

1.Client-Server Model

The Client-Server Model is a network architecture in which one device (the client) requests services or resources, and another device (the server) provides those services or resources.

Client - device that requests resources.

Server - device that provides resources.

2.OSI (Open Systems Interconnection)

Physical Layer (Layer 1)

Responsible for the actual physical connection between the devices Specifies the physical medium, signalling methods, and hardware interfaces.

Physical layer contains information in the form of **bits**.

Data Link Layer (Layer 2)

Responsible for node-to-node data transfer

error detection/correction.

Flow Control

Organizes data into **frames**.

Network Layer (Layer 3)

Routing of data packets between devices

The best route for delivery

Segment in the Network layer is referred to as **Packets**.

Transport Layer (Layer 4)

Ensures data is correctly ordered and reassembled.

The transport layer also provides the acknowledgment of the successful data transmission and re-transmits the data if an error is found.

TCP - Connection-Oriented Service

UDP - Connection-less Service

Session Layer (Layer 5)

Responsible for connections (Session Establishment, Maintenance, and Termination)

Dialog Controller

The session layer allows two systems to start communication with each other in half-duplex or full duplex.

Presentation Layer (layer 6)

Formatting

Encryption

process used to secure data by converting it from its original, readable form into an unreadable format using an algorithm and a key.

Application Layer (Layer 7)

This layer interacts directly with application software.

3.Encoding

Encoding is the process of converting data from one format into another, typically for the purpose of standardization, efficient transmission, or storage.

4. Encryption

Encryption is the process of converting data from its original form (plaintext) into a unreadable format (ciphertext) using an algorithm and a key. This transformation ensures that the data remains confidential and secure, protecting it from unauthorized access, even if intercepted during transmission.

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

IP (Internet Protocol)

Routes packets of data from the source to the destination based on IP addresses. Provides addressing and routing of packets across networks. IP is a **connection-less protocol**, which means it doesn't guarantee delivery nor does it provide error checking and correction.

TCP (Transmission Control Protocol)

TCP is a **connection-oriented protocol** that ensures **reliable, ordered, and error-free delivery** of data between devices. It breaks data into smaller packets, sends them over the network, and reassembles them at the destination.

6. ICMP (Internet Control Message Protocol)

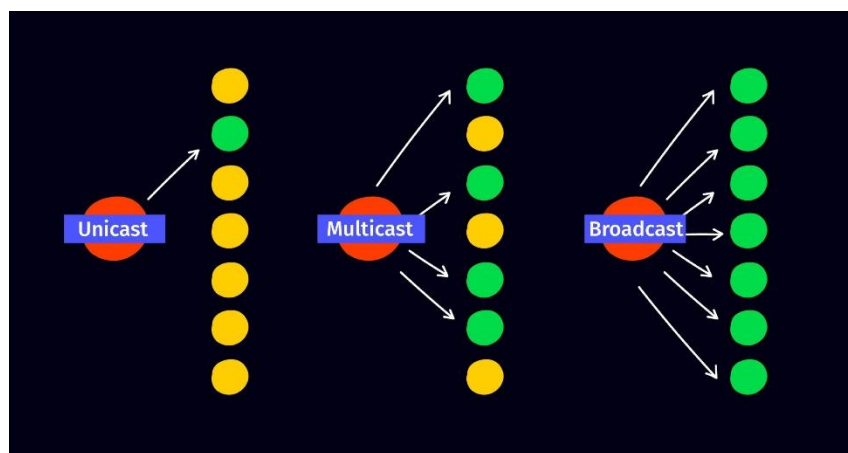
ICMP (Internet Control Message Protocol) is a **network layer protocol** used to send control messages and **error reporting in Internet Protocol (IP) networks**. It is an **essential protocol for diagnosing and troubleshooting network issues**.

7. ARP (Address Resolution Protocol)

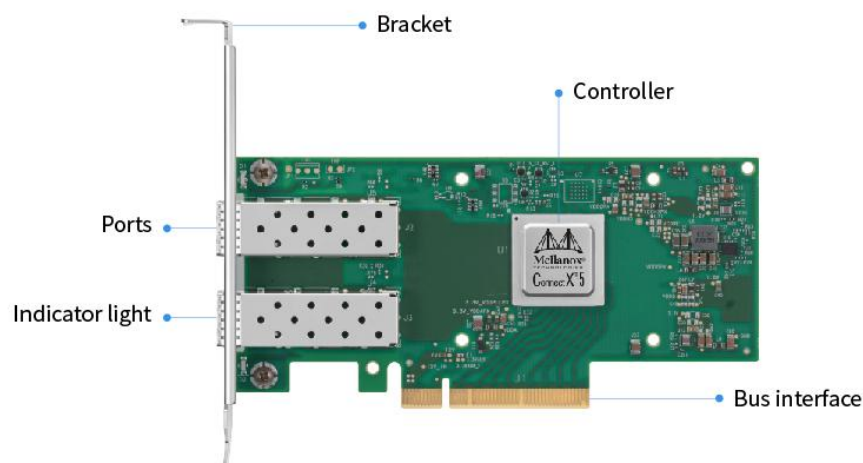
ARP (Address Resolution Protocol) is a protocol used in computer networks to map a device's **IP address** (Internet Protocol address) to its **MAC address** (Media Access Control address). This mapping is necessary because while IP addresses are used for routing data between devices on a network, the actual transmission of data between devices on the same local network (LAN) relies on MAC addresses, which are unique hardware addresses assigned to network interface cards (NICs).

8. IGMP (Internet Group Management Protocol)

IGMP (Internet Group Management Protocol) is a network-layer protocol used by hosts and adjacent routers on an **IP network** to manage the membership of **multicast groups**. Multicast allows one-to-many communication, where data is sent from a single sender to multiple receivers in an efficient manner, as opposed to broadcasting to all devices or unicasting to a single device.



9. NIC (Network Interface Card)



A NIC (Network Interface Card) is a hardware component that allows computers and other devices to connect to a network.

10.Submarine Cable

Submarine cables are specialized undersea cables that transmit data across oceans and seas, forming the backbone of the global internet infrastructure. These cables carry the majority of international communications, including internet traffic, telephone calls, and private data transfers. Submarine cables are a critical component of the global communication network, connecting continents and facilitating rapid data transmission across vast distances.

11. Fiber-optic cable

Fiber-optic cable is a high-performance cable used for transmitting data over long distances at very high speeds using light signals.

12.Nodes/Hosts/Client and servers/Protocol

Node: node is any device or point that can send, receive, or forward data

Hosts: A host is any device that has an IP address and can be a source or destination for data on a network.

Client: device that **requests** services or resources from a server.

Servers: device that **provides** resources, data, or services to clients.

Protocol: A set of rules that define how data is transmitted and handled between devices in a network.

13.LAN/MAN/WAN

LAN (Local Area Network)

A **Local Area Network (LAN)** is a network that spans a small geographic area, such as a **home, office, or building**. It is typically used to connect devices like computers, printers, servers, and other networked devices within a limited physical area.

MAN (Metropolitan Area Network)

A **Metropolitan Area Network (MAN)** is a larger network that spans a **city** or a **large campus** and is typically used to connect multiple LANs within a metropolitan area.

WAN (Wide Area Network)

A **Wide Area Network (WAN)** is a network that covers a **large geographic area**, often spanning multiple cities, regions, or even countries. WANs are used to connect multiple LANs and MANs, allowing devices from different locations to communicate with each other. The **internet** itself is the largest example of a WAN.

14. Modem/Router

Modem

A **modem** (short for **Modulator-Demodulator**) is a device that **connects your home or office network to the internet** through your **Internet Service Provider (ISP)**. It converts digital signals from your computer or network into analog signals.

Router

A **router** is a device that **routes data** between your modem and all the devices on your local network. Routers often include security features like firewalls to protect your internal network from outside threats.

15. Scalability/Elasticity

Scalability

potential to accommodate growth and increased demand without compromising performance or efficiency.

vertical - scale up/down [CPU/Resources]

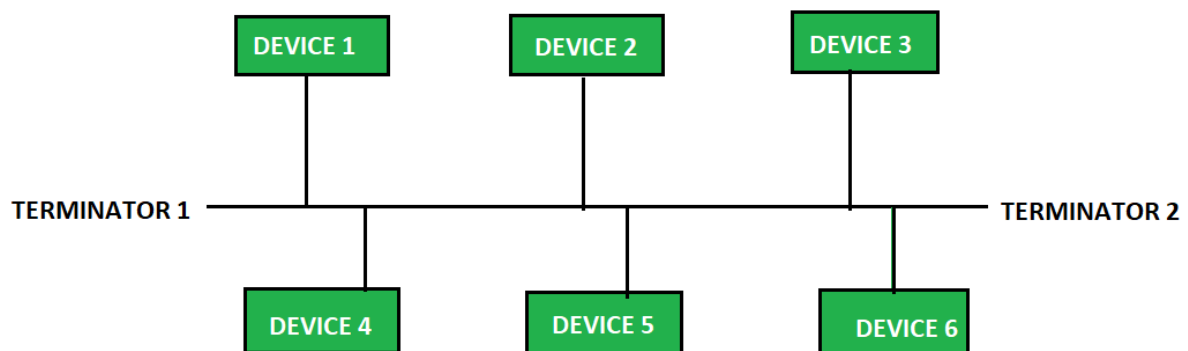
horizontal - scale in/out [adding/removing more machines]

Elasticity

Elasticity refers to a system's **ability to dynamically allocate and deallocate resources** as demand increases or decreases.

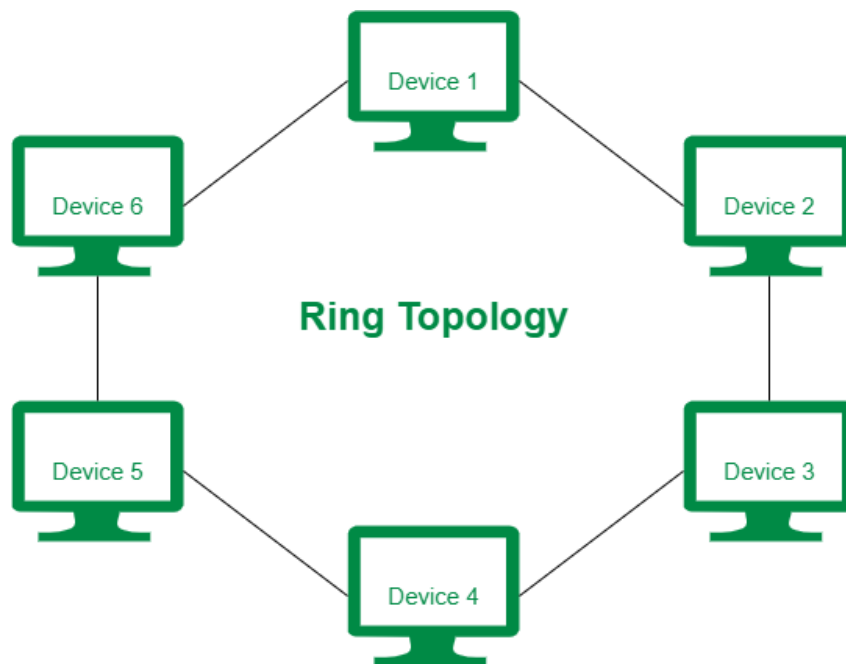
16. Network topology

Bus Topology



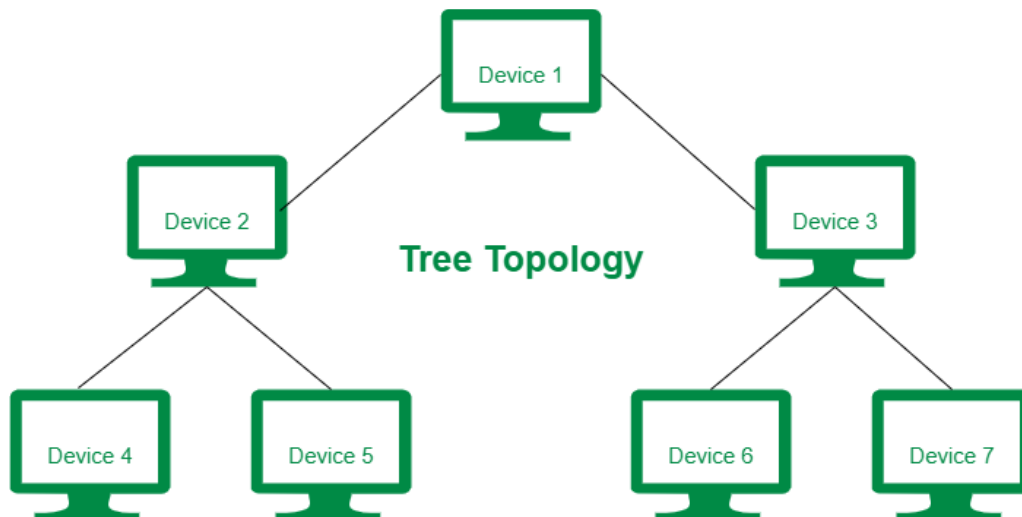
Bus topology is one of the simplest network topologies, where all devices are connected to a single central cable (called the bus or backbone). Data sent by a device is available to all other devices on the network, but only the intended recipient processes the data.

Ring Topology



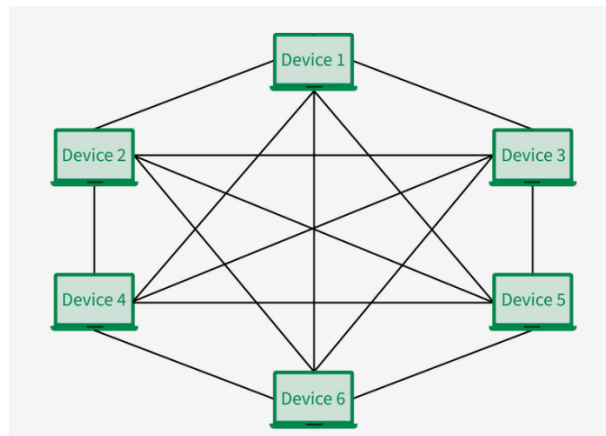
In ring topology, devices are connected in a closed loop or ring, where each device is connected to two other devices. Data travels in a circular path, passing through each device until it reaches the destination.

Tree Topology



tree topology is a type of network topology that resembles a tree. In a tree topology, there is one central node (the “trunk”), and each node is connected to the central node through a single path. Nodes can be thought of as branches coming off the trunk. Tree topologies are often used to create large networks.

Mesh Topology



Mesh topology involves each device being directly connected to every other device in the network, forming a fully interconnected network.

17. Peer-to-Peer (P2P)

Peer-to-Peer (P2P) is a type of network architecture where each device (or "peer") on the network has equal status and can act both as a **client** and a **server**. In a P2P network, devices communicate directly with each other without the need for a central server or authority to manage the interactions. Each device can share resources (like files, processing power, or storage) with other devices on the network.

18.Socket/Port

Socket: A Socket is a combination of port and IP address.

Port: A port is an access point where protocols are used in a network to locate the processes or services within a device

19. HTTP (Hypertext Transfer Protocol) Methods

GET: The **GET** method is used to request data from a specified resource.

POST: creating a resource. POST requests typically include a **request body**, which contains the data to be submitted.

PUT: The **PUT** method is used to **update** a resource on the server.

DELETE: The **DELETE** method is used to **delete** a specified resource from the server.

20. HTTP Response Code

Informational response: 100 – 199

Success response: 200 – 299

Redirection response: 300 -399

Client Errors response: 400 – 499

Server Errors response: 500 -599

21. Cookies

cookie is a small piece of data that a server sends to a client (typically a web browser) in the form of key-value pairs. The client stores these cookies locally, and they are sent back to the server with every subsequent request to the same domain. Cookies help maintain stateful information for HTTP, which is inherently stateless.

- **Session Cookies:** Temporary cookies that are deleted once the browser is closed.
- **Persistent Cookies:** Stored on the device for a set period or until deleted.

22. VPN (Virtual Private Network)

A **Virtual Private Network (VPN)** is a technology that enables a secure and private connection over the internet, providing users with a way to transmit data safely, even over unsecured networks like public Wi-Fi. VPNs encrypt the user's internet traffic and create a private tunnel through which data is transmitted, effectively masking the user's real IP address and location.

Tunnelling is a technique used in computer networks to encapsulate data in a way that allows it to be securely transmitted across a public or untrusted network.

Types of VPNs

1.Remote Access VPN (Tunnelling)

Remote Access VPN allows individual users to connect securely to a remote network (usually a corporate network) from anywhere in the world.

2. Site-to-Site VPN

A **Site-to-Site VPN** connects two or more separate private networks, typically located in different geographic locations, through a public network.

3. Cloud VPN

A **Cloud VPN** secures the connection between the client and a cloud infrastructure, allowing users to access cloud-hosted resources securely over the internet.

4. Mobile VPN

A **Mobile VPN** is designed to support mobile devices (smartphones, tablets, laptops) that may switch between different networks.

5. SSL VPN (Secure Sockets Layer VPN)

An **SSL VPN** uses SSL or its successor TLS (Transport Layer Security) protocols to provide a secure connection between the client and the server. It is commonly used for secure web-based access, allowing users to connect to a private network via a web browser without needing specialized VPN client software.

6. Double VPN

A **Double VPN** is a security feature that routes the internet traffic of the user through two VPN servers instead of just one.

23. Checksum

A **checksum** is a value calculated from a data set (such as a file, message, or packet of data) and used to verify the integrity of that data. It's essentially a form of redundancy check to detect errors that might have been introduced during the data transmission or storage process.

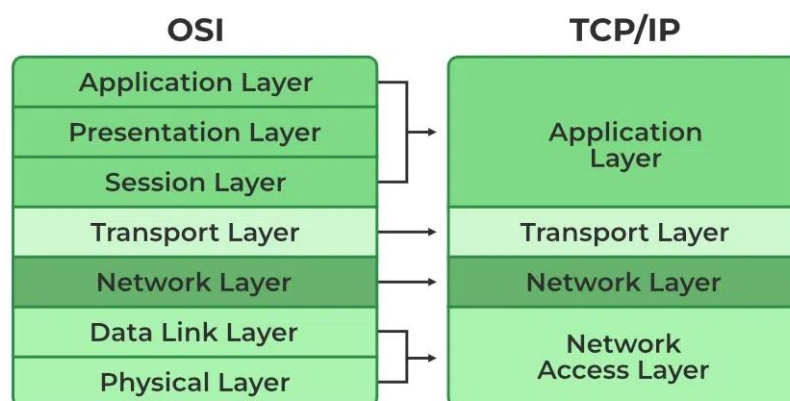
24. Ping

Ping is a network diagnostic tool used to test the connectivity between two devices on a network, typically between a computer and a remote server or another device on the internet or local network.

It works by sending a special type of message called an **ICMP Echo Request** to a target device and waiting for an **ICMP Echo Reply**.

The time it takes for the message to travel to the destination and back is measured, which is known as **round-trip time**.

25. TCP/IP



Application Layer (Layer 4 in TCP/IP)

This layer deals with **application-level protocols** and provides network services directly to end-users or applications. It enables communication between software applications running on different systems.

Transport Layer (Layer 3 in TCP/IP)

Responsible for **reliable data transfer** between two devices on a network, ensuring **error-free and complete data delivery**. This layer defines how data is packaged, transmitted, and verified between devices.

Internet Layer (Layer 2 in TCP/IP)

This layer is responsible for **routing packets** of data across different networks and ensures that data reaches its correct destination by addressing and routing it. The most important component of this layer is the **IP (Internet Protocol)**.

Network Access Layer (Layer 1 in TCP/IP)

This layer is responsible for the **physical transmission of data** over the network hardware (such as Ethernet, Wi-Fi, etc.). It deals with the **hardware addressing, data link, and physical transmission of data between devices**.

26. Subnetting

Subnetting is the process of dividing a larger network into smaller, more manageable sub-networks, known as **subnets**. This is done by borrowing bits from the host portion of an IP address to create additional network bits. Subnetting helps improve network performance, security, and address management.

27. NAT (Network Address Translation)

Network Address Translation (NAT) is a technique used in networking to modify the source or destination IP addresses in the header of IP packets as they pass through a router or firewall. NAT enables multiple devices on a local network to **share a single public IP address** for accessing external networks like the internet.

28.SSL/TSL/HTTPS

SSL: SSL is a cryptographic protocol designed to provide secure communication over a computer network. SSL encrypts the data transmitted between the client and server.

TSL: TLS is the successor to SSL, designed to improve upon SSL's weaknesses and provide stronger security for internet communications.

HTTPS: HTTPS is a secure version of the **HTTP** protocol, which is used for transferring data over the web.

29. Symmetric and Asymmetric Encryption

Symmetric: In **symmetric encryption**, the **same key** is used for both **encryption** and **decryption**.

Asymmetric: In **asymmetric encryption**, two **different keys** are used: a **public key** and a **private key**. The public key is used to encrypt data, while the private key is used to decrypt it.

30. Monolithic Architecture

Monolithic architecture refers to a traditional model of software design where an entire application is built as a single, unified unit.

31. Microservices Architecture

The key idea behind microservices is to break down a monolithic application into smaller, modular pieces, each responsible for a specific aspect of the business logic or functionality.

//**Authentication** is the process of verifying the identity of a user, system, or entity attempting to access a resource. It answers the question: *"Who are you?"*

// **Authorization** is the process of granting or denying access to a specific resource or action based on the identity that was authenticated. It answers the question: *"What can you do?"*

//10-01-2025

32.Firewall

A Firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies. At its most basic, a firewall is essentially the barrier that sits between a private internal network and the public Internet. A firewall's main purpose is to allow non-threatening traffic in and to keep dangerous traffic out.

Stateful Firewall: A stateful firewall maintains a table of the state of each active connection, so it can track and validate whether a response is from an established connection. This helps ensure that only legitimate return traffic is allowed.

Stateless Firewall: A stateless firewall does not keep track of connection states. It filters packets based solely on pre-configured rules (such as source/destination IP address or port).

33. API Gateway

An **API Gateway** is a server or service that acts as an intermediary between a client (such as a web or mobile application) and multiple backend services (often microservices). It provides a unified entry point for accessing different APIs, making it easier for clients to interact with multiple services without needing to know the specifics of each one.

34. Personal Access Token (PAT)

PATs are used as an alternative to traditional username/password authentication, and they provide a more secure and flexible way of authenticating users or applications.

35.Server Farm

A **server farm** is a large collection of servers that work together to provide a high level of computing power and storage capacity for various applications, websites, and services. These servers are usually housed in a **data center** and are configured to work as a unified system, often using load balancing, redundancy, and virtualization technologies to ensure scalability, high availability, and reliability.

36. IPSec (Internet Protocol Security)

IPSec (Internet Protocol Security) is a suite of protocols used to secure IP communications by authenticating and encrypting each IP packet in a communication session. It is commonly used to implement **Virtual Private Networks (VPNs)** and to protect data traffic across untrusted networks, such as the internet.

IPSec operates at the **Network Layer** (Layer 3) of the OSI model and can secure communication between devices such as routers, firewalls, and gateways, as well as between hosts (e.g., computers, servers). Its primary purpose is to ensure data confidentiality, integrity, and authenticity.

37. Threat, Vulnerability, and Risk in Cybersecurity

Threat: A **threat** refers to any potential danger or event that could exploit a vulnerability and cause harm to a system, network, or data.

Vulnerability: A **vulnerability** is a weakness or flaw in a system, network, application, or process that can be exploited by a threat actor to gain unauthorized access or cause harm.

Risk: **Risk** is the potential impact of a threat exploiting a vulnerability, expressed in terms of likelihood and consequences.

38. Reverse Proxy

A **reverse proxy** is a server that sits between client devices and a backend server.

39. IPV4/IPV6

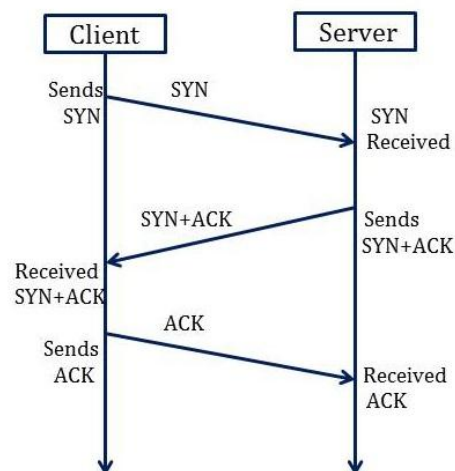
IPv4

- IPv4 is a 32-bit address.
- IPv4 is a numeric address that consists of 4 fields which are separated by dot (.).
- IPv4 has 5 different classes of IP address that includes Class A, Class B, Class C, Class D, and Class E.
- It supports manual and DHCP configuration.

IPv6

- IPv6 is a 128-bit address.
- IPv6 is an alphanumeric address that consists of 8 fields, which are separated by colon.
- IPv6 does not contain classes of IP addresses.
- It supports manual, DHCP, auto-configuration, and renumbering.

40. 3-Way Handshake



- SYN (client to server)
- SYN-ACK (server to client)
- ACK (client to server)

41. DHCP (Dynamic Host Configuration Protocol)

DHCP (Dynamic Host Configuration Protocol) is a network management protocol used to dynamically assign IP addresses and other network configuration information to devices (known as **clients**) on a network.

42. Switch and Router

Switch

A **switch** is a network device that connects multiple devices (like computers, printers, and servers) within a local area network (**LAN**) and uses **MAC addresses** to forward data packets between devices.

Router

Definition:

A **router** is a device that connects multiple networks together, typically a local area network (**LAN**) and the internet, and forwards data packets between them using **IP addresses**.

43.Port forwarding

Port forwarding is a network configuration technique used to allow external devices or services to access specific services or devices within a private internal network. It works by redirecting communication requests from an external IP address (often the public IP address of a router) to a specific device or service on the internal network, using a particular port number.

44.Hub and switch

Hub:

A **hub** is a basic networking device that connects multiple devices in a network, enabling them to communicate with each other. It operates at **Layer 1** (Physical Layer) of the OSI model and is often referred to as a "**network hub**".

Switch:

A **switch** is a more advanced network device that connects devices within a local network and uses **MAC addresses** to forward data only to the intended recipient. It operates at **Layer 2** (Data Link Layer) of the OSI model.

45. VLAN (Virtual Local Area Network)

A **VLAN** (Virtual Local Area Network) is a logical grouping of devices within a physical network, allowing them to communicate as if they are on the same local network, regardless of their actual physical location. VLANs enable network administrators to segment networks into smaller, more manageable parts, improving security, performance, and organization.

//11-01-2025

46. Data Centers

In a dedicated space with strong security levels, where enterprises or organizations store and share large amounts of data, is known as a data center.

Key components

- **Servers** (compute power)

Servers are the backbone of any data center, responsible for running applications, managing data, and providing services to clients or users. These can include web servers, database servers, application servers, and storage servers.

- **Storage systems** (data management)

Data storage systems hold and manage large volumes of data within a data center. This can include both primary data (active data) and backup or archival data.

- **Networking equipment** (communication)

Networking equipment connects all the components of a data center, ensuring communication between servers, storage systems, and external networks. (Router, Switch, Firewall)

- **Power supply and backup systems** (reliable operation)

Ensures that the data center operates continuously without interruption due to power failures.

- **Cooling systems** (temperature regulation)

Servers and other IT equipment generate a lot of heat, so cooling systems are necessary to maintain an optimal operating temperature.

- **Security systems** (physical and cybersecurity)

Protects the physical and logical infrastructure of the data center from unauthorized access, theft, and cyber threats.

Types of Data centers

On premise Data Center

- The organization owns the data center and is responsible for managing, maintaining, and upgrading the infrastructure.
- The company has complete control over how it is configured and customized to meet specific requirements.
- The data center is located within the organization's own premises.
- High initial investment.

Colocation Data Center

- A third-party data center where businesses can rent space, power, cooling, and network connectivity for their IT infrastructure.
- Lower investment
- Shared infrastructure

Cloud Data Center

- Data centers operated by cloud service providers (e.g., AWS, Google Cloud, Microsoft Azure) that host and manage virtualized resources and services.
- Pay-as-you-go model.
- High scalability

47.Storage Types

Direct Attached Storage (DAS):

DAS refers to storage devices that are directly attached to a single computer or server, without being connected to a network. It is typically used for personal or small-scale applications.

How it works: DAS storage is directly connected via interfaces such as USB, SATA, SAS, or Thunderbolt.

Local storage directly attached to servers.

Lack of scalability

Potential to lost data.

Device specific

Network Attached Storage (NAS):

NAS is a storage device connected to a network, allowing multiple users and devices to access data over the network. It's often used for centralized file storage and sharing.

How it works: NAS devices typically use Ethernet or Wi-Fi to connect to a local area network (LAN), and they present storage over protocols like SMB/CIFS (Windows), NFS (Linux/Unix), or AFP (Apple).

- File-based storage accessible over a network.
- Moderate scalability and performance
- Examples: Synology NAS, QNAP NAS, WD My Cloud.

Storage Area Networks (SAN):

SAN is a high-speed network that connects storage devices (such as disk arrays) with servers, enabling block-level data access. Unlike NAS, which provides file-level access, SAN provides block-level access to data, typically used for large-scale enterprise applications.

How it works: SANs often use Fibre Channel, iSCSI, or FCoE (Fibre Channel over Ethernet) to connect storage devices and servers. Data is accessed as blocks rather than files.

- High-speed network of storage devices, often used for large-scale enterprise data storage.
- Examples: EMC VMAX, NetApp FAS, Dell PowerMax

48. TYPES OF STORAGES

PRIMARY STORAGE

Primary storage refers to the storage that is directly accessible by the CPU and is used to store data and instructions that are actively being processed.

RAM (Random access memory)

Description: RAM is the most common type of primary storage and is used to store data and instructions that the CPU needs to access quickly while performing tasks.

Characteristics:

- Fast read and write access.
- Volatile memory (data is lost when power is turned off).
- Temporarily stores data being processed by running applications.

ROM(Read only memory)

ROM is a type of non-volatile memory used in computers and other electronic devices to store permanent data or instructions that are not meant to be altered or modified during normal operation.

Characteristics:

- Non volatile
- Read only.
- Slower access

SECONDARY STORAGE

1. **HDD (Hard Disk Drives)**
2. **SSD (Solid-State Drives)**

OPTICAL DISC

An **optical disc** is a storage medium that uses laser light to read and write data on a reflective surface. Optical discs are widely used for storing data such as software, music, videos, and backups.

49.RAID LEVELS

RAID 0 (Striping)

- **Configuration:** Data is split into blocks and distributed across multiple disks (at least 2).
- **Redundancy:** No redundancy—if one drive fails, all data is lost.
- **Performance:** High performance, as data is read and written in parallel to multiple drives.
- **Capacity:** Total capacity is the sum of the capacities of all disks.
- **Use Case:** Suitable for applications requiring high performance and where data loss is not critical (e.g., temporary data, non-essential files).

RAID 1 (Mirroring)

- **Configuration:** Data is duplicated (mirrored) across two or more disks.
- **Redundancy:** High redundancy—if one drive fails, the data is still available from the other drive(s).
- **Performance:** Good read performance (because the system can read from multiple disks), but write performance is similar to a single disk.
- **Capacity:** Total capacity is the size of one drive (since data is duplicated).
- **Use Case:** Suitable for situations where data integrity is critical and write performance is not as important (e.g., personal computers, critical data storage).

RAID 5 (Striping with Parity)

- **Configuration:** Data is striped across multiple disks (at least 3), with parity information distributed across all disks.
- **Redundancy:** Moderate redundancy—if one disk fails, the data can be rebuilt using the parity data from the remaining disks.
- **Performance:** Good read performance, but write performance is slower compared to RAID 0 and RAID 1 due to the overhead of parity calculations.
- **Capacity:** Total capacity is the sum of all disks minus one (because one disk is used for parity).
- **Use Case:** Suitable for applications that require a balance of redundancy, performance, and storage capacity (e.g., file servers, databases).

RAID 6 (Striping with Double Parity)

- **Configuration:** Similar to RAID 5 but with **two sets of parity data**, which are stored across different disks (requires at least 4 disks).
- **Redundancy:** High redundancy—can tolerate the failure of **two disks** simultaneously without data loss.

- **Performance:** Read performance is good, but write performance is slower than RAID 5 because of double parity calculations.
- **Capacity:** Total capacity is the sum of all disks minus two (because two disks are used for parity).
- **Use Case:** Suitable for environments where data protection is more important than write performance (e.g., critical business data storage).

RAID 10 (RAID 1+0)

- **Configuration:** Combines the features of RAID 1 and RAID 0. Data is mirrored (RAID 1) and then striped (RAID 0).
- **Redundancy:** High redundancy—can tolerate the failure of one disk per mirrored pair.
- **Performance:** High performance for both read and write operations, as data is striped (RAID 0) and mirrored (RAID 1).
- **Capacity:** Total capacity is the sum of half of the disks (since data is mirrored).
- **Use Case:** Suitable for applications that require both high performance and redundancy (e.g., databases, high-performance servers).

50.BACKUP AND RECOVERY

A **backup** is the process of creating a duplicate copy of data that can be restored in case the original data is lost, corrupted, or inaccessible.

TYPES OF BACKUPS

Full Backup

- **Definition:** A full backup is a complete copy of all selected data. It copies everything, including all files, folders, and system data (depending on the configuration).
- **How It Works:** Every time a full backup is performed, all data is backed up in its entirety, regardless of whether it has changed since the last backup.

Incremental Backup

- **Definition:** An incremental backup only backs up the data that has changed since the **last backup** (whether it was a full back up or the most recent incremental backup).
- **How It Works:** After an initial full backup, subsequent incremental backups only capture changes made to files since the last backup. This can be multiple times over a period.

Differential Backup

- **Definition:** A differential backup captures all the changes made since the last **full backup**. Unlike incremental backups, differential backups do not rely on previous differential backups, but only on the full backup.

Mirror Backup

- **Definition:** A mirror backup creates an exact copy (or "mirror") of the selected data. It is like a full backup but continuously synchronizes data between the source and the backup location.

51.3-2-1 BACKUP STRATEGY

The **3-2-1 backup strategy** is a widely recommended method for ensuring robust data protection and recovery. It helps mitigate the risks of data loss from various types of disasters (e.g., hardware failure, cyberattacks, accidental deletions). This strategy involves creating multiple copies of data and storing them in different locations to increase redundancy and resilience. 2 Copies stores in two different media types and one in offsite.

52.BASIC SERVER COMPONENTS

- **MOTHER BOARD:** A **motherboard** is the central printed circuit board (PCB) in a computer that connects and allows communication between various hardware components. It serves as the backbone of the computer, providing essential connections for components like the **CPU, RAM**
- **CPU**
- **RAM**
- **NIC**
- **STORAGE DRIVE**

53.LOAD BALANCING

- **ROUND ROBIN:** **Round Robin** is one of the simplest and most used load balancing algorithms. It is a method used to distribute client requests (or traffic) across a group of servers or resources in a circular order.
- **LEAST CONNECTION:** **Least Connections** is a dynamic load balancing algorithm that directs incoming traffic to the server with the **fewest active connections** at the time of the request. This method is designed to distribute load based on the number of active connections each server is currently handling, aiming to prevent overloading any single server.
- **LEAST RESPONSE TIME:** **Least Response Time** is a dynamic load balancing algorithm that routes incoming client requests to the server with the **quickest response time** now of the request. The goal of this algorithm is to optimize user experience by sending traffic to the server that is not only least loaded but also currently capable of processing requests the fastest.
- **SOURCE IP HASHING:** **Source IP Hashing** is a load balancing algorithm that uses the **client's IP address** to determine which server in the pool should handle a particular request. The key idea behind this approach is to ensure that requests from the same client IP address are always directed to the same backend server, creating session persistence (also called sticky sessions).
- **WEIGHTED ROUND ROBIN:** **Weighted Round Robin (WRR)** is an enhancement of the traditional **Round Robin** load balancing algorithm. In **Weighted Round Robin**, each server in the pool is assigned a **weight** that reflects its capacity or performance. The load balancer distributes incoming requests across the servers, but it gives more requests to servers with higher weights, effectively allowing more powerful servers to handle more traffic.

TYPES OF LOAD BALANCER

HARDWARE LOAD BALANCER

Hardware load balancer is a physical appliance designed specifically for load balancing tasks. It is a dedicated device with specialized hardware and software to handle traffic distribution efficiently.

SOFTWARE LOAD BALANCER

A software load balancer is a software application that runs on general-purpose hardware (such as a server or virtual machine) to perform load balancing tasks. It uses algorithms and protocols to distribute traffic among multiple servers.

54.Types of Firewalls:

Packet Filtering Firewall:

A packet filtering firewall works by inspecting packets (chunks of data) passing through a network. It evaluates these packets based on predefined rules such as source/destination IP address, port number, and protocol. If a packet matches the set rules, it is allowed to pass through; otherwise, it is discarded.

Stateful Inspection Firewall:

Unlike packet filtering, stateful inspection firewalls keep track of the state of active connections. They evaluate packets not only based on header information but also the context of the entire session, ensuring that packets are part of an established connection.

Application-Level Gateway (Proxy Firewall):

An application-level gateway, or proxy firewall, acts as an intermediary between a user's device and the internet. It operates at the application layer and can filter traffic based on the type of application. For example, it might filter web traffic by checking HTTP requests.

Next-Generation Firewall (NGFW):

NGFWs combine traditional firewall capabilities with additional features like deep packet inspection, intrusion prevention systems (IPS), and application awareness. They are designed to detect and block modern threats, including advanced persistent threats (APTs) and malware.

55.Common Network Security Threats:

Malware

Malware refers to malicious software that is designed to damage, disrupt, or gain unauthorized access to systems. Common types include viruses, worms, ransomware, and spyware.

Phishing Attack:

Phishing is a social engineering attack where attackers impersonate legitimate entities, such as banks or websites, to deceive individuals into disclosing sensitive information like passwords or credit card details.

Denial of Service (DoS) Attack:

A DoS attack aims to overwhelm a system, network, or service by flooding it with traffic, making it unavailable to legitimate users.

Man-in-the-Middle (MitM) Attack:

In a MitM attack, an attacker intercepts and potentially alters the communication between two parties without their knowledge. This can lead to data theft, impersonation, or information modification.

55. Git and GitHub

Types of Repositories:

1. Central Repository:

- A **central repository** is a single repository that is shared by all team members. It acts as the main hub where everyone pushes their changes and pulls updates. This type of setup is common in centralized version control systems, but in Git, it's still possible to have a central repo that acts as a common location for collaboration.
- **Example:** GitHub repositories, GitLab repositories.

2. Local Repository:

- A **local repository** is an individual's copy of the project stored on their local machine. Developers work locally, making changes and committing them without affecting the central repository until they are ready to push their changes. This is the default nature of Git's distributed model.
- **Example:** A developer's local clone of a GitHub repository.

3. Distributed Repository:

- Git is a **distributed version control system**, meaning each developer has a **complete copy of the repository** with its entire history. There is no central server required for version control, as each local repository is fully functional on its own. Developers can push and pull from others' repositories to sync changes.
- **Example:** A developer cloning a repository from GitHub, working offline, and later syncing their changes with the central repository.

Git Features:

1. Distributed:

- Git is **distributed**, meaning each developer has their own local copy of the entire repository, including its history. This allows developers to work offline and commit changes independently before syncing with others.

2. Compatible:

- Git is **compatible** with many other systems and services. It can integrate with various tools, IDEs, and CI/CD platforms, making it flexible for different workflows.

3. Non-Linear:

- Git supports **non-linear development** by allowing multiple branches to be created simultaneously. Developers can work on different features or fixes in parallel without affecting each other's work, and later merge changes back into the main branch.

4. **Branching:**

- Git makes branching extremely easy and lightweight. Developers can create **branches** to experiment with new features, fix bugs, or collaborate on isolated tasks. This enables parallel development with minimal risk of conflicts.

5. **Lightweight:**

- Git is **lightweight** in terms of both resource consumption and operations. Branches, for example, are not expensive to create or switch between, as they are simply pointers to commits, making operations like merging or branching much faster.

6. **Speed:**

- Git is known for its **speed**. Operations like commit, checkout, merge, and diff are highly optimized, even for large repositories, making Git an efficient choice for developers.

7. **Open Source:**

- Git is **open source**, meaning it is free to use and its source code is available for anyone to contribute to or modify. This has led to widespread adoption and community-driven improvements.

8. **Reliable:**

- Git is **reliable** in terms of data integrity. It uses checksums (SHA-1 hashes) to ensure that data is never lost or corrupted. Every commit has a unique hash, providing a safeguard against potential errors.

9. **Secure:**

- Git has robust **security features**, such as cryptographic hashing, which ensures the integrity of the repository and the commits. Additionally, access control can be implemented with SSH keys or personal access tokens when interacting with platforms like GitHub.

10. **Economical:**

- Git is **economical** in terms of storage. It uses efficient methods to store data, and because it's distributed, developers don't have to worry about continuously communicating with a central server to commit or retrieve changes. Additionally, GitHub and other platforms offer free repositories, especially for public projects.

Getting & Creating Projects

Command	Description
<code>git init</code>	Initialize a local Git repository
<code>git clone ssh://git@github.com/[username]/[repository- name].git</code>	Create a local copy of a remote repository

Basic Snapshotting

Command	Description
<code>git status</code>	Check status
<code>git add [file-name.txt]</code>	Add a file to the staging area
<code>git add -A</code>	Add all new and changed files to the staging area
<code>git commit -m "[commit message]"</code>	Commit changes
<code>git rm -r [file-name.txt]</code>	Remove a file (or folder)
<code>git remote -v</code>	View the remote repository of the currently working file or directory

Branching & Merging

Command	Description
<code>git branch</code>	List branches (the asterisk denotes the current branch)
<code>git branch -a</code>	List all branches (local and remote)
<code>git branch [branch name]</code>	Create a new branch
<code>git branch -d [branch name]</code>	Delete a branch
<code>git push origin --delete [branch name]</code>	Delete a remote branch
<code>git checkout -b [branch name]</code>	Create a new branch and switch to it
<code>git checkout -b [branch name] origin/[branch name]</code>	Clone a remote branch and switch to it
<code>git branch -m [old branch name] [new branch name]</code>	Rename a local branch
<code>git checkout [branch name]</code>	Switch to a branch

Command	Description
<code>git checkout -</code>	Switch to the branch last checked out
<code>git checkout -- [file-name.txt]</code>	Discard changes to a file
<code>git merge [branch name]</code>	Merge a branch into the active branch
<code>git merge [source branch] [target branch]</code>	Merge a branch into a target branch
<code>git stash</code>	Stash changes in a dirty working directory
<code>git stash clear</code>	Remove all stashed entries
<code>git stash pop</code>	Apply latest stash to working directory

Sharing & Updating Projects

Command	Description
<code>git push origin [branch name]</code>	Push a branch to your remote repository
<code>git push -u origin [branch name]</code>	Push changes to remote repository (and remember the branch)
<code>git push</code>	Push changes to remote repository (remembered branch)
<code>git push origin --delete [branch name]</code>	Delete a remote branch
<code>git pull</code>	Update local repository to the newest commit
<code>git pull origin [branch name]</code>	Pull changes from remote repository

Command	Description
<code>git remote add origin ssh://git@github.com/[username]/[repository- name].git</code>	Add a remote repository
<code>git remote set-url origin ssh://git@github.com/[username]/[repository- name].git</code>	Set a repository's origin branch to SSH

Inspection & Comparison

Command	Description
<code>git log</code>	View changes
<code>git log --summary</code>	View changes (detailed)
<code>git log --oneline</code>	View changes (briefly)
<code>git diff [source branch] [target branch]</code>	Preview changes before merging