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)

OSI Model (Open Systems Interconnection Model) :-

* The OSI Model is a conceptual framework that standardizes how computers communicates over a network.

7 layers of OSI 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 ↔ Unicode)
    (ii) Encryption / Decryption (SSL / TLS)
    (iii) Compression (JPEG, MP3).
    ~~(iv)~~
* Ex:- HTTPS encrypt 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 → 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) → Reliable, Connection-oriented.
  (ii) UDP (User Datagram Protocol) → 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 transmiss-ion of data as electrical/optical/radio signals.
* Functions :-
  (i) Transmission media (cable, fibre optics)
  (ii) Bit-by-bit delivery.
  (iii) Hardware specification (voltages, 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 networ-king 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 :-

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Network Access
Physical	

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 decim-al format.
  E.g. → 192.168.1.1

* IPV6 :- 128-bit address, written in hexadecimal
  Eg. → 2001:0db8:85a3::8a2e:0370:7334

IPV4 Octets Range Breakdown (0-255) :-

Range	Class	
~~0 - 63~~	Class 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. → 255.255.255.0 (or /24) → means first 24 bits are network and last 8 bits are host.

Why do we use subnets:-

→ ~~subnet (sub-network) is a~~

→ organize the network → easier to manage
→ Improve performance → reduce unnecessary traffic.
→ Increase security → you can isolate sensitive systems.
→ Efficient use of IP addresses → 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 n/w 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 routers for data.

For device on diff. n/w 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

	Byte 1	Byte 2	Byte 3	Byte 4
Class A	Net ID	Host ID		
Class B	Net ID		Host ID	
Class C	Net ID			Host ID
Class D	Multicast Address			
Class E	Reserved			

① Class A:-

In class A Net ID → 8 bits
Host ID → 24 bits

So to calculate no. of hosts
$2^{24} - 2 = 16,777,214$ addr

Default subnet mask = 255.0.0.0

IP range
0.0.0.0 - 127.255.255.255

First bit of class A is set to 0

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

② Class B:-

In class B

Net ID → 16 bits
Host ID → 16 bits

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

First 2 bits of first octet is 10

	14 bits	16 bits
1 0	Net	Host

Total no. of network addr. = $2^{14} = 16384$.

→ IP addr. range
128.0.0.0 - 191.255.255.255

→ Default subnet mask
255.255.0.0

3) Class C:-

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

Net ID → 24 bits
Host ID → 8 bits

Total no. of host addr. $= 2^8 - 2 = 256 - 2$
$= 254$

first 3 bits of octet 1 in class C are 110

			21 bits	8 bits
1	1	0	Net	Host

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

→ IP range 192.0.0.0 – 223.255.255.255

→ Default subnet mask → 255.255.255.0

4) Class D:-

IP addr. of class D are reserved for multicas-ting.

first 4 bits of octet 1 is 1110.

				28 bits
1	1	1	0	Host

→ remaining bits identifies diff multicast groups.

→ Multicast group is not tied to a particular network or sets of hosts insted, hosts can join or leave group dynamically.

→ class D does not passes any subnet mask.

→ IP addr. range – 224.0.0.0 – 239.255.255.255

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

→ IP addr. range → 240.0.0.0 – 255.255.255.255

				28 bit
1	1	1	1	Host

→ 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 → 255.0.0.0
/16 → 255.255.0.0
/24 → 255.255.255.0
/32 → 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 addr. = $2^{17} - 2$ = 131072 − 2
= 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 → 10101100
16 → 0010000
130 → 10000010
45 → 00101101

Full IP → 10101100 . 00010000 . 10000010 . 00101101

→ Convert CIDR to subnet :-

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

suppose with CIDR 172.16.130.45/20

Subnet mask 11111111 . 11111111 . 11110000 . 00000000

→ 255.255.240.0

→ 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 ÷ 16 = 8 → round down, 8 × 16 = 128
so, net = 172.16.128.0

→ Find Broad cast addr.

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

$128 + 16 - 1 = 143$

Broadcast addr. = 172.16.143.255

First usable = n/w + 1 = 172.16.128.1
Last usable = broadcast − 1 = 172.10.143.254

Host bits = 32 − 20 = 12 bits

Total addresses which are usable = $2^{12} - 2$ = 4096 − 2 = 4094

DNS (Domain Name System):-

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

How DNS works:-

(a) Browser cache:-

* First, the browser check 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 ISP's DNS or public ones like Google DNS 8.8.8.8, Cloud-flare 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.goggle.com → TLD server for .com is contacted.

* TLD server says: "Ask Google's auth-oritative DNS server."

(f) Authoritative DNS Server:-

* Finally, the resolver queries the auth-oritative DNS server of goggle.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 records stay cached.
* Recursive Query → Resolver does all the work to find the IP.

* Iterative Query :→ Resolver gives best hint, Client keep asking.

* Caching :→ speeds up DNS lookups, reduce load

NAT (N/w 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 → Public IP outside

* Your computer at home has a private IP (like 192.168.1.5)
* The internet does not under private IPs.

(2) Router does translation :-

* When you open goggle, 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.

<u>Why Nat is imp?</u>

* 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

<u>Port Number Masking :-</u>

* 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.

<u>How it works:-</u>

Imagine 2 laptops inside a home network:-

* Laptop A → 192.168.1.10:1234
* Laptop B → 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 → 203.0.113.5:40001
   * 192.168.1.11:1234 → 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 b/w 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. Lets 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 servers sequence number – r (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 error

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 in sent in small, independent, packets called user datagram.

Each datagram contains 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, servers) 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 :-

* Devic 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 th IP and settings.

Internet Protocol (IP):-

* A rule book that make sure every device on the internet can be uniquely identi-fied and can send / receive data packets properly.
* IP works at the N/w 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 trouble shooting tools:-

ping, traceroute → paths of packets
nslookup, dig → DNS trouble shooting
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.

This Proxy Can:-

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

In networking terms:-

Client sends a request → proxy server; forwards it → to destination server.

Destination server replies to → proxy server → client.

Types of Proxy:-

① Forward Proxy:- (client side proxy).

* works for the client
* Client → proxy → server.
* Ex:- A school proxy allow students to access only educational websites.

② Reverse proxy:- (server side proxy)

* works for the server.
* Client → reverse proxy → 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~~ 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 balance, reverse proxy or firewalls) 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 accross 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 outs
  → ~~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 pack 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 → Firewall sees TCP SYN Packet → remembers the connection → allows the returning SYN-Ack / Ack Packets.