NETWORKING FUNDAMENTALS

TOPICS COVERED:

- Network
- History of Network
- Types of Network
- How Internet Works
- Network Architecture
- Network Topology
- Network Devices
- IP Address and its Types
- OSI and TCP/IP Models
- Ports and Protocols
- DNS,DHCP,NAT
- Firewalls and Load Balancers

Network:

A network is a collection of two or more connected devices (computers, servers, printers, etc.) that can share data, resources, and services with each other.

Internet:

The internet is a global system of interconnected networks that uses standard protocols (like TCP/IP) to allow communication and data exchange worldwide.

👉 In short:

- **Network = local or limited connection of devices.**
- **Internet = worldwide connection of networks.**

History of Computer Network:

The internet started in the late 1960s with ARPANET, a U.S. military project to connect computers. In the 1970s, TCP/IP protocols were developed, creating a standard for communication. By 1983, ARPANET adopted TCP/IP, and the modern internet was formed. In 1991, Tim Berners-Lee invented the World Wide Web, making the internet accessible to the public. The 2000s brought broadband and social media, while the 2010s introduced smartphones, cloud computing, and mobile internet. Today, in the 2020s, technologies like 5G, AI, and IoT have made the internet the backbone of global communication and daily life.

Types of Computer Network

A **network** can be categorized based on its size, coverage area, and purpose. The main types are:

- **LAN (Local Area Network):** Covers a small area like a home, office, or school. Example: Wi-Fi in your house.
- **MAN (Metropolitan Area Network):** Covers a city or a large campus. Example: University networks across different buildings.
- **WAN (Wide Area Network):** Covers a large geographical area, connecting multiple LANs and MANs. The **Internet** itself is the largest WAN.
- **PAN (Personal Area Network):** Very small network for personal devices like mobile phones, laptops, Bluetooth devices.

How Internet Works:

The internet is a global network of interconnected computers and devices that communicate using a standard set of rules called **TCP/IP protocols**. When you open your browser and type something like *www.google.com*, your computer doesn't actually understand names—it only understands numbers (IP addresses). So, your request first goes to your **ISP (Internet Service Provider)**, which acts like your gateway to the internet.

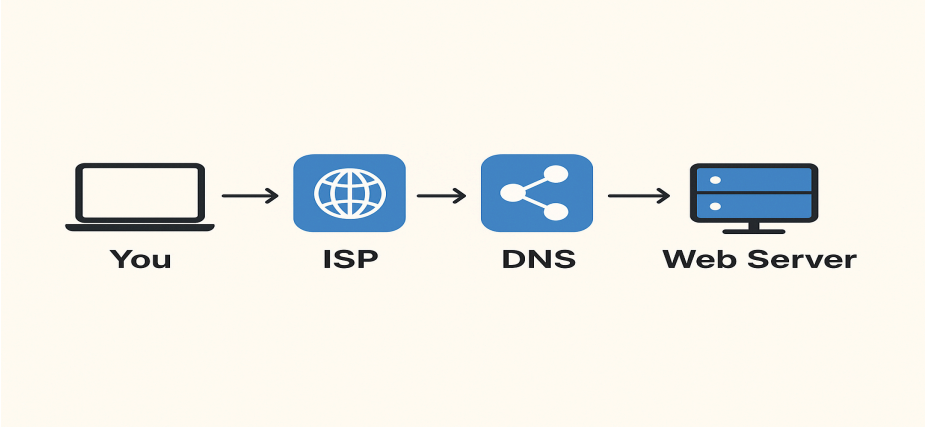
The ISP then asks the **DNS (Domain Name System)**, which works like a giant phonebook, to find the exact IP address of "www.google.com". Once the IP address is found, your computer now knows the "location" of the Google server.

Your request then travels across the internet through **routers and switches**. Think of routers as traffic managers—they check the best and fastest path to reach the destination server, even if it's thousands of kilometers away.

Finally, the request reaches Google's **server**, which stores all the information about the website. The server processes your request (for example, the search page), then sends the data back through the same path—routers → ISP → your computer.

Within seconds, your browser takes that data, organizes it, and shows you the website on your screen.

👉 **In very simple words:**
You ask → ISP forwards → DNS finds the address → Routers deliver → Server replies → Data comes back → You see the website.



Network Architecture:

Network Architecture refers to the overall design and structure of a computer network. It defines how devices (computers, servers, routers, switches, etc.) are arranged, how they communicate, and the rules (protocols) that govern data transmission.

There are two main types:

1. **Peer-to-Peer (P2P):**
    - All devices are equal and share resources directly without a central server.
    - Simple, low-cost, used in small networks like home setups.
2. **Client-Server:**

- One central server provides resources and services to multiple client devices.
- More secure, efficient, and scalable, commonly used in businesses and large organizations.

Key Components of Network Architecture:

- **Topology:** The physical or logical layout (bus, star, ring, mesh, hybrid).
- **Protocols:** Rules for communication (TCP/IP, HTTP, FTP, etc.).
- **Hardware:** Devices like routers, switches, hubs, and cables.
- **Software/Services:** Applications and services that manage data exchange (e.g., DNS, web servers).

Network Topologies:

Network Topologies describe the physical or logical arrangement of devices in a network. Each topology has its own advantages, disadvantages, and use cases. Here are the main types:

1. **Bus Topology:**
   - All devices are connected to a single central cable (the bus).
   - **Pros:** Easy to install, cost-effective for small networks.
   - **Cons:** Cable failure brings down the whole network, slow with heavy traffic.
2. **Star Topology:**
   - All devices connect to a central hub or switch.
   - **Pros:** Easy to manage, failure of one device doesn't affect others.
   - **Cons:** If the central hub fails, the whole network fails.

3. **Ring Topology:**
    - Devices are connected in a circular path, and data travels in one direction.
    - **Pros:** Simple, predictable data flow.
    - **Cons:** If one device or cable breaks, the whole network is affected.
4. **Mesh Topology:**
    - Every device is connected to every other device.
    - **Pros:** Very reliable, provides multiple paths for data.
    - **Cons:** Expensive, complex to set up.
5. **Hybrid Topology:**
    - A combination of two or more topologies (e.g., star + bus).
    - **Pros:** Flexible, can be designed to fit needs.
    - **Cons:** Expensive, harder to manage.

Common Types of Networking Devices and Their Uses:

Network devices work as a mediator between two devices for transmission of data, and thus play a very important role in the functioning of a computer network.

Modems:

Modem is also known as modulator/demodulator is a network device that is used to convert digital signal into analog signal of different frequencies and transmits these signals to a modem at the receiving location. These converted signals can be transmitted over the cable systems, telephone lines, and other communication mediums. A modem is also used to convert an analog signal back into digital signal. Modems are generally used to access the internet by customers of an Internet service provider.

Hub:

A hub is a multiport repeater. A hub connects multiple wires coming from different branches, for example, the connector in star topology which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. In other words, the collision domain of all hosts connected through Hub remains one. Also, they do not have the intelligence to find out the best path for data packets which leads to inefficiencies and wastage.

Switch:

A switch is a multiport bridge with a buffer design that can boost its efficiency(a large number of ports imply less traffic) and performance. A switch is a data link layer device. The switch can perform error checking before forwarding data, which makes it very efficient as it does not forward packets that have errors and forward good packets selectively to the correct port only.

Router:

A router is a device like a switch that routes data packets based on their IP addresses. The router is mainly a Network Layer device. Routers normally connect LANs and WANs and have a dynamically updating routing table based on which they make decisions on routing the data packets. The router divides the broadcast domains of hosts connected through it.

Gateway:

A gateway, as the name suggests, is a passage to connect two networks that may work upon different networking models. They work as messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switches or routers.

NIC:

NIC is a network adapter that is used to connect the computer to the network. It is installed in the computer to establish a LAN. It has a unique ID that is written on the chip, and it has a connector to connect the cable to it. The cable acts as an interface between the computer and the router or modem. NIC is a layer 2 device which means that it works on both the physical and data link layers of the network model.

What is an IP Address?

An **IP (Internet Protocol) address** is a unique numerical label assigned to every device connected to a network. It identifies the device and allows communication over the internet or a local network. Example: **192.168.1.1**

Types of IP Addresses

1. **IPv4 (Internet Protocol version 4):**
    - Uses **32 bits** (written as 4 numbers separated by dots, e.g., 192.168.0.1).
    - Provides about **4.3 billion unique addresses**.
    - Most commonly used.
2. **IPv6 (Internet Protocol version 6):**
    - Uses **128 bits** (written in hexadecimal, e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
    - Provides an almost unlimited number of addresses.
    - Designed to replace IPv4 due to shortage of addresses.

IP Address Classes (IPv4):

IPv4 addresses are divided into **five classes (A–E)** based on range:

1. **Class A:**
    - Range: **1.0.0.0 – 126.255.255.255**
    - Used for very large networks.

    - Default subnet mask: **255.0.0.0**

2. **Class B:**
    - Range: **128.0.0.0 – 191.255.255.255**
    - Used for medium-sized networks.
    - Default subnet mask: **255.255.0.0**
3. **Class C:**
    - Range: **192.0.0.0 – 223.255.255.255**
    - Used for small networks.
    - Default subnet mask: **255.255.255.0**
4. **Class D:**
    - Range: **224.0.0.0 – 239.255.255.255**
    - Used for **multicasting** (sending data to multiple devices).
5. **Class E:**
    - Range: **240.0.0.0 – 255.255.255.255**
    - Reserved for **experimental purposes**.

Subnetting:

Subnetting is the process of dividing a large network (IP address block) into **smaller, manageable subnetworks (subnets)**.

- It improves **network performance** and **security**.

- Each subnet has its own **network address** and **host addresses**.
- Example: The network **192.168.1.0/24** can be split into smaller subnets like **192.168.1.0/26**, **192.168.1.64/26**, etc.

👉 In short: Subnetting = breaking one big network into smaller networks.

CIDR (Classless Inter-Domain Routing):

CIDR is a method of representing IP addresses and their subnet masks.

- Written as: **IP Address / Prefix length**.
- Example: **192.168.1.0/24** → `/24` means **255.255.255.0** (first 24 bits are the network part).

CIDR Ranges (IPv4 Common Blocks)

CIDR Notation	Subnet Mask	Hosts per Subnet
/8	255.0.0.0	16,777,214
/16	255.255.0.0	65,534
/24	255.255.255.0	254

/25	255.255.255.128	126
/26	255.255.255.192	62
/27	255.255.255.224	30
/28	255.255.255.240	14
/29	255.255.255.248	6
/30	255.255.255.252	2

OSI Model (Open Systems Interconnection):

The OSI model is a **7-layer conceptual framework** developed by ISO to standardize how different computer systems communicate over a network. It breaks down communication into layers, where each layer has a specific function and interacts only with the layers directly above and below it. This model helps in designing, troubleshooting, and understanding networks.

---

1. Physical Layer:

- The lowest layer of the OSI model.

- Responsible for the **physical transmission of raw data (bits: 0s & 1s)** over a medium.
- Deals with hardware components like **cables, switches, network cards, hubs, and Wi-Fi signals**.
- Defines standards for **voltages, frequencies, data rates, connectors, and physical topology**.
- **Example:** Ethernet cables, fiber optics, radio signals in Wi-Fi.

---

2. Data Link Layer:

- Ensures **error-free node-to-node delivery** of data across the physical medium.
- Converts raw bits from the Physical Layer into **frames** (structured packets).
- Uses **MAC (Media Access Control) addresses** to identify devices on the same network.
- Has two sub-layers:
    - **MAC (Media Access Control):** Controls how devices use the medium (Ethernet, Wi-Fi).
    - **LLC (Logical Link Control):** Manages error detection and flow control.
- **Example:** Ethernet protocol, Switch operation, Wi-Fi access.

---

3. Network Layer:

- Handles **logical addressing and routing** of data between different networks.
- Data at this layer is in the form of **packets**.

- Uses **IP addresses** to identify devices globally.
- Responsible for choosing the best path for data to travel from source to destination (routing).
- Devices: **Routers, Layer 3 switches**.
- **Protocols:** IPv4, IPv6, ICMP.
- **Example:** Sending an email from your laptop in Pakistan to a server in the USA.

---

4. Transport Layer:

- Ensures **end-to-end communication** between devices.
- Breaks data into **segments** and ensures reliable delivery.
- Provides:
    - **Error detection & correction**
    - **Flow control** (prevents data overload)
    - **Retransmission** of lost data
- Two main protocols:
    - **TCP (Transmission Control Protocol):** Reliable, connection-oriented (used in web browsing, emails).
    - **UDP (User Datagram Protocol):** Fast but connectionless (used in streaming, gaming, video calls).
- **Example:** Watching a YouTube video (uses UDP for speed), downloading a file (uses TCP for reliability).

---

5. Session Layer:

- Manages and controls the **dialogue (sessions)** between computers.
- Establishes, maintains, and terminates connections.
- Keep track of ongoing sessions so communication is not mixed up.
- Supports features like **synchronization and checkpoints** (important in file transfers).
- **Example:** Logging into a website, video conferencing sessions.

---

6. Presentation Layer:

- Acts as a **translator** between the application and the network.
- Ensures that data sent from the application layer of one system can be read by the application layer of another.
- Handles:
    - **Data formatting** (JPEG, GIF, MP3, MP4).
    - **Compression** (reduces data size).
    - **Encryption & Decryption** (security).
- **Example:** SSL/TLS encryption in HTTPS, converting a text file into readable format, video/audio codecs.

---

7. Application Layer:

- The **topmost layer** closest to the end user.

- Provides network **services directly to applications** like browsers, emails, and messengers.

- Interfaces directly with software applications to implement communication functions.

- **Protocols:** HTTP (web browsing), FTP (file transfer), SMTP (email), DNS (domain name system).

- **Example:** When you open Google in Chrome, the Application Layer uses HTTP to request the webpage.

TCP/IP Model (Transmission Control Protocol / Internet Protocol):

The **TCP/IP model** is a simplified 4-layer model developed by the U.S. Department of Defense (DoD) in the 1970s. It explains how data is transmitted across the internet and is the foundation of modern networking. Unlike OSI's 7 layers, TCP/IP has **4 layers** that map to OSI layers.

---

1. Network Access Layer (Link Layer):

- Responsible for the **physical transmission of data** between devices.

- Combines OSI's **Physical + Data Link layers**.

- Defines how devices access the medium and send frames.

- Uses **MAC addresses, Ethernet, Wi-Fi, ARP**.

- **Example:** A switch delivering frames within a LAN.

---

2. Internet Layer:

- Provides **logical addressing & routing** of data across multiple networks.

- Uses **IP addresses** to deliver data packets from source to destination.
- Protocols: **IPv4, IPv6, ICMP, ARP**.
- Device: **Routers**.
- **Example:** When you open a website, IP ensures your request reaches the correct server.

---

3. Transport Layer:

- Ensures **end-to-end communication** between applications.
- Breaks data into segments, ensures reliable delivery, error checking, and flow control.
- Protocols:
    - **TCP:** Reliable, connection-oriented (web browsing, email, file transfer).
    - **UDP:** Fast, connectionless (online games, video streaming).
- **Example:** WhatsApp call uses UDP; downloading a file uses TCP.

---

4. Application Layer:

- Closest to the **user**, provides network services directly to applications.
- Covers OSI's **Session, Presentation, and Application layers**.
- Protocols: **HTTP, HTTPS, FTP, SMTP, POP3, DNS, DHCP**.
- **Example:** When you search Google, your browser uses HTTP/HTTPS at this layer.

OSI vs TCP/IP (Comparison Table):

OSI Model (7 Layers)	TCP/IP Model (4 Layers)
Application	Application
Presentation	Application
Session	Application
Transport	Transport
Network	Internet
Data Link	Network Access
Physical	Network Access

✅ **Key Difference:**

- **OSI** is a **theoretical model** (good for learning concepts).
- **TCP/IP** is a **practical model** (used in real networks today).

Networking Ports & Protocols (DevOps Notes):

Protocol	Stands For	Port Number	Function / Use Case
HTTP	HyperText Transfer Protocol	80	Standard web browsing, APIs (not secure).
HTTPS	HyperText Transfer Protocol Secure	443	Secure web communication (encrypted websites, APIs).
SSH	Secure Shell	22	Secure remote login, server management, Git over SSH, automation.
FTP	File Transfer Protocol	21	Upload/download files (not secure, plain text).
SFTP	Secure File Transfer Protocol	22	Encrypted file transfers (built on SSH).
Telnet	Telecommunications Network	23	Remote login (insecure, replaced by SSH).
DNS	Domain Name System	53 (UDP/TCP)	Converts domain names into IP addresses, service discovery in cloud/K8s.

SMTP	Simple Mail Transfer Protocol	25 (or 587/465 secure)	Sending outgoing emails (alerts, notifications).
IMAP	Internet Message Access Protocol	143 (or 993 secure)	Retrieves emails while keeping them on the server.
POP3	Post Office Protocol v3	110 (or 995 secure)	Downloads emails from the server (removes from server).
DHCP	Dynamic Host Configuration Protocol	67 (server), 68 (client)	Automatically assigns IP addresses in networks/clouds.
UDP	User Datagram Protocol	*No fixed port*	Fast, connectionless transport protocol (used by DNS, streaming, VoIP, Kubernetes

DNS (Domain Name System):

- **Port:** 53 (TCP/UDP)
- **Stands for:** Domain Name System
- **Function:** DNS translates human-readable domain names (like `www.google.com`) into IP addresses (like `142.250.190.14`) so that computers can locate and communicate with each other.

- **How it works:**

    1. You type a URL in your browser.

    2. The DNS resolver asks the nearest DNS server for the IP.

    3. If the server doesn't know, it queries root servers, TLD servers (.com, .org, etc.), and authoritative name servers until the IP is found.

    4. The result is cached for future requests.

DHCP (Dynamic Host Configuration Protocol)

- **Port:** 67 (UDP for server), 68 (UDP for client)

- **Stands for:** Dynamic Host Configuration Protocol

- **Function:** DHCP automatically assigns IP addresses and other network settings (like subnet mask, default gateway, and DNS server) to devices on a network. This saves time and avoids conflicts compared to assigning IPs manually.

- **How it works (steps):**

    1. **Discover:** The client broadcasts a request asking for an IP address.

    2. **Offer:** The DHCP server responds with an available IP and configuration.

    3. **Request:** The client requests to use that IP.

    4. **Acknowledge:** The DHCP server confirms and leases the IP for a specific time.

NAT (Network Address Translation)

- **Stands for:** Network Address Translation

- **Port:** NAT itself doesn't use a fixed port (it works at the network layer), but it translates ports dynamically when needed.

- **Function:** NAT allows multiple devices in a private network (like your home Wi-Fi) to share a single public IP address when accessing the internet. It modifies the IP addresses in packet headers as they pass through a router or firewall.

- **Types of NAT:**

    1. **Static NAT:** Maps one private IP to one public IP.

    2. **Dynamic NAT:** Maps private IPs to any available public IPs from a pool.

    3. **PAT (Port Address Translation) / Overloading:** Maps many private IPs to a single public IP by using different port numbers (most common).

Firewall

A **firewall** is like a security guard for your network. It sits between your internal network (trusted) and the external world (untrusted, like the internet) and decides which traffic is allowed or blocked. Firewalls can be hardware devices (routers, appliances) or software-based (installed on servers).

They work by applying **rules** that filter traffic based on IP addresses, ports, and protocols. Modern firewalls (Next-Gen Firewalls) go beyond this by inspecting the actual data, detecting intrusions, and preventing cyberattacks.

For DevOps and cloud environments, firewalls are essential to secure applications, control access to servers, and protect against unauthorized users. Without a firewall, your systems are exposed directly to threats on the internet.

Load Balancer

A **Load Balancer** is a networking device or software that distributes incoming traffic across multiple servers to ensure no single server is overloaded. It improves performance, reliability, and availability of applications by balancing the workload.

- **How it works:** When users send requests (like opening a website), the load balancer receives them and forwards each request to the least busy or healthiest server. If one server goes down, the load balancer automatically redirects traffic to the remaining servers.

- **Types of Load Balancing:**

    1. **Hardware Load Balancer** – Physical device used in large-scale data centers.

    2. **Software Load Balancer** – Runs on servers or cloud (e.g., Nginx, HAProxy).

    3. **Application Load Balancer (Layer 7)** – Distributes traffic based on content (e.g., URL, cookies).

    4. **Network Load Balancer (Layer 4)** – Distributes traffic based on IP and port.