

Computer Networking

What is Computer Network?

A computer network is a collection of interconnected computing devices that can communicate and share resources (such as files, printers, internet connection, or applications) by following a set of rules called network protocols.

Key features of a computer network :-

- ① Connectivity
- ② Resource sharing
- ③ Data communication
- ④ Scalability
- ⑤ Reliability
- ⑥ Security

Types of computer Network :-

- * LAN (Local area Network)
- * MAN (Metropolitan Area Network)
- * WAN (Wide Area Network)
- * PAN (Personal Area Network)

O/I Model (Open Systems Interconnection Model) :-

- * The O/I Model is a conceptual framework that standardizes how computers communicate over a network.

7 layers of O/I Model :-

(i) Application layer (Layer 7) :-

- * Closest to the user - provides services directly to applications.
- * Handles: emails, browsing, file transfers, chatting
- * Protocols: HTTP, HTTPS, SMTP, FTP, DNS, SNMP.
- * Ex: - When you open Chrome and visit Google, the browser works at this layer.

(ii) Presentation Layer (layer 6) :-

- * Responsible for data translation, encryption, and compression.
- * Ensure data from the application layer is in a readable format for the receiving system.
- * Functions :-
 - (i) Data translation (ASCII \leftrightarrow Unicode)
 - (ii) Encryption / Decryption (SSL/TLS)
 - (iii) Compression (JPEG, MP3).
- * Ex: - HTTPS encrypts your data before sending.

(iii) Session layer (layer 5) :-

- * Manages sessions (connections) b/w two systems.
- * Functions :-
 - (i) Establishes, manages, and terminates sessions.
 - (ii) Handles authentication and authorization.

Ex:- Logging into a banking website \rightarrow session is created until you log out.

(iv) Transport layer (layer 4):-

* Ensures reliable data transfer b/w two devices.

* Functions:-

- (i) Error detection & correction
- (ii) Flow control (avoiding overload)
- (iii) Segmentation & reassembly of data.

* Protocols:-

(i) TCP (Transmission control protocol) \rightarrow Reliable, Connection-oriented.

(ii) UDP (User Datagram Protocol) \rightarrow Faster, no guarantee.

* Ex:- WhatsApp call uses UDP, but sending a file uses TCP.

(v) Network layer (layer 3):-

* Responsible for routing data packets from source to destination.

* Uses logical addressing (IP addresses).

* Functions:-

(i) Packet forwarding.

(ii) Routing via routers.

(iii) Fragmentation (breaking large packets)

* Protocols:- IP (IPv4/IPv6), ICMP, OSPF, BGP.

(vi) Data link layer (layer 2):-

* Provides error detection and correction from node to node.

* Uses MAC addresses.

* Divided into

(i) LLC (Logical Link Control)

(ii) MAC (Media Access Control).

* Devices :- switches, bridges

* Protocols :- Ethernet, PPP, ARP, VLAN

* Ex :- when your laptop connects to wifi, the data link layer handles communication with the router.

(vii) Physical layer (layer 1) :-

* Lowest layer - deals with physical transmission of data as electrical/optical/radio signals.

* Functions :-

(i) Transmission media (Cable, fibre optics)

(ii) Bit-by-bit delivery.

(iii) Hardware specification (voltage, pin layout)

* Devices :- Hubs, Repeaters, Modems, Cable.

* Example :- The actual Ethernet cable or wi-fi signal transmitting binary data (0s and 1s).

TCP/IP Model :-

* The TCP/IP model (Transmission Control Protocol/Internet protocol model) is a practical networking model that defines how data is transmitted over the internet.

Layers of TCP/IP Model :-

* It has 4 layers :-

① Application layer (Top layer)

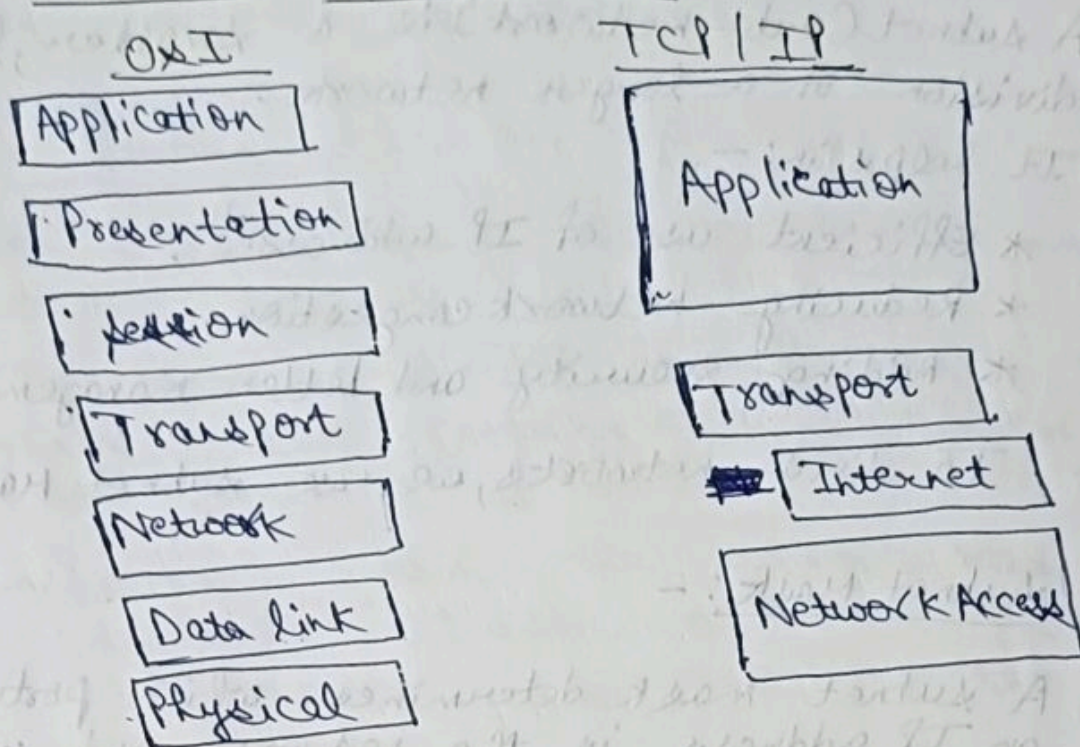
② Transport layer

③ Internet layer → Network layer

④ Network Access layer → Data link layer



TCP/IP Reference Model :-



IP address :-

An ip address (Internet Protocol address) is a unique numerical identifier assigned to each device connected to a network.

There are two main version :-

* **IPv4** : 32-bit address, written in dotted decimal format.

E.g. $\rightarrow 192.168.1.1$

* **IPv6** :- 128-bit address, written in hexadecimal.

E.g. $\rightarrow 2001:0db8:85a3::8a2e:0370:7334$

IPv4 Octet Range Breakdown (0-255) :-

Range	Class	Range
0 - 63	A	0 - 127
64 - 127	B	128 - 191
128 - 191	C	192 - 223
192 - 255	D	224 - 239
	E	240 - 255

Subnet :-

A subnet (sub-network) is a smaller, logical division of a larger network.

It helps in:-

- * Efficient use of IP addresses.
- * Reducing network congestion.
- * Adding security and better management.

To divide networks, we use Subnet Masks.

Subnet Mask :-

A subnet mask determines which portion of an IP address is the network and which is the host.

E.g. $\rightarrow 255.255.255.0$ (or /24) \rightarrow means first 24 bits are network and last 8 bits are host.

Why do we use subnets :-

~~\rightarrow Subnet (sub-network) is a~~

- \rightarrow Organize the network \rightarrow easier to manage
- \rightarrow Improve performance \rightarrow reduce unnecessary traffic.
- \rightarrow Increase security \rightarrow you can isolate sensitive systems.
- \rightarrow Efficient use of IP addresses \rightarrow avoid

Every device in a n/w has an IP address.
It has 2 parts:-

- ① Network Part: tells which n/w the device belongs to
- ② Host Part: tells the specific device.

→ A subnet mask is used to separate the new part from the host part.

Data routing:-

When a device sends info. to another device over the internet.

→ The data is divided into packets.

→ Each packet contains the IP address of the device it is destined for.

→ Routers within the network read the destination IP addr. on each packet and determine the best path for the packet to travel.

Routers communicate with each other to update & maintain records of the fastest and most efficient routes for data.

For device on diff. new data must travel through multiple routers across the internet.

Each router makes independent decisions about the best route for the packet based on destination IP address.

Need for classful Addressing:-

→ Simplified IP Allocation.

→ Faster Routing.

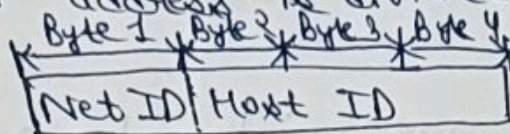
→ Scalability.

→ Interoperability.

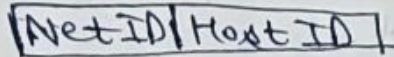
Classes of IP Addressing:-

32 bit IP address is divided into 5 classes

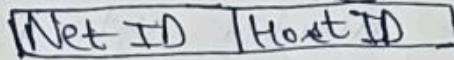
Class A



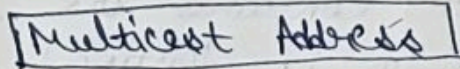
Class B



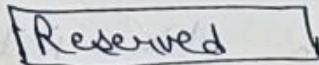
Class C



Class D



Class E



① Class A:-

In class A Net ID \rightarrow 8 bits

IP range

Host ID \rightarrow 24 bits

0.0.0.0 -
127.255.255.255

To calculate no. of hosts

$$2^{24} - 2 = 16,777,214 \text{ address}$$

First bit of class A is

Default subnet mask = 255.0.0.0

set to 0

Total no. of net. address
 $2^7 = 128$

② Class B:-

In class B

Net ID \rightarrow 16 bits

\rightarrow IP address range

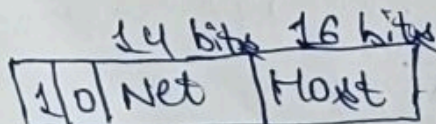
Host ID \rightarrow 16 bits

128.0.0.0 -
191.255.255.255

$$\text{No. of Host add} = 2^{16} - 2 = 65534$$

First 2 bits of first octet is 10

\rightarrow Default subnet mask



255.255.0.0

$$\text{Total no. of network address} = 2^{14} = 16384.$$

3) Class C:-

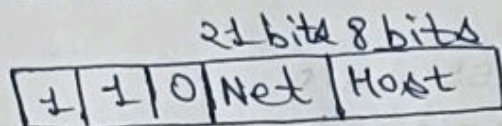
IP addr. belonging to class C are assigned to small sized networks.

Net ID \rightarrow 24 bits

Host ID \rightarrow 8 bits

$$\text{Total no. of host addr.} = 2^8 - 2 = 256 - 2 = 254$$

first 3 bits of octet 1 in class C are 110



$$\text{Total no. of net. addr} = 2^{21} = 2097152$$

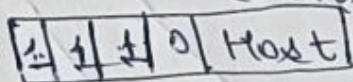
\rightarrow IP range 192.0.0.0 - 223.255.255.255

\rightarrow Default subnet mask \rightarrow 255.255.255.0

4) Class D:-

IP addr. of class D are reserved for multicasting.

first 4 bits of octet 1 is 1110.



\rightarrow remaining bits identifies diff multicast groups.

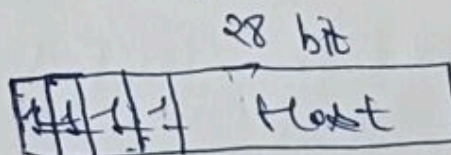
\rightarrow Multicast group is not tied to a particular network or set of hosts insted, hosts can join or leave group dynamically.

\rightarrow class D does not posses any subnet mask.

\rightarrow IP addr. range - 224.0.0.0 - 239.255.255.255

5) Class E:- IP addr. are reserved for experimental & research purposes

\rightarrow IP addr. range \rightarrow 240.0.0.0 - 255.255.255.255



\rightarrow Do not have any subnet mask.

CIDR (Classless Inter Domain Routing): -

CIDR is a notation (IP/Prefix) and a system that replaced old class-based addressing, making IP addr. allocation and routing more flexible & efficient.

e.g. 192.168.1.0/24

Convert or find subnet mask of given cidr range:-

| 8 \rightarrow 255.0.0.0

| 16 \rightarrow 255.255.0.0

| 24 \rightarrow 255.255.255.0

| 32 \rightarrow 255.255.255.255

Subnet mask for 172.16.130.45/15

11111111.11111110.00000000.00000000

| 15 means 15 n/w bits

so host bits = $32 - 15 = 17$ bits

Subnet mask will be - 255.254.0.0

11111111.11111110.00000000.00000000

No. of usable host = $2^{17} - 2 = 131072 - 2$
addr. = 131070

Every IP addr. is 32 bit long, grouped into 4 octet each 8 bits,

~~For ex~~ For ex:-

IP = 172.16.130.45

In Binary

172 \rightarrow 1010 1100

16 \rightarrow 0010000

130 \rightarrow 100000 10

45 \rightarrow 00101101

Full IP \rightarrow 1010 1100 . 0010000 . 1000 0010 .
00101101

\rightarrow Convert CIDR to subnet :-

IP /n = first n bits are 1 w/h, rest are 0 (host)

suppose with CIDR 172.16.130.45/20

Subnet mask 1111 1111 . 1111 1111 . 1111 0000
00000000

\rightarrow 255.255.240.0

\rightarrow Block size trick :-

Find the first octet that is not 255 in subnet mask

Block size = 256 - that octet
= 256 - 240 = 16

IP = 172.16.130.45/20

Block size = 16

Interesting octet = 130

130 \div 16 = 8 \rightarrow round down, $8 \times 16 = 128$

So, net = 172.16.128.0

→ Find Broadcast addr.

start with network address, add block size - 1 in the interesting octet and set all later octets to 255.

$$128 + 16 - 1 = 143$$

$$\text{Broadcast addr.} = 172.16.143.255$$

$$\text{First usable} = \text{nw} + 1 = 172.16.128.1$$

$$\text{Last usable} = \text{broadcast} - 1 = 172.16.143.254$$

$$\text{Host bits} = 32 - 20 = 12 \text{ bits}$$

$$\text{Total addresses} = 2^{12} - 2 = 4096 - 2$$

$$\text{which are usable} = 4094$$

DNS (Domain Name System):-

- * DNS is the internet phonebook.
- * It translates human name like www.example.com.
- * Without DNS, you'd need to remember raw IPs instead of easy domain names.

How DNS works:-

(a) Browser cache:-

- * First, the browser checks its cache.
- * If it already knows the IP, it uses that directly.

(b) Operating System Cache:-

- * If not found in the browser, it asks the OS (your computer's DNS cache)

(c) Local DNS Resolver (ISP/DNS Provider):-

- * If still not found, your request goes to a DNS Resolver (usually your IP's DNS or public ones like Google DNS 8.8.8.8, Cloudflare 1.1.1.1)

(d) Root DNS server:-

- * The resolver asks a Root DNS server.
- * Root servers don't know the exact IP but know where to find the TLD servers.

(e) TLD (Top-level Domain) server:-

- * Ex:- For `www.google.com` → TLD server for `.com` is contacted.
- * TLD server says: "Ask Google's authoritative DNS server."

(f) Authoritative DNS server:-

- * Finally, the resolver queries the authoritative DNS server of `google.com`
- * This server responds with the actual IP address (e.g., 142.250.182.14).

(g) Response Back:-

- * The resolver caches the IP
- * It sends the IP back to your browser.
- * Browser connects to that IP and loads the website.

Important concepts:-

- * TTL (Time to Live) → How long DNS record stays cached.
- * Recursive Query → Resolver does all the work to find the IP.

* Iterative Query : \rightarrow Resolver gives best kind, Client keep asking.

* Caching : \rightarrow speeds up DNS lookups, reduce load.

NAT (Network address translation) :-

* NAT (Network Address Translation) is a process where one or more private IP addresses are mapped to a public IP address, so devices inside a private network can access the internet.

How NAT works :-

① Private IP inside \rightarrow Public IP outside

* Your computer at home has a private IP (like 192.168.1.5)

* The internet does not understand private IPs.

② Router does translation :-

* When you open google, your request goes to your router.

* The router changes your private IP into its public IP (given by ISP).

③ Internet sees only Public IP :-

* Google sees the request from your router's public IP.

④ Reply comes back :-

* Google replies to the router's public IP.

- * The router checks its records and sends the reply back to your computer.

Why Nat is imp?

- * It lets private IP devices access the internet
- * It hides internal IPs for better security
- * It saves public IP addresses.
- * Without NAT, private networks couldn't connect safely to the internet

Port Number Masking :-

- * Port number masking happens in PAT (Port Address Translation), a type of NAT.
- * When many devices in a private network share one public IP, NAT uses different port numbers to keep track of each device's connection.
- * This way, the real private IP and port are hidden (or "masked") behind the router's public IP and a translated port number.

How it works :-

Imagine 2 laptops inside a home network:-

* Laptop A \rightarrow 192.168.1.10:1234

* Laptop B \rightarrow 192.168.1.11:1234

Both want to visit google at the same time.

① NAT router has only one public IP (203.0.113.5).

② It creates unique mappings by changing ports:-

* 192.168.1.10:1234 \rightarrow 203.0.113.5:40001

* 192.168.1.11:1234 \rightarrow 203.0.113.5:40002

- ③. Google sees requests only from 203.0.113.5, but with different masked ports.
- ④. When replies come back, the router uses its NAT table to send them to the right laptop.

TCP (Transmission Control Protocol): -

→ a connection oriented transport layer protocol, that ensures reliable, ordered and error-free delivery of data plus application over a n/w.

Before sending data TCP creates a connection using 3-way handshake.

SYN → SYN-ACK → ACK

3 way handshake: -

Imagine 3 way handshake as making a phone call.

Step 1: - SYN (Synchronize): You dial someone
⇒ Hello, can you hear me? I'd like to talk.

Step 2: - SYN-ACK: They respond
→ Yes I can hear you clearly, can you hear me too?

Step 3: - ACK: You confirm
→ Perfect I can hear you too.
Let's start ~~over~~ our conversation.

Actual

Step 1 :- SYN (Client to server):

Client says: I want to connect

Sends SYN flag = 1

Includes initial sequence no. 100

→ I'll start numbering my data from 100.

Step 2 :- SYN - ACK (server to client):

Server says: I am ready too.

~~Sends~~ SYN flag = 1 AND ACK flag = 1

Acknowledges client's sequence no (100 + 1) = 101

Sends its own sequence no 300

I got your 100, expecting 101 next.

I'll start from 300.

Step 3 :- ACK (client to server):

Client says = great let's start

Sends ACK flag = 1

Acknowledges server's sequence number - x (300 + 1 = 301)

I got your 300, expecting 301 next

Connecting established.

→ TCP Protocol ensures all data reaches destination. If packets are lost, TCP resends them.

If packet arrives out of order, TCP rearranges them.

Every packet has checksum to check errors.

Used in critical application like

- * web browsing.
- * Emails (SMTP, IMAP, POP3).
- * File transfer (FTP, SFTP).

UDP (User Datagram Protocol):-

→ a simple, connectionless and lightweight transport layer protocol that enables fast, low-latency data exchange over Internet Protocol suite.

→ UDP prioritizes speed over reliability.

→ Data is sent in small, independent, packets called user datagram.

Each datagram contains ~~the~~ source and destination port no. in its 8 byte header to identify the sending and receiving application.

* video & audio streaming

* VoIP (Voice over IP): enables realtime voice communication.

(Zoom, Skype)

* DNS (lookups).

DHCP (Dynamic Host Configuration Protocol) :-

- * It is a network protocol that automatically assigns IP addresses and other network settings to device (like laptops, phones, server) when they connect to a network.
- * Without DHCP, you'd have to manually configure IP addresses on every device.

How DHCP works :-

① DHCP Discover :-

- * Device broadcasts :- "I need an IP address!"

② DHCP Offer :-

- * DHCP server replies: "I can give you IP: 192.168.1.50, subnet mask, gateway, DNS."

③ DHCP Request :-

- * Device responds: "Yes, I want to use that IP."

④ DHCP Acknowledgement :-

- * Server confirms: "Okay, 192.168.1.50 is yours for now."
- * The device configures itself with the IP and settings.

Internet Protocol (IP):-

- * A rule book that make sure every device on the internet can be uniquely identified and can send / receive data packets properly.
- * IP works at the NW layer of OSI model.
- * Without IP, your data wouldn't know where to go.
- * IP decides the path your data takes across routers to reach destination.

Monitoring and troubleshooting tools:-

ping, traceroute → paths of packets
nslookup, dig → DNS troubleshooting
netstat, ss → open ports, connection
curl, telnet → testing connectivity

Load balancing → L4 → TCP/UDP
L7 → HTTP/HTTPS

Proxy:-

- * ~~##~~ In CN, a proxy server is an intermediate system that sits b/w a client and a server.
- * It acts as a bridge b/w clients & servers.

Thix Proxy can:-

- * Hide your real IP address
- * Filter or control access.
- * Load balance or provide security.

In networking terms:-

Client sends a request \rightarrow Proxy server,
forwards it \rightarrow to destination server.

Destination server replies to \rightarrow proxy server
 \rightarrow client.

Types of Proxy:-

① Forward Proxy:- (client side proxy).

* works for the client

* Client \rightarrow Proxy \rightarrow server.

* Ex:- A school proxy allow students
to access only educational
websites.

② Reverse proxy:- (server side proxy)

* works for the server.

* Client \rightarrow reverse proxy \rightarrow server(s).

* Ex:- Nginx or HAProxy handling
requests for multiple backend
servers.

③ Transparent proxy:-

* User is unaware of proxy usage

* often used for monitoring/filtering
traffic

SSL/TLS Termination :-

* Secure socket layer / Transport layer security

* SSL/TLS encrypts data b/w client and a server.

e.g. when you see https:// instead of http://, TLS is protecting the connection

→ SSL/TLS termination means the points where encrypted HTTPS traffic is decrypted into plain HTTP.

→ It happens at special device or software (like load balancer, reverse proxy or firewall) before the traffic reaches your actual web server.

Why do we use SSL/TLS termination:-

① Performance (offloading):-

* Encryption / Decryption is CPU heavy.

* Instead of every backend server doing it, a dedicated device (load balancer) does it once.

② * Centralized Certificate management.

③ Scalability:-

easier to scale backend server because they don't need SSL config.

④ Inspection / security tools :-

Since traffic is decrypted at termination point, security tools can inspect it more easily.

Variations :-

① SSL/TLS termination :-

decrypts at load balancer → forwards plain HTTP to backend

② SSL/TLS Passthrough :-

Load balancer does not decrypt, Encrypted traffic goes straight to backend, which decrypts.
(More secure end to end, but heavier on server)

③ SSL/TLS bridging (re-encryption) :-

Load balancer decrypts, inspects, then re-encrypts before sending to backend.

Load balancing :-

* Distributing incoming client requests across multiple backend servers to improve capacity & availability.

Basic algorithm :-

① Round robin :- sends requests sequentially across servers.

② Least Connection:- Send to the server with fewest active connections.

③ Weighted:- send more traffic to more powerful servers.

④ IP hash / sticky session → send same client IP to same backend

Firewalls:-

* Firewall is like security guard at the entrance of your building (network)

→ it checking who is coming or who is going out

~~→ It doesn't care if packets are~~

~~part of an existing~~

→ decides whether to allow or block the traffic based on rules.

Firewalls can be stateless or statefull

Types:-

① stateless firewalls:-

* Treats every packet independently

* Makes decisions based only on source / destination IP, port, protocol.

Ex:- Client sends a req → Firewall checks packet → allows or blocks it.

② Statefull firewalls :-

- * keeps tracks of the state of active connections.
- * knows whether a packet is part of an existing, valid connection.

Ex:- Client req a web page \rightarrow Firewall sees TCP SYN Packet \rightarrow remembers the connection \rightarrow allows the returning SYN-Ack / Ack Packets.