COMPUTER NETWORKING

What is a Network? A network is a collection of interconnected devices that can communicate with each other. These devices can be computers, servers, printers, or any other electronic device capable of sending and receiving data. **Types of Networks** Networks are classified based on their geographical extent and the number of devices connected. Here are some common types:

Based on Geographical Extent

- Personal Area Network (PAN):
  - o Covers a small area, typically within a person's workspace.
  - o Examples: Bluetooth connection between a smartphone and headphones, USB connection between a computer and a mouse.
- Local Area Network (LAN):
  - o Connects devices within a limited geographical area, such as a home, office, or school building.
  - o Typically uses Ethernet cables or Wi-Fi.
  - o Example: A network connecting computers in a school computer lab.
- Metropolitan Area Network (MAN):
  - o Spans a larger geographical area than a LAN, covering an entire city or metropolitan region.
  - o Often used to connect multiple LANs together.
  - o Example: A university campus network.
- Wide Area Network (WAN):
  - o Covers a large geographical area, such as a country or even the entire world.

- The Internet is the largest WAN.
- Uses various technologies like fiber optic cables, satellite links, and DSL.

Other Network Types

- Wireless Local Area Network (WLAN):
  - A type of LAN that uses wireless technology (Wi-Fi) to connect devices.
- Storage Area Network (SAN):
  - A dedicated network for storing and accessing data.
  - Primarily used in enterprise environments for high-performance data storage.
- Virtual Private Network (VPN):
  - Creates a secure connection over a public network, allowing remote access to private networks.

Network Protocols

Network protocols are sets of rules that govern communication between devices. Some common protocols include:

- **TCP/IP:** The foundation of the internet.
- **HTTP:** Used for transferring data on the web.
- **FTP:** Used for transferring files between computers.
- **SMTP:** Used for sending emails.

What is a Server?

A server is a computer program or device that provides a service to other computers, known as clients.

Two Main Concepts of a Server:

1. **Hardware Server:** This is the physical computer itself. It's typically equipped with powerful processors, ample memory, and large storage capacity to handle the demands of running server software.

2. **Software Server:** This is the program that runs on the hardware server and provides a specific service. Examples include:

    - **Web server:** Delivers web pages to users.
    - **Mail server:** Handles email sending and receiving.
    - **Database server:** Stores and manages data.
    - **File server:** Stores and shares files.
    - **Game server:** Hosts online games.

How Does it Work?

The **client-server model** is the foundation of how servers operate. Clients request services from servers, and servers fulfill those requests. For example, when you visit a website, your computer (client) sends a request to the website's server, which then sends the webpage back to your computer.

How Data is Transferred

Data is transferred through networks using a process called **packet switching**.

Packet Switching

1. **Data Breakdown:** Data is divided into smaller pieces called packets. Each packet contains information about the sender, receiver, and the sequence number of the packet.
2. **Routing:** Packets are sent through various routers on the internet. Routers determine the best path for each packet to reach its destination.

3. **Transmission:** Packets travel through networks, using different communication channels like fiber optic cables, copper wires, or wireless signals.
4. **Reassembly:** At the destination, the packets are reassembled in the correct order to form the original data.

TCP/IP: The Backbone of the Internet

TCP/IP is the fundamental protocol suite for internet communication. It's a set of rules that govern how data is transmitted across networks. While often referred to as a single protocol, it's actually a collection of protocols working together.

The Two Main Protocols

- **TCP (Transmission Control Protocol):**
  - o Ensures reliable data delivery.
  - o Breaks data into packets and reassembles them at the destination.
  - o Acknowledges the receipt of data and retransmits lost packets.
  - o Used for applications that require error-free data transfer, such as email and file transfer.
- **IP (Internet Protocol):**
  - o Handles the addressing and routing of packets.
  - o Assigns unique IP addresses to devices on the network.
  - o Determines the best path for packets to reach their destination.
  - o Less concerned with reliability than TCP.

The TCP/IP Model

To better understand how TCP/IP works, it's helpful to visualize it as a four-layer model:

1. **Application Layer:** This layer is closest to the user and includes protocols like HTTP, FTP, and SMTP.

2. **Transport Layer:** Handles end-to-end communication, using TCP or UDP (User Datagram Protocol).
3. **Internet Layer:** Responsible for packet routing and addressing, using IP.
4. **Network Access Layer:** Deals with physical network connections.

How it Works

When you send an email, for example:

1. **Application Layer:** Your email client uses SMTP to format the email as a series of packets.
2. **Transport Layer:** TCP takes these packets and adds information like sequence numbers and checksums for error detection.
3. **Internet Layer:** IP assigns IP addresses to your computer and the recipient's server, and determines the route the packets will take.
4. **Network Access Layer:** The packets are converted into electrical signals and sent over the network.

At the receiving end, the process is reversed, with TCP reassembling the packets and the email client displaying the message.

TCP/IP is the foundation of the internet, enabling seamless communication between billions of devices worldwide.

IPv4 and IPv6: A Comparison

IPv4 and **IPv6** are the two primary versions of the Internet Protocol (IP) used to identify devices on a network.

IPv4

- **32-bit address:** Limited address space, leading to the current shortage.
- **Dotted decimal notation:** Example: 192.168.1.100

- **Classful addressing:** Addresses were divided into classes (A, B, C, D, E) based on the first octet.
- **Older protocol:** Widely deployed but facing address exhaustion.

IPv6

- **128-bit address:** Vastly larger address space, capable of addressing billions of devices per square meter.
- **Hexadecimal notation:** Example: 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- **Stateless addressing:** No predefined address classes.
- **Newer protocol:** Being gradually adopted to address IPv4 limitations.

Key Differences

Feature	IPv4	IPv6
Address space	Limited	Vast
Notation	Dotted decimal	Hexadecimal
Addressing scheme	Classful	Stateless
Header size	Larger	Smaller
Security	Less secure	More secure
Autoconfiguration	Limited	Extensive

Static vs. Dynamic IP Addresses

- **Static IP Address:**
  - o A fixed, permanent IP address assigned to a device.
  - o Doesn't change over time.
  - o Commonly used for servers, network devices, and applications requiring consistent accessibility.
  - o Usually requires manual configuration.
- **Dynamic IP Address:**
  - o An IP address assigned temporarily to a device when it connects to a network.

- Can change each time a device connects or reconnects.
- Commonly used for home and small office networks to conserve IP addresses.
- Assigned automatically by a DHCP (Dynamic Host Configuration Protocol) server.

Public vs. Private IP Addresses

- **Public IP Address:**
  - A unique address assigned to a device connected to the internet.
  - Used to identify devices on the global network.
  - Can be either static or dynamic.
  - Assigned by an Internet Service Provider (ISP).
- **Private IP Address:**
  - An IP address used within a private network (LAN).
  - Not routable on the internet.
  - Conserves public IP addresses.
  - Ranges defined by RFC 1918:
    - 10.0.0.0 - 10.255.255.255
    - 172.16.0.0 - 172.31.255.255
    - 192.168.0.0 - 192.168.255.255

Summary Table

Type	Description	Public/Private	Static/Dynamic
Public Static	Fixed IP address accessible from the internet	Public	Static
Public Dynamic	IP address changes over time, accessible from the internet	Public	Dynamic
Private Static	Fixed IP address within a private network	Private	Static
Private Dynamic	IP address changes within a private network	Private	Dynamic

In essence

- **Public IP addresses** are used for internet communication.
- **Private IP addresses** are used for internal network communication.
- **Static IP addresses** remain constant.
- **Dynamic IP addresses** can change.

Modem

A **modem** (modulator-demodulator) is a device that converts digital data into a format suitable for transmission over a physical medium such as a telephone line, cable, or fiber optic cable, and vice versa. Essentially, it's the bridge between your digital devices and the analog world of your internet connection.

Router

A **router** is a networking device that forwards data packets between computer networks. It determines the best path for data to travel and directs it accordingly. In a home network, it typically connects your modem to your devices (computers, smartphones, etc.) and assigns IP addresses to each device.

Gateway

A **gateway** is a network point that acts as an entrance to another network. It translates information between incompatible systems. In a home network, your router often acts as a gateway between your local network and the internet.

Mesh Network

A **mesh network** is a network topology where nodes (devices) communicate with each other directly, without the need for a central access point. It creates a self-healing network where if one node fails, the network can still function.

Mesh networks are often used in wireless environments and are becoming increasingly popular for home Wi-Fi systems.

To summarize:

Modem: Connects your home to the internet.
Router: Connects your devices to your home network and the internet.
Gateway: The point where your home network connects to the external network
Mesh network: A type of network where devices communicate directly with each other.

DHCP: Dynamic Host Configuration Protocol

DHCP is a network protocol that automatically assigns IP addresses and other network configuration parameters to devices on a network. Think of it as the automatic address assigner for your network.

How does it work?

1. **DHCP Server:** This is a designated device on the network that manages a pool of IP addresses.
2. **DHCP Client:** Any device that needs an IP address to connect to the network.
3. **Request:** When a device (client) connects to the network, it sends a DHCP . request to the server
4. **Assignment:** The DHCP server assigns an available IP address from its pool to the client, along with other network information like subnet mask, default . gateway, and DNS server addresses
5. **Lease:** The IP address is assigned for a specific period (lease time). After the lease expires, the client can renew the lease or the IP address is returned to . the pool

Why is DHCP important?

- **Efficiency:** It simplifies network administration by automating IP address assignment.

- **Flexibility:** IP addresses can be reused when devices are not connected to the network.
- **Scalability:** It can handle a large number of devices on a network.

In essence, DHCP makes it easier to manage networks by eliminating the need for manual IP address configuration for every device.

Network Address Translation (NAT)

NAT is a method of mapping one IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device. Essentially, it allows multiple devices on a private network to share a single public IP address.

How it works:

- A device on a private network sends a packet with its private IP address to a NAT router.
- The router replaces the private IP address with its public IP address and modifies other header information.
- The packet is sent to the internet.
- When a response packet arrives, the router uses the information in the packet header to determine the original private IP address and forwards the packet to the correct device.

Types of NAT:

- **Static NAT:** Maps a single private IP address to a single public IP address.
- **Dynamic NAT:** Maps multiple private IP addresses to a single public IP address, using a pool of public IP addresses.
- **Port Address Translation (PAT):** Maps multiple private IP addresses and ports to a single public IP address and port.

IP Mapping

IP mapping refers to the process of associating a private IP address with a public IP address. This is typically done through NAT, but it can also be used in other contexts, such as VPNs or load balancing.

Types of IP Mapping:

- **One-to-one mapping:** A single private IP address is mapped to a single public IP address (static NAT).
- **Many-to-one mapping:** Multiple private IP addresses are mapped to a single public IP address (dynamic NAT or PAT).

In summary, NAT is a crucial technology that allows multiple devices to share a single public IP address, while IP mapping is the process of associating private and public IP addresses.

Ports and Sockets: The Digital Doorways

Ports

Think of a port as a logical doorway on a computer. It's a number assigned to a specific application or service running on a device. This number helps differentiate between different services running on the same computer.

- **Range:** Ports are numbered from 0 to 65535.
- **Types:**
    - o **Well-known ports:** Used for standard services (e.g., HTTP on port 80, FTP on port 21).
    - o **Registered ports:** Used for specific services or applications.
    - o **Dynamic ports:** Used for temporary connections.

Sockets

A socket is a combination of an IP address and a port number. It uniquely identifies an application process on a network. Imagine a socket as a specific door within a building (the IP address) that leads to a particular room (the port).

- **Purpose:** Sockets enable communication between different devices on a . network
- **Components:** An IP address specifies the computer, and the port number specifies the application.

To summarize:

- **Port:** A number that identifies a service on a computer.
- **Socket:** A combination of IP address and port number that identifies a . specific application process on a network

Example: If you want to access a web page (HTTP), your computer will create a socket using your IP address and port 80 (the standard HTTP port) to connect to the web server.

Subnet Mask: Dividing Your Network

A subnet mask is a 32-bit number that divides an IP address into two parts: the network address and the host address.

How it works:

- **Network address:** Identifies the network to which a device belongs.
- **Host address:** Identifies the specific device within that network.

To determine the network and host parts of an IP address, you apply a bitwise AND operation between the IP address and the subnet mask.

Example:

- **IP address:** 192.168.1.100
- **Subnet mask:** 255.255.255.0

In binary, this would be:

- IP address: 11000000.10101000.00000001.01100100
- Subnet mask: 11111111.11111111.11111111.00000000

Applying the bitwise AND operation:

- Network address: 11000000.10101000.00000001.00000000 (192.168.1.0)

- Host address: 00000000.00000000.00000000.01100100 (0.0.0.100)

So, in this example, 192.168.1.0 is the network address, and 0.0.0.100 is the host address.

Why is it important?

- **Efficient IP address usage:** By dividing a network into smaller subnets, you can maximize the number of devices that can be connected.
- **Network segmentation:** Subnets allow you to isolate different parts of your network for security or performance reasons.
- **Routing:** Subnet masks help routers determine the best path for data packets.

In essence, a subnet mask is a crucial tool for network administrators to organize and manage IP addresses effectively.

MAC Address: Your Device's Unique Identifier

MAC address stands for **Media Access Control address**. It's a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment.

Key points about MAC addresses:

- **Unique:** Each device has a unique MAC address.
- **Physical:** It's a physical address embedded in the network interface card). NIC(
- **Format:** Usually expressed as a 12-character hexadecimal number separated by hyphens (e.g., 00-15-8D-00-01-13).
- **Purpose:** Used for communication within a local network (LAN).

How does it work?

When a device wants to send data to another device on the same network, it uses the recipient's MAC address to determine the destination. The data is

encapsulated in a frame with the sender's and receiver's MAC addresses, allowing devices on the network to communicate directly.

Note: While MAC addresses are used within a local network, IP addresses are used for communication across different networks.

Internet vs. WWW:

Internet and **WWW** are often used interchangeably, but they are not the same thing.

Internet

- **Infrastructure:** The internet is the global network of interconnected computers.
- **Hardware-based:** It's the physical infrastructure that allows devices to . communicate
- **Services:** It supports various services like email, file transfer, and the World Wide Web .

WWW (World Wide Web)

- **Service:** The WWW is a service built on top of the internet.
- **Software-based:** It uses protocols like HTTP to access and display . information
- **Content:** It's a collection of interconnected web pages accessible . through the internet

To summarize:

- The internet is the highway, while the WWW is a car driving on that highway.
- You need the internet to access the WWW, but you can use the internet for other purposes as well.

HTTP vs HTTPS: A Security Comparison

HTTP (Hypertext Transfer Protocol) is the foundation for data communication on the World Wide Web. It's how web browsers and servers communicate with each other.

HTTPS (Hypertext Transfer Protocol Secure) is a more secure version of HTTP. It uses SSL/TLS encryption to protect data transmitted between a web server and a user's browser.

Key Differences:

- **Security:** HTTPS is the secure version of HTTP. It encrypts data, making it much harder for hackers to intercept and read information.
- **Encryption:** HTTPS uses SSL/TLS certificates to establish a secure connection .
- **Trust:** Websites using HTTPS are generally considered more trustworthy by users and search engines .
- **URL:** Websites using HTTP have "http://" in their address, while HTTPS websites have "https://".

Why HTTPS is Important:

- **Data Protection:** Protects sensitive information like passwords, credit card numbers, and personal data.
- **Trust Building:** Users are more likely to trust websites that use HTTPS.
- **Search Engine Ranking:** Google and other search engines prioritize HTTPS websites in search results .

In summary, while HTTP is the basic protocol for web communication, HTTPS is the standard for secure data transfer. It's essential for protecting user information and building trust .

Domain Name: Your Website's Address

A domain name is essentially the human-readable address of a website. It's the part you type into a web browser to access a specific website, like `google.com or amazon.com.`

How it works:

- **Unique identifier:** Each domain name is unique on the internet.
- **IP address connection:** Behind the scenes, a domain name is linked to a numerical IP address, which is the actual address computers use to . communicate
- **DNS:** The Domain Name System (DNS) is responsible for translating . domain names into IP addresses

Domain Structure:

A domain name is typically divided into parts:

- **Top-Level Domain (TLD):** The part after the last dot (e.g., .com, .net, .org).
- , **Second-Level Domain (SLD):** The part before the TLD (e.g., google). amazon
- ., **Subdomain:** Optional part before the main domain (e.g).mail.google.com

Example:

- **google.com:**
    - o Top-Level Domain: .com
    - o Second-Level Domain: google

In summary, a domain name is a user-friendly way to access websites on the .internet. It simplifies the process of finding and accessing online resources

DNS: The Internet's Phonebook

DNS stands for **Domain Name System**. It's essentially the internet's phonebook, translating human-readable domain names (like google.com) into machine-readable IP addresses (like 192.168.1.1).

How does it work?

1. **User enters a domain name:** When you type a website address into your browser, it sends a query to a DNS server.
2. **DNS lookup:** The DNS server checks its records to find the
. corresponding IP address for that domain name
3. **IP address returned:** The DNS server returns the IP address to your browser.
4. **Connection established:** Your browser uses the IP address to connect to the website's server.

Why is DNS important?

- **User-friendly:** Allows people to use easy-to-remember domain names instead of complex IP addresses.
- **Efficiency:** Improves internet performance by reducing the need to
.manually configure IP addresses
- **Scalability:** Handles the increasing number of websites and internet users.

In essence, DNS is a critical component of the internet, making it possible for us to easily navigate the web.

URL: Your Web Address

URL stands for **Uniform Resource Locator**. It's essentially the address of a specific resource on the internet. Think of it as the street address for a website or a file online.

Structure of a URL

A typical URL consists of several parts:

- **Protocol:** Specifies the method used to access the resource (e.g., http, https, ftp).
- **Domain name:** The address of the website (e.g., google.com).
- **Path:** Indicates the location of the resource within the website (e.g., /search).
- **Parameters:** Optional information added to the URL). e.g., ?q=search+term(
- **Fragment:** Specifies a particular part of a web page (e.g., #section1).

Example:

- **[invalid URL removed]**
  - o https://: Protocol (Hypertext Transfer Protocol Secure)
  - o `www.example.com`: Domain name
  - o /products/shoes: Path to the product category
  - o ?color=red: Parameter specifying the color
  - o #sale: Fragment indicating a specific section on the page

In essence, a URL provides the necessary information for a web browser to locate and retrieve a specific resource from the internet.

FTP: File Transfer Protocol

FTP stands for **File Transfer Protocol**. It's a standard network protocol used to transfer computer files from one server to a client on a computer network. Essentially, it's a way to move files from one place to another over the internet.

How it works:

- **Client-server model:** FTP operates on a client-server architecture. A user (client) connects to an FTP server to transfer files.

- **Data transfer:** FTP uses two separate connections: one for control commands and another for data transfer. This allows efficient file transfer.
- **Authentication:** Users typically need a username and password to access an FTP server.

Security Concerns:

- **Plaintext transmission:** FTP transmits data in plain text, making it vulnerable to interception.
- **Secure alternatives:** To address security concerns, FTP can be secured using SSL/TLS (FTPS) or replaced by SFTP (SSH File Transfer Protocol).

Common uses of FTP:

- Uploading website files to a web server.
- Transferring large files between computers.
- Backing up data to a remote server.

In summary, FTP is a basic protocol for transferring files, but its security limitations have led to the adoption of more secure alternatives like FTPS and SFTP for sensitive data

API: The Intermediary of Software

API stands for **Application Programming Interface**. It's essentially a set of rules or protocols that allows different software applications to communicate and interact with each other. Think of it as a messenger that carries requests and responses between two systems.

How does it work?

- **Request:** An application sends a request to an API with specific instructions.
- **Processing:** The API processes the request and interacts with the requested system.
- **Response:** The API sends a response back to the original application with the requested data or result.

Example:

Imagine you're using a weather app. When you check the weather, the app doesn't have its own weather station. Instead, it uses an API to fetch weather data from a weather service like AccuWeather or OpenWeatherMap.

Why are APIs important?

- **Efficiency:** APIs streamline data exchange between applications.
- **Innovation:** They enable developers to create new applications and services by leveraging existing platforms.
- **Integration:** APIs facilitate the integration of different systems and data sources.

In essence, APIs are the backbone of modern software development allowing for seamless interaction and data sharing between different applications