Computer Networking

Computer Network: Bunch of computers connected together.

Internet : collection of computer networks

How did networking start ?

After the soviet union launched the first satellite into space, and won the race in doing so.

America set up ARPA (Advanced Research Projects Agency)

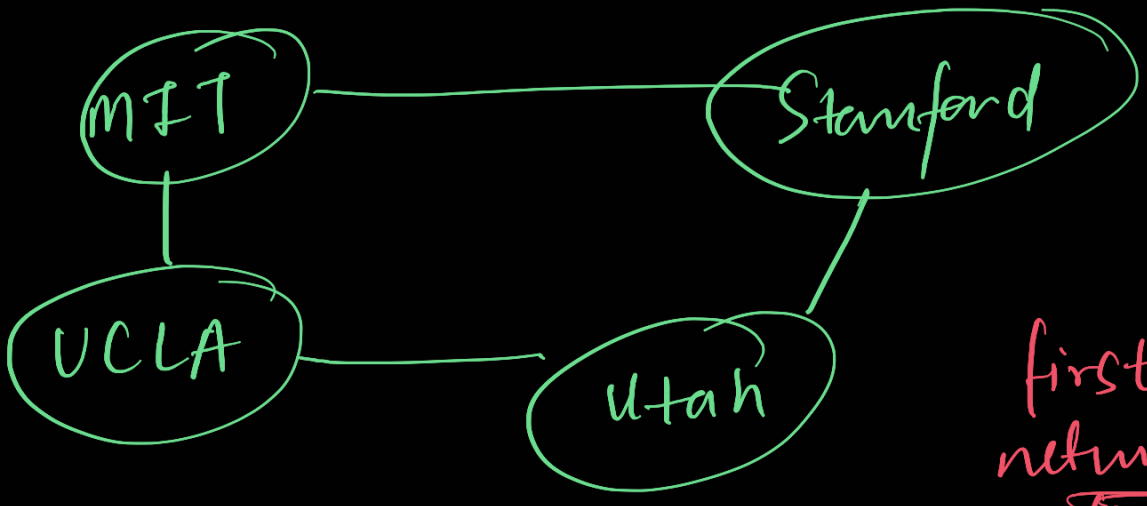
⇓

To carry out advanced scientific research, so as to be world's #1 superpower & not miss out on achievements like "launching the first satellite."

As the "ARPA" facility was set up,

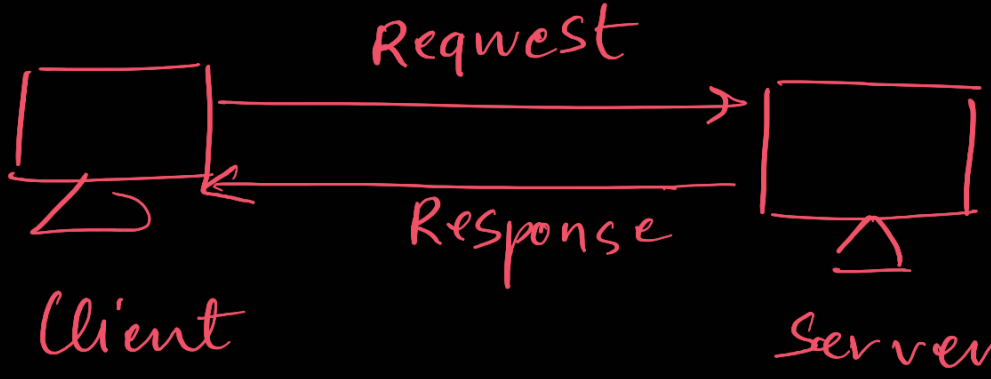
scientists needed a way to communicate with each other across buildings.

⇒ ARPA NET was set up.

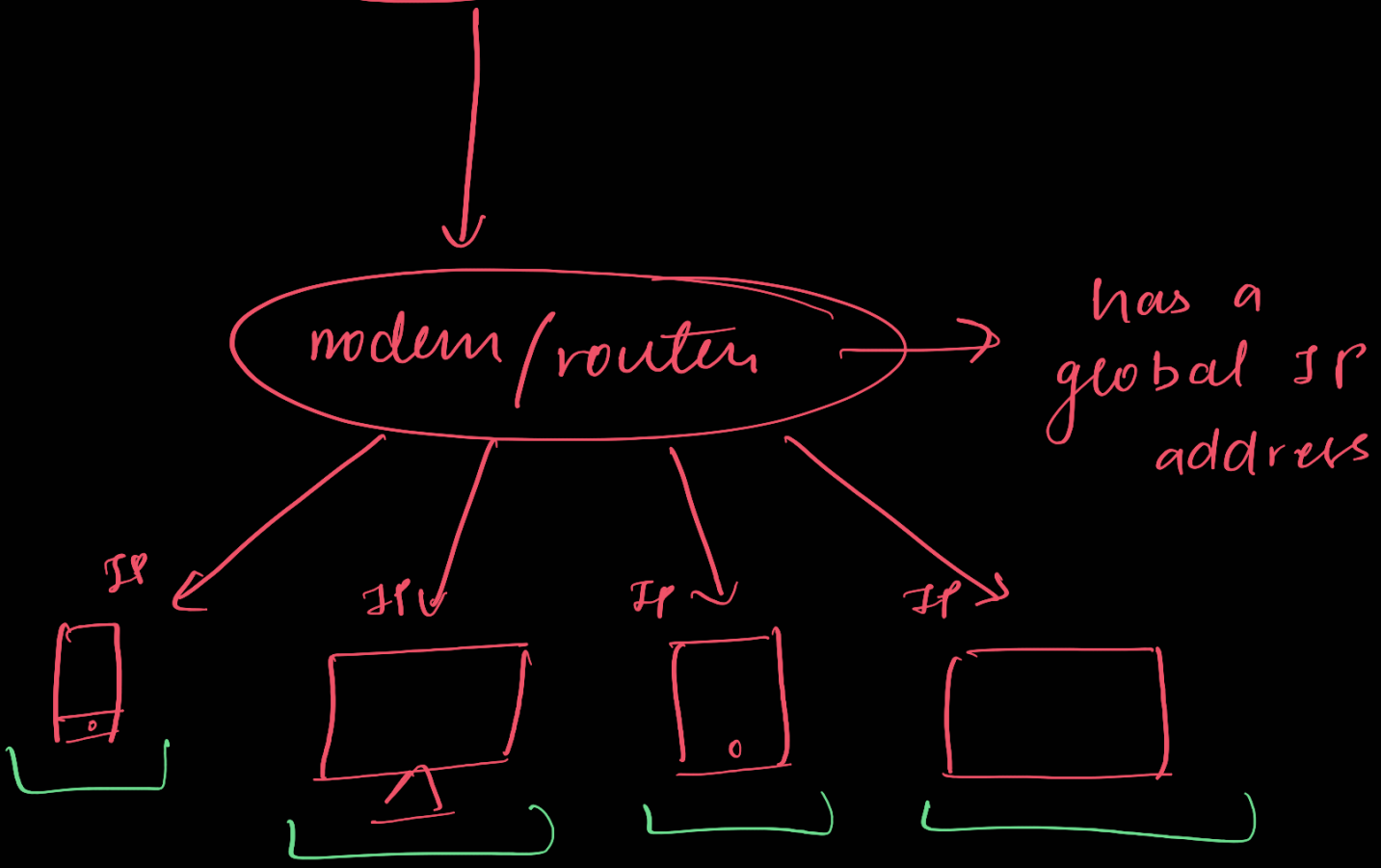


first network using TCP

IP as well, later



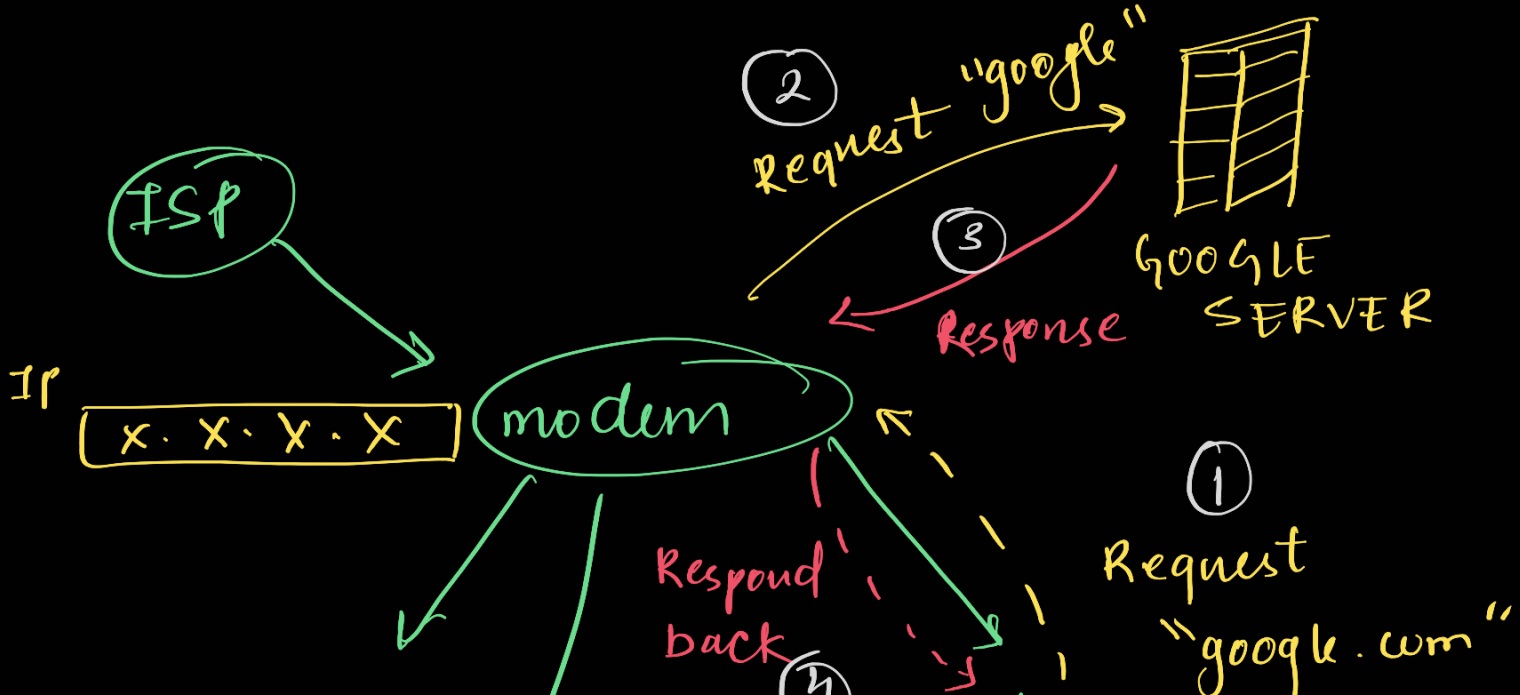
ISP — Internet Service Provider



=> All 4 devices will have same IP address given to them by the router

→ Local IP address

→ Done using DHCP protocol



① device connected to modem requests for any website.

② Router uses the url to request at specific IP addr

③ website's server responds with suitable data

④ IMP Router decides where to respond back, which device requested the website out of all that are connected.

⇒ NAT — Network Access Translator

⇒ Devices differ by IP addresses

⇒ Applications on the same device differ by port numbers.

Ports are 16-bit numbers — 0 or 1

→ 2^{16} possible ports : ≃ 65,000

HTTP ⇒ port : 80

MongoDB ⇒ port : 27017

SQL ⇒ 1433

0 - 1023 ⇒ reserved ports.

1024 — 49152 ⇒ for applications

eg. mongo, SQL

remaining : you can use.

Network categories

LAN : for houses / offices

using : ethernet cables, wi-fi

MAN : across cities

WAN : across countries.

① SONET
Synchronous optical networking
using optical fibre cables, cover larger distances.

② Frame relay
↳ may to connect LAN to widen networks like internet.

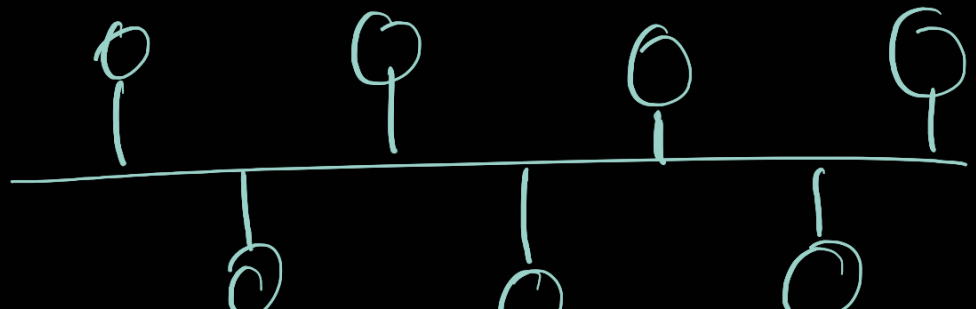
INTERNET : collection of LAN, MAN, WAN

MODEM : convert digital signals to analog & vice versa.

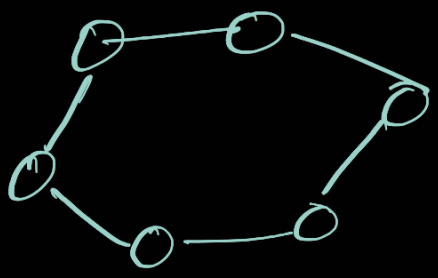
ROUTER : routes the data packets based on the IP addresses.

Network Topologies

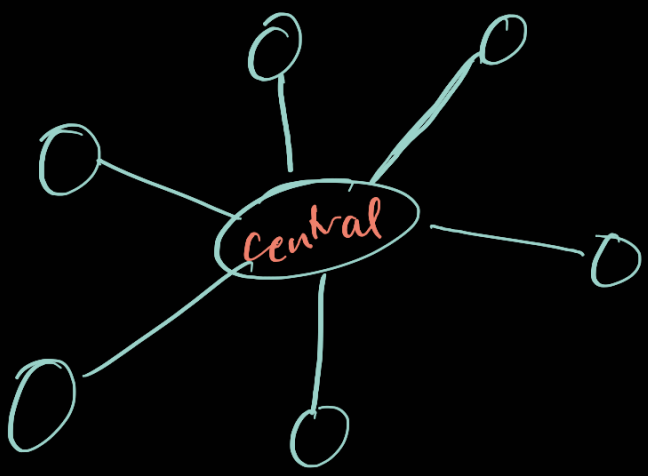
① Bus



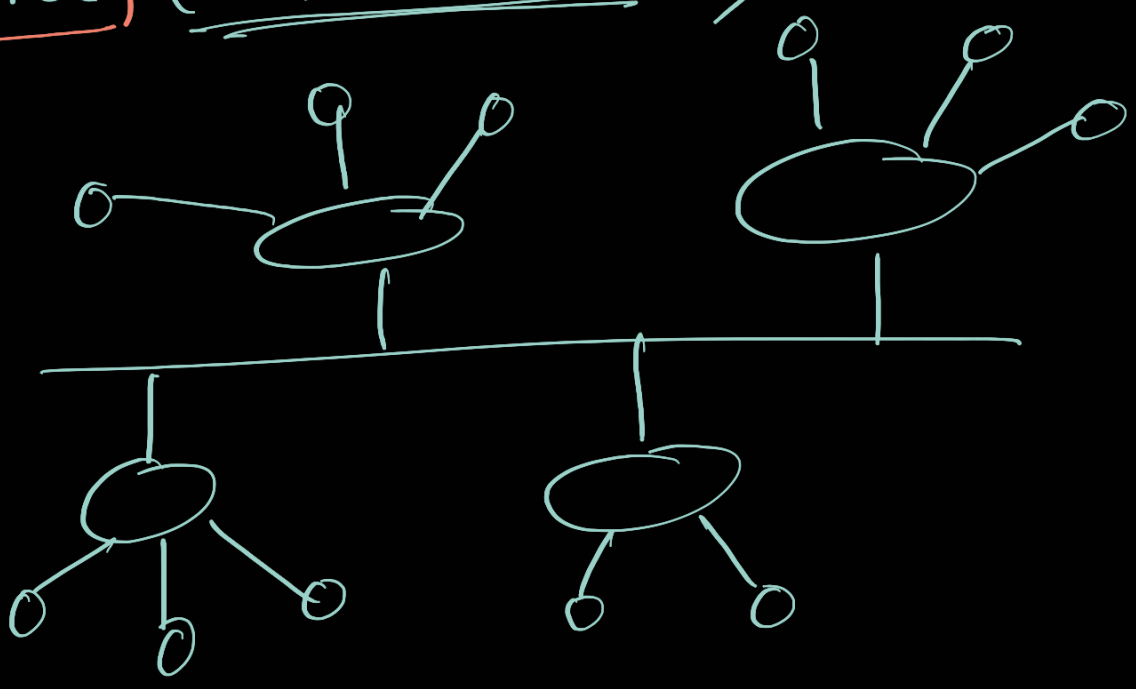
② Ring



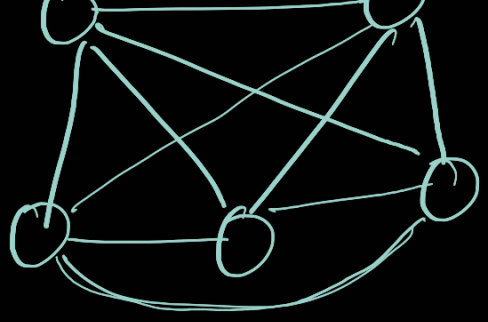
③ Star



④ Tree (bus & star like)



⑤ Mesh



Structure of the network

OSI Model

Open System Interconnection

→ implemented in software

Application

↓ data

(SSL) Presentation → Translation
→ convert data format
eg. ASCII to EBCDIC

↓

Session → helps in setting up & managing connection
enables sending & receiving,
Authentication, Authorization

(UDP) (TCP) Transport → segmentation
→ flow control
→ Error control
Check sum

Network → transmission of received data segments from 1 computer to another, located in different networks

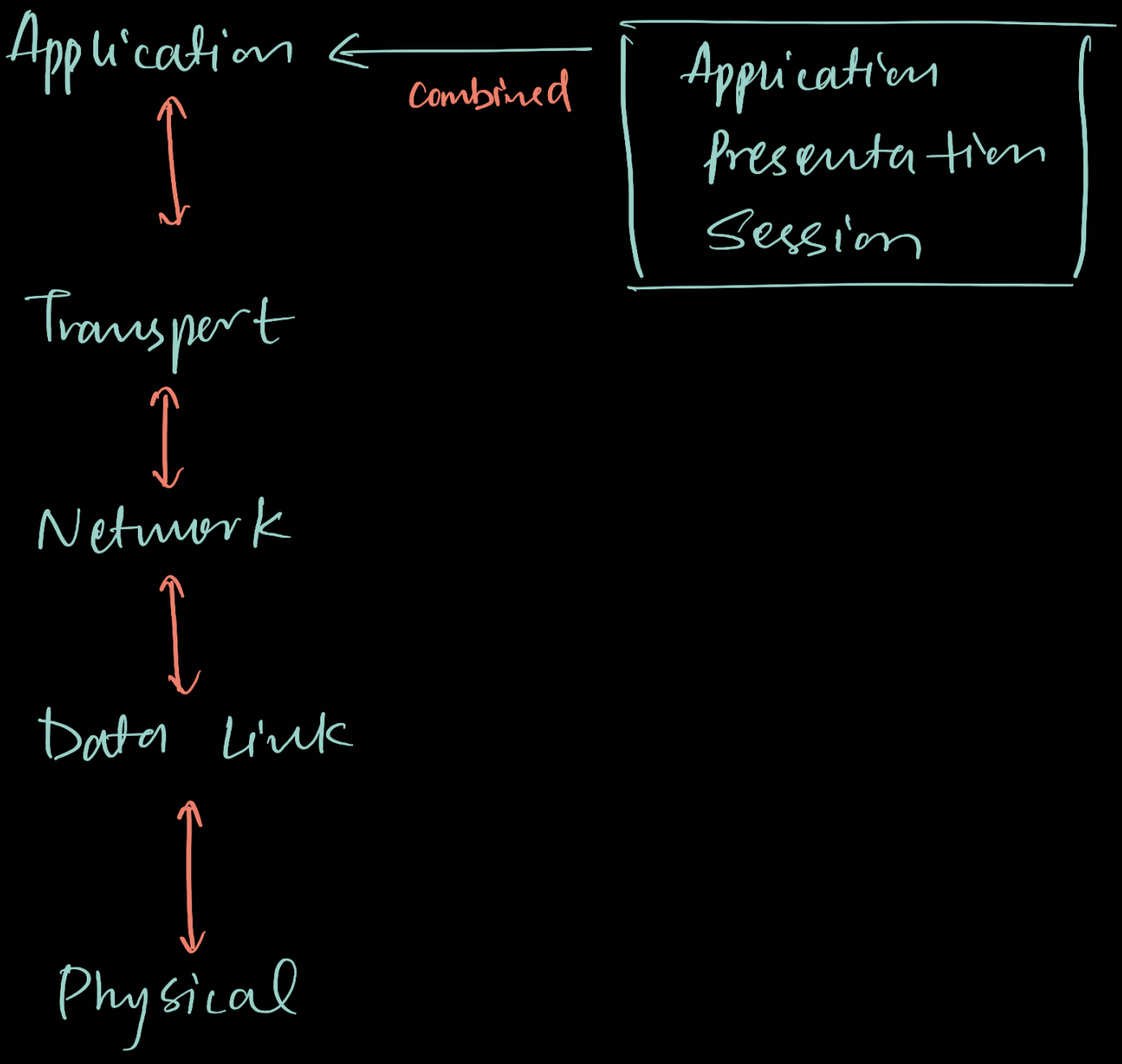
- router → routing
- logical addressing
- Load Balancing

Data Link → directly communicate to computers

- Physical addressing (MAC addresses)
  12 digit alphanumeric network interface
- media access control

Physical → working with cables

TCP/IP Model



#Application Layer

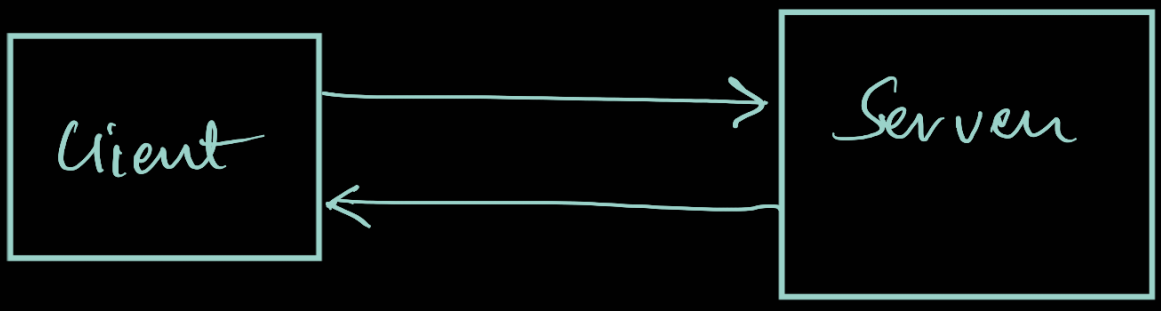
→ Users interact with it.

→ Whatsapp, browsers etc.

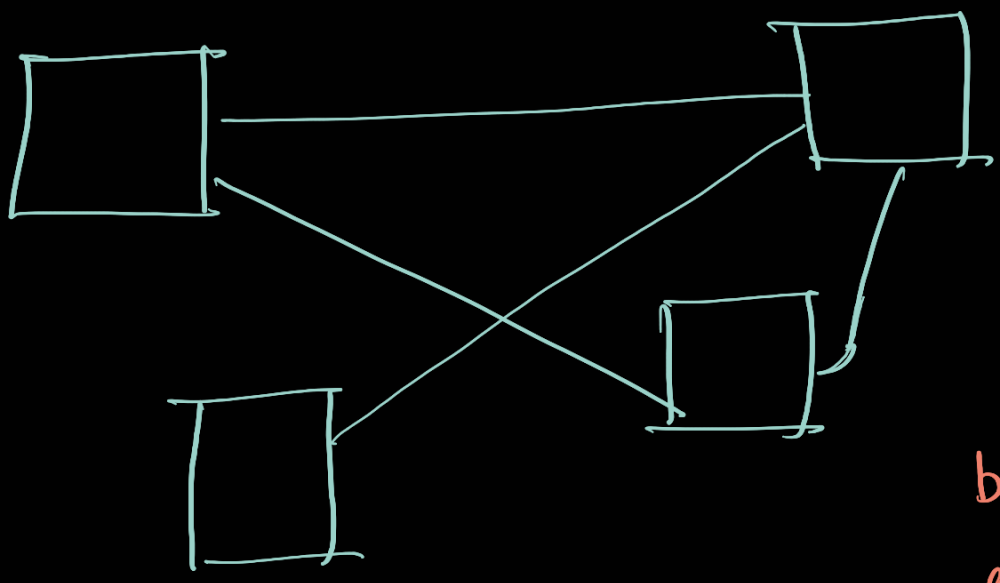
→ Lies on our devices

→ Protocols

→ Client-server Architecture



→ Peer to peer architecture



every system can be server as well as client

eg. Torrents, seeding etc.

Protocols :

web protocols

TCP / IP :

★ HTTP: html pages

★ DHCP: dynamic host control
allocates ip add^r to devices connected in the same network.

★ FTP: file Transfer Protocol

★ SMTP: Simple Mail Transfer

★ POP3 & IMAC: receive mails

★ SSH: secure shell.

★ VNC: virtual network computing

Telnet: terminal emulation
enables the user to connect to a remote host using the telnet client.

Port: 23

UDP: a connectionless session
does not maintain state
data may be lost.

Program: Whatsapp

(Application)

Send a message

Record a video

Process: (features/ running instance of the program)

Thread: a process completing a simple small task.

Sockets: interface b/w processes and the internet.

Ports: Ephemeral Ports

↓

distinguish which application requested the data

distinguish which instance of the applicat" requested the data.

eg: ports will distinguish

whether chrome, discord or MS teams is requesting the data,

furthermore: ephemeral ports will distinguish which tab of google chrome requested the data

---

→ Client server protocol

→ Tells us how you request data & tells us how server responds.

Application Layer

↑

HTTP uses TCP → transport layer

→ ensures no data is lost

↳ stateless protocol, server will not store any info about client by default.

HTTP methods

1) GET
2) POST

Status codes

1XX → Informational

2XX → Successful

3) PUT

4) DELETE

⋮

3XX → Redirection

4XX → Client Error

5XX → server Error

Cookies : Unique String

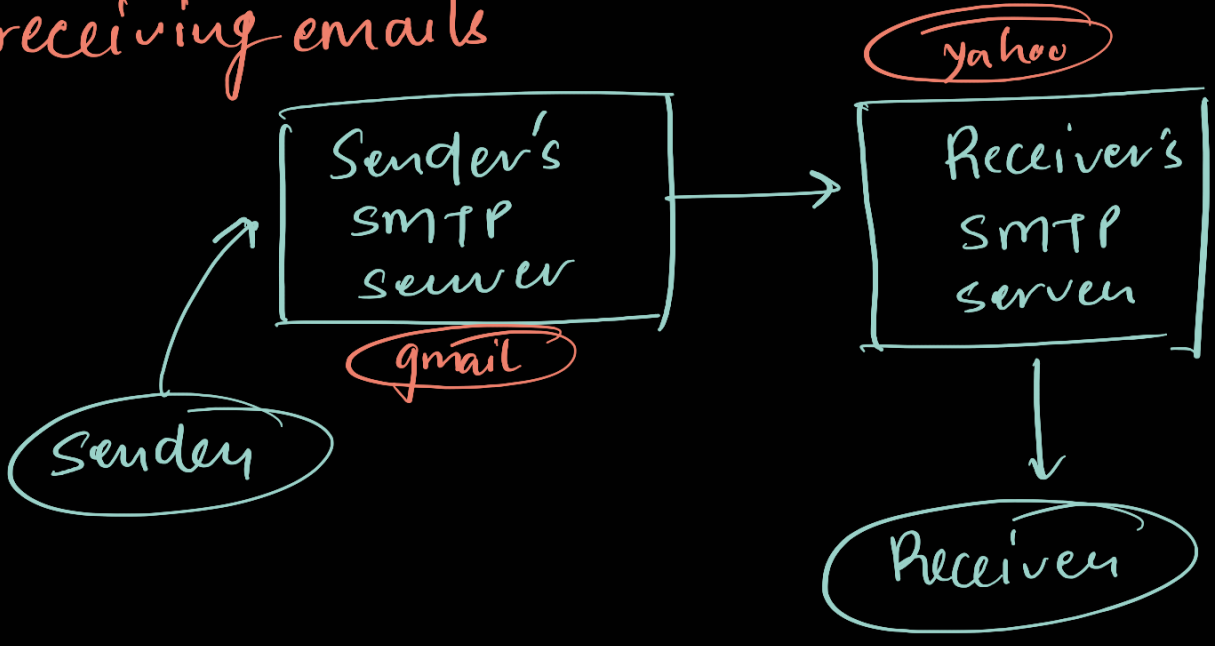
Stored in client's browser

☆ third party cookies

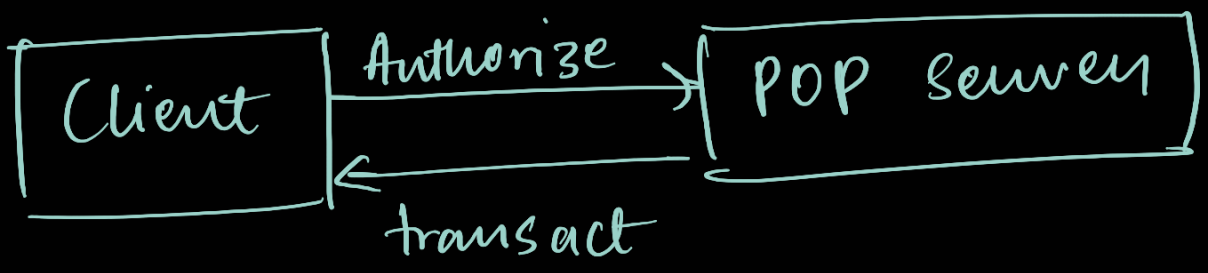
How emails work

SMTP : sending mails

POP 3 : receiving emails



POP : Post Office Protocol [Port : 110]



IMAP : Internet Message Access Protocol

Also used to receive emails.
Allows to get e-mails on multiple devices.

DNS : Domain Name System

enter "www.google.com"

HTTP will take this domain & look up its IP address in DNS & process the request

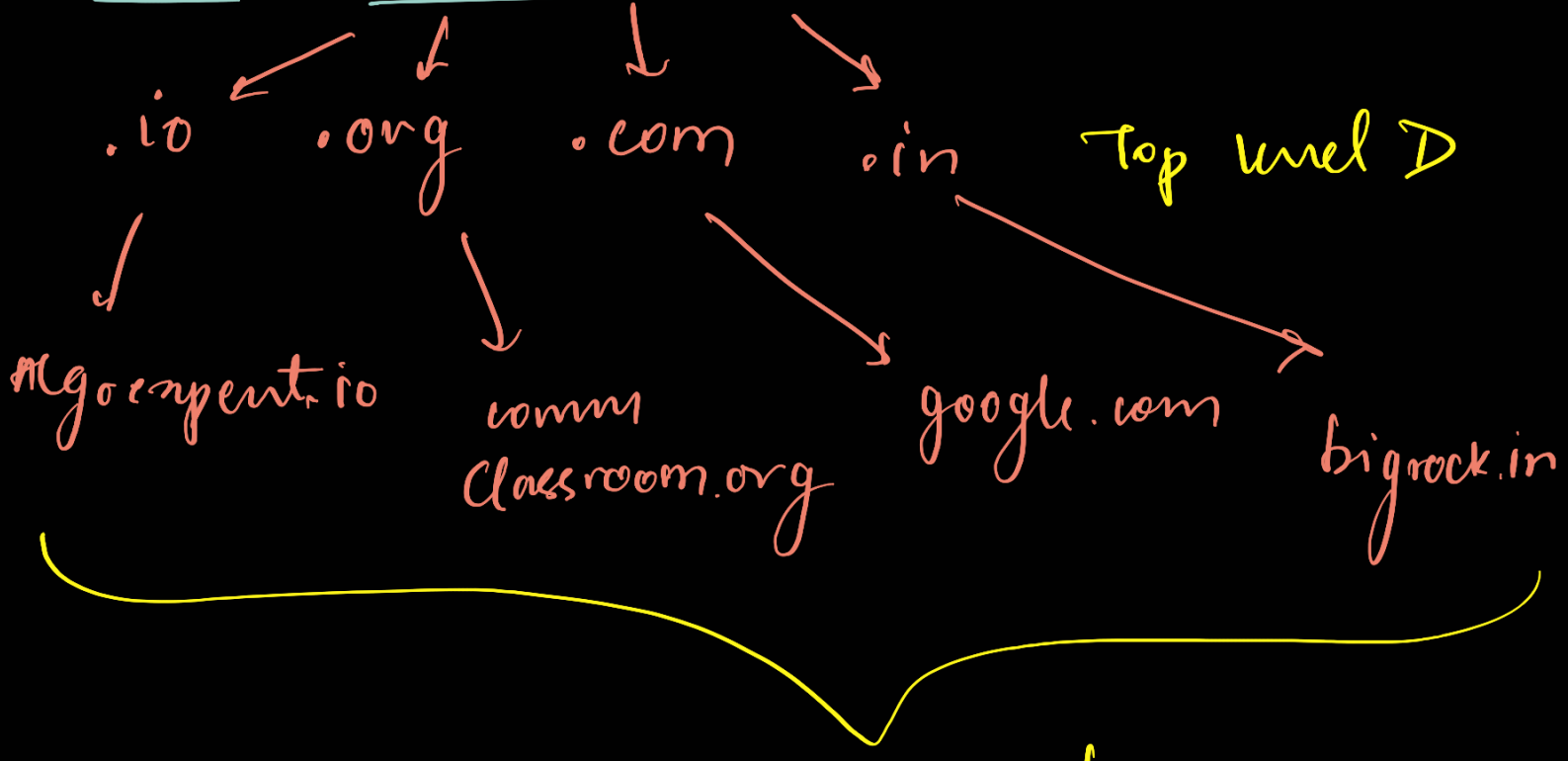
second level domain

mail.google.com

sub domain

top level domain

Root DNS servers



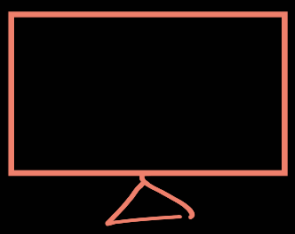
ICANN : Internet Corporation for Assigned Names and Number

→ Corporation that registers top level domains.

Process

looking for "google.com"

①



Your Computer

→ first step: check local storage
(cache data on laptop / PC / phone)

② Not able to find in cache

⇒ Check local DNS server.

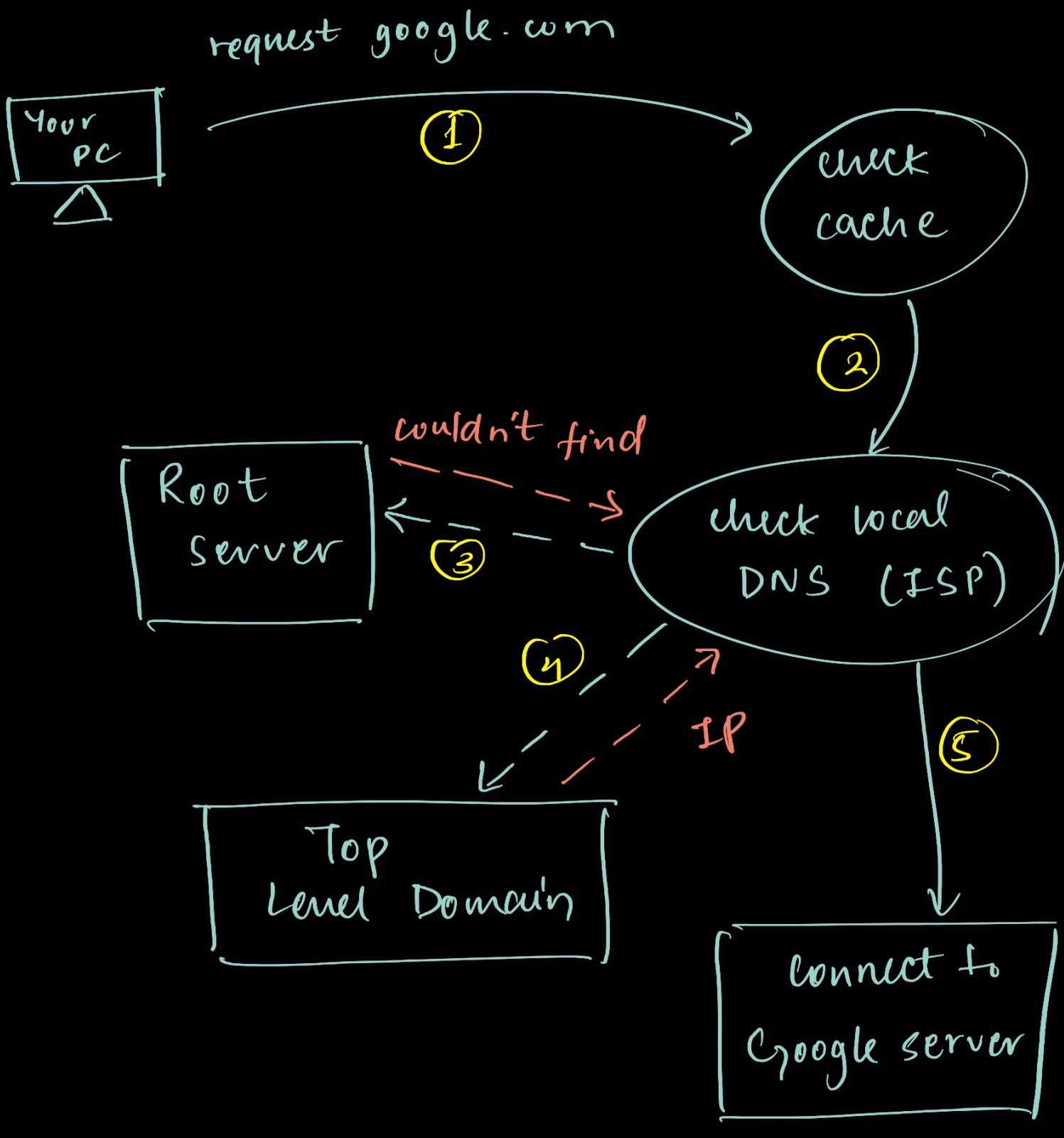
first point of contact
(usually ISP)

③ ISP not able to find the domain

⇒ They will look up in root server

④ Root Server does not have it

→ Check Top Level Domain.

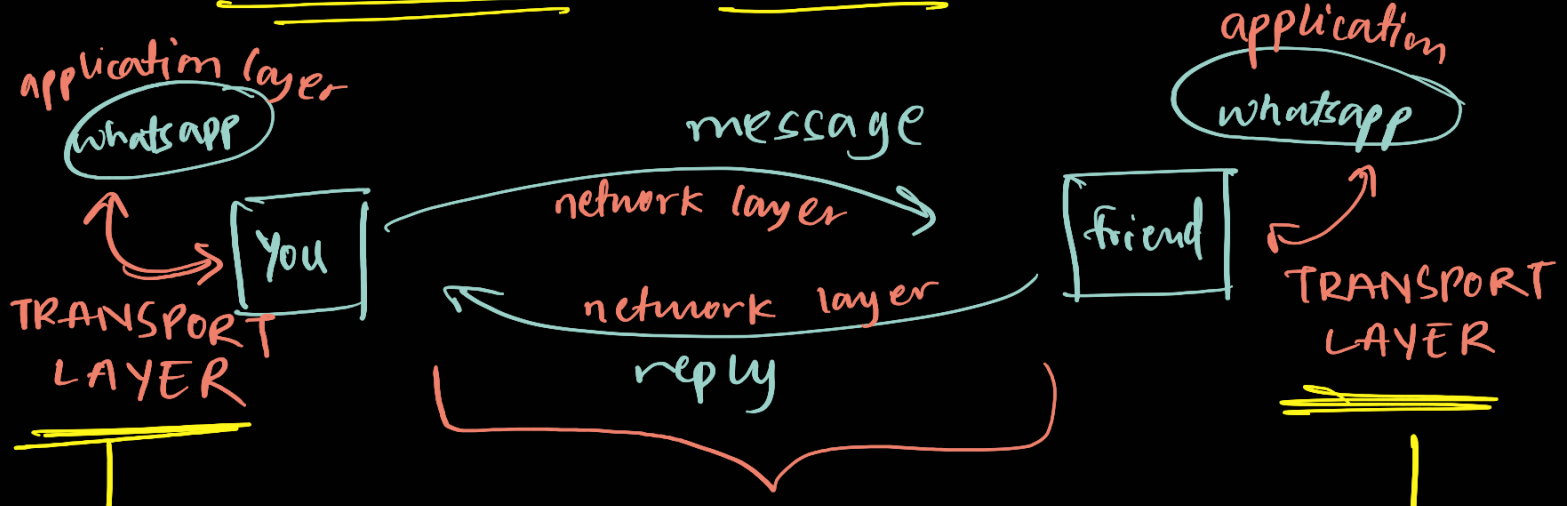


Side Note : We only rent a domain name, we don't buy it.

We pay providers like godaddy & they

pay to ICANN.

TRANSPORT LAYER



actual transport between two devices is done by NETWORK LAYER

⇒ one network to another

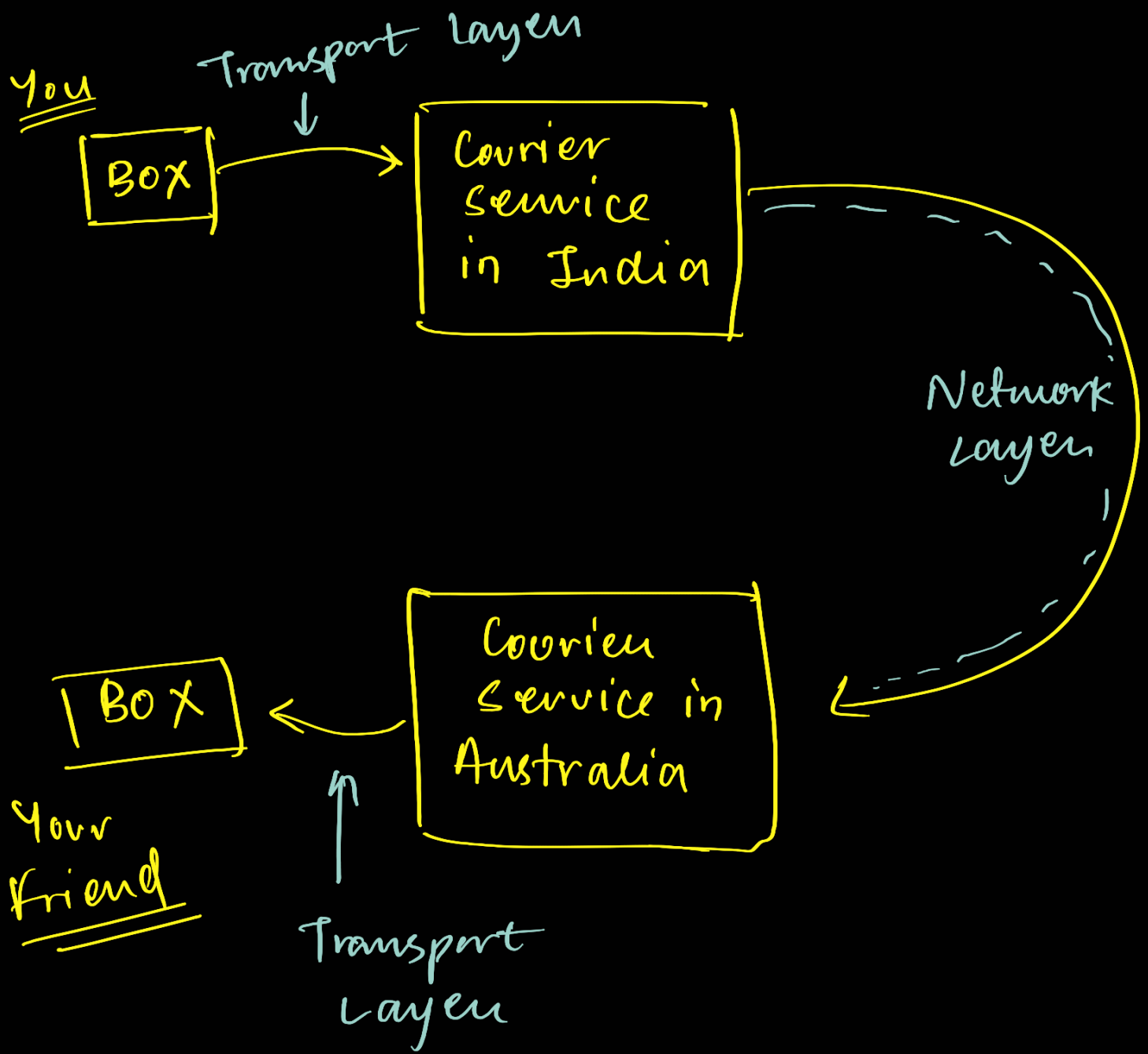
Transportation of data from application to the network layer is done by TRANSPORT LAYER.

To Reiterate: Network layer will take care of delivering message from one device to another.

BUT, once the message is received, Transport Layer will determine which application's data this is.

Example in layman terms:

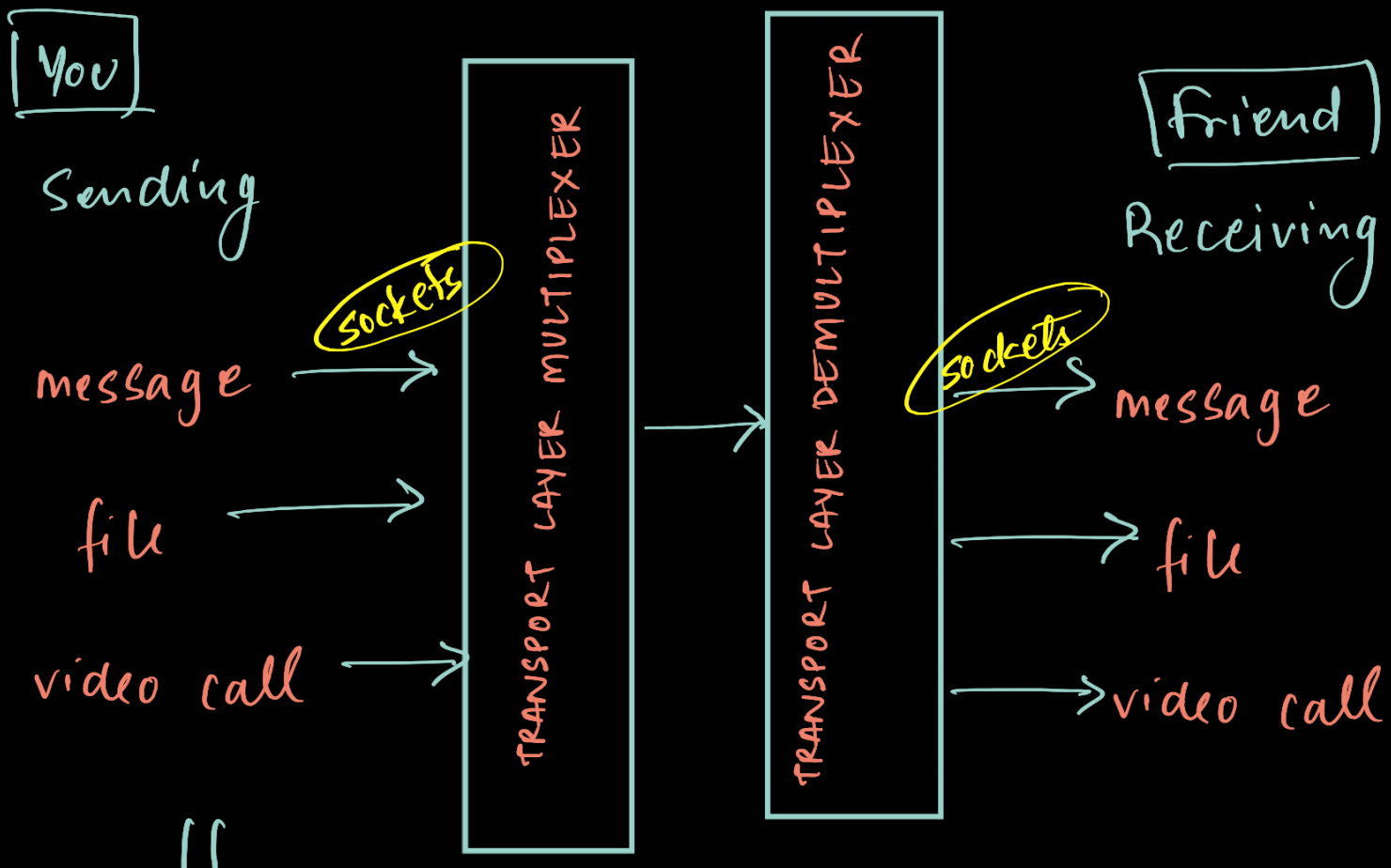
⇒ Sending a package to your friend living in Australia.



Transport Layer Protocols

TCP

UDP



Sockets have port numbers

Data travels in packets

⇓

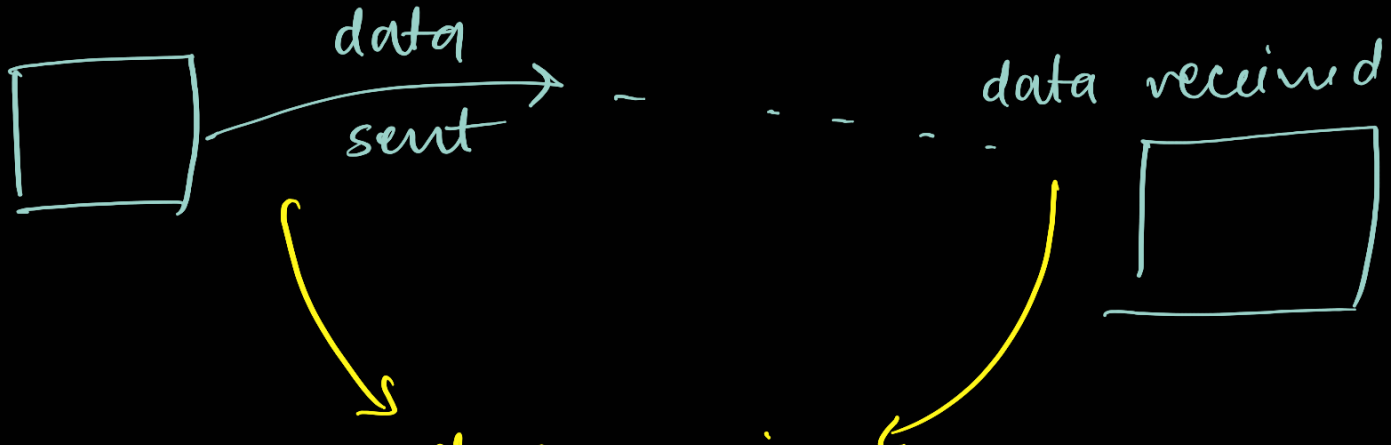
Transport Layer will attach these socket port numbers.

This is how it knows what application is

Sending the data & trained to deliver it.

* Transport Layer takes control of congestion control (traffic)
  - adjustment of speed & bandwidth etc.
  - using congestion control algorithms built in TCP

Transport Layer Protocols perform error correction using checksums.



checksum is calculated at sending time as well as at receiving time.

If checksum value matches ⇒ NO ERROR

If not, there is some error, loss of data.

Timers : As data packet are sent from one device to another, retransmission timers are started, which only end at acknowledgement of receiving

If timer does not end, it expires and implies that data packet was not received.

However timers can cause sending of duplicate data packets

* This problem is solved using Sequence Numbers

UDP : User Datagram Protocol (Transport layer Protocol)

→ Data may or may not be delivered.

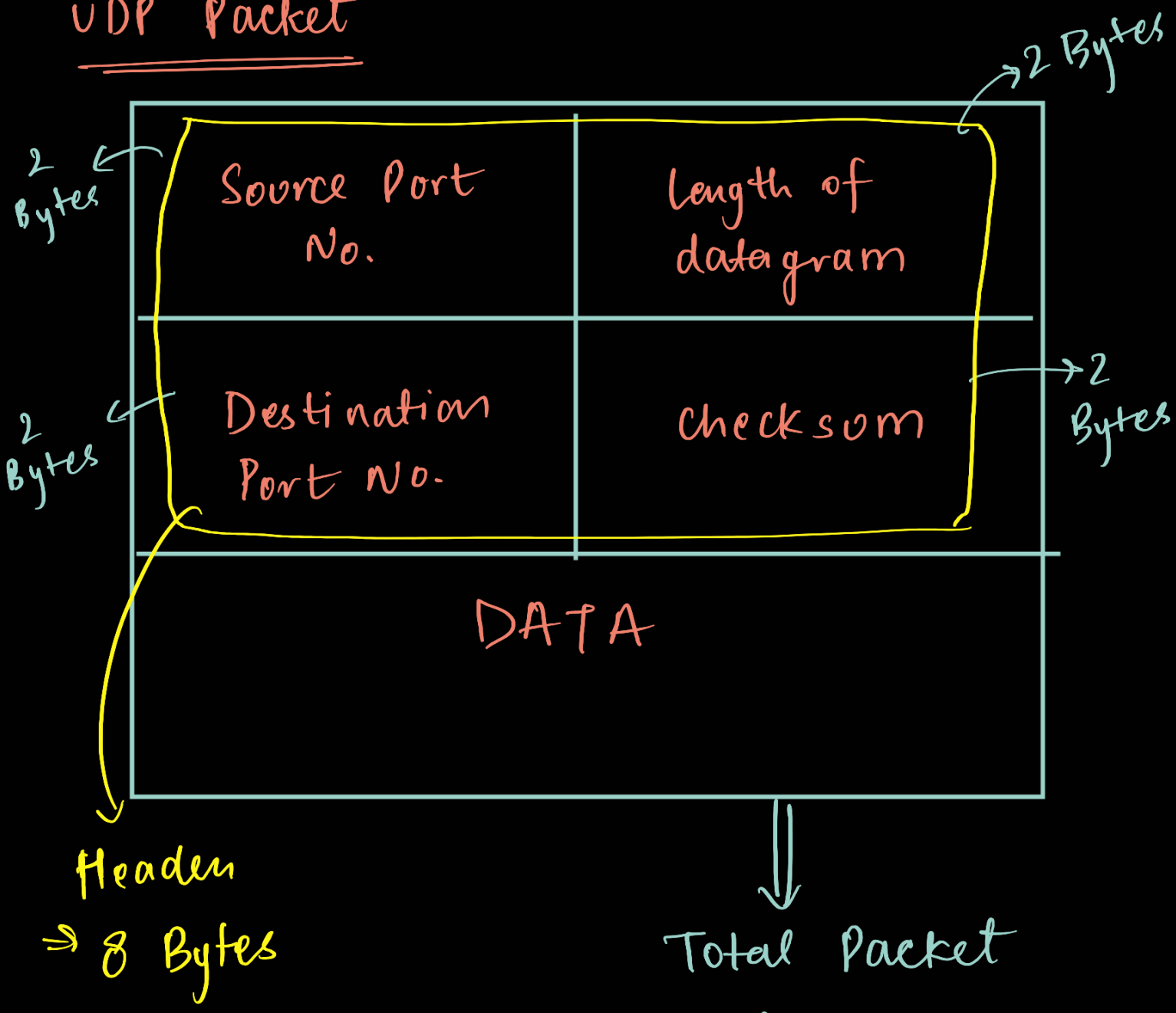
→ Data may change.

→ Data may not be in order.

UDP is a connectionless protocol.

UDP also uses checksums but doesn't do anything about data getting corrupted.

UDP Packet



Data = $2^{16} - 8$ Bytes

$\approx$ 65000 Bytes

USE CASES

→ UDP is fastest,

→ Video Conf. Apps., Gaming.

→ DNS uses UDP because it's fast.

TCP : Transmission Control Protocol

→ Transport Layer Protocol.

→ Application layer sends a lot of raw data
TCP creates segments of this raw data.
↳ divides it into chunks, add headers.
↳ may also collect data from network layer

⇓

when data is passed from application layer to network layer, transport layer helps by creating small segments/packets.

BUT, network layer can further divide these

segments into even smaller segments.

So, the transport layer at the receiving end will first patch these smaller segments into original size segments & then forward them to receiver's application layer.

⇒ Provides congestion control.

→ Takes care of:

→ proper (complete) delivery of data.

→ delivery of data in correct order. (sequence numbers)

Features

→ connection oriented.

→ error control

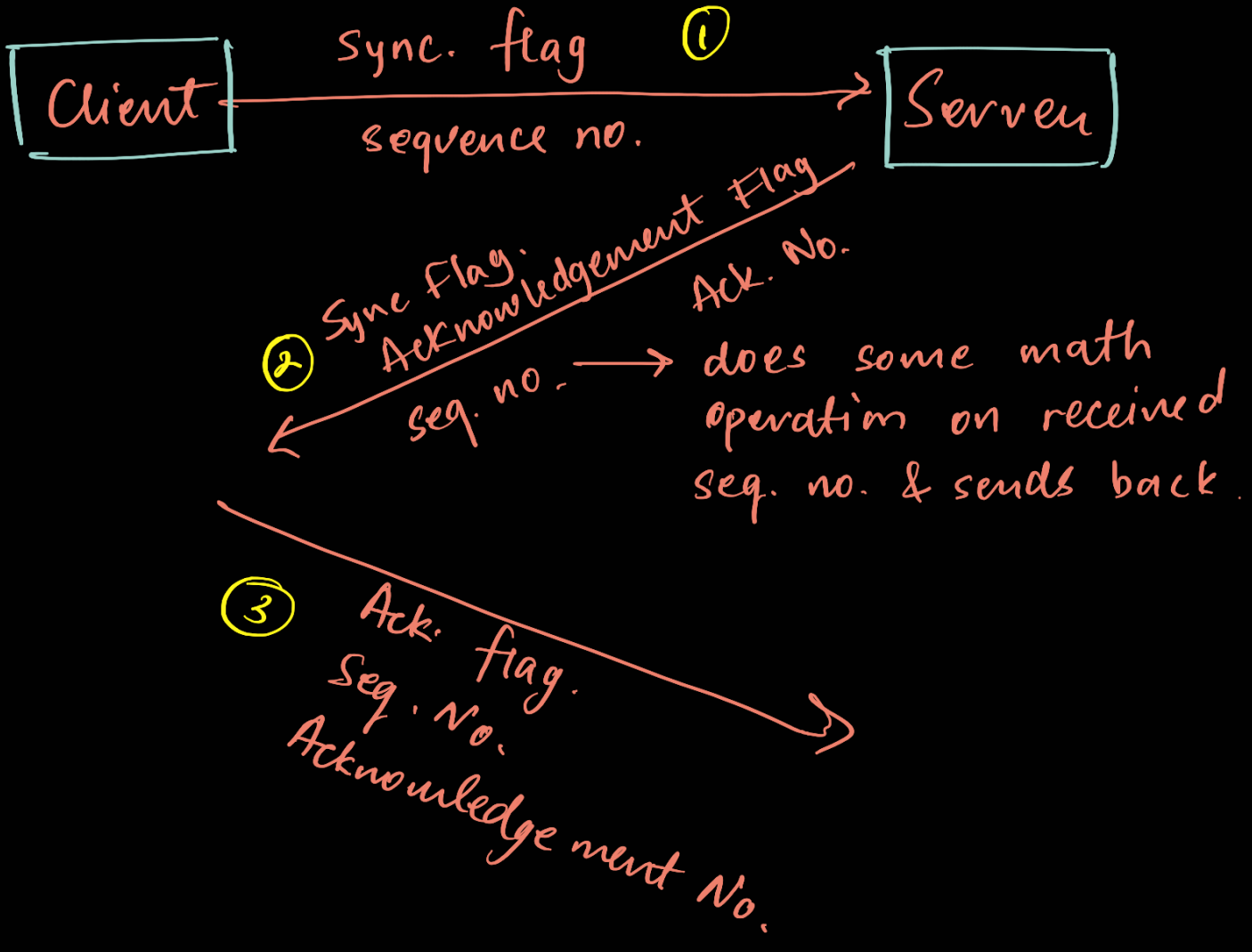
→ congestion control

→ full duplex (bi directional communication)

NOTE: at a time, one TCP connection can only be between two devices.

CONNECTION ORIENTED meaning

→ 3-way handshake



NETWORK LAYER

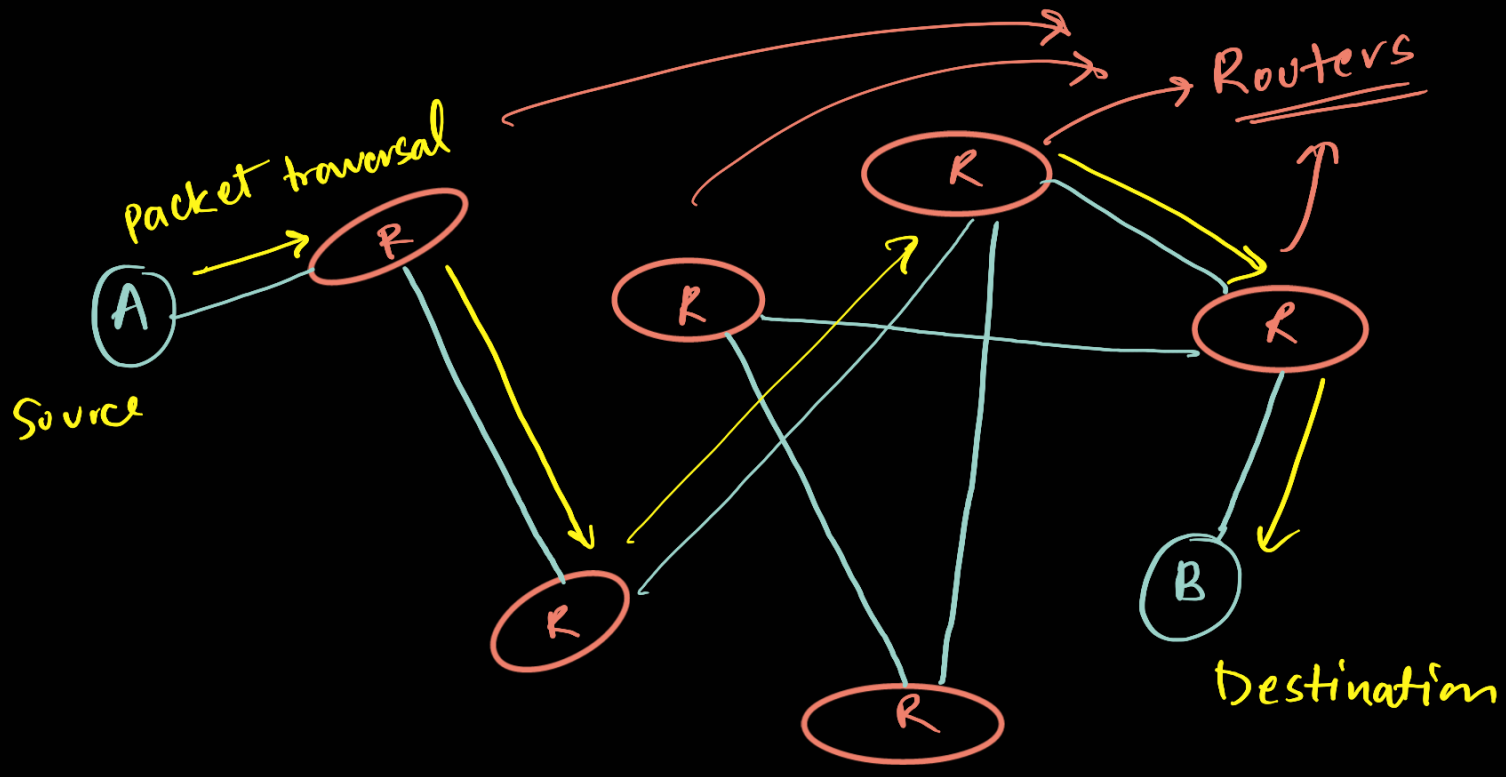
Here, we work with routers.

Network layer works with packets

Packet contains → Network layer addr of destination

Network layer addr of sender

Info



hop by hop forwarding

packet hops from source to nearest router containg all info about source & destination.

router checks if packets destination addr matches its own addr.

yes → receives packet

NO → checks list of routers in forwarding list,

forwards packet to suitable router.

⇒ packet moves ahead.

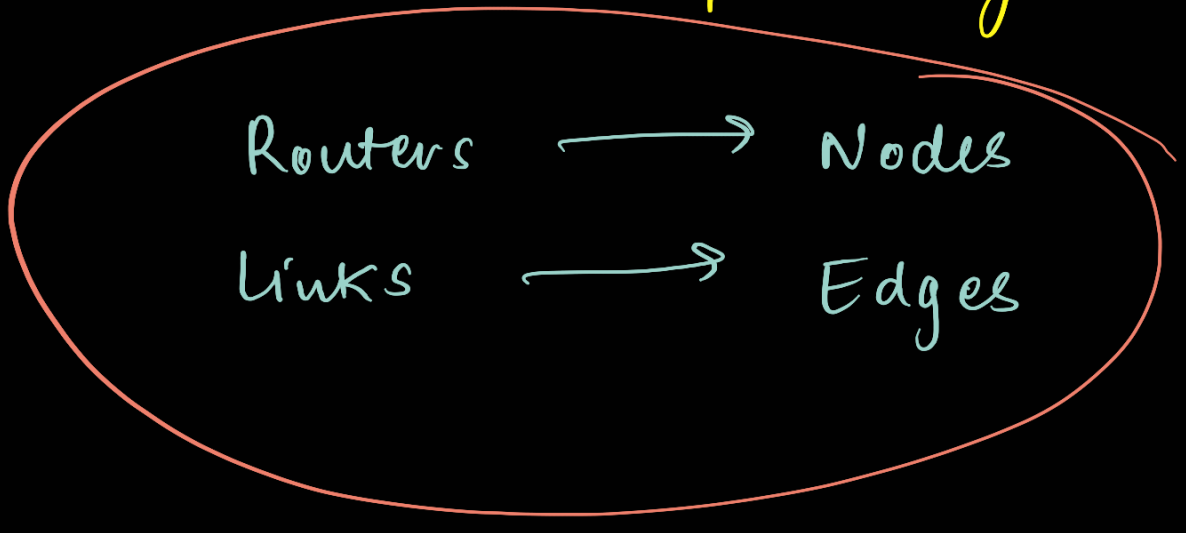
IP addr:

eg. 192.168.2.30

192.168.2. → network addr

30 → device addr

#Control Plane: takes care of all these routing lists and forwarding lists.



① Static Routing: Manually adding addr. in routing table.

→ time consuming.

② Dynamic Routing : Automatically evolves

→ uses algorithms

Dijkstras, Bellman ford

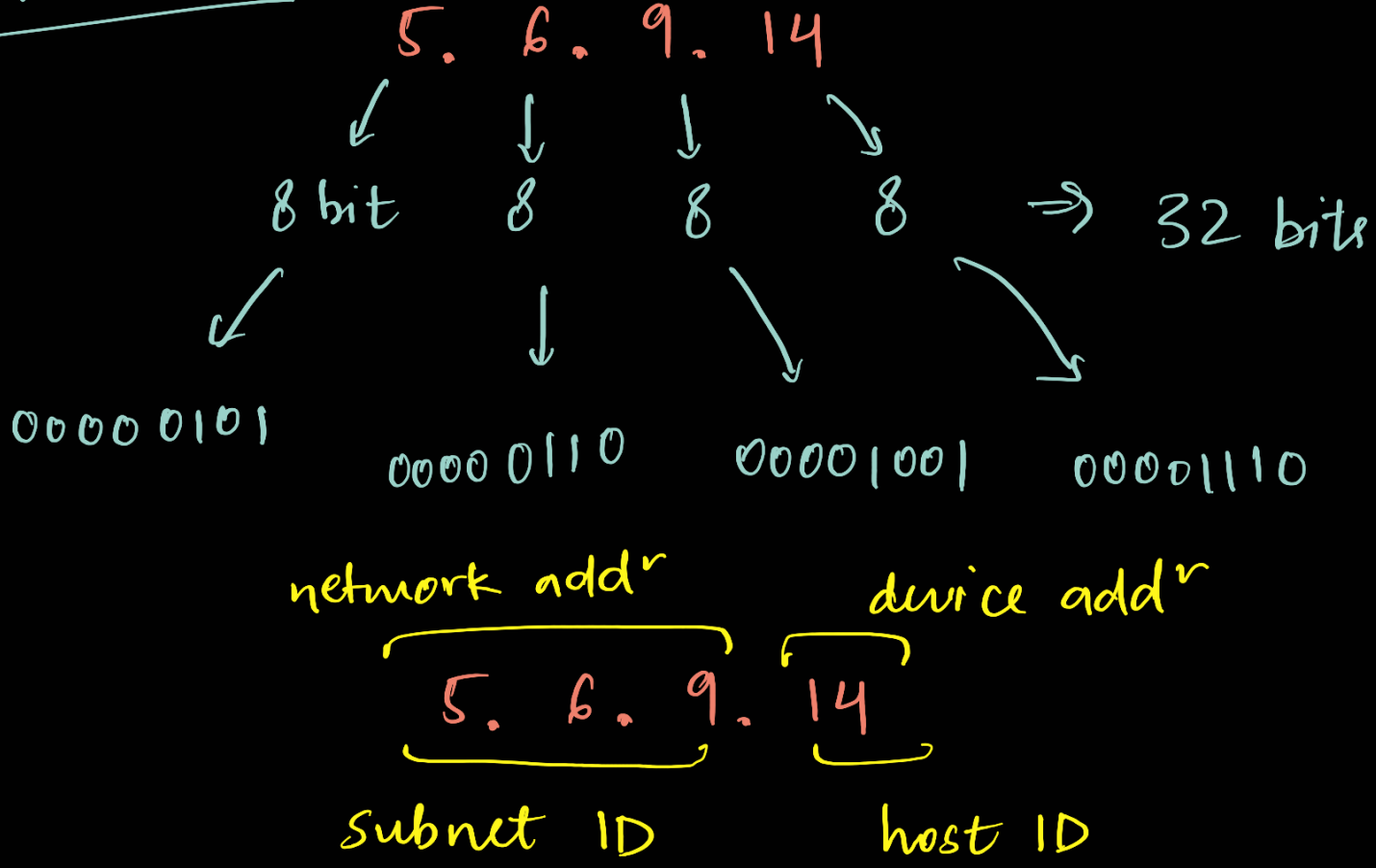
Internet Protocol (IP)

→ Network layer protocol

IPV4 : 32 bits, 4 words

IPV6 : 128 bits, 4 words

IPV4 example



classes of IP addresses

A	0.0.0.0	———	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	———	239.255.255.255
E	240.0.0.0	———	255.255.255.255

Reserved Addresses

127.0.0.0 / 8

ex: localhost : 127.0.0.1 —

loop back addresses

- allows same machine to be client and server
- for testing purposes.

Packets → Header is of 20 Bytes

contains : IP version
total length
Identification No.
flags,
Protocols
Checksum,
Addresses
TTL

} google the terms.

IPV6 : 128 bits

2^{128} possible IP addresses.

$\Rightarrow$ 3.4×10^{38} unique IPs

cons: not backward compatible.

→ shifting from ipv4 to ipV6 will take a lot of effort.

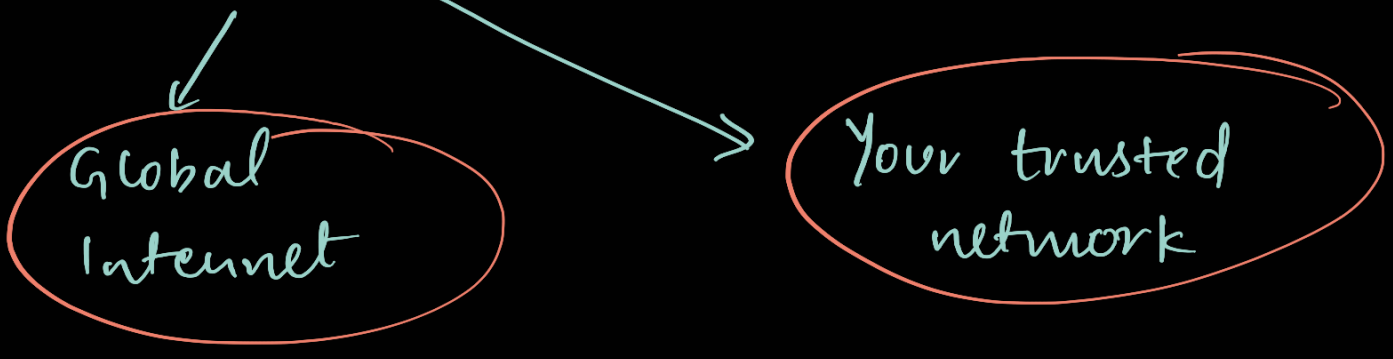
representation :

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

hexa decimal
(16-bit)

#1 Middle boxes : devices apart from computers & routers that also interact with packets.

① Firewall



filters out IP packets based on various rules.

like:

→ filter based on address.
→ modify addr or headers of packets.
→ restrict some port nos
→ set flags.
→ protocols.

Stateless firewall

Stateful firewall

↓

More Efficient

→ uses cache memory

Resides in both network & transport layer.

② Network Address Translation [NAT]

→ Modifies IP addr of devices.

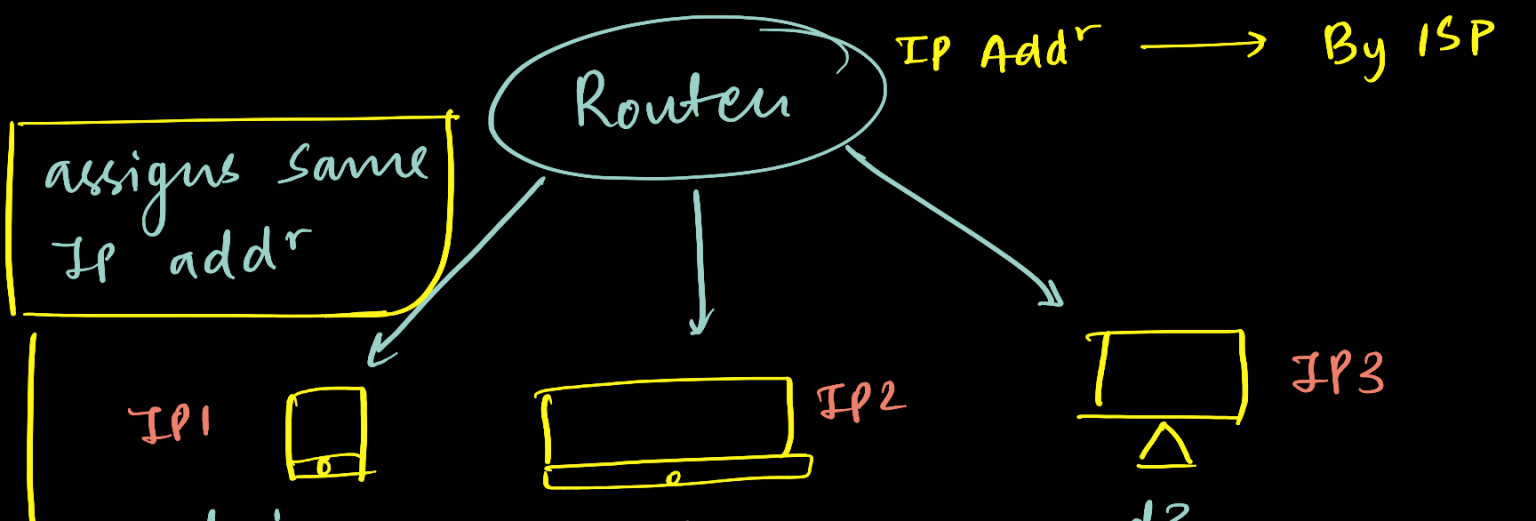
→ To slow down consumption of IP addr.

DATA LINK LAYER [works with frames]

→ receives packets from network layer.

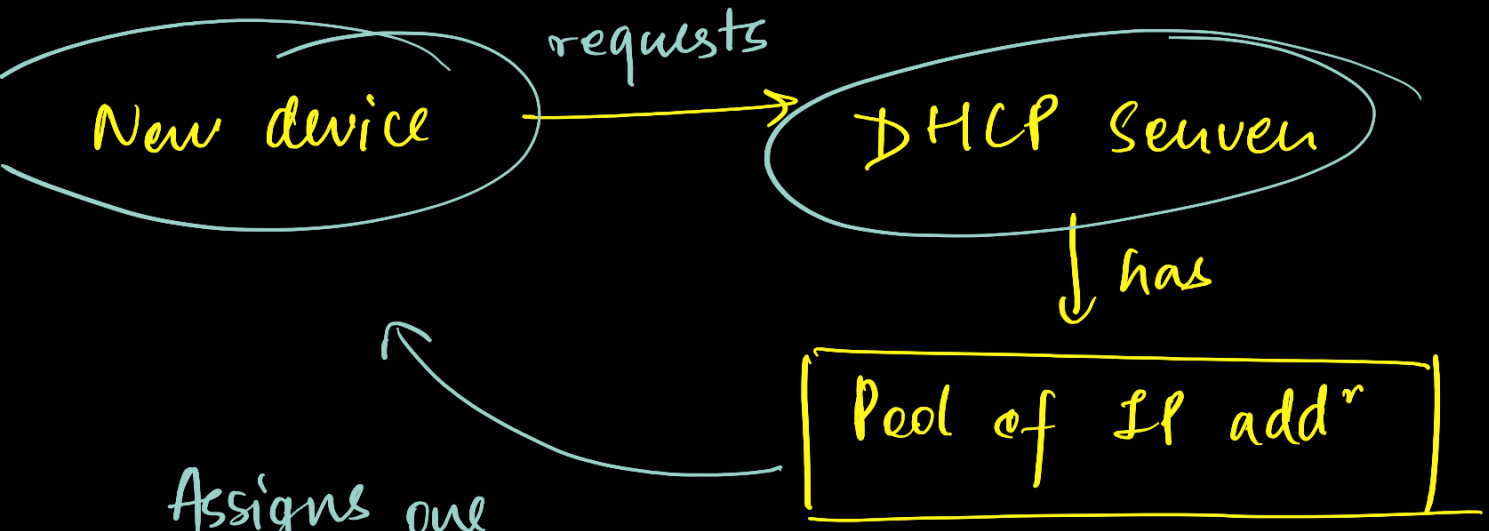
→ responsible for sending these packets over a physical link.

eg. House / Office router.



device 1 d2 d3

using DHCP : Dynamic Host Configuration Protocol



* Many devices connected to each other in a LAN, will have a data link layer address.

MAC Address

ARP Cache

Address Resolution Protocol