Networking

Reading Material

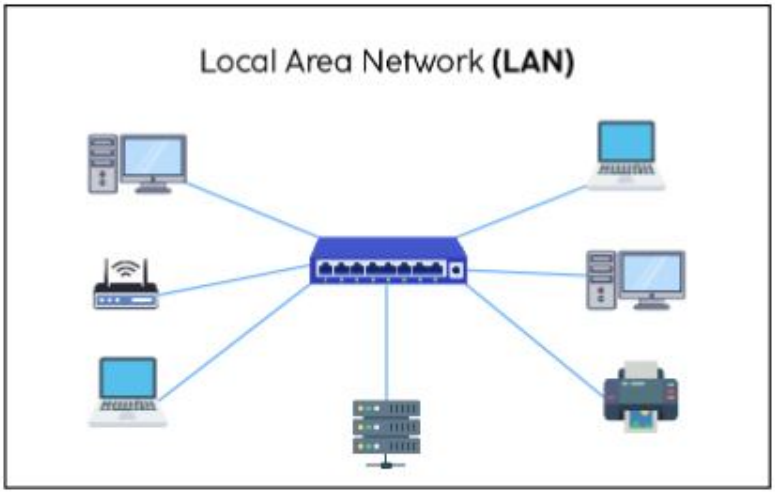
PW SKILLS

Topics

- Overview of computer networks
- IP Addressing
- Security Protocols (SSL/TLS)
- Network Troubleshooting Tools (ping, traceroute, tcpdump)
- Client Server Architecture

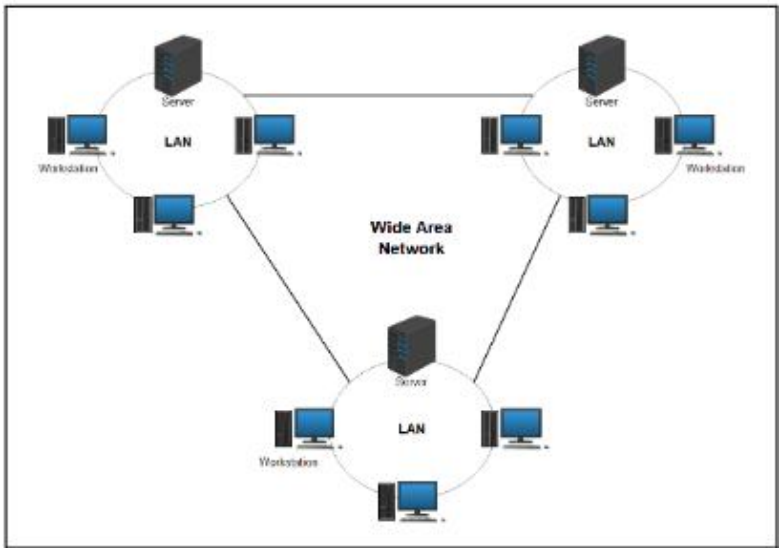
Overview of computer networks

Definition: A computer network is a system that links multiple separate computers together to exchange resources and data. Communication between users is made easier by the integration of computers and other technologies.

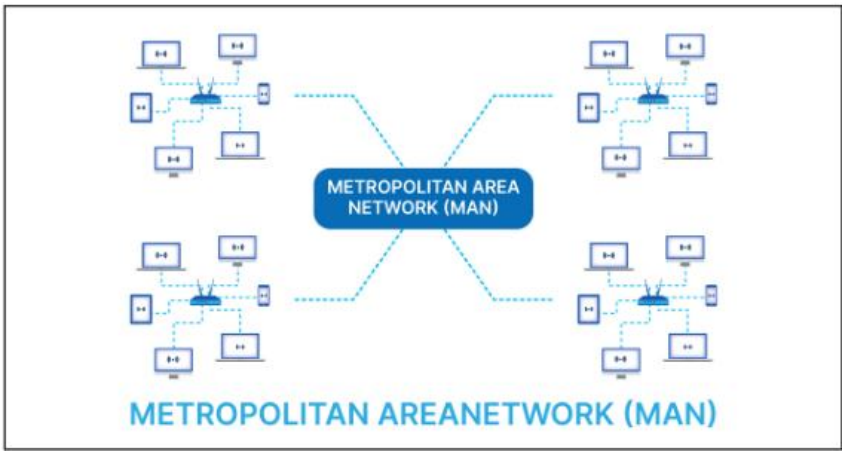


Types of Networks :

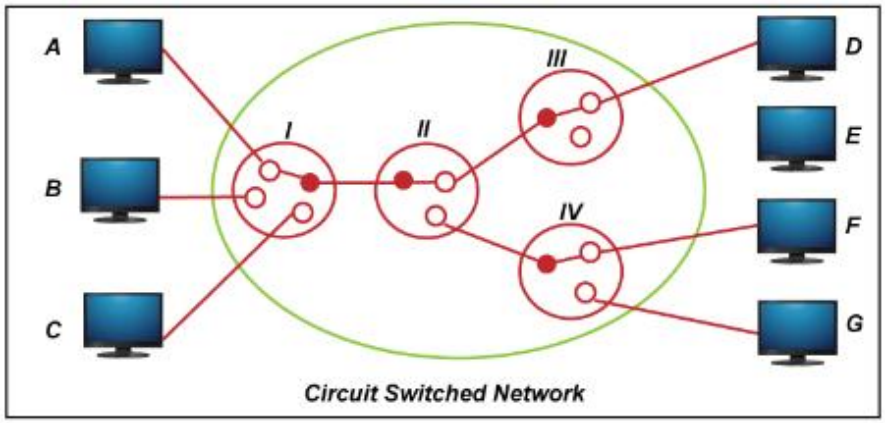
- **LAN (Local Area Network):** Connects devices within a limited area, such as a home, school, or office building.
- **WAN (Wide Area Network):** Spans over a large geographical area, connecting multiple LANs or other networks together.



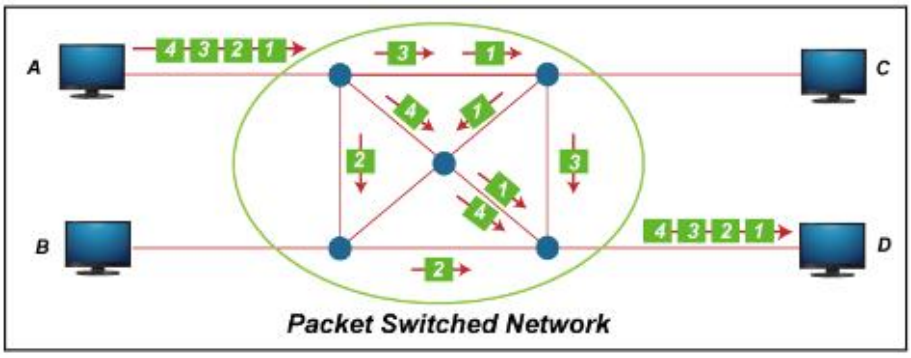
MAN (Metropolitan Area Network): Covers a larger area than a LAN but smaller than a WAN, typically within a city or metropolitan area.



Circuit Switching



Packet Switching



Question Time

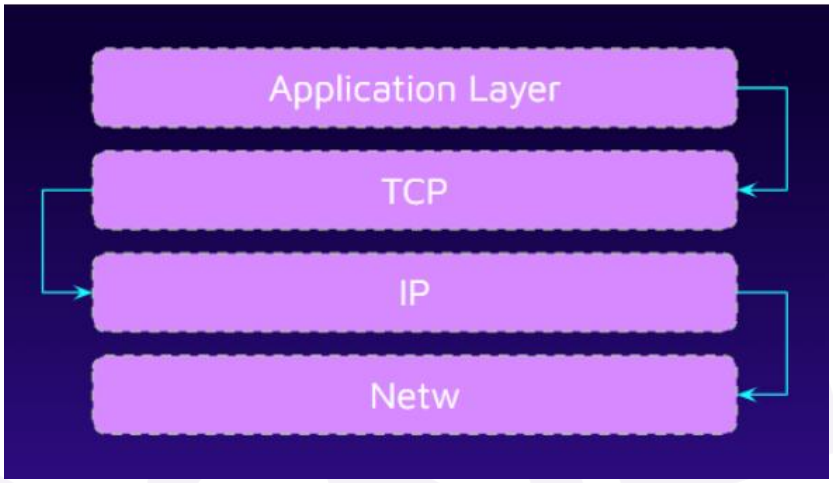
Question: What is the main difference between circuit switching and packet switching?

TCP/IP Stack and Protocol Suite

TCP (Transmission Control Protocol): Ensures reliable, connection-oriented data transmission between devices.
Internet Protocol (IP): Facilitates data transfer between devices using unique IP addresses.

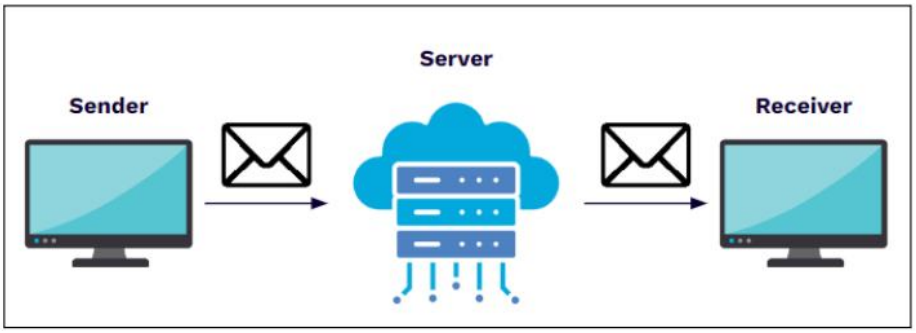
TCP/IP stack

The TCP/IP stack is a framework of protocols used for communication over the internet and similar networks. It consists of four layers, each with specific functions:



Features of TCP

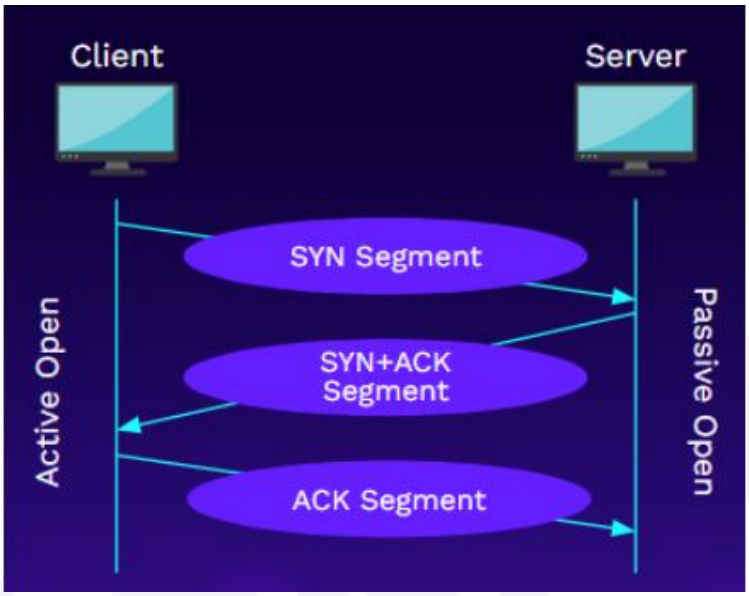
- Connection-Oriented
- Reliable Delivery
- Ordered Data Transfer
- Flow Control
- Congestion Control
- Error Detection and Correction



Working of TCP

Three-Way Handshake: Establishes the connection.

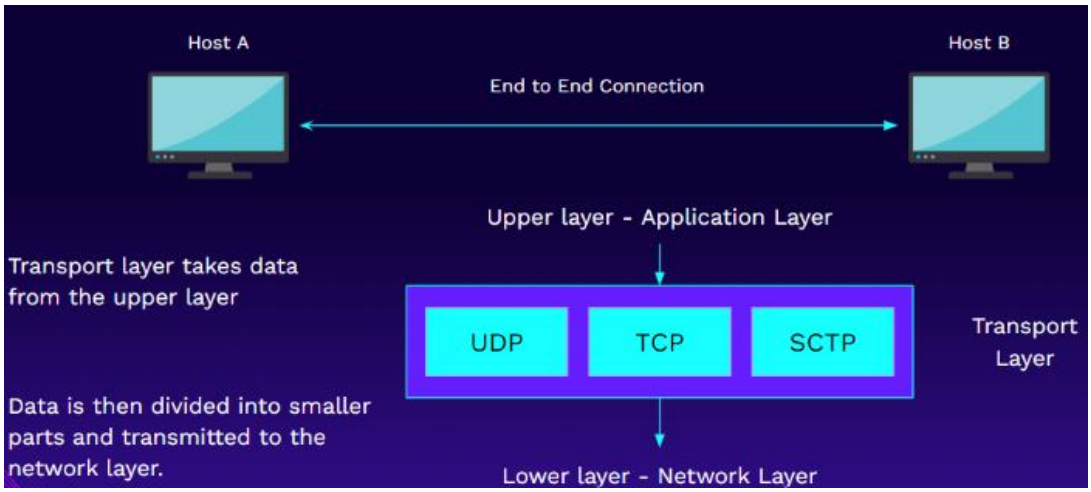
- **Step 1:** The client sends a segment with its sequence number to the server.
- **Step 2:** The server responds with its own sequence number and an acknowledgment number, which is the client's sequence number plus one.
- **Step 3:** The client sends an acknowledgment back to the server, confirming receipt.
- **Result:** The connection between the client and server is established



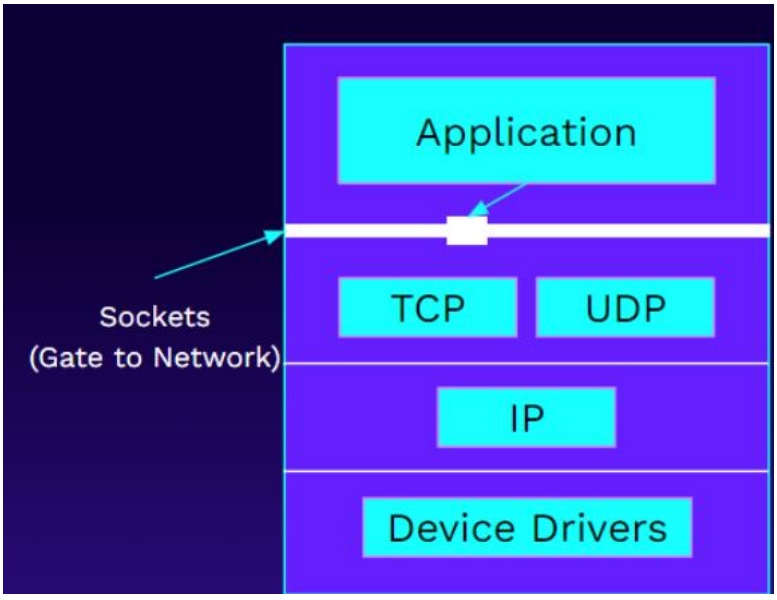
Need of TCP

- Reliable Communication:
- Data Integrity:
- Flow and Congestion Management:
- Error Handling

Purpose of transport layer



TCP/IP Protocol stack



TCP:

- Establish connection
- Maintain connection during the communication
- Release connection
- Reliable (Acknowledge, in order)

UDP:

- No need to setup a link
- Independent packets
- Not reliable (no acknowledgement)

Advantages of the TCP/IP Model

- Promotes interoperability among diverse systems and networks
- Highly scalable for both small and large networks
- Based on open standards ensuring compatibility
- Flexible with various routing protocols and communication methods
- Reliable with error-checking and retransmission features

Disadvantages of the TCP/IP Model

- Complex configuration, especially for large networks
- Security vulnerabilities due to initial design without security
- Inefficient for very small networks due to overhead and complexity
- Limited address space in IPv4, leading to address exhaustion
- Data overhead in TCP reduces efficiency, especially for small packets

OSI Model (Open Systems Interconnection)

The OSI model, developed by ISO, provides a universal framework for creating networking standards and protocols.

7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network

Key Differences

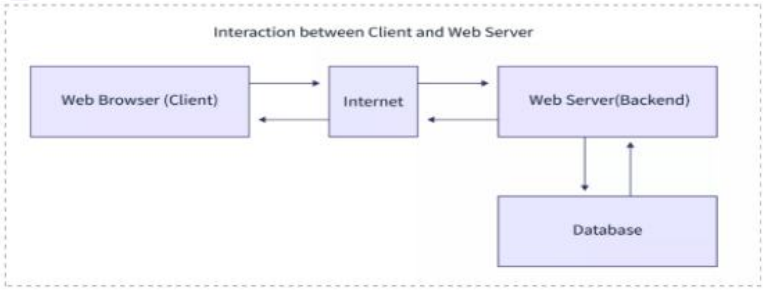
- The TCP/IP model has four layers, while the OSI model has seven layers.
- The TCP/IP model is a practical implementation used in the Internet, whereas the OSI model is more of a theoretical framework.
- The layers in the TCP/IP model are more loosely defined compared to the OSI model, which has strict definitions for each layer's functions and protocols.
- Both models are valuable for understanding networking concepts, and each has its strengths in different contexts.

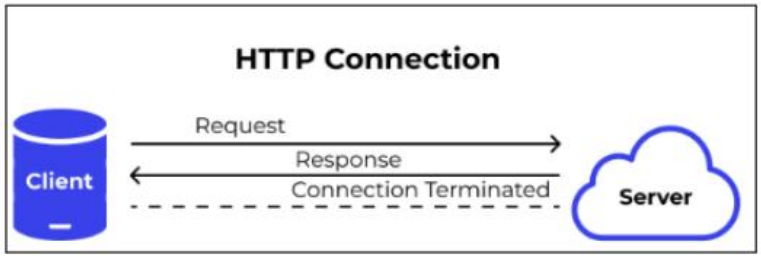
Question Time

Question: What type of communication protocol is TCP?
Question: What is the primary function of the Internet Layer in the TCP/IP model?
Question: What is the function of the Transport Layer in the OSI model?

HTTP/HTTPS Protocols

HTTP (HyperText Transfer Protocol) and HTTPS (HyperText Transfer Protocol Secure) are protocols used for transferring data over the web.





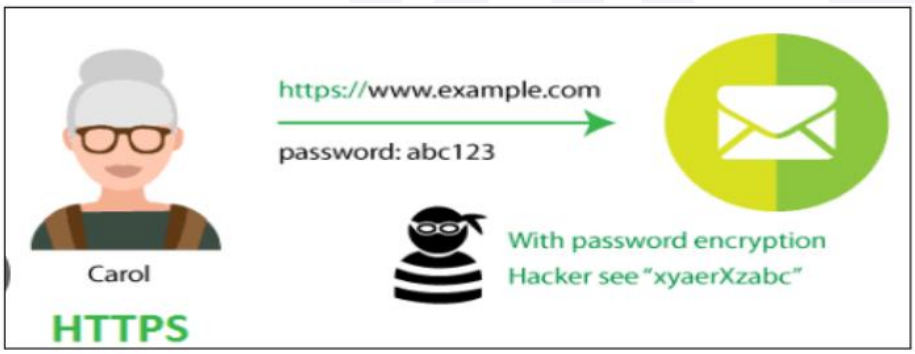
HTTP

- **HTTP (HyperText Transfer Protocol):**
- **Stateless:** Each client request is independent, with no server memory of previous requests.
- **Methods:** Common HTTP methods include GET (retrieve data), POST (submit data), PUT (update data), DELETE (remove data).

HTTPS

HTTPS (HyperText Transfer Protocol Secure) : HTTPS is the secure version of HTTP, encrypting data between client and server using SSL/TLS for privacy and security.

SSL/TLS Encryption: SSL and TLS are cryptographic protocols that provide secure communication over networks, with TLS being the successor to SSL.



Key Differences Between HTTP and HTTPS

Security:

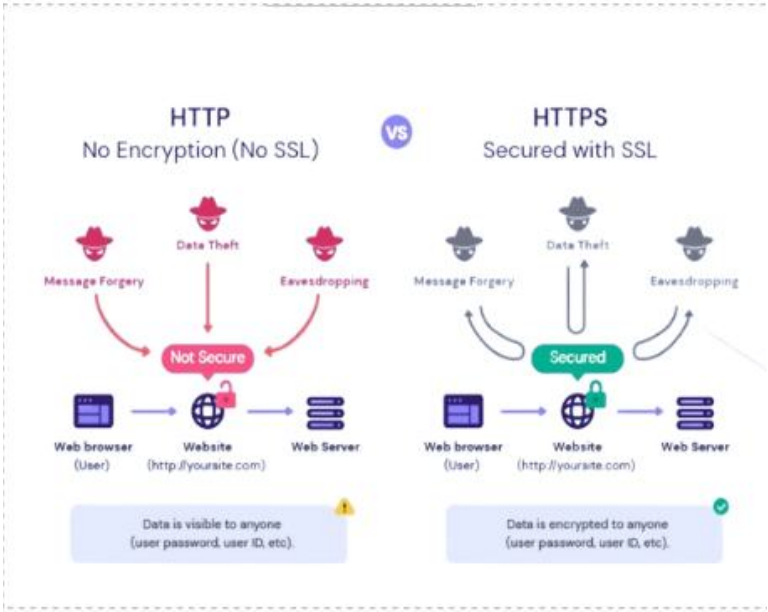
- **HTTP:** Data is transmitted in plain text and can be intercepted.
- **HTTPS:** Data is encrypted, ensuring privacy and security.

URL Prefix:

- **HTTP:** URLs start with http://.
- **HTTPS:** URLs start with https://.

Port Number:

- **HTTP:** Typically uses port 80.
- **HTTPS:** Typically uses port 443.



Importance of HTTPS

- Security: HTTPS protects sensitive information like passwords and credit card numbers.
- Trust: HTTPS instills user trust with a padlock icon in the browser's address bar.
- SEO: HTTPS improves search engine rankings, favoring secure websites.

Example Scenario: Logging into a Website

Using HTTP:

- You go to http://example.com/login.
- Your login credentials (username and password) are sent as plain text.
- Anyone intercepting the communication can read your credentials.

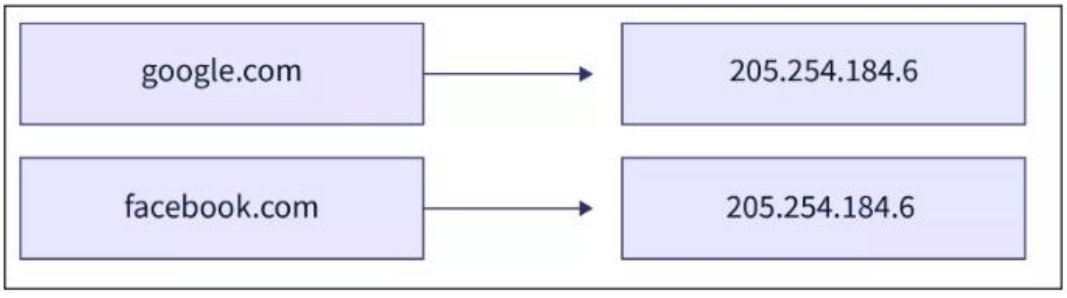
Using HTTPS:

- You go to https://example.com/login.
- Your login credentials are encrypted before being sent.
- Even if someone intercepts the communication, they cannot read your credentials because they are encrypted.

Working with HTTP Protocol.

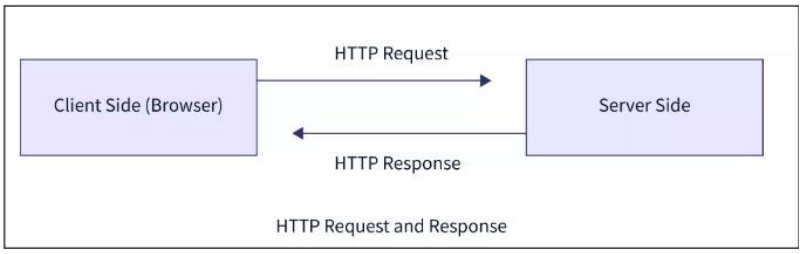
step-by-step process of how HTTP works, from DNS resolution to display the web page content on your browser. So, let's dive in and explore the inner workings of HTTP!

1. **DNS resolution :**-DNS Resolution: Browser resolves the domain name to an IP address via DNS query.



2. TCP connection :- Is established between the client's browser and the server on port number 80 .

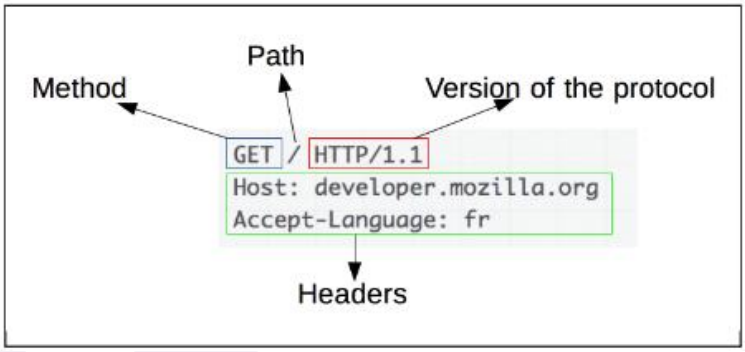
3. Establishing an HTTP connection:- The HTTP protocol is used to facilitate communication between the client (browser) and the server.



Request and response

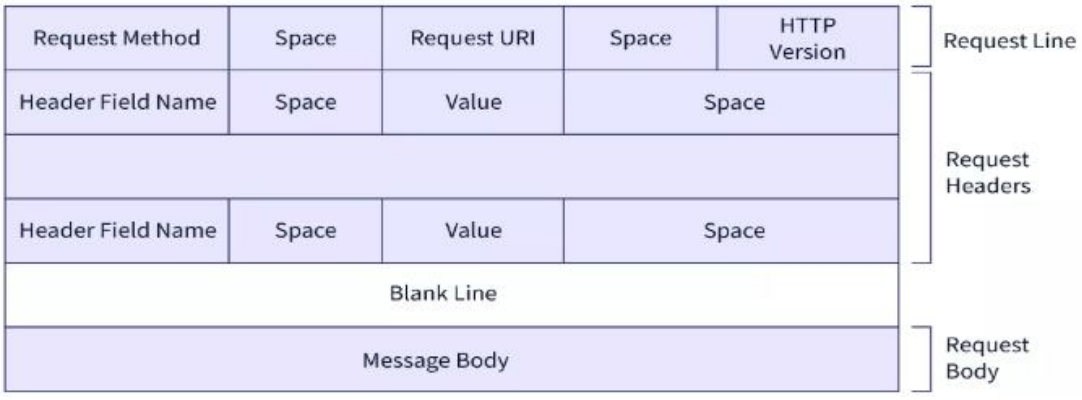
- **Request and response:-** Browser sends request (method, URL, headers) to server; server responds with web page content, status code, and headers; browser displays the web page.

1. HTTP Request.



Method: Specifies the action (GET, POST, PUT, DELETE).
Path: Identifies the resource (e.g., web page URL).
Headers: Include additional details (user agent, content types, cookies).
Body (optional): Sends extra data (e.g., form submissions, JSON payloads).

- Request line.
- Request headers.
- Request body (optional).



2. Response

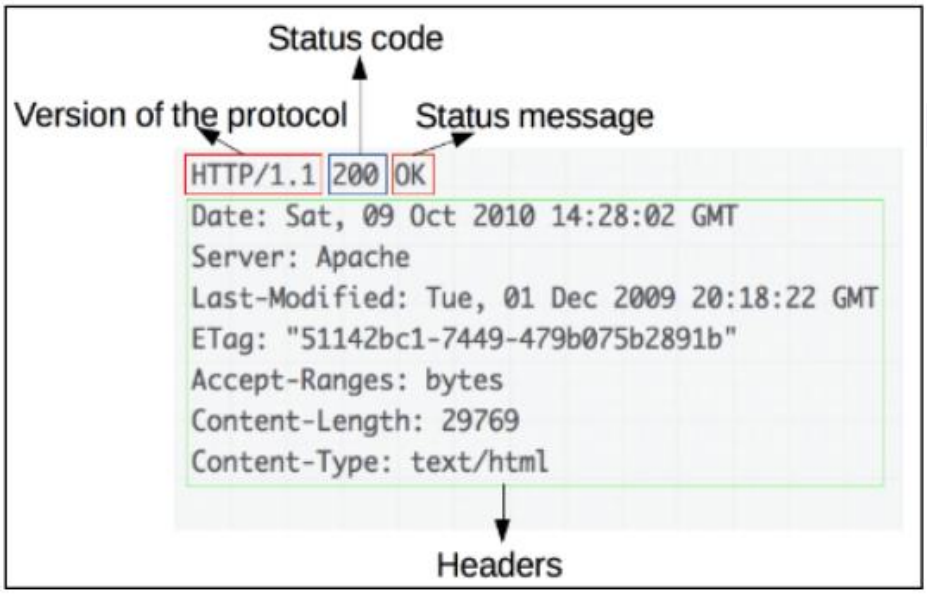
Status Code:

- Informational responses (100 – 1 99)
- Successful responses (200 – 2 99)
- Redirection messages (300 – 3 99)
- Client error responses (400 – 4 99)
- Server error responses (5 00 – 599)

Headers:

- **User-Agent:**
- Accept
- Authorization
- Content-Type
- Cache-Control
- Cookie

Body: Contains the actual content of the response, such as HTML, JSON, or images; may be empty for some responses.



Question Time

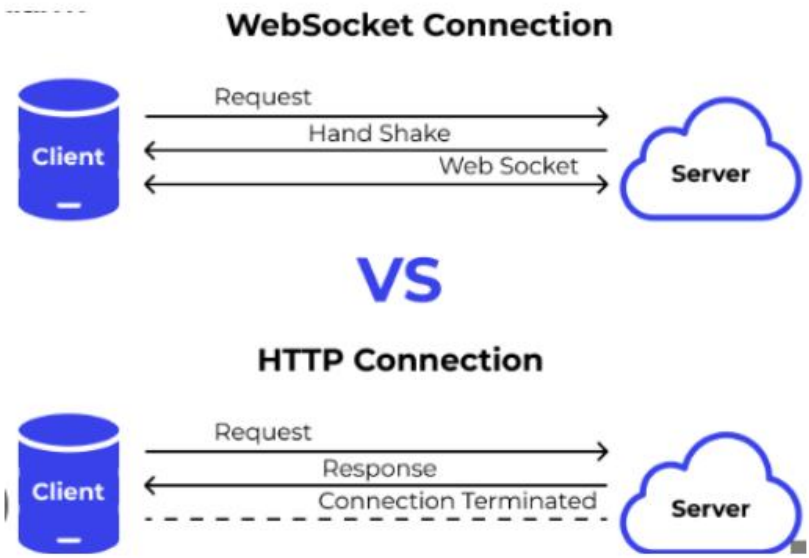
Question: Why do search engines prefer HTTPS over HTTP?
Question: What are the potential vulnerabilities in HTTPS

WebSockets for Real-Time Communication

WebSockets enable real-time, two-way communication between a client and a server, unlike HTTP which requires repeated requests.

Differences between WebSockets and HTTP

- **Connection Model:** HTTP is request-response, while WebSockets provide a persistent, full-duplex connection.
- **Overhead:** WebSockets have significantly less overhead compared to HTTP.
- **Latency:** WebSockets offer lower latency due to the persistent connection.

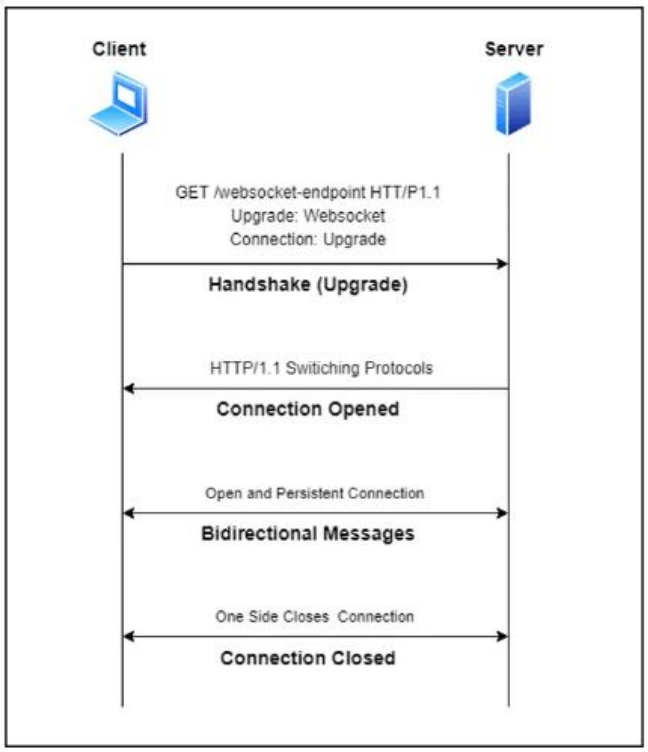


WebSocket Protocol and Handshake:

WebSocket Protocol: Operates over a single, long-lived connection allowing independent message sending by both client and server.

Handshake Process: Begins with an HTTP request from the client and a server response with an upgrade header to establish the connection.

Connection Upgrade: The server responds with a 101 Switching Protocols status code, upgrading to a persistent WebSocket connection.



Benefits of Using WebSockets

- **Low Latency Communication:** Enables real-time updates with minimal delay.
- **Full-Duplex Communication:** Allows simultaneous sending and receiving of data, improving efficiency.
- **Reduced Overhead Compared to HTTP:** Maintains a single, long-lived connection, reducing the need for multiple connections and enhancing performance.

WebSockets for Real-Time Communication

Setting Up WebSocket Communication

Client-Side Setup (JavaScript)

```
const socket = new WebSocket('ws://example.com/socketserver');
socket.onopen = function(event) {
    socket.send('Hello Server!');
};
socket.onmessage = function(event) {
    console.log('Message from server ', event.data);
                                        };
```

Server-Side Setup (Node.js, Python, etc.)

```
const WebSocket = require('ws');
const server = new WebSocket.Server({ port: 8080 });

server.on('connection', function(socket) {
    socket.on('message', function(message) {
        console.log('Received: ' + message);
        socket.send('Hello Client!');
    });
});
```

WebSocket Communication Patterns

- **One-to-One Communication:** Direct communication between two parties, such as a private chat between two users.
- **One-to-Many Communication (Broadcast):** A single sender transmits messages to multiple recipients, useful for live streaming or notifications.
- **Many-to-Many Communication (Chat Rooms):** Multiple users communicate with each other in a shared environment, like a chat room or collaborative application.

Security Considerations: WebSockets support TLS (wss://) for encrypted communication.

Question Time!

Question: How does WebSockets communication differ from HTTP?
Question: Name three common use cases for WebSockets.

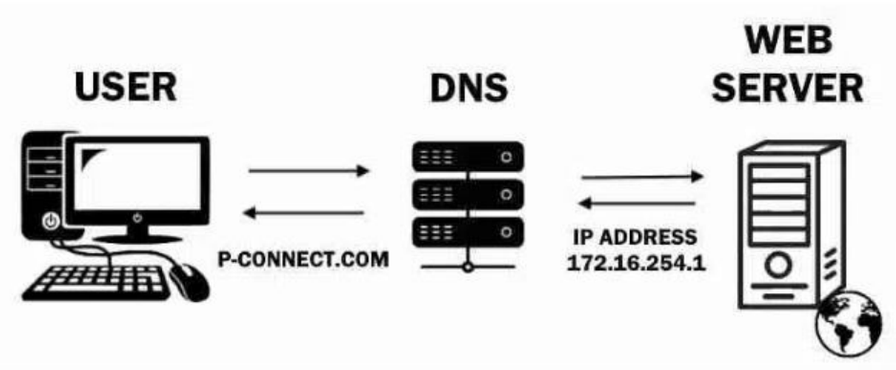
DNS and DHCP Protocols

- DNS and DHCP are both client-server architectures.
- DNS maps domain names to IP addresses.
- DHCP allocates IP addresses to hosts in a network, either statically or dynamically.
- DHCP connects hosts to DNS servers.

DNS (Domain Name System):

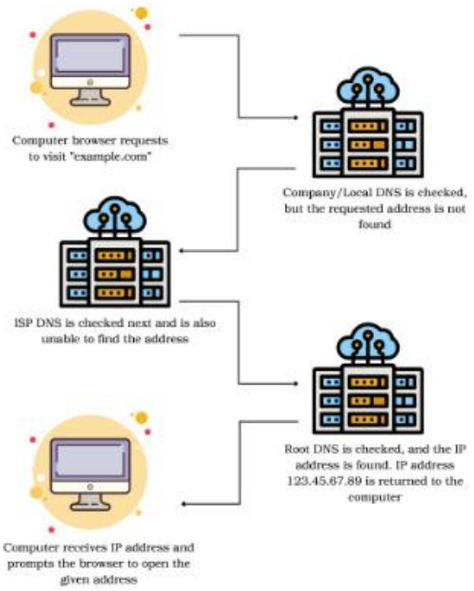
DNS translates domain names (e.g., www.example.com) to IP addresses (e.g., 192.168.1.1), making it easier for users to access websites by name rather than numerical IP addresses, acting as the internet's address book.

- **Purpose**: Converts domain names to IP addresses.
- **Components**: Domain names, IP addresses, DNS servers, and resolvers.
- **Process**: When a user types a URL, the DNS server resolves it to the corresponding IP address.



How does DNS work?

- **Information Request:** When you enter a website's domain in your browser, it checks local cache first. If not found, a DNS query is sent out.
- **Recursive DNS Servers:** Your ISP's servers handle the query, possibly using cached data. If not cached, they find the information or direct further.
- **Root Name Servers:** Directs queries to servers that know where to find the requested data, but don't store it themselves.
- **TLD Name Servers:** Guides to servers holding specific domain information (.com, .org, etc.).
- **Authoritative DNS Servers:** Hold specific DNS records, like IP addresses.
- **Retrieve the Record:** Recursive servers fetch and cache the record for future use.
- **The Final Answer:** Once obtained, the IP address is sent back, allowing your browser to connect and display the website.



Advantages and Disadvantages of DNS

Advantages

- Internet Speed
- IP address Conversion
- Internet Dependency
- Stability
- Security

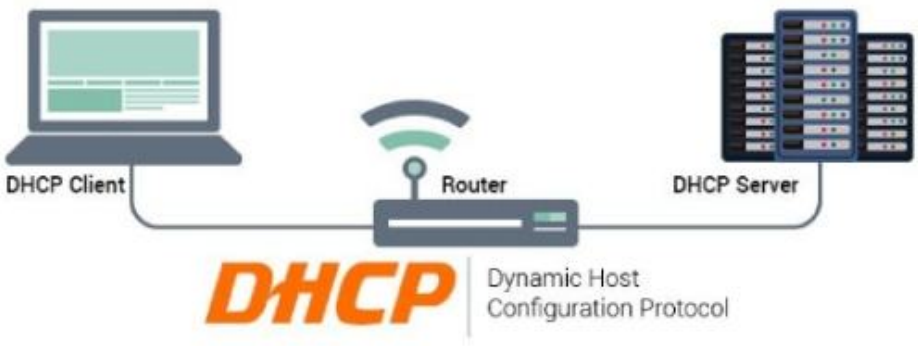
Disadvantages

- DNS Attacks
- Registry Control
- Server Breakdown
- Troubleshooting
- Client Information

DHCP (Dynamic Host Configuration Protocol):

DHCP automates IP network device configuration by assigning IP addresses and network settings, facilitating network communication without manual setup.

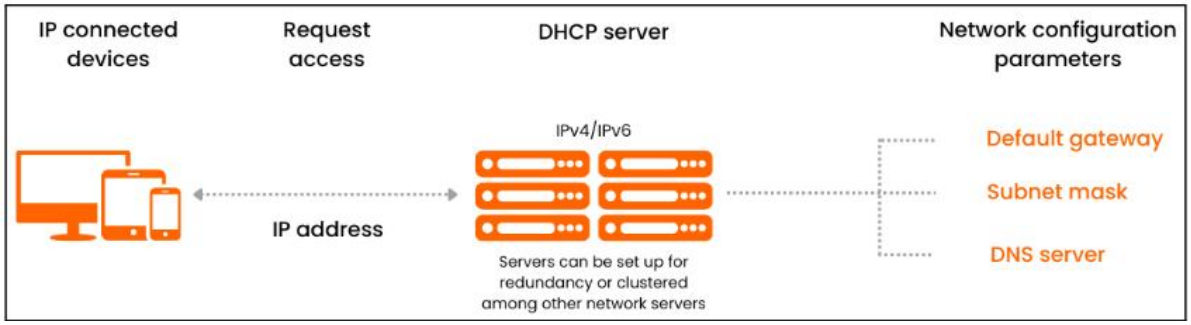
- **Purpose:** Automatically assigns IP addresses and network settings.
- **Components:** DHCP server, DHCP client, and IP address pool.
- **Process:** When a device connects to the network, the DHCP server assigns it an IP address from a predefined range and provides other necessary configuration details, such as subnet mask, default gateway, and DNS server address



How does DHCP work?

- **DHCP Discover:** When a new device connects without an IP address, it broadcasts a request for one. The DHCP server receives this request.
- **DHCP Offer:** The DHCP server responds by offering an IP address to the device.
- **DHCP Request:** The device accepts the offered IP address and formally requests to use it.
- **DHCP Pack:** Upon receiving the request, the DHCP server assigns the IP address, along with subnet mask and DNS server details. It logs this information, including the device's MAC address and lease expiration time. DHCP leases are temporary, and the IP address returns to the pool after expiration.

Communication typically uses UDP ports 68 for clients and 67 for servers, though specifics may vary by network equipment vendors.



Advantages and Disadvantages of DHCP

Advantages

- Simplifies Network Management
- Efficient Resource Allocation
- Centralized Network Configuration
- Reduces Configuration Errors
- Supports Mobility

Disadvantages

- Dependency on DHCP Server
- Potential for IP Address Exhaustion
- Security Concerns
- Limited Control Over IP Addressing
- Complex Troubleshooting

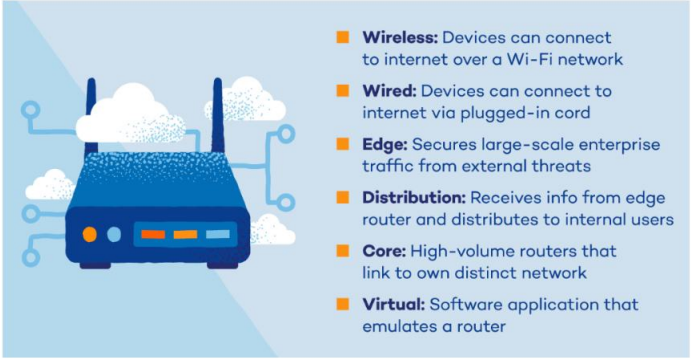
Question Time

Question: What is the primary function of DNS?
Question: Give an example of a DNS function in action.
Question: How does DHCP enhance the connection between a host and a DNS server?
Question: How does DHCP enhance the connection between a host and a DNS server?

Network Routing, Switching, and Addressing

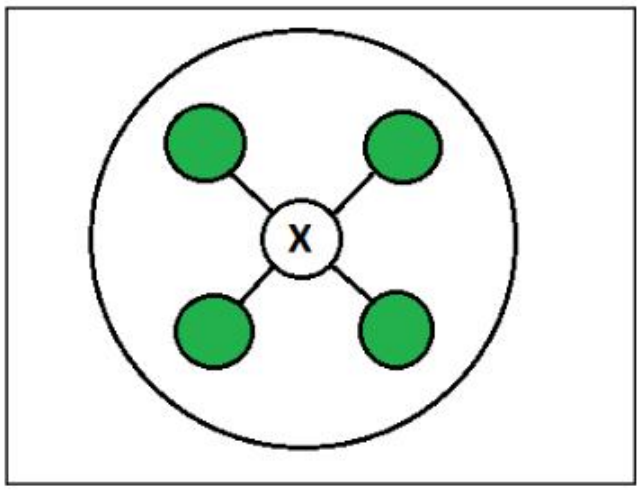
Network Routing : Networking routing directs data between networks via routers, optimizing transmission by selecting paths based on destination addresses. Protocols such as RIP, OSPF, and BGP manage routing tables dynamically or statically to ensure efficient data flow.

Types of Routers:



Routing Tables

A routing table in a router or network device lists available paths to network destinations, including addresses (destination and next hop), route metrics or costs, and the interface to reach each destination.



Key components of a routing table include:

- **Destination Network:** The IP address or network prefix of the destination network.
- **Next Hop:** The IP address of the next router or gateway on the path to the destination network.
- **Interface:** The network interface through which packets should be forwarded to reach the next hop.
- **Metric:** A value used to determine the cost or preference of the route, which helps routers select the best path among multiple options.

Routing Protocols

Routing protocols are used to determine the best paths for data packets to travel across networks.

Key characteristics of routing protocols include:

- **Route Calculation:** Routing protocols compute and update routes based on network topology, link costs, and administrative preferences.
- **Route Exchange:** Routers exchange routing information regularly or reactively to network changes, ensuring all routers know current network paths.
- **Types of Protocols:** Includes IGPs (e.g., OSPF, RIP,EIGRP) for internal networks and EGPs (e.g., BGP) for inter-network connections.
- **Dynamic Updates:** Routing protocols adjust routes dynamically in response to network events like link failures or new devices.
- **Routing Metrics:** Each protocol uses metrics (e.g., hop count, bandwidth) to find the optimal path based on speed, reliability, or cost considerations.

Static vs Dynamic Routing :

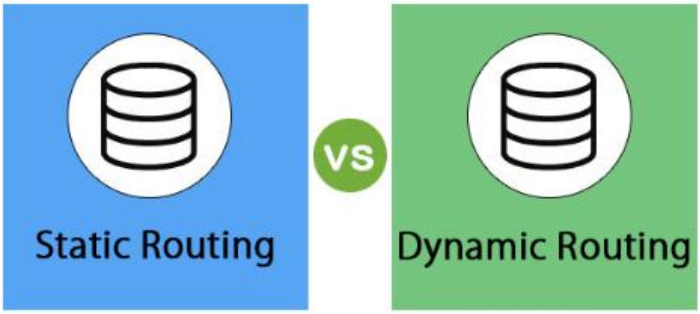
Configuration:
Static: Manually set routes.
Dynamic: Automatically adjusted routes.

Scalability:
Static: Best for small networks.
Dynamic: Best for large networks.

Maintenance:
- **Static:** Manual updates.
- **Dynamic:** Automatic updates.

Adaptability:
- **Static:** Fixed routes.
- **Dynamic:** Adapts to changes.

Resource Usage:
- **Static:** Low resource usage.
- **Dynamic:** Higher resource usage.



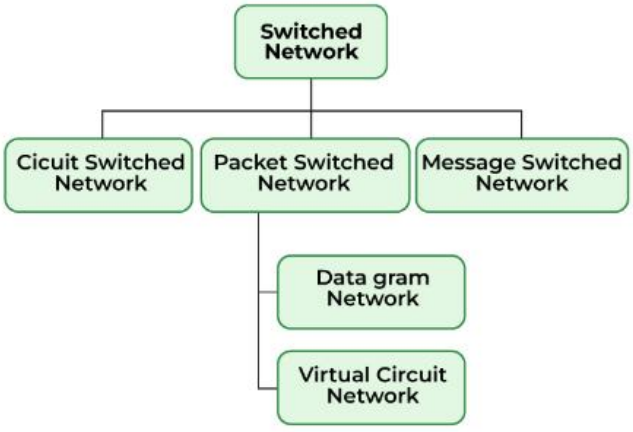
Network Switching

Network switching refers to the process of forwarding data packets from one network device to another within a computer network. It involves making decisions based on the destination address of each packet to determine the most appropriate path for transmission. Switching operates at Layer 2 (Data Link Layer) of the OSI model

Types of Switches:

There are three types of switching methods:

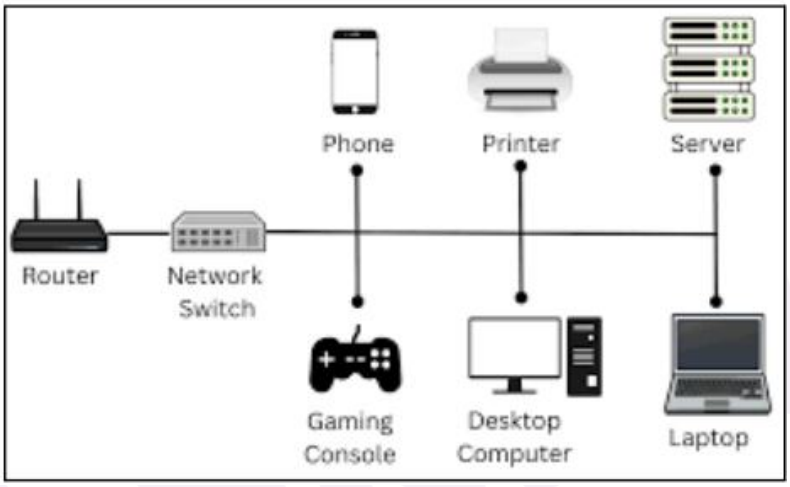
- **Message Switching:** Entire messages are stored and forwarded by intermediate nodes.
- **Circuit Switching:** A dedicated path is established for the entire session.
- **Packet Switching:** Data is broken into packets, sent independently, and reassembled.
  - **Datagram Packet Switching:** Each packet is routed independently without a predefined path.
  - **Virtual Circuit Packet Switching:** A predefined path is set up for all packets, ensuring order and reliability.



How Does a Network Switch Work?

- **Frame Reception:** The switch receives a data frame from a connected device.
- **MAC Address Extraction:** The switch reads the frame header and extracts the destination MAC address.
- **MAC Address Table Lookup:** The switch looks up the MAC address in its switching table to find the corresponding port.

- **Forwarding Decision and Switching Table Update:**
  - If the destination MAC address is found, the switch forwards the frame to the appropriate port.
  - If not found, the switch floods the frame to all ports except the source port and updates its table with new MAC addresses.
- **Frame Transition:** The switch sends the frame to the identified port and forwards it to the target device.

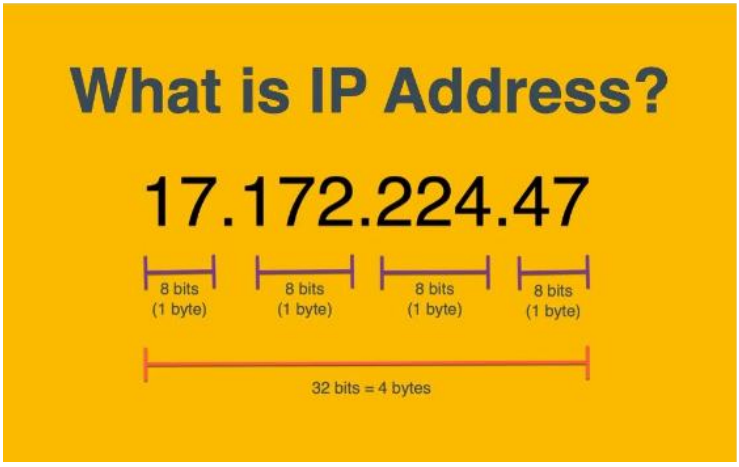


Network addressing:

Refers to the systematic assignment of identifiers to devices and nodes within a computer network. These identifiers enable devices to communicate with each other over the network. Key components of network addressing include

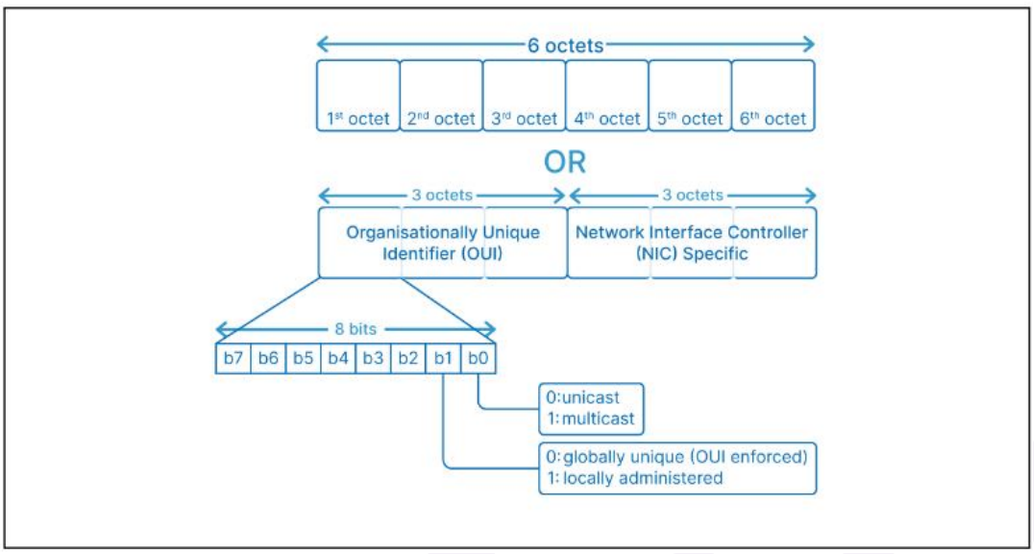
IP Addressing:

IPv4: Uses a 32-bit address expressed in dotted-decimal format (e.g., 192.168.1.1).
IPv6: Uses a 128-bit address expressed in hexadecimal format (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
IP addresses are assigned to devices (like computers, servers, routers) to uniquely identify them on a network.



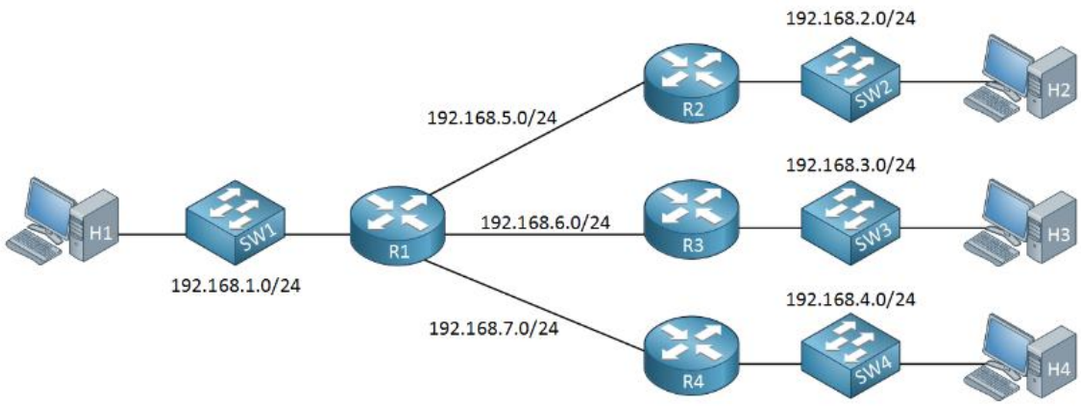
MAC Addressing:

Also known as the physical address or hardware address.
Assigned to network interface controllers (NICs) by manufacturers.
Used for communication within a local network segment (Layer 2 of the OSI model).



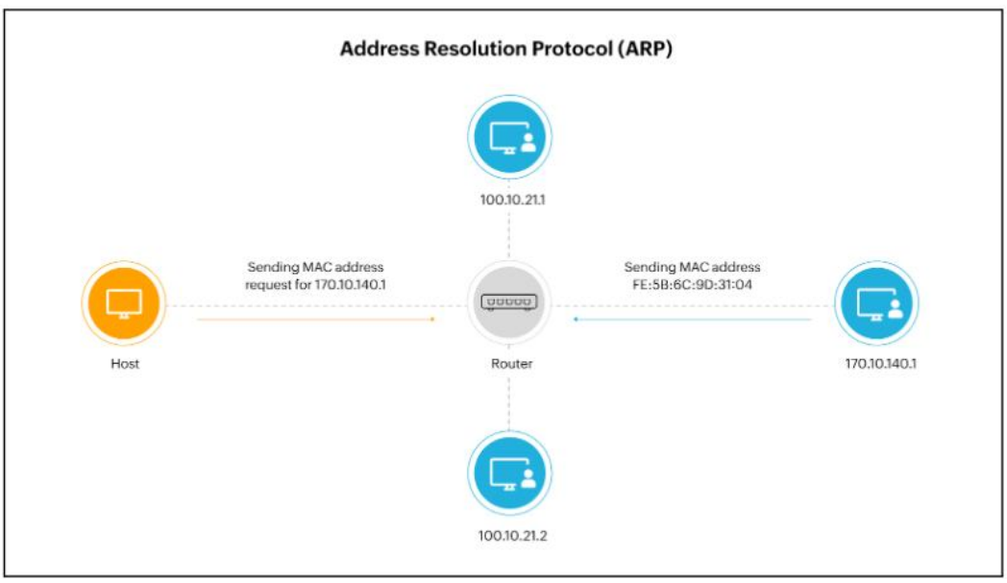
Subnetting:

Divides a single IP network into multiple smaller sub-networks (subnets).
Helps in efficient management of IP addresses and routing within larger networks.



Address Resolution Protocol (ARP):

Translates IP addresses to MAC addresses within a local network.
Helps devices find the hardware address of a device based on its IP address.



Question Time

Question: Explain the difference between static and dynamic routing.
Question: What are Virtual LANs (VLANs), and why are they used?
Question: What are Virtual LANs (VLANs), and why are they used?
Question: What are the main switching techniques, and how do they differ?

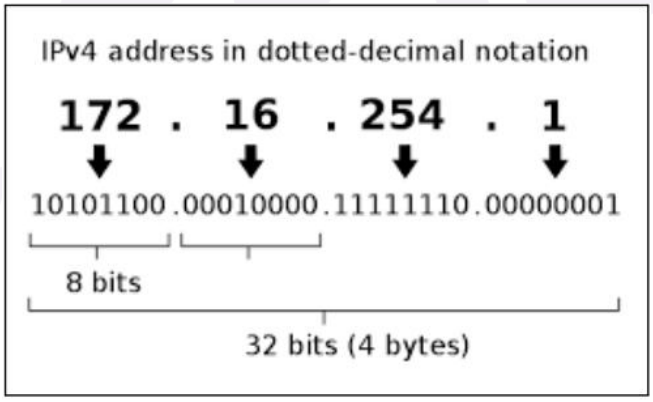
IP Addressing

Introduction to IP Addressing
IP Addressing assigns unique numerical identifiers to devices on a network, enabling communication and facilitating data routing over IP networks.

Purpose and Function
IP addresses facilitate the routing of data between devices on different networks, ensuring that information reaches the correct destination

IPv4 Addressing
IPv4 addresses are 32-bit numerical identifiers divided into four octets, separated by periods (e.g., 192.168.1.1).

Classes of IPv4 Addresses
IPv4 addresses are categorized into five classes (A, B, C, D, and E) based on their leading bits and intended usage.

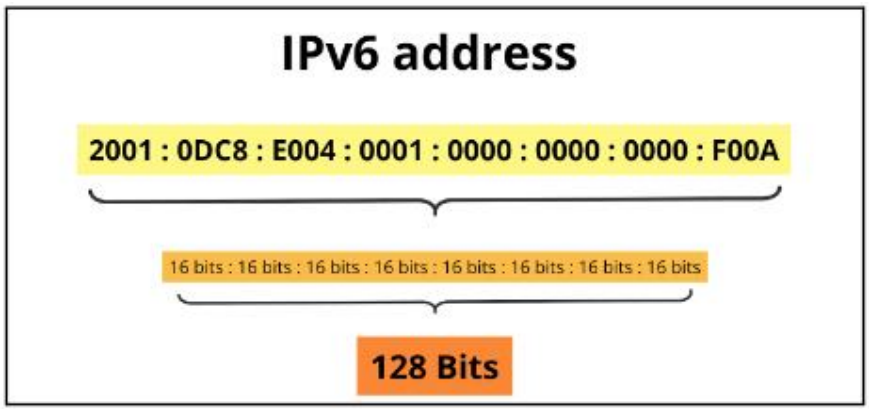


IPv6 Addressing

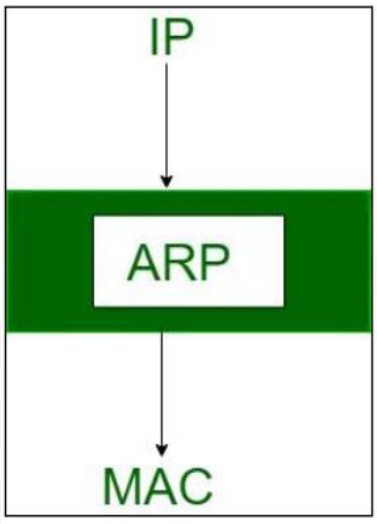
IPv6 addresses are 128 bits long, written in hexadecimal notation and separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

Benefits over IPv4
IPv6 offers a larger address space, improved routing efficiency, and better security features.

Address Types
- **Global Unicast:** Publicly routable addresses.
- **Link-Local:** Used for communication within a single network segment.
- **Multicast:** Addresses used to deliver packets to multiple destinations.
- **Anycast:** Addresses assigned to multiple interfaces, with data delivered to the nearest one.



Address Resolution Protocol (ARP): ARP resolves IP addresses (logical addresses) to MAC addresses (physical addresses) within a local network segment.



IP Address Management (IPAM):

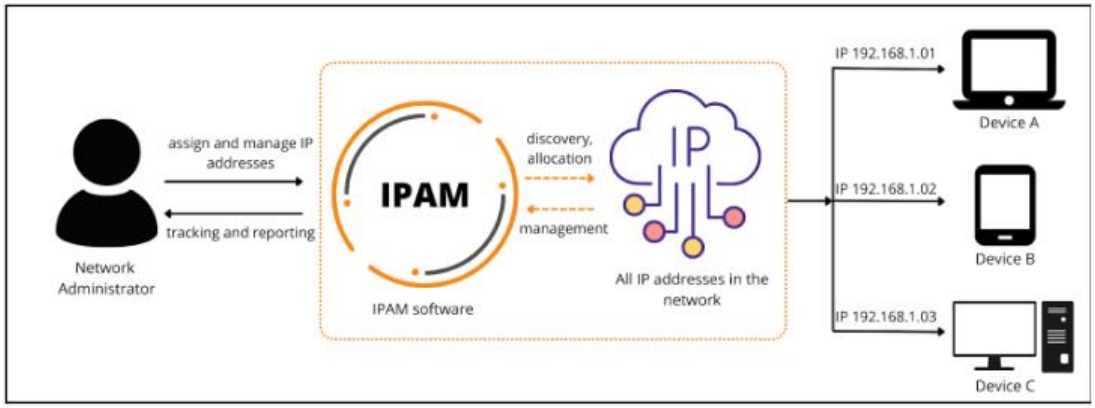
IPAM ensures efficient allocation, tracking, and management of IP addresses across large networks.

A typical IPAM software can:

- Show you the available IPs at the moment.
- The status of the IPs – permanent or temporary.
- The hostname related to each IP address.
- Routers in use by each device.
- Show the subnets, who is using them and how large they are.

Key Functions of IPAM

- **Discovery**: Identifies and records all IP addresses and their associated devices in the network. This can include automatic discovery through network scanning.
- **Allocation**: Manages the assignment of IP addresses to devices, ensuring that addresses are assigned from the correct pools and following policies.
- **Utilization Monitoring:** Tracks IP address usage to ensure efficient use of address space and to identify underutilized or overutilized addresses.
- **Conflict Detection:** Detects and resolves IP address conflicts, which can occur when two devices are assigned the same IP address.
- **Reporting:** Provides detailed reports on IP address usage, trends, and potential issues, helping administrators make informed decisions.



Question Time

Question: Differentiate between IPv4 and IPv6 addressing.

Security Protocols (SSL/TLS)

Introduction to SSL/TLS

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are cryptographic protocols.
Ensure secure communication over a computer network.
Essential for securing data transmission over the internet.

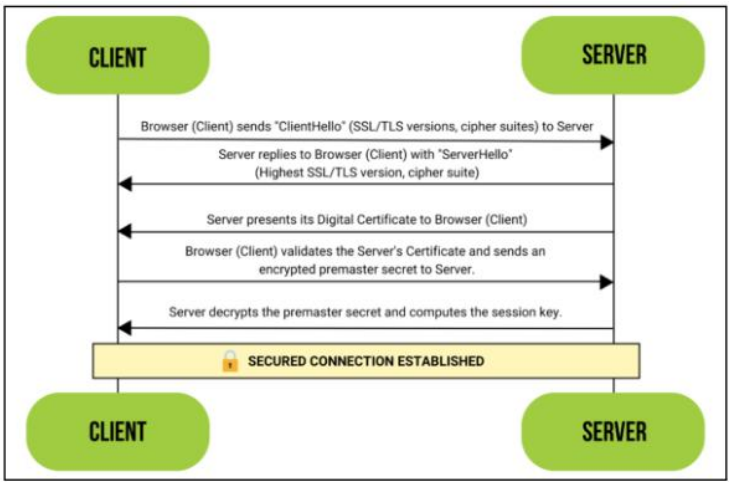
Key Objectives

- **Encryption:** Encrypts data to prevent unauthorized access during transmission.
- **Authentication:** Verifies the identity of parties involved in the communication.
- **Integrity:** Ensures data remains intact and unaltered during transmission.
- **Trust:** Establishes trust between communicating parties through certificates.

SSL/TLS Handshake Process

Overview

- **Client Hello:** Initiates the handshake by sending supported cryptographic algorithms.
- **Server Hello:** Responds with selected cryptographic algorithms and server certificate.
- **Key Exchange:** Client and server agree on session keys for encryption.
- **Encryption:** Secure data transmission begins using agreed-upon keys.



Components of SSL/TLS

Key Components

- Certificates: Digital certificates issued by Certificate Authorities (CAs) to verify identities.
- Cipher Suites: Sets of encryption algorithms and key exchange methods.
- Protocols: Versions such as SSL 3.0, TLS 1.0, TLS 1.2, TLS 1.3 (latest).

Importance of SSL/TLS in Security

Significance

- Data Protection: Safeguards sensitive information like passwords, credit card numbers.
- Trustworthiness: Builds user trust through secure connections.
- Compliance: Meets regulatory requirements for data protection (e.g., GDPR, PCI DSS).

Question Time

Question: What is the purpose of the SSL/TLS handshake protocol in secure communication?
Question: What is the difference between SSL and TLS?

Network Troubleshooting Tools (ping, traceroute, tcpdump)

Introduction to Network Troubleshooting

Network troubleshooting involves diagnosing and resolving issues within a network. This can include connectivity problems, performance bottlenecks, and security breaches. Effective troubleshooting requires a good understanding of the network architecture and the tools available to identify and solve problems.

Ping

Ping (Packet Internet Groper) is used to test connectivity between two network nodes by sending ICMP echo request packets and receiving ICMP echo reply packets.

How Ping Works

Ping uses the Internet Control Message Protocol (ICMP) to send echo request packets to the target host and listens for echo response replies. The time taken for the round trip is recorded, along with any packet loss.

Practical Uses of Ping

Connectivity Testing: Verify if a device is reachable over the network.
Latency Measurement: Determine the round-trip time to a destination.
Packet Loss Detection: Identify if packets are being lost in transit.

Traceroute

Traceroute (or traceroute6) tracks the route packets take across an IP network to reach a specific destination, showing each hop and measuring transit delays.

How Traceroute Works

Traceroute sends packets with gradually increasing Time to Live (TTL) values. Each router along the path decreases the TTL by 1 until it reaches zero, then sends an ICMP "Time Exceeded" message back to the source. This process reveals each hop on the path to the destination

Practical Uses of Traceroute
Path Discovery: Identify the route taken by packets to reach a destination.
Latency Measurement: Determine the delay at each hop along the path.
Network Troubleshooting: Identify where delays or failures occur.

Tcpdump
Tcpdump captures and analyzes network packets in real-time, providing detailed information about traffic, protocols used, and specific packet contents.

How Tcpdump Works
Tcpdump captures packets as they pass through the network interface, displaying them in real-time or saving them for later analysis. It can filter packets based on criteria such as IP addresses, protocols, and ports.

Practical Uses of Tcpdump
- **Traffic Analysis:** Examine network traffic to understand data flow and detect anomalies.
- **Troubleshooting:** Identify issues such as packet loss, delays, and errors.
- **Security Monitoring:** Detect suspicious activities and potential security breaches.

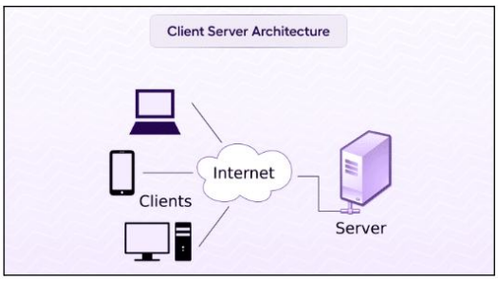
Comparison and Selection of Tools:

- Ping for basic connectivity checks.
- Traceroute for path analysis and latency measurements.
- Tcpdump for detailed packet inspection and troubleshooting.

Question Time
Question: What is the purpose of network troubleshooting tools?
Question: What is tcpdump and how is it used in network troubleshooting?

Client Server Architecture

Client-Server Architecture is a computing model where client and server devices interact to perform tasks and share resources over a network. It enables efficient distribution of workload and centralized management of resources.



Components of Client-Server Architecture

- The client is a device or software application that requests services or resources from a server. It initiates communication and handles user interactions.
- The server is a centralized system or software application that provides services, resources, or data to clients over a network. It processes client requests and manages resources.
- n-Tier Architecture: A multi-layered architecture that includes multiple layers (beyond three) to separate different aspects of the application for scalability and manageability.

Types of Client-Server Architectures
Two-Tier Architecture: A client-server architecture where the client communicates directly with the server.
Three-Tier Architecture: An architecture consisting of the presentation layer (client), business logic layer (application server), and data layer (database server).

n-Tier Architecture: A multi-layered architecture that includes multiple layers (beyond three) to separate different aspects of the application for scalability and manageability.

Two-Tier Architecture

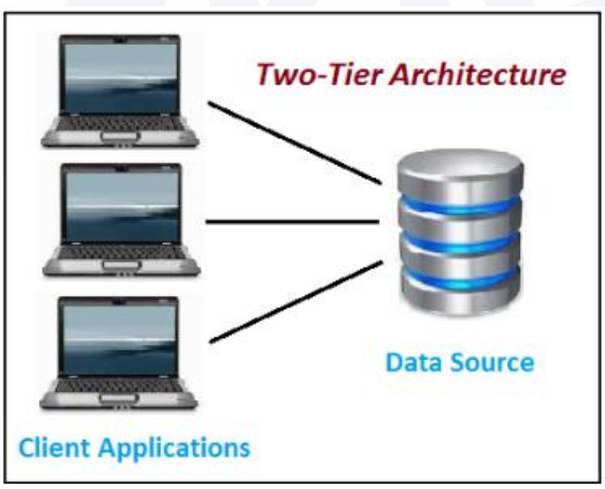
Two-tier architecture is a basic client-server architecture model where the user interface runs on the client and the database is stored on the server. This architecture is characterized by a direct communication path between the client and the server.

Components of Client-Server Architecture

Client (Presentation Layer):
- User Interface: Presents data and collects user input through GUI or CLI.
- Application Logic: Manages tasks like input validation, user authentication, and session management.

Server (Data Layer):
- DBMS: Hosts and manages the database, handling data storage, retrieval, and SQL queries.
- Data Access Logic: Manages data operations such as retrieval, updates, inserts, and deletes.

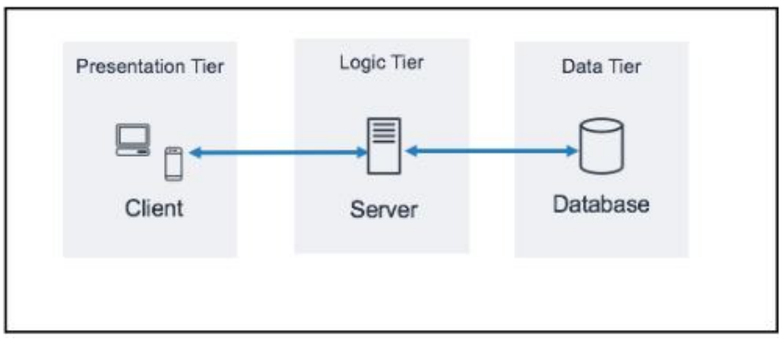
How It Works
- **Client Request:** The user interacts with the client application, which sends a request to the server. This request typically involves querying the database or submitting data to be stored.
- **Server Processing:** The server receives the request, processes it, and performs the necessary operations on the database.
- **Response:** The server sends the response back to the client, which processes and displays the information to the user.



Three-Tier Architecture
Three-tier architecture is a well-established software design pattern that separates applications into three logical and physical computing tiers: the presentation tier, the application tier (business logic layer), and the data tier. This separation improves scalability, manageability, and reusability of applications.

Components of n-Tier Architecture
- **Presentation Tier (Client Tier):**
- **UI:** User interface components (web, desktop, mobile) that present data and capture user input.
- **Interaction:** Communicates with the application tier via HTTP/HTTPS or other protocols.
- **Application Tier (Business Logic Tier):**
- **Business Logic:** Core application logic that processes data, handles business rules, and workflows.
- **Application Server:** Hosts the business logic, using technologies like Java EE, .NET, and Spring Framework.

How It Works
- **Client Request:** The user interacts with the client application (presentation tier), which sends a request to the application server (application tier).
- **Business Processing:** The application server processes the request using business logic, which may involve accessing or updating data in the database (data tier).
- **Database Interaction:** If data access is required, the application server communicates with the database server to retrieve or modify data.
- **Response:** The application server sends the processed data back to the client application, which presents it to the user.



n-Tier Architecture

n-Tier architecture is a software design pattern that separates an application into multiple layers or "tiers," each responsible for a specific aspect of the application's functionality. This architecture extends beyond the traditional three-tier model, introducing additional layers to enhance modularity, scalability, and manageability.

Components of n-Tier Architecture

Presentation Tier:
- UI: Manages user interfaces and presentation logic (web, desktop, mobile).

Application Tier:
- Business Logic: Processes data with business rules on application servers.

Service Tier (Optional):
- APIs: Exposes business logic as web services or APIs.

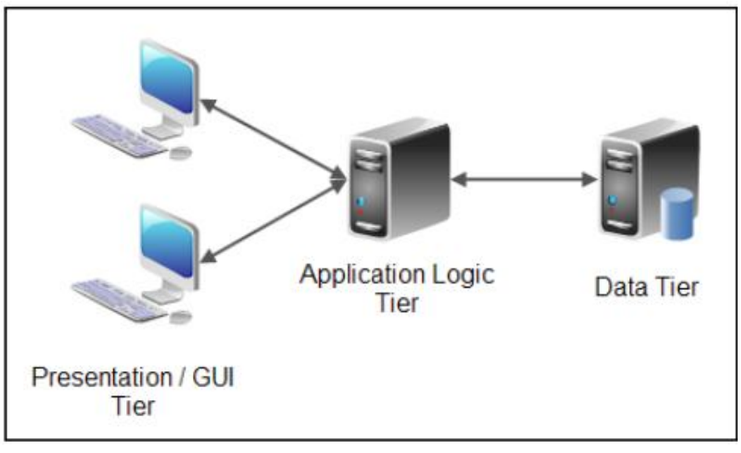
Persistence Tier:
- Data Access: Handles database interactions, including queries and transactions.

Database Tier:
- DBMS: Stores, retrieves, and manages data on database servers.

Integration Tier (Optional):
- ESB/Middleware: Facilitates communication and data exchange between systems.

How It Works

- **Client Request:** A user interacts with the presentation tier, which sends a request to the application tier.
- **Business Processing:** The application tier processes the request using business logic and may call services from the service tier.
- **Service Calls:** If a service tier is present, it handles requests from the application tier and interacts with the persistence tier for data access.
- **Database Interaction:** The persistence tier communicates with the database tier to retrieve or update data.
- **Integration:** If an integration tier is present, it handles communication between the application and other systems or services.
- **Response:** Data and results are passed back through the layers to the presentation tier, which displays them to the user.



Question Time

Question: What are the advantages of Client-Server Architecture?
Question: What are some common examples of Client-Server applications?