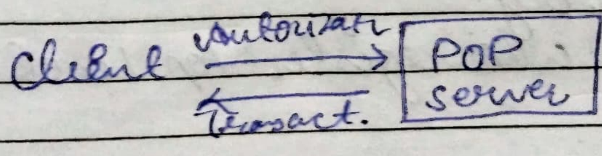
POP 3 → Revicieve

TCP protocol



:type=mx gmail.com

POP → Post office Protocol



Send items

Drafts are not sync in POP.

IMAP

Internet Message Access Protocol

View Mails on Multiple devics

DNS
Domain Name system.

Google.com → DNS do find IP address.

mail.google.com
second Level domain
Top Level domain.
sub-domain

Root DNS servers

.io .org .com
student.io comm.org google.com

commerical → .com
Non-profit → .org
Education → .edu
Country → .uk, .US, .in

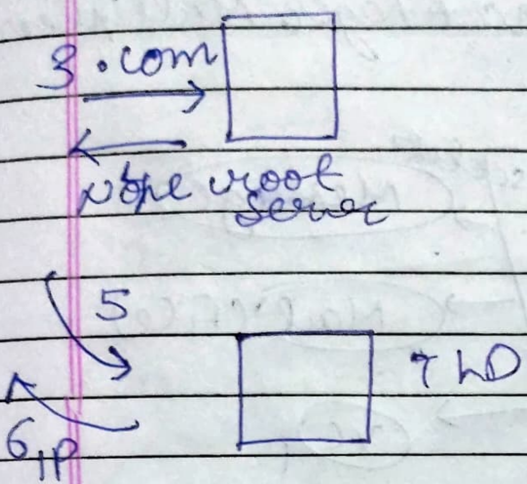
rootservers.org.

Ican.org
who register top level domains

(check in own computer)

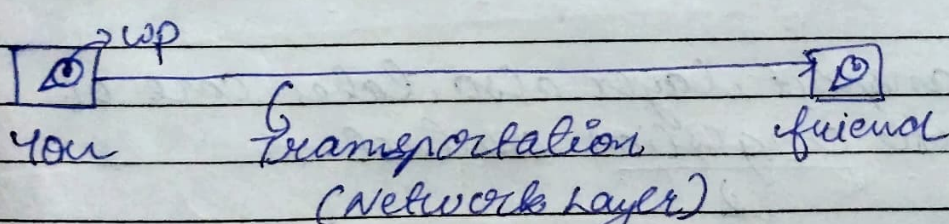
Local cache
IP will be stored → 1
They know everything
connected
7
2
Local DNS server (ISP)

n.dig

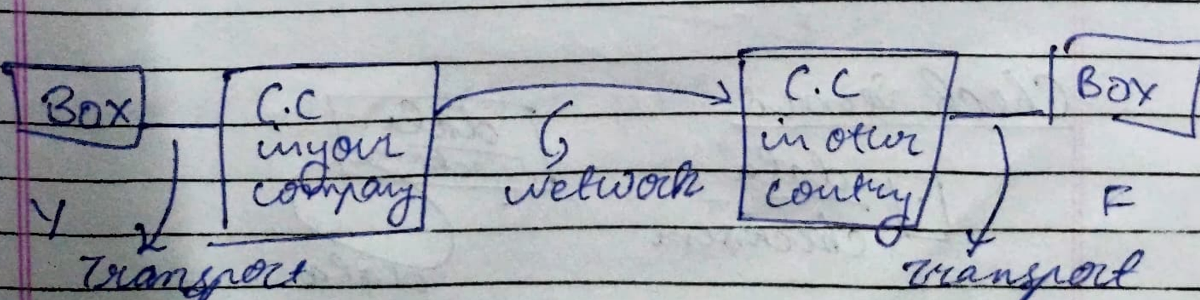


can't buy a domain name
You can just rent.

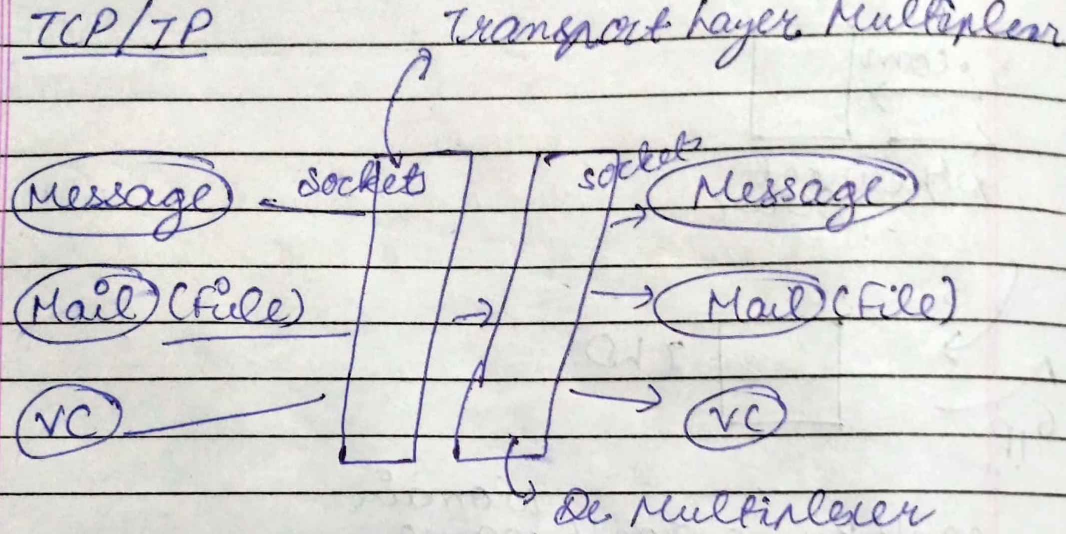
TCP/IP (Transport Layer)



Network to Application → Transport Layer



TCP/IP



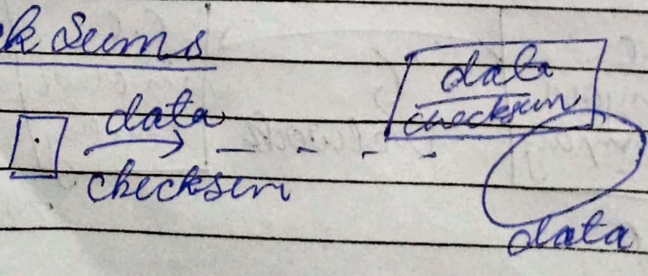
Multiplexing
DeMultiplexing

Data travels in Packets → Transport layer will attach these Port Nos.

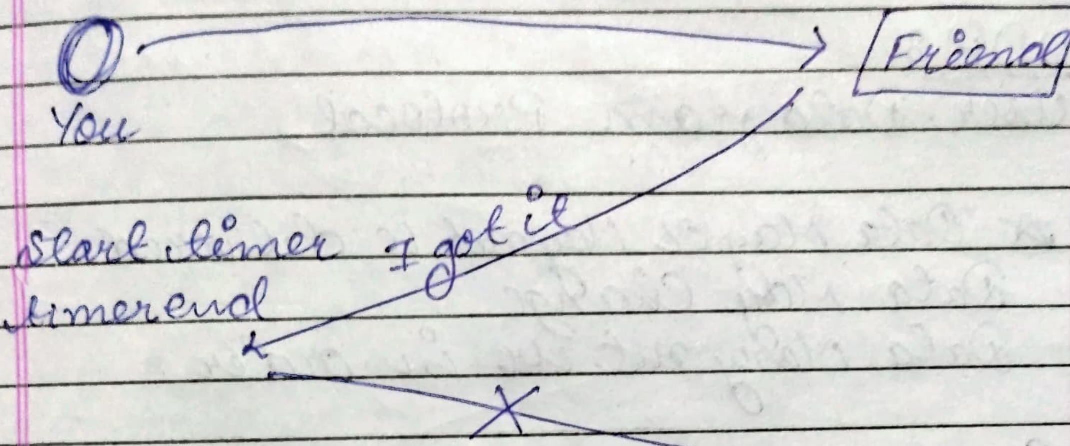
Transport layer also takes care of ~~conte~~ congestion control
&
low high data

Congestion control algorithms build in TCP

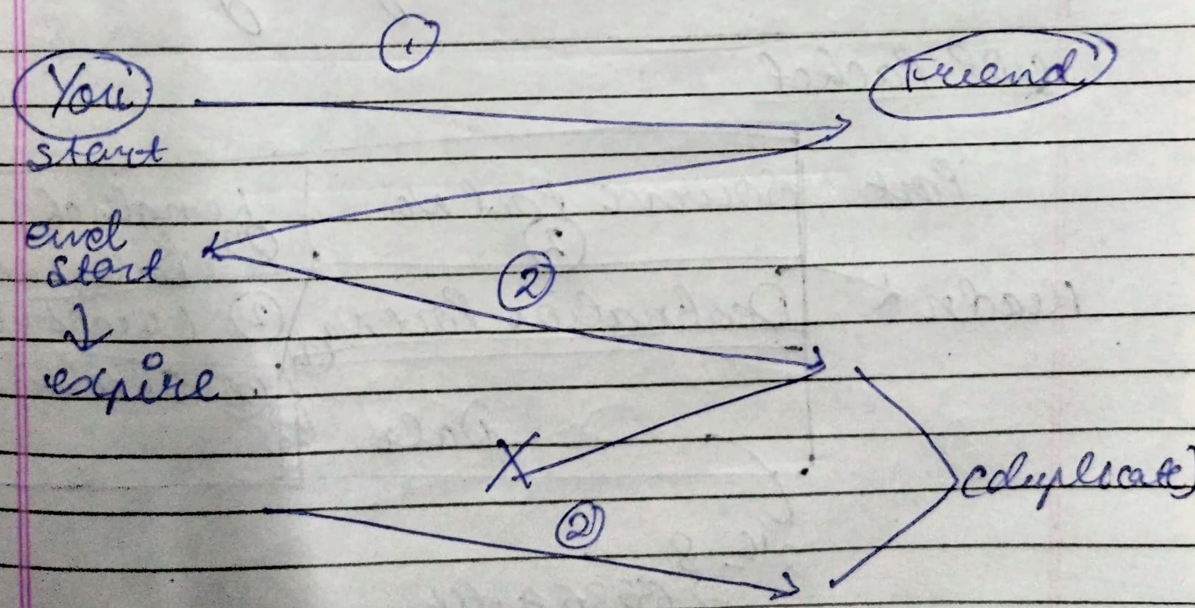
Check Sums



<u>Timers</u>



timer start
timer expire.



solve this problem using sequence nos.

Protocols

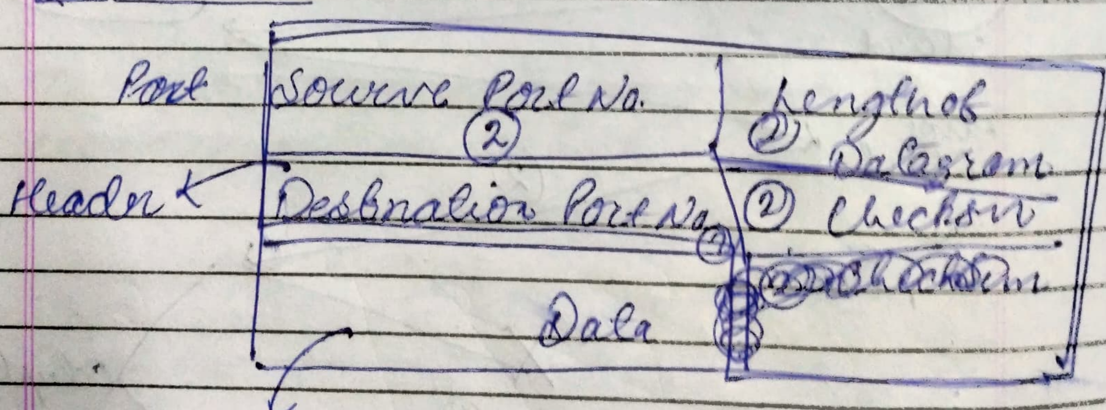
UDP:
User Datagram Protocol

* Data May or May not be delivered
  Data May change
  Data Many not be in order

Connection less Protocol

Uses Checksums
↳ UDP won't do anything.

UDP Packet



$2^{16} - 8$
$\Rightarrow$ 65536 bytes

Total $= 2^{16}$.

Use Cases:
Its very fast
Video conferencing app
Gaming
DNS - UDP

tcpdump -c 5

TCP
Transmission Control Protocol

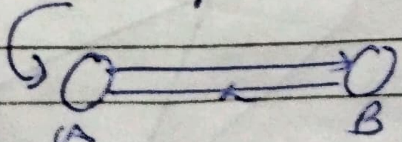
Transport Layer Protocol

Application layer sends lots of rawdata
TCP segments this data
↳ divide in chunks & add headers
It may also collect data from Network layer
□□□□ → □□

Congestion control

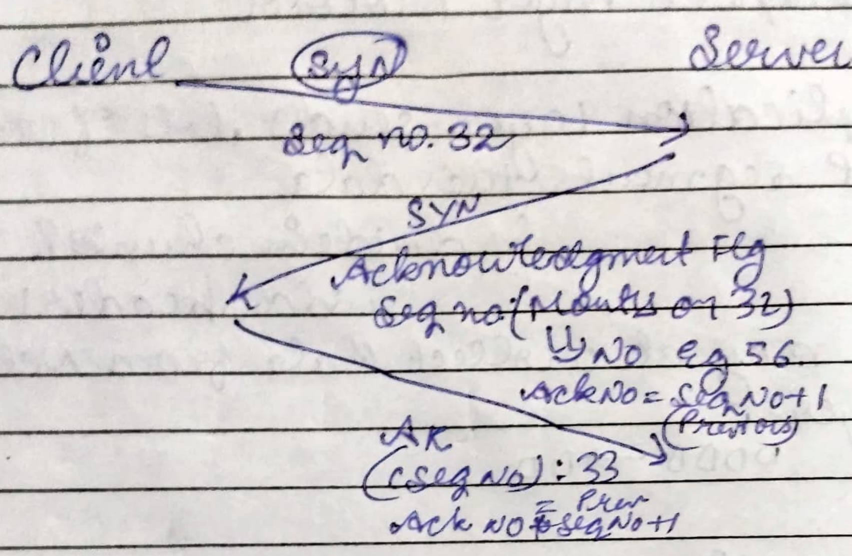
Takes cares of:
- When data does not arrive
- Maintains the order of data

Features:
- connection oriented
- Error control
- Congestion control
- ~~Full~~ Full Duplex



Synchronization Flag

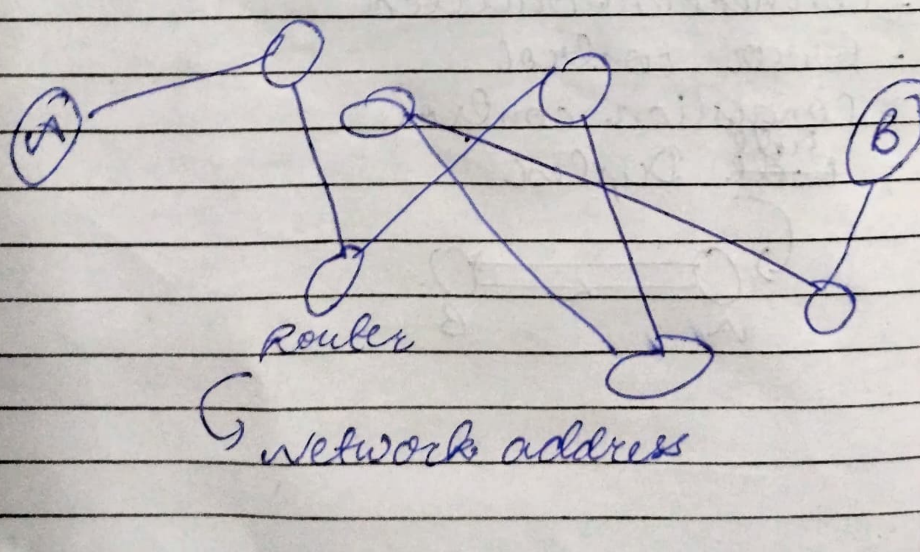
3-Way Handshake



Network layer

Here we work with Routers

Transport → Segment
Network → Packets
Data Link → Frames



Packet
- Network Add of A
- B
- Info

Hop-by-Hop Forwading.

Routing Table → Full
Forwarding Table → single

$\underbrace{\text{192.168.2.}}_{\text{Network Address / Subnet ID}}$30 → Device Address (Host ID)

Network Address
Subnet ID

Control Plane
Build this routing Tables

Routers → nodes.
Edges

① static routing
(Manually)

② Dynamic Routing

Internet Protocol (IP)

Network Layer

IPv4 → 32 bit, 4-words
IPv6 → ~~32 bits~~, 128-bits

5. 6. 9. 14

↓ ↓ ↓ ↓

00000101

Hoping happends at ISP level.

Class A, Class B, Class C ; D, E

A	0.0.0.0	127. 2255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.25

Submet Mask

12. 0. 0. 0/31

192. 0.100/24

Start
192.0.1.0

end.
192.0.1.255

IETF

Reserved addresses

127.0.0.0/8

Eg: local host: 127.0.0.1
Loophost addresses

Packets: Header is of 20 bytes

IPv, length, identification, flags, Protocols checksums, addresses, TTL, etc

~ping google.com

IPv4 v/s IPv6

IPv4 → 32 bits → 2^{32} → 4.3 Billion

DHCP IPv6 → 128 bits → 2^{128} → 4 x IPv4 → 3.4×10^{38}

Cons:

- Not Backward Compatible
- ISPs would have to shift, lot of hardware work

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

Hexadecimal

16 bit number

Eg:

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

127.0.0.0/8 → Prefix

We can do the same representation

1 : 0000 : 0000 : 0000 : 9

= 1 : 0 : 0 : 0 : 9

= 1 : : 9

Middle Boxes

Extra devices that also interact with IP packets

① Firewall → Global Internet
→ Your trusted network.

Filter out IP packets based on various rules
→ Address
→ modify Packets
→ Port nos
→ Flags
→ Protocols

Stateless v/s statefull firewall
↳ More efficient

Have this in network find system

Date
Page

IPv4 v/s IPv6

DHCP

IPv4 → 32 bits → 2^{32} → 4.3 Billion
IPv6 → 128 bits → 2^{128} → 4 X IPv4 → 3.4×10^{3}

Cons:
- Not Backward Compatible
- ISPs would have to shift, lot of hardware work

a : a : a : a : a : a : a : a
Hexadecimal
16 bit number

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

127.0.0.0/8 → Prefix

We can do the same representation

1 : 0000 : 0000 : 0000 : 9
= 1 : 0 : 0 : 0 : 9
= 1 : : 9

Middle Boxes

Extra devices that also interact with IP packets

① Firewall → Global Internet
→ Your trusted network

Filter out IP Packets based on various rules
→ Address
→ Modify Packets
→ Port nos
→ Flags
→ Protocols

Stateless v/s statefull firewall
↳ More efficient

Have this in network. find system

NAT

Network Address Translation

Private IP

10.0.0.1 → 150.150.0.1

Data Link Layer

Router → IP add → ISP

(subnet)

IP1 IP2 IP3

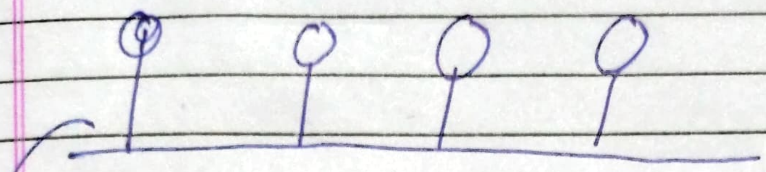
DHCP

New device → DHCP Server

Pool of I.P address

Many devices connected in LAN.

data link layer address



cache

DLLA of sender & IPaddress of destination

ARP cache → Address resolution
Protocol

MAC address
→ data link layer add

Framing
Error detection