Computer Network (CN)

- Network:
  It is a group of system of interconnected people or items.
  Computers connected with each other with cables or wireless is called computer networks.
  (Need of CN to share files, for connecting with others, etc)

- Internet:
  In a nutshell, internet is a network of computer networks. Complex web of interconnected CN.

History
1957 → US → ARPA (Advanced Research Project Agency) → 1960's–1970's →
- 1957 → Soviet Union → Sputnik (satellite)
- 1960's–1970's → Communication system for ARPA's computers to talk → 1969 → ARPANET (used TCP at that time)
- → 1980's (started using TCP/IP) → Internet
- → 1990's → CERN (European org. for nuclear research) → Hyperlink based docs → Tim Berners Lee introduced WWW (MOSAIC, NETSCALE)
- ~~1990~~ ~~Tim Berners Lee introduced www~~

- Protocols:

Network protocols -

These are set of rules and regulations setup to communicate and share data over network.
eg. HTTP, UDP, TCP, etc.

Packets -

In order to share data, we can't send big chunk of data over the network. So we divide the data in smaller chunks called as packets.

Address -

Sending messages over network requires the destination details. This detail uniquely identify the end system is called as address.

Ports -

Any machine could be running many network apps. In order to distinguish these apps for recieving messages we use ports (port number).

(IP Addr + port)
↑ machine ↑ software/app/program
socket

port helps you to get packets to specific process on host.
Every process has 16-bit port number.
range: $0 - 2^{16} = 65535$

0 - 1023 - well known ports (for specific apps)
ex Port 80 → http
Port 443 → https

1024 - 49152 → Registered ports
These are used by specific, potentially proprietary app/process that are known but not system defined.
eg. SQL server - 1433
mongoDB - 27017

49153 - 65535 → dynamic ports (Future purpose)

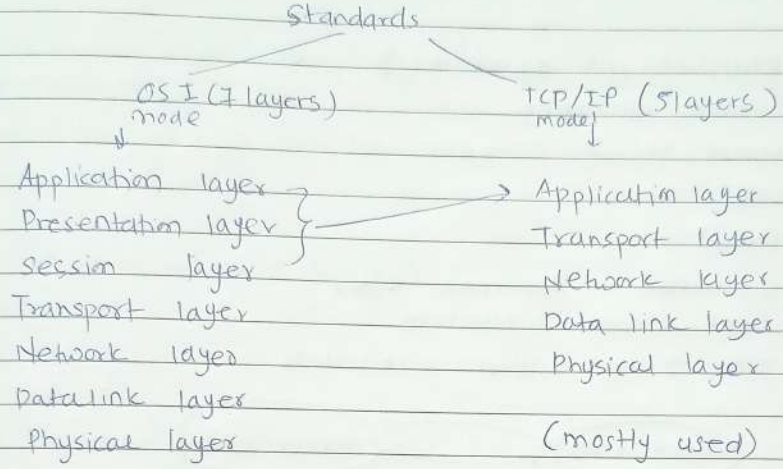
- Access Networks.
These are media using which end systems connect to the internet.

Network interface adapter:
It enables computer to attach to a network. As there are different types of networks, it acts as a single unit to connect to any network.

DSL (Digital subscriber line)
DSL uses existing telephone groundwork lines for internet connection. Generally DSL is provided by same company which supplies telephone service.

ISP (Internet service provider)
It is just a company that provides end users internet. ex. AT&T.

- Network protocol stack



Application layer - email serice, chat service, etc
Presentation - Presentation of data, compression, encryption
Session - user session management.
Transport - Divides big chuncks of data coming from above to small chuncks and manages these chuncks
Network - How routing of packets will be done on the network.
Data link - error/flow control, multiplexing and demultiplexing. handles addressing
Physical - optic fibre, cables, satellite.

(senders scenario)

1) Application layer

Roles →

1) writing/providing data off to the network.
2) Reading data from user.
3) Contains applications that helps users to interact with network (social media, api app, etc)

4) Error handling and recovery can be done

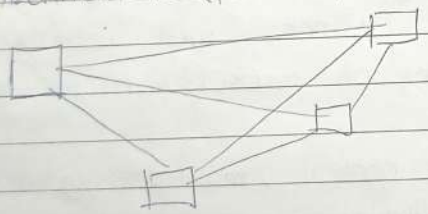
(Exists on end system only)
- Instant messaging
- WWW
- VoIP
- Email

~~Architecture in Application layer~~

1) Client - server architecture:
It is a 2 level architecture

server - This process controls access to a centralised resource or service such as a website / webapp.
client - Frontend where user interacts.

2) P2P Architecture (peer to peer)



better for speed ~~for ...~~

3) Hybrid architecture
combo of P2P + client-server

HTML HTTP
Hyper text, transfer protocol

It is set of rules that are used to transfer hypertext
(developed by (ERN)

data which

objects → web
other objects
some other
object has

URL (Uniform
parts
- protoco
- hostnam
- location
- argume

http ; //
protocol

Http de
will inte

clie

first m
second r

It is sto
about f
(As it d
accessing

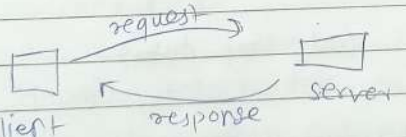
data which contains links to other files/objects.

objects → web pages are the main objects that contains other objects.
Some other objects can be mp3 files, .pdf, .jpg, etc and every object has a URL.

URL (Uniform resource locator)
parts
- protocol
- hostname
- location of file/object
- arguments

http://flipkart.com/image/23.jpg?q=50
protocol | hostname | filepath | extra args

Http defines whole procedure on how client and server will interact.



First msg → http request
second msg → http response
} Http is categorised in req-resp protocols.

It is stateless protocol i.e. servers don't store any information about the client.
(As it doesn't keep known anything about client and if client is accessing/requesting the same resource again and again then

server will send the resource again and again, there is nothing like optimisation)

A lot of application layer protocols depends on lower layer protocols of transport layer.
In transport layer there are 2 main protocols
1) TCP
2) UDP
(HTTP depends on TCP i.e. first TCP connection is set the HTTP connection is set on it and req-res cycle can be started)
There are two type of HTTP connections -
(web socket (ex. typing..)) → 1) persistent http (here remains persistant for req-res cycle)
2) non-persistent http (after one req-res cycle http connection breaks)
(browser refreshes website for every request)

HTTP req-resp msgs:
Any http msg are plain ASCII text.
(request object)
↳ host
method
status code (only imp data)
referral-policy
⋮

request header (metadata) | response header (metadata)

(The data that can't be sent in req object broz it get populated and against convention but is required to fulfill the request is sent through headers)

http methods
1) GET
2) POST
3) PUT (updating) Full

User-Agent → It sp
has different webpa
Accep-language → It s
Connection → close (no
Open (p
status code → 100-19
For responses 200-2
300-39
400-49
500-59

-Cookies
These are mainly
Http is a stat
session is requi
cookies are uniqu
server through
it is sent alon
server. This allo
it and hence

set cookie heade
When a serve
"set-cookie: v
This value is s

Http methods

1) GET
2) POST
3) PUT (updating) Full
4) PATCH (updating) Partial
5) DELETE
6) HEAD

User-Agent → It specifies the client. useful when server has different webpages that exists for different devices
Accep-language → It specifies the preferred language.
Connection → close (non persistent http connection)
Open (persistent)

Status code → 100-199 (Informational res)
for responses 200-299 (successful)
300-399 (Redirection)
400-499 (client error)
500-599 (server error)

-Cookies

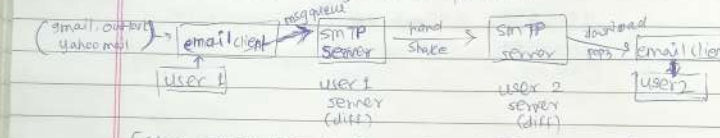
These are mainly concerned towards privacy.
Http is a stateless protocol but a lot of times user session is required.
cookies are unique identifier strings. These are set by a server through http headers. As soon as a cookie is stored it is sent along with subsequent http req to same server. This allows server to know who is contacting it and hence server sends the content accordingly.

set cookie header:
When a server wants to set a cookie it include "set-cookie: value" in http response.
This value is stored in the cookie file of browser

email - process of sending/recieving digital messages
gmail - email service provider

- SMTP (Email) (Push protocol) only
- for executing the functionality of email, SMTP is used one more protocol named pop 3 is used in combination with SMTP. one is used to send emails that are stored in users inbox and other is used to retrieve emails sent to user.
- SMTP also uses TCP protocol from transport layer
- Connection for SMTP is setup on port 25
- mail clients gives the actual UI for end users to send and recieve e mail viz gmail, outlook, etc
- working -

1) When an email is sent it is sent to the senders SMTP server using SMTP protocol.



(Also SMTP server is configured in the mail clients)

2) SMTP server places the email on message queue. Then SMTP server initates a connection with recievers SMTP server and conducts an initial SMTP handshake

3) Then finally it sends the email to recipients SMTP server

4) The email is downloaded from recievers SMTP server and then the client shows the mail.

(SMTP is only push protocol so we required POP3/IMAP to download the mail from server as they are pull protocol)

sending email — SMTP → push protocol

downloading email — POP3 / IMAP → pull protocol

gmail port - 587

(If recievers SMT
tries again and
a set threshold
and marks as "

- POP3 (post office
It downloads in
1) connect to the
2) Authorize (Chec
3) Transaction (Pro
4) Update (Upda

IMAP (Internet m
Emails are also
local copies of H
If an email is
it gets deleted

- Transport layer
- key responsibili
- Transport layer
Responsibility:
1) Segment Data: To
manageable piec
- can allow multi
application conne
multiple conversa
3) multiplexing Data
application pro
with 'headers' and

gmail port - 587 / 465

(If recievers SMTP server is offline, the sender SMTP sererver tries again and again after some ~~time~~ delta mins. There is a set threshold after which it stop sending the email and marks as "undelivered")

- POP3 (Post office protocol 3)

It downloads in 4 phases

1) connect to the server via TCP
2) Authorize (checks if it is valid reciever)
3) Transaction (Data transfer)
4) Update (Updates as delivered)

2 phases
download → keep
download → delete

IMAP (Internet message access protocol)

Emails are ~~also~~ kept on the server and not deleted.
local copies of the emails are cached on each devices.
If an email is deleted by the user manually then only it gets deleted from server.

- ~~Transport~~
- Transport layer

layer

- key responsibilities extends network ↑ to applications layer.
- Transport layer and its protocols ~~layer~~ reside on end system only.

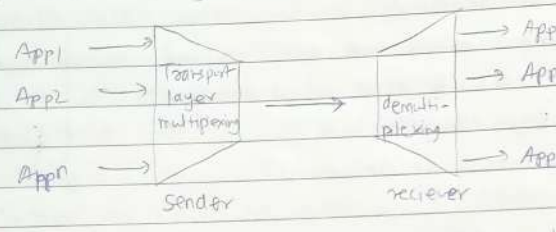
Responsibility:

1) Segment Data: Transport layer also divides the data into manageable pieces called "SEGMENTS" (TCP) or "DATAGRAMS" (UDP)
2) Can allow multiple conversations: Tracks each application to application connection or 'conversations' seperately, which can allow multiple conversations to occur at once.
3) Multiplexing Data - Gathering ~~multiple~~ data from multiple ~~processes~~ application processes of sender and enveloping that data with 'headers' and sending them as a cohole to intended reciever.

It allows the messages to be sent to more than one destination host via single medium.

4) Demultiplexing Data: Delivering recieved segments at recievers side to the correct app layer process is called as demultiplexing

(Headers are key part here)



5) Reliable Data transfer: There are a lot of network layer imperfections that transport layer is supposed to deal with

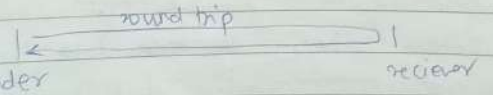
↳ Segment can be corrupted
↳ Segment can be lost
↳ segment can be reordered or duplicated

(TCP is reliable as it provides acknowledgment and UDP is unreliable as it doesn't provide acknowledgement)

Solutions for imperfections:

1) Checksum: It is error detection mechanism. Checksum (arithmetic sum) are attached to the segment which is verified by reciever. (for data corruption)

2) Retransmission Timers: segments may be lost. To resolve this retransmission timers are used. The value of retransmission timers is round trip time + delta



Case 1) Segment was not rec
Then the timer will expi
Case 2) segment was recie
ack got lost then we
will cause duplicacy.

3) Sequence number: This is
each segment to ident

Sliding window: It is used
are set of consecutive n
to ensure waitless data
sender and reciever agg

- TCP (Transmission Contro
Features
↳ Send data at appro
an ideal state).
↳ To segment data (recie
divides it into appropr
↳ Congestion control
↳ Identify and retransm

Applications (based on reliab
data
1) FTP (port 20 and
2) SSH (port 22) secu
3) email
4) Web browsing (H

features:
• Connection oriented (re
• Full duplex (sender

ore than one

segments at

er process is

t of network

is supposed to

ted

ment and UDP

de acknoledgement

ism. Checksum

segment which

(corruption)

- To resolve this

of retransmission

case 1) Segment was not recieved to reciever or got lost in betn
Then the timer will expire and we will retransmit the segment.
case 2) segment was recieved but no ack was sent or
ack got lost then we have to send the segment again which
will cause duplicacy.

3) Sequence number: This is an identification attached with
each segment to identify duplication or solve the order prob.

Sliding window: It is used to avoid overload on the reciever there
are set of consecutive numbers that sender uses in transmitting
to ensure wailless data transfer. At the beginning of sessim,
sender and reciever aggrees on window size.

{

- TCP (Transmission control Protocol)
  - Features:
    - Send data at appropriate transmissim rate (to avoid congestim on ideal state).
    - To segment data (recieves big chunck from app layer and then divides it into appropriat segments)
    - Congestion control
    - Identify and retransmit message (ack based protocol)

Applications. (based on reliability)
data control
1) FTP (port 20 and 21) File transfer protocol
2) SSH (port 22) secure shell protocol
3) email
4) Web browsing (Http/Https)

features:
- Connection oriented (persistent connectim until certain terminatim is followed)
- full duplex (sender <-> reciever)

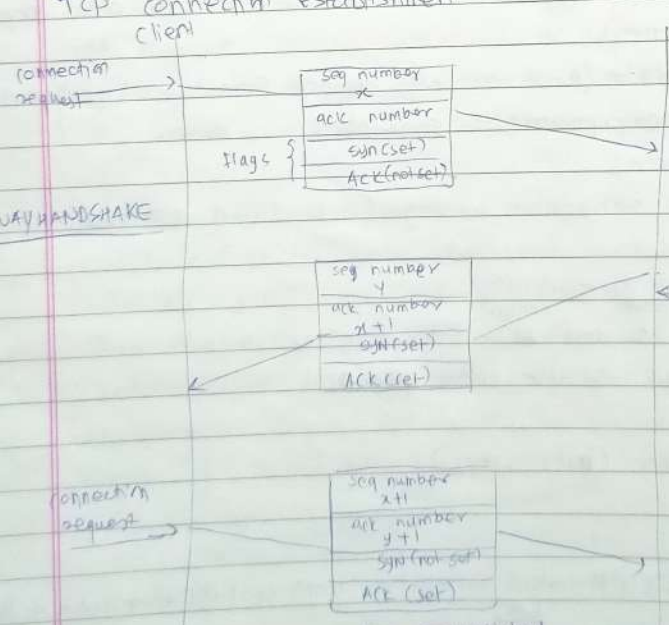
1 byte = 8 bit

- Point to point transmission (no broadcasting/multicasting)
- Error control
- Congestion control

Segment Header

Source Port No. (2 bytes)	Destination Port No. (2 bytes)
Seq Number (4 bytes)	
Ack Number (4 bytes)	
Header length (4 bits) / Reserved (4 bits)	8 Flags (8 bits) / Window size (2 bytes)
checksum (2 bytes)	Urgent pointer (2 bytes)
Options and Padding (upto 40 bytes)	

TCP connection Establishment



- UDP (User Datagra
- Used by apps tha
- either beoz applica
doesn't need rel
- much simpler
- It gets the dat
and then sends it

UDP header

Source Port No (2 bytes)
length (2 bytes)
Data

Features:
1) Faster (no rick
2) Reliability can
3) small header
transmission ove

Applications:
- Xbox
- Name translati

- UDP (User Datagram Protocol)
- Used by apps that don't need guaranteed delivery service of TCP.
- Either app itself handles it on its own or it just doesn't need reliable delivery.
- Much simpler than TCP and Fast
- It gets the data from application, converts it into UDP datagram and then sends it to network layer.

UDP header

Source Port No (2 bytes)	Destination Port No (2 bytes)
length (2 bytes)	Checksum (2 bytes)
Data	

Header 8 bytes

$2^{16} - 8 = 65528$ bytes

2^{16} bytes max datagram packet size

Features:
1) Faster (no ack)
2) Reliability can be built seperately.
3) Small header size gives an edge in terms of reduced transmission overhead and quicker transmission times.

Applications.
- Xbox
- Name translation is DNS

- Internet Protocol : IPv4

* IP in network layer protocol

It's design was based on following assumptions—

1) IP should provide an unreliable connectionless service. (so that packets are not bound to specific route and can be transfered using diff routes due to connectionless)

service → definition

2) Must have 32 bit fixed size add (x y z w)
8 bits 8bits 8bits 8bits
4 octets 32 bits

3) IP hosts should be able to exchange variable length packets.

Address Assignment -

Appropriate network layer address allocation is key to the effeciency and scalability of internet.

Note: The brute force approach to allocate IPv4 address to each host is attached to internet is FCFS with this solution the host in belgium will have 2.3.4.5 as IP while another in India would use 2.3.4.6.

Disadvantage: This force all the routers to maintain a specific route to all ≈ 1 billion hosts on internet which is not scalable. (32 bits, $2^{32} \approx 10^9$)

Subnetting (better than FCFS)

One soln is that router should only maintain route towards "Blocks of address" and not to individual host (as done in FCFS). for this blocks of IP addresses are assigned to ICP's. the ISP's then assign sub block of the assigned address space in a hefrarchical manner. These sub blocks are called as subnets.

A typical subnet groups all the hosts that are part of same enterprise. An enterprise network is usually

composed of several LAN
block of IP address fro
assigned to LAN.

IP has 2 parts
10001010 00011000 0000
subnetwork ID

(In simple words hefrarch
large block of IP add
called as subnets. This
tables size and effec

- Classful Addressing

• when a router nee
the subnet of desti
routing table to fin
• It was proposed
encode length of s
• we have 5 classes.

Class A :

• In class A first bit
• In class A, we have
• In class A first 8 b
Of bits represents
• In class A, no. of n
In class A, for each
• Default mask of
the network add
ex 13.7.13.2
255.0.0.0

composed of several LAN's interconnected by routers. A small block of IP address from the enterprise block is usually assigned to LAN.

IP has 2 parts

10001010 00011000 00001101, 0 0 0 0 0 0 0 1

subnetwork ID | Host ID

(In simple words heirarchical distribution of IP's is done to divide large block of IP add into smaller, more manageable groups called as subnets. This helps organise networks, reduce routing tables size and efficiently allocate IP add)

- Classful Addressing

- When a router needs to send a packet it must know the subnet of destination address to be able to consult its routing table to forward the packet.
- It was proposed to use higher order bits of address to encode length of subnet identifier.
- we have 5 classes → A, B, C, D, E

Class A:

- In class A first bit is always zero.
- In class A, we have 2^{31} IP add bcz 1st bit is fixed 0.
- In class A, first 8 bit octet represent network ID and rest of bits represents host ID.
- In class A, no. of network IP is 2^7 as first bit is fixed 0.
- In class A, for each network there are 2^{24} hosts (3 octets)
- Default mask of class A → 255.0.0.0 (This helps to identify the network add for class A by doing bitwise AND)

ex 13.7.13.2 → 13.0.0.0

255.0.0.0 network ID

Class A : 0 → 127
Class B : 128 → 191
Class C : 192 → 223
Class D : 224 → 239
Class E : 240 → 255

(27)

- The IP add will range from 0.0.0.0 → 127.255.255.255 for class A
- Every network has 2 special IP add
  ex X.0.0.0 → net id (x = 0 to 127)
  X.255.255.255 → broadcast add
  (so $(2^{24}-2)$ these are exact unique host per network in class A)
  (IF we send msg to X.255.255.255 it will be recieved by all hosts in particular network)

Class B:
- first two bits are always 10 so we can have 2^{30} unique IP address for class B.
  128.0.0.0 → 191.255.255.255
- First two octets are used to represent network ID, so 2^{14} unique networks IDs can be created for class B as first two bits are fixed (10)
- last two octets will represent host or 2^{16} hosts IDs can be created for class B
- Every network has special IP add
  X.Y.0.0 → net ID
  X.Y.255.255 → broadcast add
  (so $2^{16}-2$ unique hosts)
- Default mask for class B 255.255.0.0
  ex 184.3.180.0 → 184.3.0.0 (net ID)
  255.255.0.0

class C
- The first 3 bits are fixed as 110
- Range 192 → 223
- Total IP add → 2^{29}

- first 3 octets represent
- last octet is used
- Default mask → 255.2
  X.Y.Z.0 → net ID
  X.Y.Z.255 → broadca
  (2^8-2^{1st} For each ne

Class D
- First four bits are fixe
  For class D
- Range 224 – 239
- No network or host
  purpose (multicasting)

Class E
- First four bits are
  class E.
- Range 240 – 255
- These are reserved

Disadvantage of classf
- Wastage of IP add
- High maintenance
- More prone to error

Classless Addressing
- No class.
- only blocks.
  Notation:
  X.Y.Z.n/n

- First 3 octets represent network ID so 2^{21} unique networks
- last octet is used to represent host ID so 2^8 unique hosts
- Default mask → 255.255.255.0
  X.Y.Z.0 → net IP
  X.Y.Z.255 → broadcast IP
  ($2^8 - 2$ not for each network)

Class D:
- First four bits are fixed 1110 so 2^{28} unique IP add for class D
- Range 224 – 239
- No network or host are reserved as this is for special purpose (multicasting)

Class E:
- First four bits are fixed 1111 so 2^{28} unique IP add for class E
- Range 240 – 255
- These are reserved for special purpose (military)

Disadvantage of classful addressing
- wastage of IP add
- High maintainance and time consumity
- more prone to error

- Classless Addressing
- No class.
- only blocks.

Notation:
X.Y.Z.W/n → (mask / no. of bits to represent block)

ex. 200.10.20.40 /28

How to calculate mask

11111111 11111111 11111111 0000 => 28 | 4

28 bits are set — Block Id — not set — Host ID — block ID — host ID

255.255.255.240

(So 2^{28} block ID can exist and (2^4-2) unique host can exist)

Rules for classless addressing:

- Address should be continuous
- No. of addresses in a block must be in power of 2
- First add or network add should be divisible by size of block.

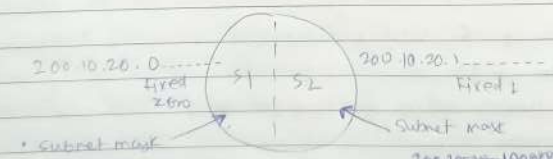
Subnetting in classful addressing

- Dividing a big network to small networks.
- No effect should be done on network ID.

(Suppose we have class C IP add and want to create two subnets)

IP = 200.10.20.x

$2^{\text{bits}} = 2^1 = 2$ bits host



Subnetting in mask is gene

- IPV6

Goal of IP

- Stick many
- 32 bit add
- So total I
- General

IPV6 entry

- Work start
- Basic pack

Address Stru

1) 128 bit a
2) Seperated
3) Used 8
as a blo
4) can omit
5) use h
ex

Address
Similar to
Note: We can a
address
and last
Address

IPV4 to IPV6 — transfo

subnetting in classless:
mask is denoted by /n

→ IPV6
Goal of IP address (v4)
- Stitch many network together
- 32 bit address
- So total IP address available → 2^{32}
- General utilization is ~35%

IPV6 entry:
- work started in 1996
- Basic protocol published in 1998

Address structure of IPV6
1) 128 bit address → 2^{128} IP addresses exists
2) Seperated into 2 parts → Subnet prefix and host suffix
3) uses 8 octets and can be seperated using hexadecimal as 8 blocks of 16 bits each.
4) can omit single row of zeros using ::
5) use brackets in URL:
ex. http://[2001:470:806:1::9]:80

Address assignment:
Similar to IPV4 and depends on ISP (Regional Internet registry)
We can generate IPv6 address from subnet/64 and ethernet address. Ethernet has 48 bit address, initial 24 bits denotes manufacturer and last 24 bits for device. Convert 48 bit ethernet address to 64 bit host ID by sticky oxiffe in the middle.