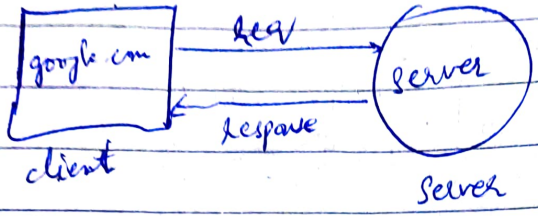
Computer Networking

Client - Server Architecture :-



Protocols :- Rules defined by Internet Society.

TCP :- Transmission Control Protocol.
→ Data will reach destination and be like corrupted in midway.

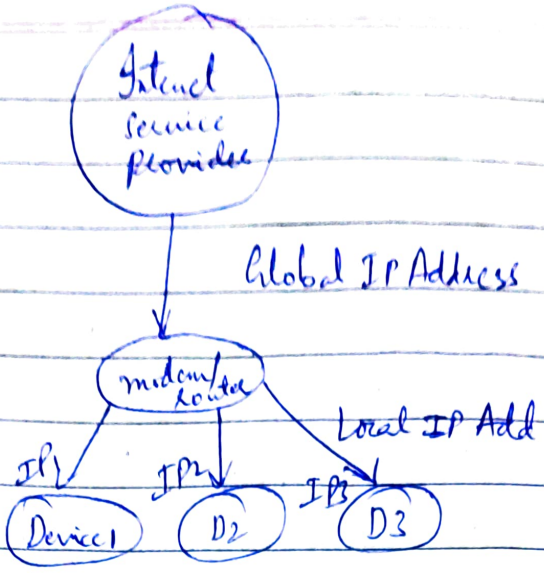
UDP :- All data may not be reaching
(User Datagram Protocol)

HTTP :- Hyper Text Transfer Protocol
→ The format of the data being tranferred b/w client and server.

How data is transferred?
→ Data is not sent at one single go but comes in packets or small chunks.

IP Address :-

(X.X.X.X) → format
↓
(0-255)



Local IP Address set by DHCP.

D1 ⇄ modem (Google.com) → ISP → Internet

Modem decides who has requested (which device) using NAT.

Once we get data (response) back to device (D1)
→ which application in D1 has requested data is identified by port numbers.

IP Add. identify the computer.
Port identify the application.

Port Numbers:-
→ 16 bit number.
→ Total port number $2^{16} \approx 65,000$ (approx)

All the HTTP stuff will happen on PORT = 80
MongoDB = 27017

→ Ports from 0-1023 are reserved ports.
(if we want to host application b/w 0-1023 we cant do that)
→ Port from 1024-49152 for some specific applications (Ex:- Mongodb, SQL)

1mbps = 1000000 bits/s

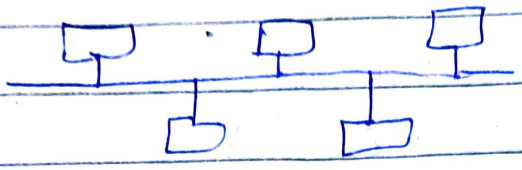
LAN :- Local Area Network (for small house/office).
Ex:- Ethernet cable, wifi.

MAN:- Metropolitan Area Network (across city use)

WAN:- Wide Area Network (across countries)
Ex:- Optical fibre cables
→ SONET :- Synchronous Optical N/w.
→ Carries data using optical fibre cables.
→ Frame Relay:- Connect the LAN to WAN.

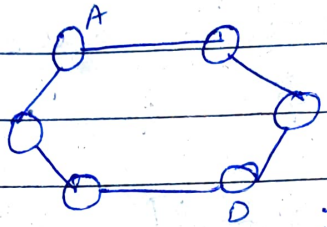
Topologies: (How computers are connected)

→ BUS



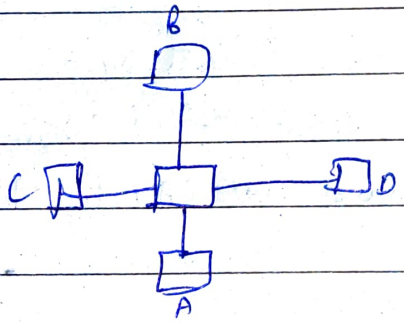
→ RING

Every system communicates with each other



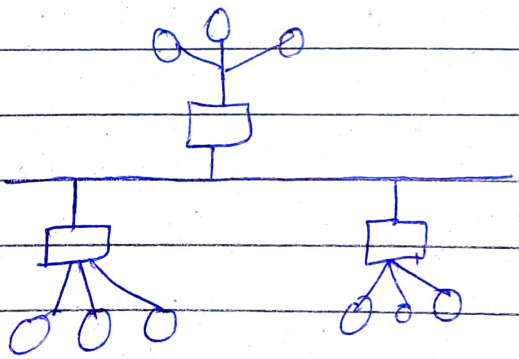
If we want to communicate to A from D it should go through all systems.

→ STAR



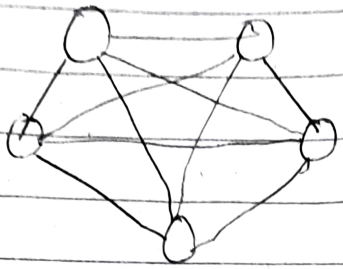
One central devices connected to all devices

→ TREE (BUS - STAR)



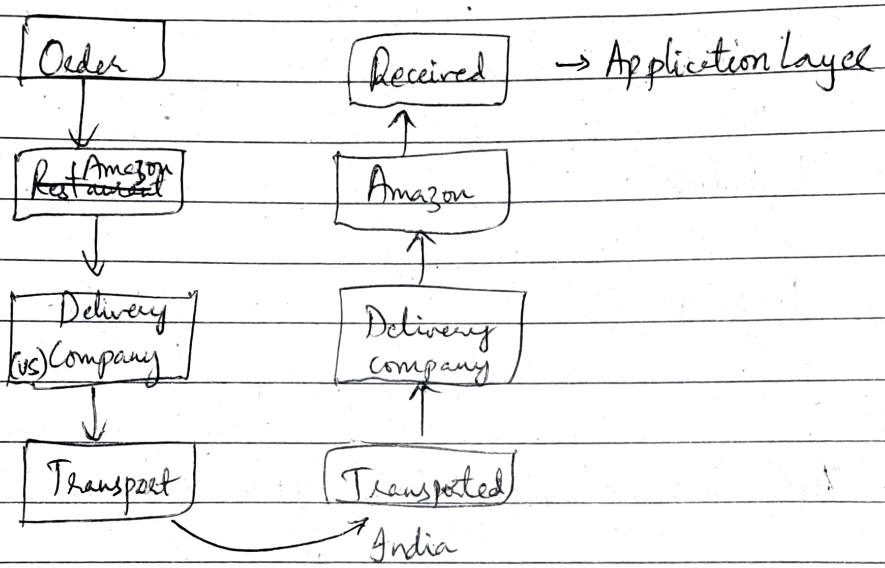
→ MESH

Every Single system connected to every single system



→ Expensive
→ Scalability Issues.

Structure of the Network:



OSI Model:

Open system Interconnection Model

Application

Presentation

Session

Transport

Network

Data Link

Physical

① Application → implemented in software (like apps).

② Presentation → data was sent from app layer.
Data which we get will be in the form of numbers, char, words etc → converted to machine representable binary format. (from ASCII to EBCDIC)
This is called translation.
→ Data also compressed here to reduce traffic.

Session ⇒ helps in setting up and managing the connections and enables sending and receiving data followed by termination of connected sessions.
Before session is established it do some sort of authentication.
Authorization takes place like whether you have permission to access file or not.

Transport :- It has its own protocol like how data is transported like UDP and TCP. (in 3 ways).
Segmentation :- data received divided into small data units called segments. Every segment will contain source and destination's port number and sequence number (seq. no helps to reassemble segments in exact order)
Flow control :- Controls amount data is being transported
Error control :- Works with data which is lost/corrupted

Network :- Works for transmission of received data segments from one computer to another that is located in diff. network.
→ Router lives in network.
Logical addressing → IP addressing is done.
→ Assigns sendess and receives IP address to every data segment and forms an IP packet
⇒ Routing & transportation of packets happen here.
→ load balancing also happens here.

- Data Link:- Allows to directly communicate with computer or host.
  Physical Addressing is done here which means appln data to be sent.
  Ex:- MAC Address. (12-digit alphanum of N/w interface of comp)
  MAC Add of sender and receiver are assigned to data packet to form frame.

- Physical → Contains hardware like wires.

TCP/IP Model:- (most used practically)

Application

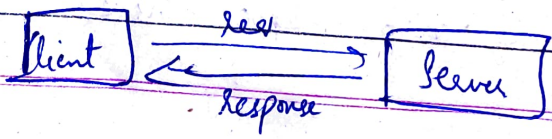
Transport

Network

Data Link

Physical

- Application:-
  → Users interact with these. (Ex:- whatsapp, browser etc)
  → Have some protocols.
  → Client - Server Architecture.



Protocols :-

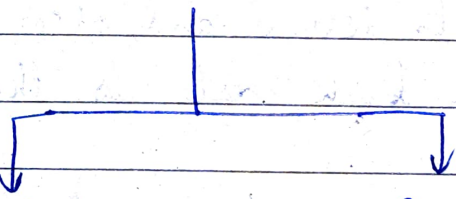
TCP/IP :-

- → HTTP
- → DHCP
- → FTP
- → SMTP
- → POP3 & IMAC
- → SSH
- → VNC
- → ~~TELNET~~

→ TELNET :- Terminal emulation that enables user to connect to remote host or device. (PORT :- 23)

→ UDP :- Stateless connection (data may be lost in this).

Program :- whatsapp



Process (one of feature of program)

Send a message

Record a video

Thread (Part of process) (process can have multiple thread)

Set up page

Open Camera

Sockets:-

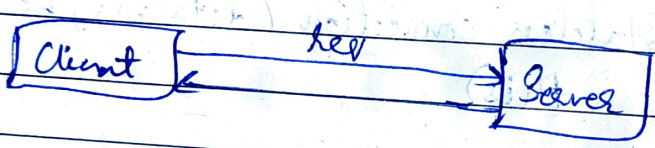
Interface between process and Internet.

Ports:-

→ Tells which application we are working with.

→ If there are multiple tabs opened in chrome. If data is requested, how do you know that which instance of chrome it should go like tab1, tab2. There is some called EPHEMERAL PORTS.

HTTP:-



→ It is a client-server protocol which tells how u send data to server and also tells how the server send back data to the client.

→ Every application layer protocol also requires some transport layer protocol.

→ HTTP uses TCP (app layer → transport layer)

→ It is a stateless protocol (server does not store any info of client by default)

HTTP Methods:-

1) GET 2) POST 3) PUT 4) DELETE

Status Codes:

1xx → Informational codes.
2xx → Success codes.
3xx → Redirecting codes.
4xx → Client error codes.
5xx → Server error codes.

Cookies:-

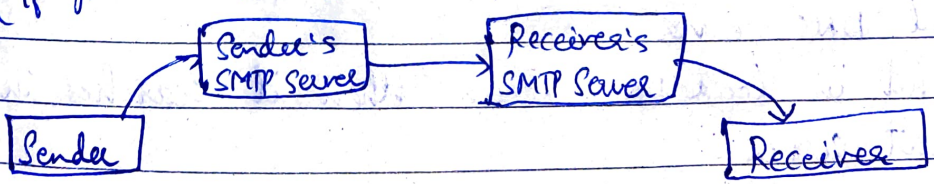
→ It is a unique string.
→ It is stored on a client's browser.

→ When we visit application for first time it will set a cookie. ~~Whenever~~ After that, whenever we make request, a cookie will be sent in request's header. Then server checks the user from which request is coming and set the state for that.

How EMAIL works?

→ It uses SMTP for sending email.

(if gmail to yahoo)



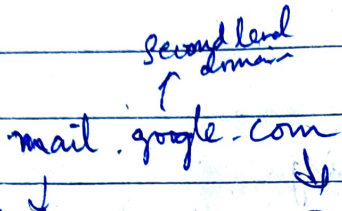
if both are same the connection does not happen.

→ POP3 & IMAP (download and delete mails)

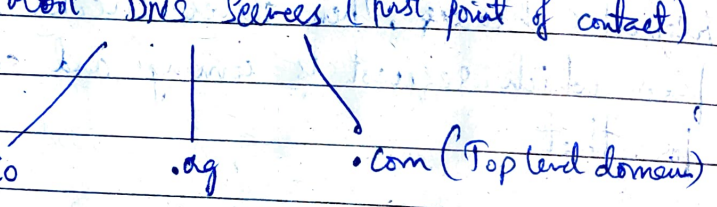
DNS (Domain Name System):

When we type www.google.com, it will use DNS to find IP address of google server.

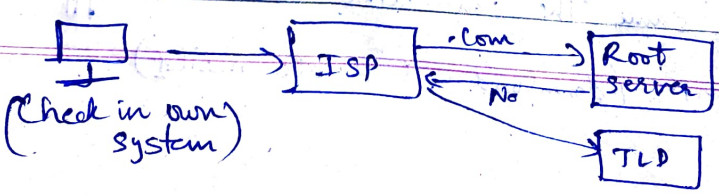
→ HTTP going to use DNS and convert URL to IP Address.



→ Instead of storing in a single database, all three are stored in different database.



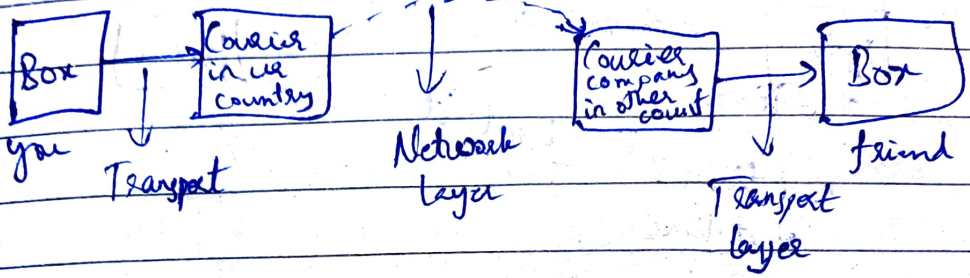
→ When we visit any website first time, it stores value of IP Address in local cache in our system.
→ If not able to find in local system, it uses local DNS server.
→ If not in local DNS server then it searches in root servers.
→ If it dont have then checks in TLD.

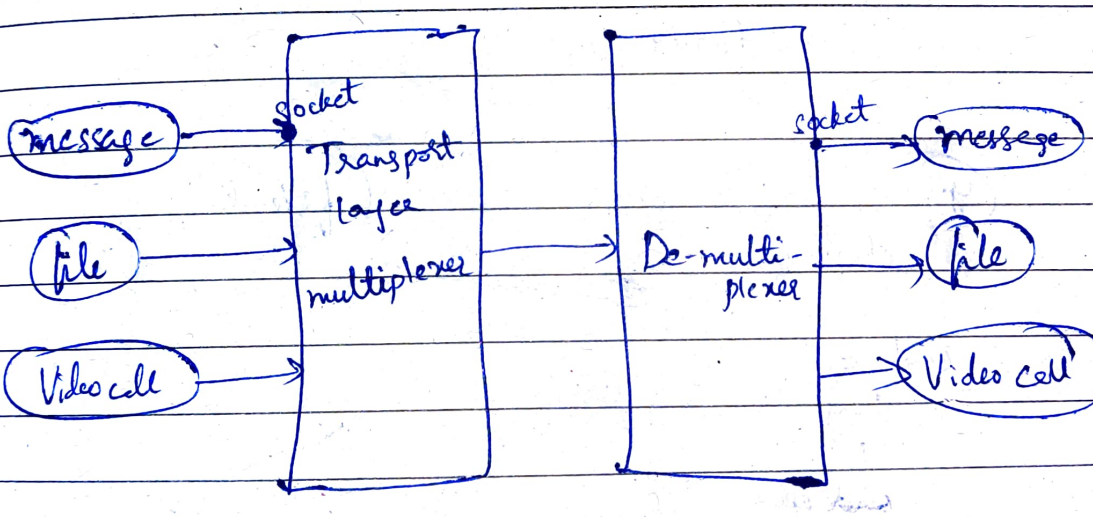


Transport layer:-

→ Data transfer from network to the applications is done by transport layer.

Ex:





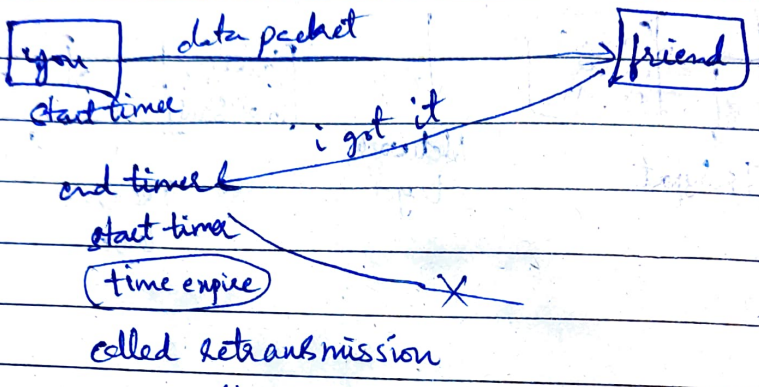
→ message will be sent to sockets.
→ Sockets have port numbers.
→ Transport layer will attach port numbers to sockets to data (as data travel in packets)
→ Transport layer also takes care of conjestion control (traffic)
→ Conjestion control algorithms built in TCP.

Checksums:

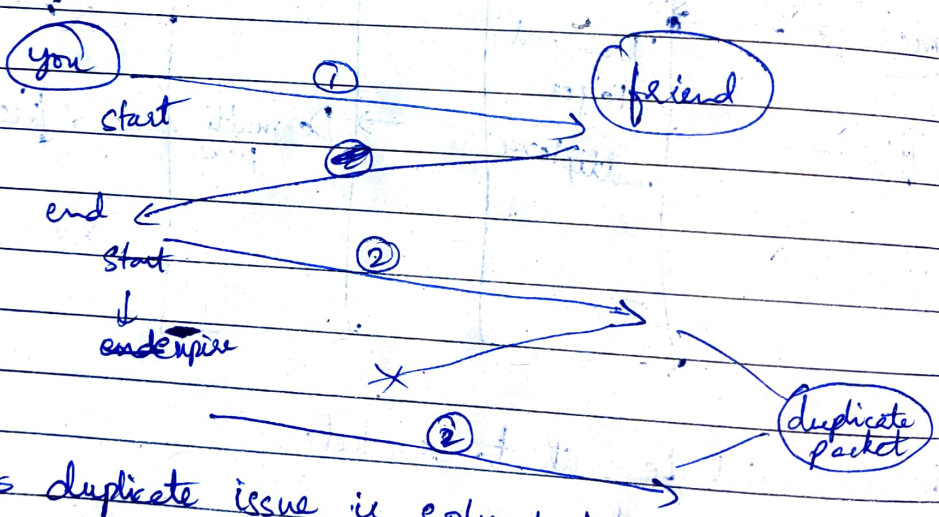
→ It is used to prevent data corrupt or data being delayed or data is not sent properly.

Timers:

→ If data is lost.



called retransmission timer.

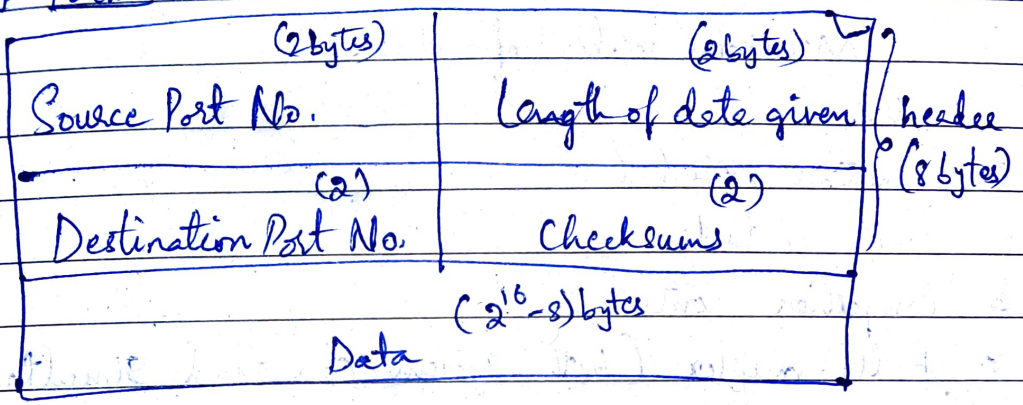


This duplicate issue is solved by sequence numbers.

UDP (User Datagram Protocol):

→ Data may / may not, be delivered.
→ Data may change
→ Data may not be in order.
→ It is connectionless protocol.
→ UDP uses checksums.

UDP Packet:-



Use Cases:-

→ Its fast
→ Video conferencing apps.
→ DNS → UDP
→ Gaming.

TCP (Transmission Control Protocol):-

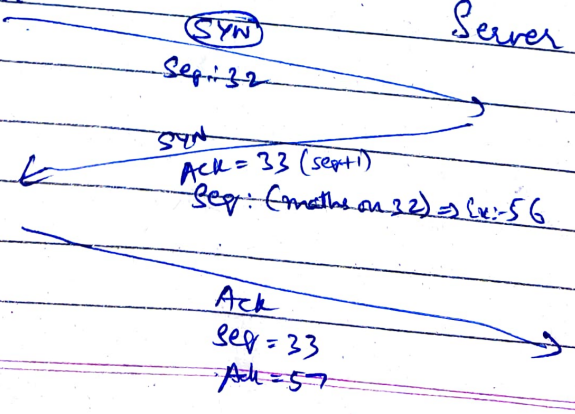
→ Application layer sends lots of raw data.
→ TCP segments this data → divide in chunks, add header, checksum etc.
→ It may also collect the data from other layers.
→ Congestion control.
→ Takes care of 2 things:-
  (When data does not arrive.
  (Maintains order of data.

Features:-

- Connection oriented.
- Error control.
- Congestion control.
- Full duplex. (both system can send simultaneously)

3-Way Handshake:-

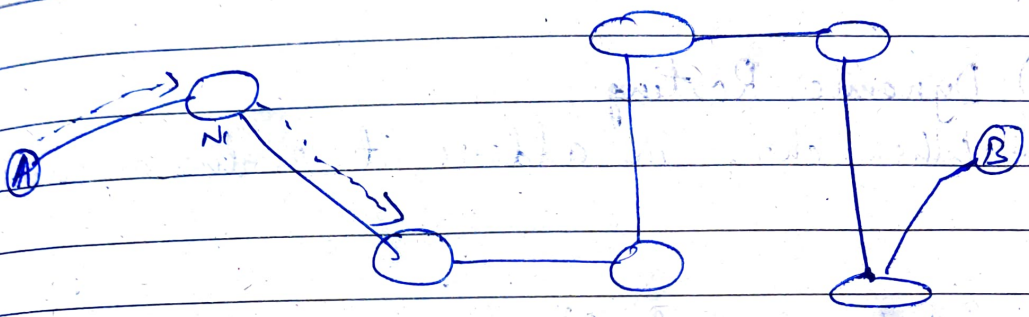
→ Client send connection req. as flag called synchronization flag and sequence number.
→ Then server send acknowledge flag and seq. number.



Network Layer:

Transport layer → segments (data)
N/W → packets
Data link → frames

→ Here we work with routers.



→ Data packets (N1) will be sent to router.
→ The router checks the N/wing address in its forwarding table.
(as packets contain receiver add, sender add and data).
→ If router finds that data to be send is somewhere to east, then it will send to next router.
This is called hop-by-hop forwarding.
→ Every router has its own N/w address.

192.168.2.30
N/W Add (Subnet ID) — Device add (Host ID)

Control Plane:

→ It creates tables like (routing table, forwarding table)

→ Two types of routing that creates tables.

1) Static Routing:-

→ Adding address manually.

→ Time consuming.

2) Dynamic Routing:-

→ When change in address, it evolves accordingly.

IP (Internet Protocol):-

IPV4:-

→ 32 bit number with 4 words.

IPV6:-

→ 128 bits.

Class of IP Address:-

A → 0.0.0.0 - 127.255.255.255

B → 128.0.0.0 - 191.

C → 192.0.0.0 - 223.

D → 224.0.0.0 - 239.

E → 240.0.0.0 - 255.255.255.255

→ Subnet masking is masking the n/w part of IP Address.

→ Variable length subnet :- we can set our own length of subnet network.

Reserved Address:

127.0.0.0/8

Ex:- localhost 127.0.0.1

These are loopback addresses.

Packets:-

Header is of 20 bytes.

IPv, length, Identification no., flags, checksum, etc.

TTL etc.

TTL ⇒ time to leave.

(after so many hops, packets won't reach destination)

IPv6 :-

IPv4 :- 2^{32} ≈ 4.3 billion IP Add.

IPv6 :- $2^{32\times4}$ ≈ lot of 3.4×10^{38}.

Cons:-

→ Not backward compatible.

→ ISP would have to shift, lot of hardware work.

→ ~~These are 8 numb~~

→ There are 8 numbers, every ~~digit~~ number is hexa decimal digit.

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

↓

hexadecimal
(16 bit)

Middleboxes:-

Ex:- Firewall —connected→ Global Internet

—connected→ Your own network

Filter the IP packets based on some rules:-

→ Address
→ Modify packets
→ Flags
→ Protocols.
→ Port nos.

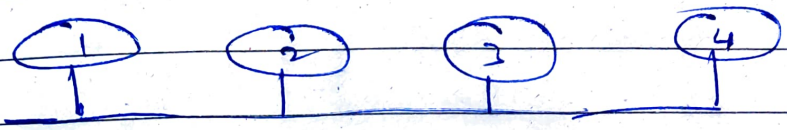
NAT (Network Address Translation):-

→ Mapping an IP Add space into another by modifying IP Add information in header of packet.

Data Link Layer:-

→ Data packets which we received from network layer, data link layer is responsible to send this packets over a physical link.

→ Whenever a new device is added, it connects to DHCP server.
DHCP server have pool of IP address.



→ Something needs to send to device 4.
→ Device 1 will check in its own cache for data link layer Address of 4.
→ If not, it asks the devices connected. it is known as ARP cache (Address Resolution Protocol)
→ 1 will send frame to other devices.

(Frame)
DLLA of sender
IP Add of destination.