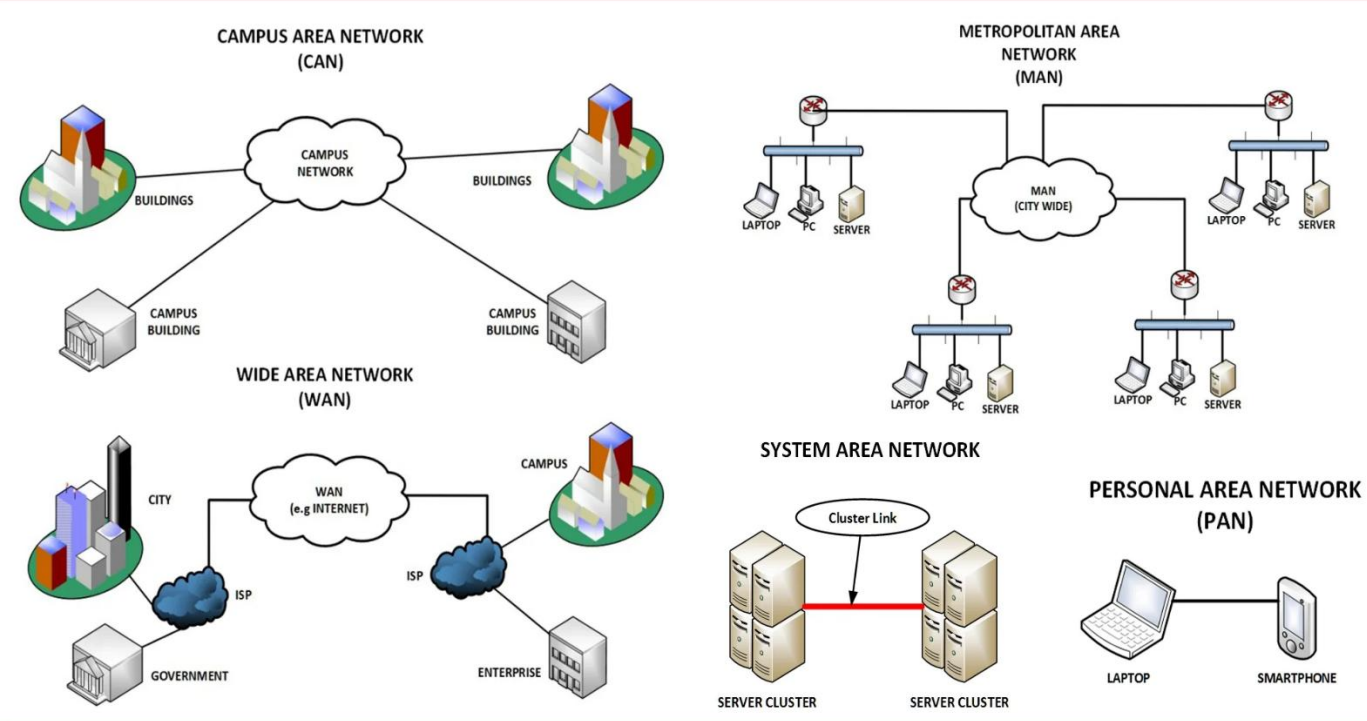
What is a Network?

- Networking is the process of connecting multiple computers and devices together to share resources like files, internet access, and printers, etc and communicate through protocols.
- It is composed of two main aspects
  - Physical connection for ex: wires,cables,wireless media,etc
  - Logical connection

There Are Some Basic Networking Rules

- Data must be delivered uncorrupted. If it is corrupted, it's useless.
- Computers in a network must be capable of determining the origin and destination of a piece of information, i.e., its IP and Mac Address.

Types of Computer Networks



1. **Personal Area Network (PAN)** : Ultra-small networks used for personal use to share data from one device to another. Ex: (smart phone to laptop, smart watch to smart phone,etc)

2. **Local Area Network (LAN)** : A computer network within a small geographical area such as a single room, building or group of buildings. Ex: (Home Network, Small Business or Office Network).
3. **Wireless Local Area Network (WLAN)** : A LAN that's dependent on wireless connectivity or one that extends a traditional wired LAN to a wireless LAN. Most home networks are WLAN's
4. **Campus Area Network (CAN):** A computer network of multiple interconnected LAN's in a limited geographical area, such as a corporate business park, government agency, or university campus. Typically owned or used by a single entity.
5. **Metropolitan Area Network (MAN) :** A computer network that interconnects users with computer resources in a city and it's larger than a campus area network but smaller than a wide area network.
6. **Wide Area Network (WAN) :** A computer network that extends over a large geographical distance, typically multiple cities and countries and typically use leased telecommunications lines from **Internet Service Providers** (ISP). Ex: (The internet and corporate offices in different states).

Network Topologies

1 . Star topology : All devices are connected to a central hub or switch communication passes through this central device

Advantages :

- Easy to install and manage.
- Failure of one cable doesn't affect others.
- Easy to detect faults.

Disadvantages:

- Central hub Failure = entire network down.
- Require more cables means higher cost.

2 . Mesh Topology : Every device connects to every other device directly or partially.

Advantages :

- Highly reliable (no single point of failure).
- Data can take multiple paths.
- Excellent performance and privacy.

Disadvantages :

- Very expensive .
- Difficult to set up and manage.

3 . Ring Topology : Each device connects to two devices forming a ring. Data travels in one or both directions.

Advantages :

- Easy to install and expand
- Equal access for all devices (no collision).

Disadvantages :

- Failure of one device breaks tha ring.
- Difficult to troubleshoot.

4 . Tree Topology : A combination of star and bus topologies, forming a hierachicak structure.

Advantages :

- Easy to expand.
- Supports large networks.
- Fault isolation is easier.

Disadvantages :

- Central hub failure affects entire branch.
- More cable and maintenance cost.

5 . Fully Connected Topology : Every node has a direct link to every other node.

Advantages :

- Maximum reliability.

- No network congestion.
- High security-each connection is private.

Disadvantages :

- Extremely high cost and complexity.
- Not scalable for large networks.

6 . Bus Topology : All devices share a single central cable(bus).

Advantages :

- Cheap and simple setup
- Require less cable.

Disadvantages :

- Entire network fails if backbone breaks.
- Difficult to troubleshoot.
- Performance drops with more devices.

7 . Line Topology : devices are connected in a linear sequence – one after another.

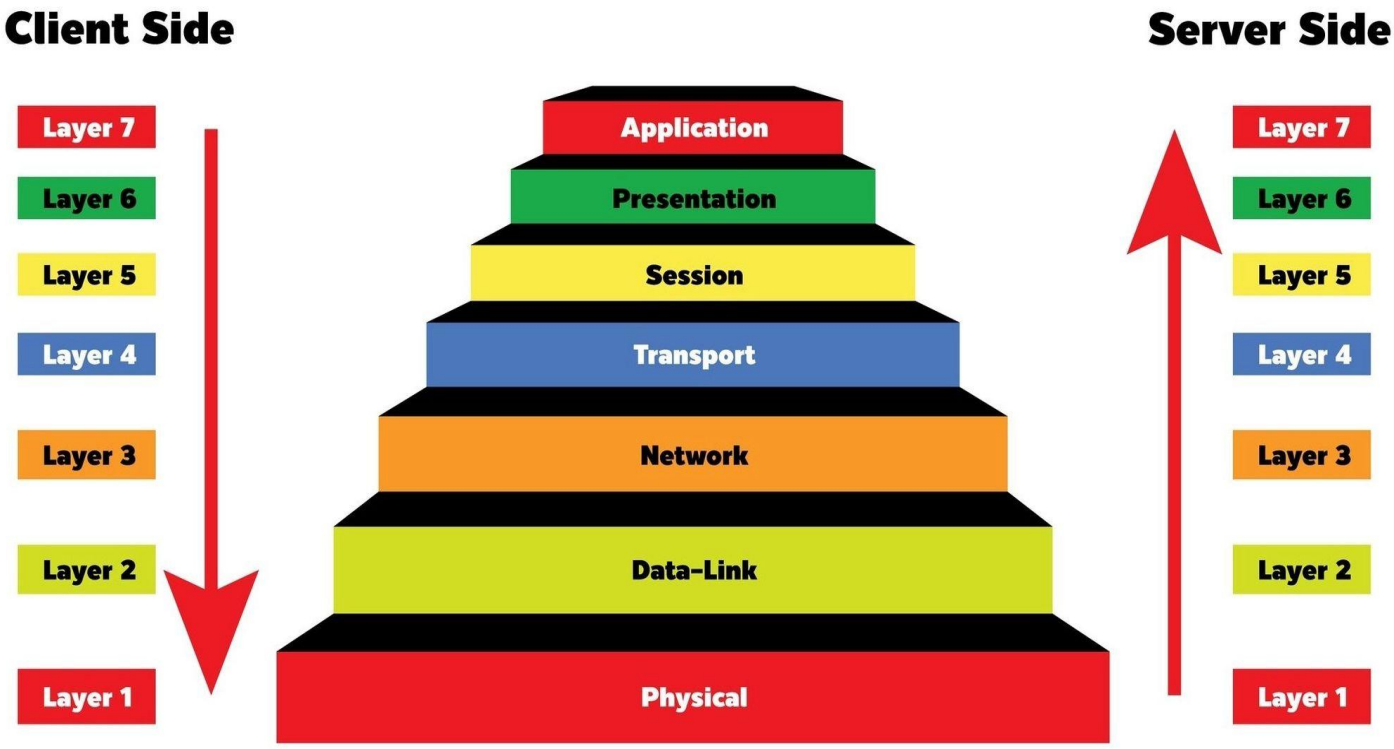
Advantages :

- Simple and inexpensive
- Easy to extend.

Disadvantages :

- If one link fails, communication stops beyond that point.
- Slower as network grows.

OSI Model



1. **Physical Layer :** Deals with hardware transmission – raw bits (0s and 1s) are converted into signals (electrical, optical, or radio).
2. **Data link Layer :** Responsible for error detection, flow control and MAC addressing. It ensures data frames are delivered to the correct device on the same network.
3. **Network Layer :** Handles logical addressing (ip) and routing – decides the best path for data to travel.
4. **Transport Layer :** Provides end-to-end communication, error recovery, and flow control. It Segments large messages into smaller units.
5. **Session Layer :** Manages sessions between applications – opens, maintains, and closes connections.
6. **Presentation Layer** : Ensures that data is in readable format for the application. Handles encryption, compression and data conversion (like ASCII to binary).

7. **Application Layer :** Closest to the user – provides network services directly to applications.

Encapsulation and de-encapsulation

Encapsulation in OSI Model:

Encapsulation is the process of **adding control information** to data as it moves **down through the OSI layers** from the **Application Layer to the Physical Layer** before being transmitted over the network.

When a user sends data, it starts at the **Application Layer** as simple information. As it passes through each layer, that layer adds its own header (and sometimes a trailer) containing important details needed for communication — such as addressing, routing, or error detection.

Encapsulation = Adding headers as data is prepared for transmission.

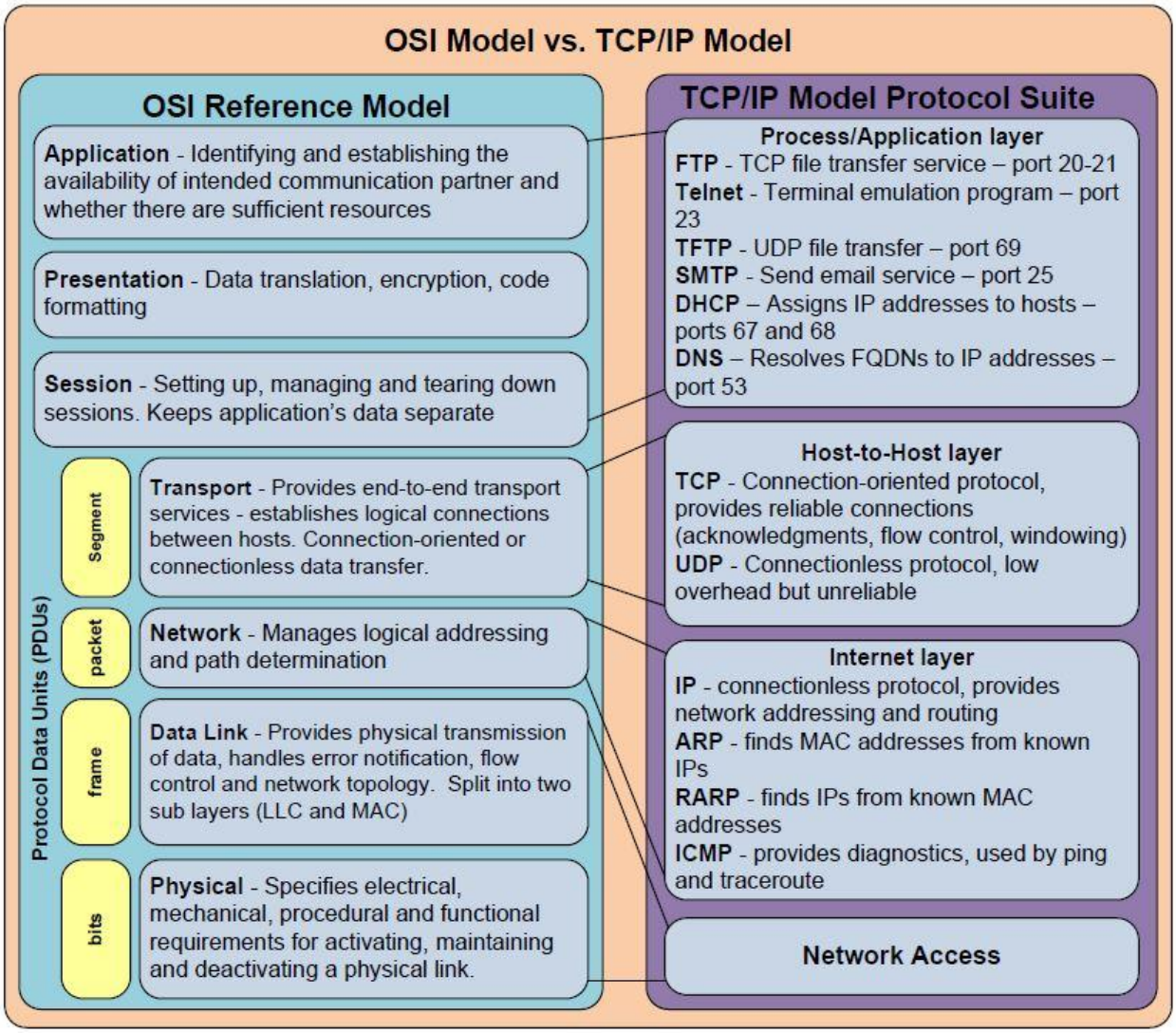
Decapsulation in OSI Model:

De-encapsulation is the **reverse process** of encapsulation. It occurs when data is **received** by the destination device and moves **up the OSI layers** from the **Physical Layer to the Application Layer**.

As the data travels upward, each layer **removes its corresponding header and trailer** and processes only the information meant for that layer. The **Physical Layer** converts signals into bits, the **Data Link Layer** verifies frames, the **Network Layer** checks the IP address, and so on — until finally, the **Application Layer** receives the original data in its usable form.

De-encapsulation = Removing those headers as data is received and interpreted.

TCP/IP Model



The TCP/IP Model (Transmission Control Protocol / Internet Protocol) is the fundamental framework that defines how data is transmitted over the internet.

It was developed by the U.S Department of Defence (DoD) in the 1970s and is sometimes called the DoD Model.

The TCP/IP model is simpler than the OSI model and has four layers, each performing specific functions to ensure successful data transmission.

1 . Application Layer : The Application Layer is the top layer of the TCP/IP model.

It provides network service directly to users and applications, enabling communication through softwares like web browsers, email clients, and file transfer tools.

This layer combines the OSI model's Application, Presentation, and Session layer into one.

Examples of Protocols :

- HTTP / HTTPS – Web Browsing.
- FTP / SFTP – File transfer
- SMTP / POP3 / IMAP – Email services
- DNS – Domain name resolution
- Telnet / SSH – Remote login

2 . Transport Layer : The Transport layer ensures end-to-end communication between devices. It is responsible for data segmentation, flow control, and error recovery. This layer decides how much data should be sent, checks for errors, and ensures that the data arrives correctly and in order.

It corresponds to the Transport Layer (layer 4) of the OSI model.

Main Protocols :

TCP (Transmission Control Protocol) : Reliable, connection-oriented, ensures all data packets arrive correctly and in sequence. Used for web, email, and file transfer.

UDP (User Datagram Protocol) : Unreliable, Connectionless, faster but does'nt guarantee delivery. Used for streaming , gaming, and voice calls.

3 . Internet Layer Protocols : These protocols handle logical addressing and routing of data packets across different networks.

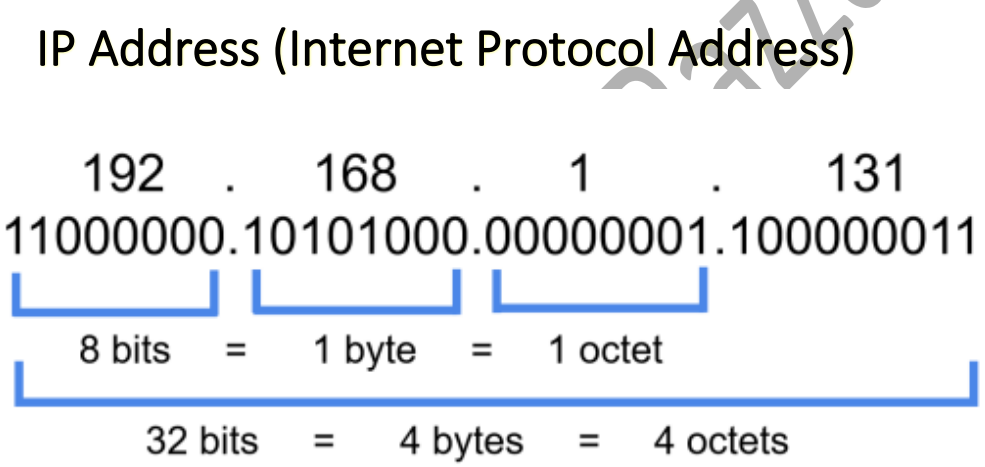
Examples :

- IP (Internet Protocol)
- ICMP (Internet Control Message Protocol)
- ARP (Address Resolution Protocol)
- RARP (Reverse Addresses Resolution Protocol)

4 . Network Access (Data Link And Physical) Layer Protocols : These protocols work at the lowest layers, defining how data is transmitted over physical media such as cables, Wi-Fi, or fiber optics.

Examples :

- Ethernet
- Wi-Fi
- PPP (Point-to-Point Protocols)
- Frame Relay
- HDLC (High-Level Data Link Control)



IP Address : An IP Address is a unique string of number assigned to every device on a network. It identifies where data should be sent and where it came from.

- IPv4 : 32-bit, looks like 192.168.1.1 (4 billion addresses)
- IPv6 : 128-bit, looks like 2001:0db8:85a3:0000:0000:8a2e:0370:7334 (billions of trillions of addresses)

MAC Address (Media Access Control) : A MAC address is a hardware ID assigned to the network interface card (NIC) of a device. It's permanent and unique like a fingerprint.

Format : 00:1A:2B:3C:4D:5E.

IP Address To Binary

What is the binary 11111111 in decimal?

	128		64		32		16		8		4		2		1		
Binary	1		1		1		1		1		1		1		1		
Decimal	128	+	64	+	32	+	16	+	8	+	4	+	2	+	1	=	**255 Decimal**

Add the number where there is a "1".
Add zero, when there is a "0".

Binary to IP Address Examples :

1 . 10101100.01001101.10111000.00011010

Ans : 172.77.184.26

2 . 10110011.01000110.00101011.10101111

Ans : 179.70.43.175

3 . 00100100.10010010.00110011.10010011

Ans : 36.146.51.147

4 . 11100101.01110101.00110010.10001001

Ans : 227.117.50.137

IP Address To Binary :

1 . 184.126.139.72

Ans : 10111000.01111100.10001011.01001000

2 . 112.214.52.142

Ans : 01110000.11010110.00110100.01111011

3 . 129.214.119.123

Ans : 10000001.11010110.01110111.01111011

Network Security

Network Security is the practice of **protecting computer networks** and the data that travels across them from **unauthorized access, misuse, modification, or cyberattacks**.

It ensures that your network — whether it's a home Wi-Fi, an office LAN, or the Internet — remains **secure, reliable, and available** for legitimate users.

In simple terms:

Network Security = Protecting data *while it moves* from one device to another.

Why Network Security Is Important

Every organization (and even every individual) relies on networks to store, share, and access information.
Without security, hackers can:

- Steal personal or financial data
- Infect systems with malware
- Disrupt services (like websites or servers)
- Spy on network traffic

So, **network security** protects both people and systems from digital threats.

Key Components of Network Security

1. Firewalls

Act as a barrier between trusted internal networks and untrusted external ones (like the Internet).
They filter traffic based on security rules and block suspicious data.

2. Intrusion Detection and Prevention Systems (IDS/IPS)

Monitor network traffic for suspicious activity or known attack patterns and alert or block them.

3. Antivirus and Anti-malware Software

Detect and remove malicious programs that can damage systems or steal data.

4. Virtual Private Network (VPN)

Encrypts data for secure remote connections over public networks.

5. Access Control

Defines who can access what within a network. Uses authentication (passwords, biometrics) and authorization policies.

6. Encryption

Converts data into unreadable form so even if intercepted, it remains private.

7. Network Segmentation

Divides a large network into smaller sections to contain attacks and limit damage.

8. Security Policies

Guidelines and rules that define how users, devices, and data are protected.

How Network Security Works

1. **Monitoring** – Observe all incoming and outgoing data packets.
2. **Detection** – Identify unusual or suspicious behavior.
3. **Prevention** – Block harmful traffic or access.
4. **Response** – Take action (alert admins, isolate systems, etc.).

All this is managed using security tools like **firewalls**, **routers**, **IDS/IPS**, and **security policies**.

In Simple Words:

Network Security is like a multi-layered shield 🛡️ that protects data, devices, and communication from being hacked, stolen, or disrupted.

Bonus Tips :

- Use sudo carefully — it gives **root-level access**.
- Combine commands using pipes: | (e.g., ps aux | grep apache).