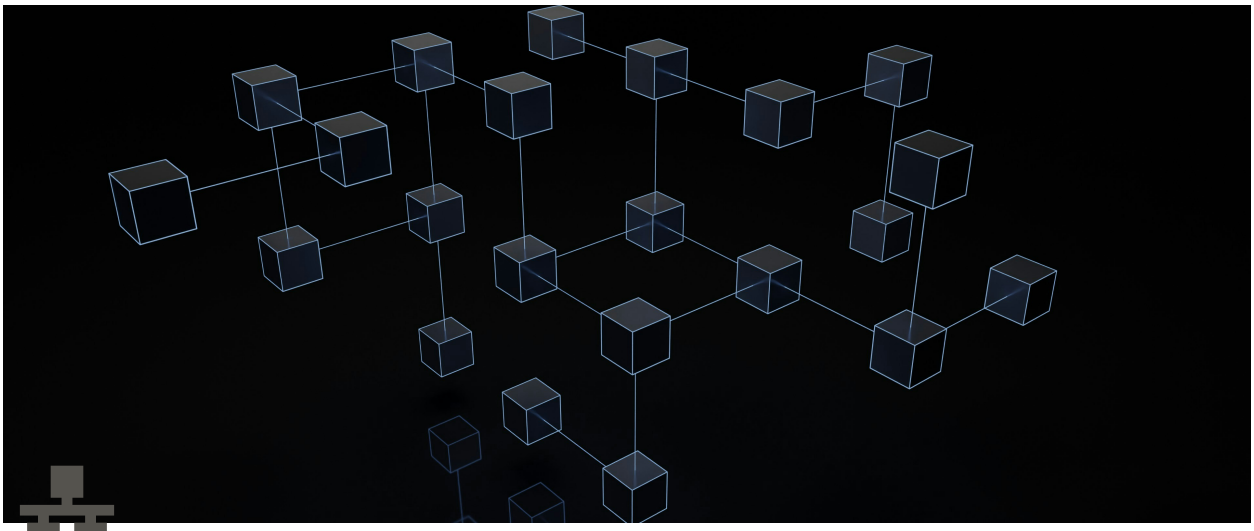
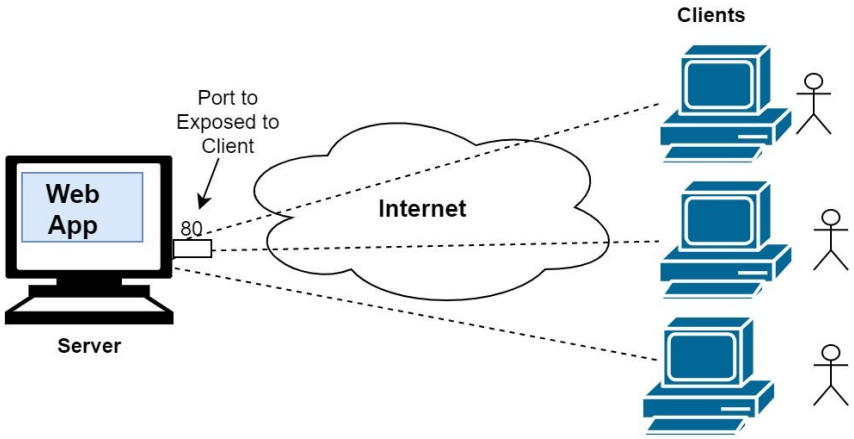


Networking Notes

1. Fundamentals of Networking

- **Networking**: The practice of connecting devices to share resources and information.
- **Nodes**: Devices on a network (e.g., computers, printers).
- **Hosts**: Devices with IP addresses (e.g., laptop).
- **Client-Server Architecture**: Clients request services; servers provide them (e.g., web browsers requesting data from web servers).



2. OSI Model

The **OSI Model (Open Systems Interconnection)** is a framework for understanding how different network protocols and devices interact. It consists of **7 layers**, each responsible for specific tasks.

1. Physical Layer

- **Function**: Deals with the physical connection between devices. Responsible for transmitting raw bits (0s and 1s) over physical media.
- **Examples**:
  - Cables (Ethernet, Fiber Optic)
  - Connectors (RJ45)
  - Network Interface Cards (NICs)
- **Real-World Scenario**: When you plug in an Ethernet cable, you're interacting with the Physical Layer. It ensures data travels as electrical signals.

---

2. Data Link Layer

- **Function**: Establishes, maintains, and terminates the connection between physically connected nodes. Handles **MAC addresses** and ensures data integrity with error detection and correction.
- **Examples**:
  - Ethernet
  - Wi-Fi (802.11 standards)
- **Real-World Scenario**: When you connect to Wi-Fi, the Data Link Layer manages your device's MAC address and ensures your data frames are error-free before forwarding them.

---

3. Network Layer

- **Function**: Handles routing and forwarding of data packets. Responsible for logical addressing (**IP addresses**) and determining the best path for data to travel.

- **Examples**:
  - IPv4 and IPv6
  - Routers
- **Real-World Scenario**: When you visit `google.com`, the Network Layer determines the best route to Google's server using your device's IP address.

---

4. Transport Layer

- **Function**: Ensures reliable data transfer between devices, managing end-to-end communication. Uses protocols like:
  - **TCP**: Reliable, ensures all data reaches the destination.
  - **UDP**: Faster but less reliable, often used for streaming.
- **Examples**:
  - TCP (e.g., for downloading a file)
  - UDP (e.g., for a video call)
- **Real-World Scenario**: When you download a file, TCP ensures all packets arrive and are reassembled in the correct order.

---

5. Session Layer

- **Function**: Manages sessions or connections between applications. Responsible for opening, closing, and managing communication sessions.
- **Examples**:
  - Login sessions on a website
  - File transfer sessions
- **Real-World Scenario**: When you log in to your bank's website, the Session Layer manages the connection between your browser and the bank's server.

---

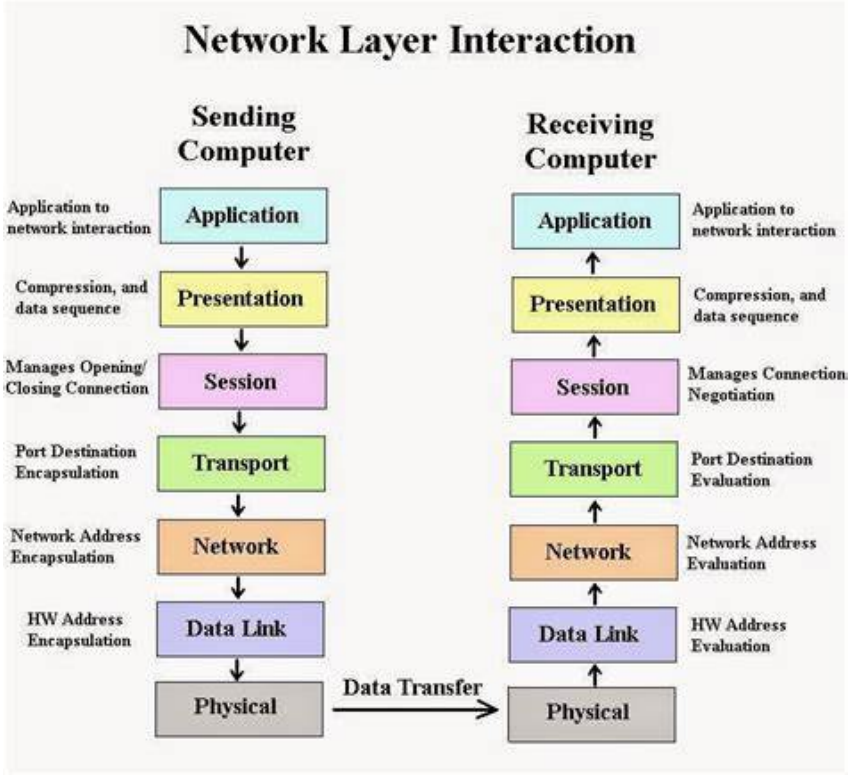
6. Presentation Layer

- **Function**: Translates data into a format the application can understand (e.g., encryption, compression).
- **Examples**:
  - Data encryption (SSL/TLS)
  - JPEG, PNG (image compression)
- **Real-World Scenario**: When you watch a movie on Netflix, the Presentation Layer ensures the video is compressed and decrypted correctly for viewing.

---

7. Application Layer

- **Function**: Provides network services to end users. Interfaces directly with applications.
- **Examples**:
  - HTTP/HTTPS (web browsing)
  - FTP (file transfer)
  - SMTP (email)
- **Real-World Scenario**: When you type `www.google.com` in your browser, the Application Layer interacts with the HTTP protocol to retrieve the webpage.



How Data Travels Through the OSI Model (Example: Sending an Email)

1. **Application Layer**: Your email application (e.g., Gmail) uses SMTP to compose and send the message.
2. **Presentation Layer**: Encrypts the email content for security.
3. **Session Layer**: Establishes a connection with the recipient's email server.
4. **Transport Layer**: Breaks the email into smaller packets and ensures reliability using TCP.
5. **Network Layer**: Adds the sender and recipient's IP addresses and determines the route.
6. **Data Link Layer**: Adds the MAC address of the recipient's network device.
7. **Physical Layer**: Converts data into electrical signals and transmits them through cables or Wi-Fi.

3 Protocols

Protocols are standardized rules that define how data is transmitted, received, and understood across networks. Each protocol operates at a specific layer of the OSI or TCP/IP model.

Common Protocols:

1. **HTTP (HyperText Transfer Protocol)**:
   - Port: **80**
   - Use: Transfer web pages.
   - Example: Typing `http://example.com` in a browser retrieves a webpage.
2. **HTTPS (HTTP Secure)**:
   - Port: **443**
   - Use: Secure version of HTTP using SSL/TLS for encryption.
   - Example: Logging into a bank website (`https://bank.com`).
3. **FTP (File Transfer Protocol)**:
   - Port: **21**
   - Use: Transfer files between a client and a server.
   - Example: Uploading a file to a web server.
4. **SMTP (Simple Mail Transfer Protocol)**:
   - Port: **25**
   - Use: Sending emails.
   - Example: Sending an email via Gmail or Outlook.
5. **DNS (Domain Name System)**:
   - Port: **53**
   - Use: Resolving domain names to IP addresses.
   - Example: Translating `www.google.com` to `142.250.190.14`.

---

4. Port Numbers

Ports act as logical communication endpoints for differentiating services on a device.

Categories of Ports:

1. **Well-Known Ports (0–1023)**: Reserved for common protocols like HTTP, FTP, etc.
2. **Registered Ports (1024–49151)**: For specific applications (e.g., database services).
3. **Dynamic/Private Ports (49152–65535)**: For temporary or client-side connections.

Real-World Example:

If your computer accesses a web server at `192.168.1.1`:

- The destination port is **80** (for HTTP).
- The source port is dynamically assigned (e.g., **49200**) to track the session.

5. What is TCP (Transmission Control Protocol)?

TCP is a **reliable, connection-oriented protocol** used in networking. It operates at the **Transport Layer** of both the **TCP/IP model** and the **OSI model**. TCP ensures data is delivered accurately and in the correct order between devices over a network.

Key Features of TCP

1. **Connection-Oriented**:
   - TCP establishes a connection before transferring data, ensuring reliability.
   - It uses the **3-Way Handshake** to set up the connection.
2. **Reliable Data Transfer**:
   - TCP ensures all data packets are delivered.
   - If packets are lost or corrupted, TCP retransmits them.

3. **Sequencing**:
    - Packets are numbered, ensuring they are reassembled in the correct order at the destination.
4. **Flow Control**:
    - TCP prevents overwhelming the receiver by adjusting the data flow based on the receiver's capacity.
5. **Error Detection and Correction**:
    - TCP uses checksums to detect errors in data and requests retransmissions if necessary.
6. **Full-Duplex Communication**:
    - Data can be sent and received simultaneously between two devices.

TCP 3-Way Handshake

The **3-Way Handshake** is a process used to establish a reliable connection between a client and a server.

1. **SYN (Synchronize)**:
    - The client sends a SYN packet to the server, requesting to establish a connection.
2. **SYN-ACK (Synchronize-Acknowledge)**:
    - The server acknowledges the client's request by sending a SYN-ACK packet back.
3. **ACK (Acknowledge)**:
    - The client responds with an ACK packet, confirming the connection.

Advantages of TCP

1. **Reliable**: Ensures error-free data delivery.
2. **Orderly**: Ensures data is received in the correct order.
3. **Widely Used**: Essential for applications requiring accuracy, like file transfers and web browsing.

Disadvantages of TCP

1. **Slower**: The reliability mechanisms introduce latency.
2. **Overhead**: TCP's features require additional resources, such as memory and processing power.

Common Applications Using TCP

- **Web Browsing**: HTTP/HTTPS
- **Email**: SMTP, IMAP, POP3
- **File Transfer**: FTP
- **Remote Access**: SSH, Telnet

TCP in Real-World Example

Scenario: Downloading a file from Google Drive

1. **Establish Connection**:
    - Your computer establishes a TCP connection with Google's server using a 3-Way Handshake.
2. **Data Transfer**:
    - The file is broken into TCP segments and sent. Each segment is acknowledged.
    - If a segment is lost, TCP retransmits it.
3. **Termination**:
    - Once the file transfer is complete, the connection is closed using a 4-Way Handshake.

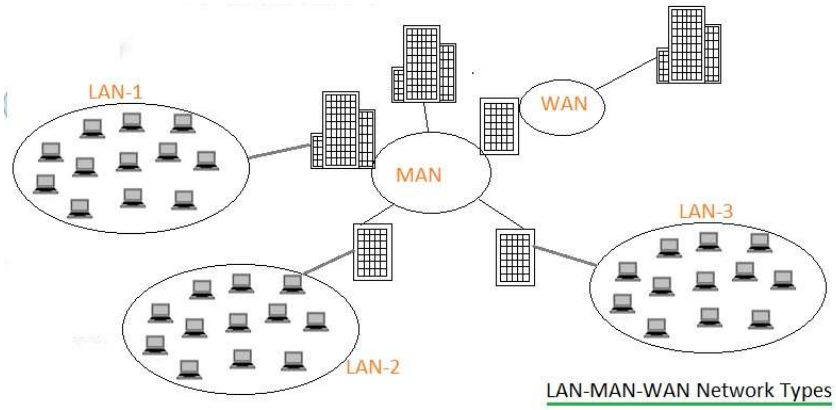
Comparison: TCP vs. UDP

Feature	TCP	UDP
Connection	Connection-oriented	Connectionless

Reliability	Reliable, ensures delivery	Unreliable, no delivery guarantee
Speed	Slower due to reliability checks	Faster, minimal checks
Use Case	File transfers, web browsing	Video streaming, gaming

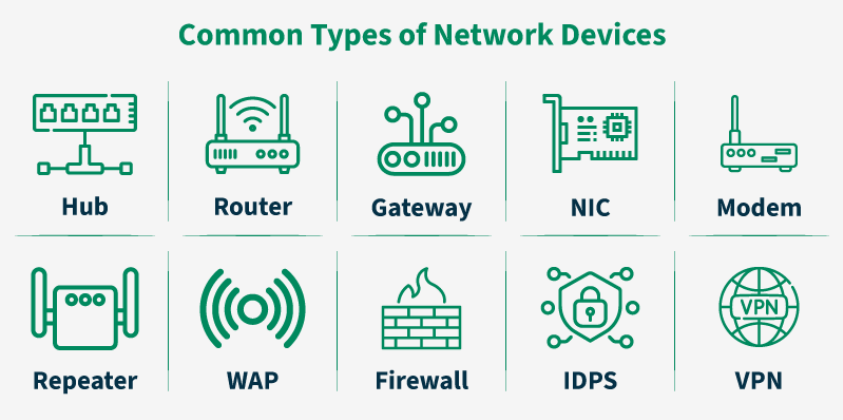
6. Network Types

- **LAN (Local)**: Small area (e.g., office Wi-Fi).
- **MAN (Metro)**: City-wide networks (e.g., cable TV network).
- **WAN (Wide)**: Large areas (e.g., the internet).



7. Devices and Terminology

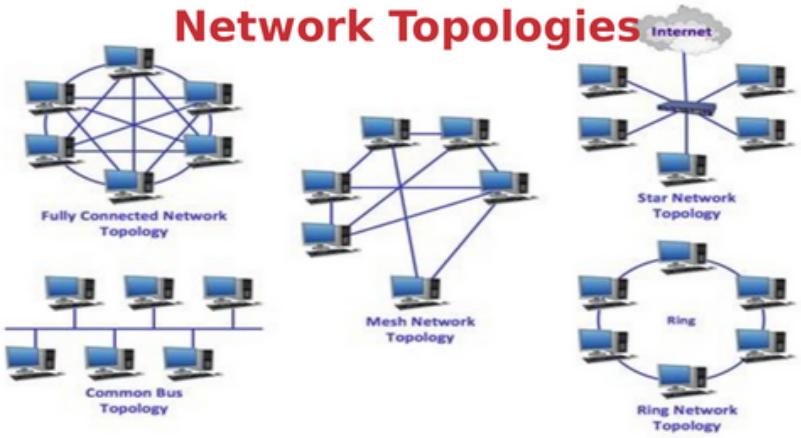
- **Router**: Connects networks (e.g., home Wi-Fi to the internet).
- **Modem**: Converts digital data to analog for transmission.
- **Switch**: Connects devices within a LAN.
- **Hub**: Broadcasts incoming data to all connected devices.
- **Gateway:** Acts as an entry/exit point between networks with different protocols.**Topologies**:



1. **Bus Topology**:
    - **Structure**: All devices are connected to a single cable.
    - **Advantages**: Simple and cheap.
    - **Disadvantages**: Failure of the main cable disrupts the network.
    - **Example**: Early Ethernet networks.
2. **Star Topology**:
    - **Structure**: All devices connect to a central hub/switch.
    - **Advantages**: Easy to manage; a failure of one device doesn't affect others.
    - **Disadvantages**: If the central hub fails, the network goes down.
    - **Example**: Office networks.
3. **Mesh Topology**:
    - **Structure**: Devices are interconnected.
    - **Advantages**: Highly reliable; multiple paths for data.
    - **Disadvantages**: Expensive and complex.
    - **Example**: Modern wireless networks.
4. **Ring Topology**:
    - **Structure**: Devices are connected in a circular manner.
    - **Advantages**: Predictable performance; easy to troubleshoot.

- **Disadvantages**: A single failure can disrupt the entire network.
- **Example**: Token Ring networks.

5. **Tree Topology**:
   - **Structure**: Hierarchical connection of multiple star topologies.
   - **Advantages**: Scalable; easier to manage.
   - **Disadvantages**: Expensive; central node failure affects the network.
   - **Example**: Used in corporate networks.



8. HTTP Methods

- **GET**: Retrieve data (e.g., searching on Google).
- **POST**: Send data (e.g., login forms).
- **PUT**: Update data.
- **DELETE**: Remove data.
- **Status Codes**:
  - **200**: OK
  - **201** Created
  - **404**: Not Found
  - **500**: Server Error

9. DNS (Domain Name System)

DNS is like the internet's phonebook. It maps human-readable domain names (e.g., `www.google.com`) to their corresponding IP addresses (e.g., `142.250.190.14`) so computers can communicate.

How DNS Works:

1. **User Action**: You type `www.google.com` in your browser.
2. **DNS Query**:
   - The browser checks its local cache for the IP address.
   - If not found, it sends a request to a **DNS Resolver** (usually provided by your ISP or public DNS like Google DNS at `8.8.8.8`).
3. **Recursive Lookup**:
   - The resolver queries the **Root DNS Server**, which points to the **Top-Level Domain (TLD) Server** for `.com`.
   - The TLD Server points to the **Authoritative DNS Server** for `google.com`.
4. **Response**: The resolver retrieves the IP address (e.g., `142.250.190.14`) and sends it to your browser.
5. **Connection**: Your browser uses the IP address to connect to the web server hosting `www.google.com`.

Real-World Example:

When you enter `youtube.com`:

- DNS translates `youtube.com` to its IP address (e.g., `142.250.190.46`) so your browser can access YouTube.

10. VPNs (Virtual Private Networks)

VPNs create secure, encrypted tunnels for data transmission over the internet, ensuring privacy and security.

Types of VPNs:

1. **Remote Access VPN**:
    - For individuals connecting to a private network remotely.
    - Example: A remote employee accessing their company's network securely.
2. **Site-to-Site VPN**:
    - Connects entire networks, often used between branches of an organization.
    - Example: A company's New York office connects to its London office via a VPN.
3. **Consumer VPNs**:
    - Used by individuals to hide their IP address, encrypt internet traffic, or bypass geo-restrictions.
    - Example: Using NordVPN to watch region-locked Netflix content.
4. **SSL VPNs**:
    - Utilizes SSL (Secure Sockets Layer) to provide secure access.
    - Example: Logging into a secure web portal from a browser.

Real-World Example:

- **Without VPN**: Your ISP can see which websites you visit, and your data is vulnerable to interception.
- **With VPN**: Your traffic is encrypted and routed through the VPN server, masking your activity and IP address.

Key Benefits of VPNs:

1. **Privacy**: Hides your online activity.
2. **Security**: Protects data on public Wi-Fi.
3. **Bypassing Restrictions**: Access blocked websites in specific regions.

Day 2

1. TCP/IP (Transmission Control Protocol/Internet Protocol)

The **TCP/IP** model is the foundation of internet communication, enabling data transfer between devices. It consists of four layers:

1. **Application Layer**: Handles protocols like HTTP, FTP, DNS.
    - Example: Browsing a website (HTTP).
2. **Transport Layer**: Ensures reliable data delivery using **TCP** (reliable) or **UDP** (fast, less reliable).
    - Example: Video streaming uses UDP.
3. **Internet Layer**: Routes data between networks using **IP**.
    - Example: Data is routed from your computer to a server using IP addresses.
4. **Network Interface Layer**: Handles hardware-level data transfer.
    - Example: Ethernet or Wi-Fi.

---

2. ARP (Address Resolution Protocol)

ARP maps an **IP address** to a **MAC address** (physical address) on a local network.

Example:

- You want to send data to `192.168.1.5`.
- ARP resolves the IP to the corresponding MAC address like `00:1A:2B:3C:4D:5E`.
- The data is then delivered to the correct device.

Commands:

- `arp -a` (view ARP cache).

---

3. ICMP (Internet Control Message Protocol)

ICMP is used for error reporting and diagnostic purposes.

Common ICMP Tools:

1. **Ping**: Tests if a host is reachable.

- Example: `ping google.com` sends ICMP Echo Requests and waits for replies.

2. **Traceroute**: Traces the path packets take to a destination.
   - Example: `tracert google.com` shows each hop along the way.

---

4. Switch

A **switch** connects devices in a local network and forwards data only to the intended device.

How It Works:

- Builds a **MAC address table** to determine which port to send data.
- Operates at Layer 2 (Data Link) of the OSI model.

Example:

In an office network, a switch ensures only the printer receives a print request from your computer.

---

5. Modem

A **modem** converts digital signals from your computer into analog signals for transmission over telephone or cable lines (and vice versa).

Types of Modems:

1. **DSL Modem**: Uses telephone lines.
2. **Cable Modem**: Uses coaxial cables.
3. **Fiber Optic Modem**: Uses light signals.

Example:

Your internet connection relies on a modem to communicate with your ISP.

---

6. Socket/Port

A **socket** is an endpoint for communication, combining an IP address and a port number.

- **Port**: A logical access point for specific services (e.g., HTTP uses port 80).

Common Port Numbers:

- **80**: HTTP
- **443**: HTTPS
- **22**: SSH
- **25**: SMTP (email)

Example:

When you access `www.google.com`, your browser connects to Google's server using **IP:Port (e.g., 142.250.190.14:443)**.

---

7. Cookies

Cookies are small pieces of data stored on your browser by websites to track user behavior and maintain sessions.

Types of Cookies:

1. **Session Cookies**: Deleted after you close the browser.
2. **Persistent Cookies**: Stored for a longer duration.
3. **Third-Party Cookies**: Used by advertisers to track activity across websites.

Example:

When you log in to Amazon, a session cookie keeps you logged in while browsing.

---

8. Checksum

A **checksum** is a value calculated from data to ensure its integrity during transmission.

How It Works:

1. Sender calculates a checksum and sends it along with the data.
2. Receiver recalculates the checksum.
3. If the two checksums match, the data is intact; otherwise, it's corrupted.

Example:

Downloading files often involves checksum validation to ensure the file isn't corrupted.

---

9. Subnetting

Subnetting divides a network into smaller, manageable sub-networks to improve efficiency and security.

Subnet Mask:

Defines which part of an IP address is the network portion and which part is the host portion.

Example:

- IP: `192.168.1.0/24` (Subnet Mask: `255.255.255.0`).
  - Network: `192.168.1`
  - Hosts: `0-255`.

Dividing this network into 2 subnets:

- Subnet 1: `192.168.1.0/25` (128 hosts).
- Subnet 2: `192.168.1.128/25` (128 hosts).

Day 3

1. Data Center Technology

Overview of Data Centers

A **Data Center** is a facility that houses servers, storage, networking equipment, and other IT infrastructure for managing and storing data.

- **Purpose**: Provide reliable access to data, applications, and IT services.
- **Key Features**: High availability, scalability, and security.

Types of Data Centers

1. **On-Premises**:

- Owned and managed by an organization within its premises.
- **Example**: A company’s in-house server room.

2. **Colocation**:
    - Organizations rent space in a third-party data center for their servers.
    - **Benefits**: Shared power, cooling, and security.
3. **Cloud**:
    - Data is hosted and managed by cloud service providers (e.g., AWS, Azure).
    - **Benefits**: Flexible, scalable, and no need for physical infrastructure.

Data Center Infrastructure

- **Power**: Backup generators and UPS (Uninterruptible Power Supply) ensure continuous power.
- **Cooling**: Prevents overheating of servers. Methods include CRAC (Computer Room Air Conditioning) units and liquid cooling.
- **Space Management**: Efficient use of racks and physical layout to optimize performance and access.

---

2. Storage

Basics of Data Storage

Storage systems hold data and make it accessible to users or applications.

Types of Storage

1. **DAS (Direct-Attached Storage)**:
    - Storage is directly connected to a server (e.g., internal hard drives).
    - **Use Case**: Small setups or local storage.
2. **NAS (Network-Attached Storage)**:
    - Shared storage connected over a network.

- **Use Case**: File sharing in a small office.

3. **SAN (Storage Area Network)**:
   - High-speed, dedicated network for connecting storage to servers.
   - **Use Case**: Enterprise environments needing fast, scalable storage.

Introduction to RAID (Redundant Array of Independent Disks)

- Combines multiple disks for redundancy or performance.
- **RAID Levels**:
  - **RAID 0**: Striping for performance, no redundancy.
  - **RAID 1**: Mirroring for redundancy.
  - **RAID 5**: Striping with parity for performance and redundancy.

Backup and Recovery Concepts

- **Backup**: Creating a copy of data to restore in case of loss.
- **Recovery**: Restoring data from backups after an event like hardware failure.

---

3. Servers

What is a Server?

A server is a computer or system that provides resources, data, or services to other devices (clients).

Types of Servers

1. **File Servers**: Manage and store files.
2. **Web Servers**: Host websites and handle HTTP requests.
3. **Database Servers**: Store and manage databases.

Basic Server Hardware Components

- **CPU**: Handles processing tasks.
- **RAM**: Temporary data storage for quick access.

- **Storage**: Hard drives or SSDs for data storage.
- **NIC**: Network Interface Card for communication.

Introduction to Virtualization

- Virtualization allows multiple virtual servers to run on a single physical server.
- **Benefits**: Optimized resource usage, scalability, and cost-efficiency.

---

4. Firewalls

Overview of Firewalls

A firewall monitors and controls incoming and outgoing network traffic based on security rules.

Types of Firewalls

1. **Packet Filtering**: Inspects individual packets based on predefined rules.
2. **Stateful Inspection**: Tracks the state of active connections and decides based on connection context.
3. **Proxy Firewall**: Acts as an intermediary between client and server, inspecting traffic at the application layer.

Basic Firewall Configurations

- **Allow Rules**: Permit specific traffic.
- **Deny Rules**: Block certain traffic.
- **Example**: Allowing SSH traffic (port 22) while blocking other ports.

Introduction to Network Security

- Protects data and resources from unauthorized access or attacks.
- **Tools**: Firewalls, intrusion detection/prevention systems, encryption.

---

5. Load Balancing

What is Load Balancing?

Distributes incoming traffic across multiple servers to ensure no single server is overloaded.

Types of Load Balancers

1. **Hardware Load Balancers**: Dedicated devices for load balancing.
2. **Software Load Balancers**: Applications that run on general-purpose servers.

Basic Load Balancing Algorithms

1. **Round Robin**: Distributes requests sequentially to servers.
2. **Least Connections**: Sends requests to the server with the fewest active connections.

Understanding High Availability

- Ensures continuous service availability even during failures.
- Achieved through redundancy (e.g., multiple load balancers, servers).