


Computer Networks

⇒ Networking Fundamentals

 Networking

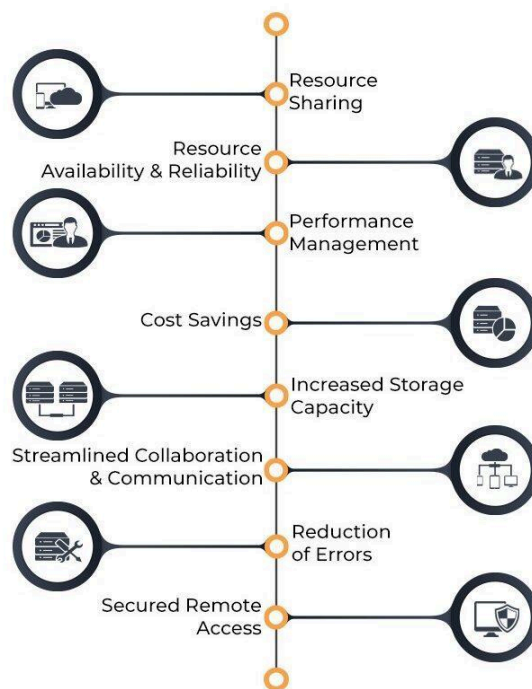
- **Computer network** A collection of computing devices that are connected in various ways in order to communicate and share resources

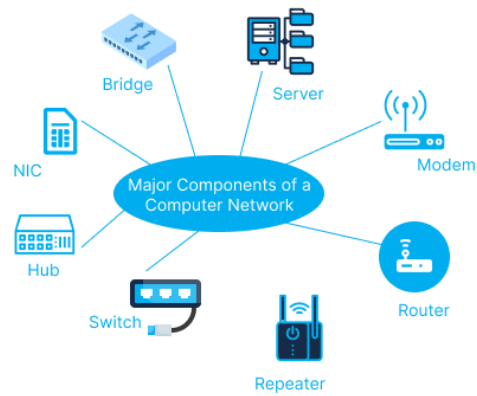
Usually, the connections between computers in a network are made using physical wires or cables

However, some connections are **wireless**, using radio waves or infrared signals

[5-1]

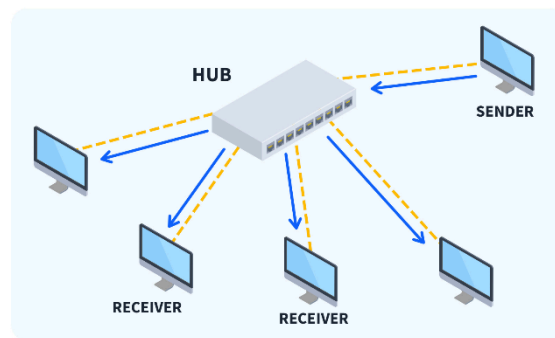
KEY OBJECTIVES OF DEPLOYING A COMPUTER NETWORK





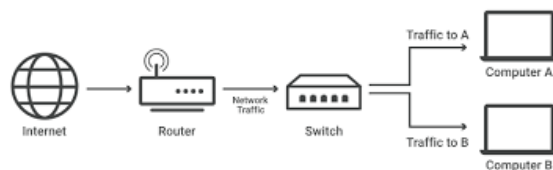
1. Network Interface Card (NIC)

- It's like a **door** for your computer to connect to the internet or other computers.
 - It can use **wires (Ethernet)** or **wireless signals (Wi-Fi)** to send and receive data.
 - Every NIC has a **unique ID** called a **MAC address** to identify your device in the network.
 - It's needed for **communication** between devices.
-



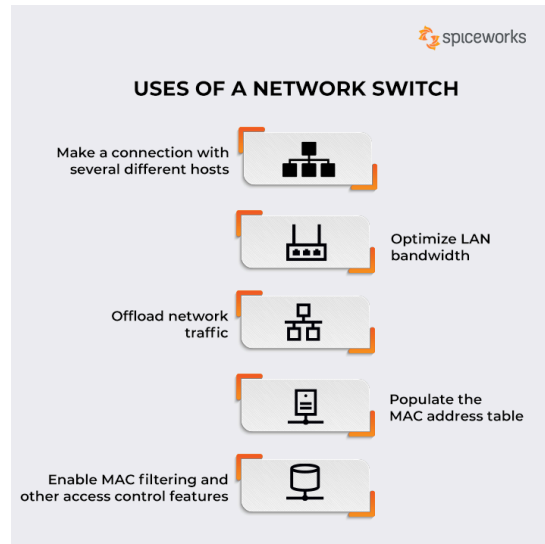
2. Hub

- Imagine a **meeting room** where one person speaks, and **everyone listens**, even if the message isn't for them.
 - That's how a **hub** works—it sends the same data to **all devices** connected to it.
 - **Simple and cheap**, but it can cause **traffic jams** if too many devices are connected.
-



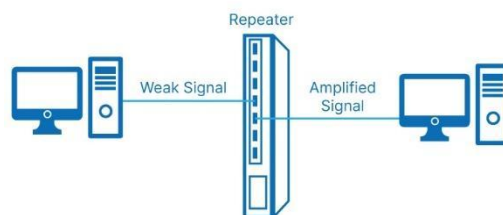
3. Switch

- A switch is like a **postman** who delivers letters to the **right address** instead of giving them to everyone.
- It checks the **destination address** (MAC address) and sends data only to the **intended device**.
- It's **faster and more efficient** than a hub.
- Used in **offices and big networks** to avoid **data collisions**.

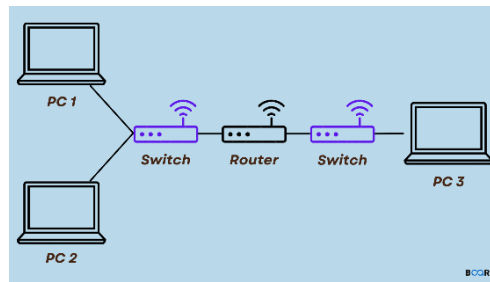


4. Repeater

- Think of a **lighthouse** that shines light farther to help ships see.
- A **repeater** takes a **weak network signal**, **boosts it**, and sends it further.
- It's useful when the **distance between devices** is long, like in **large buildings**.
- Example: **Wi-Fi range extenders** in homes.

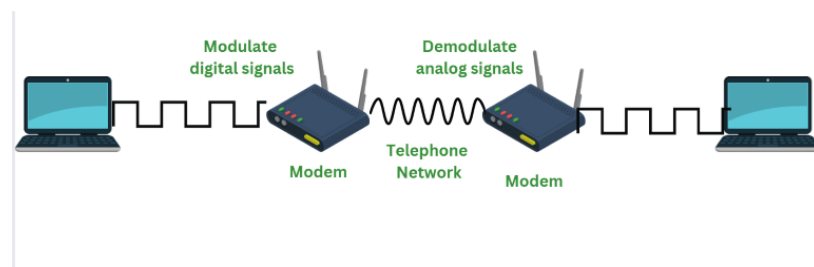


5. Router



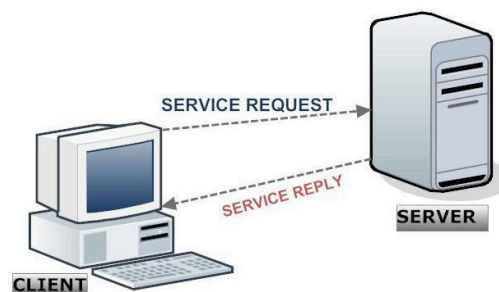
- A router is like a **traffic cop** that directs cars (data) to the **right roads (networks)**.
- It connects **your home network** to the **internet** and ensures data gets to the **correct destination**.
- It also provides **Wi-Fi** so you can connect phones, laptops, and smart devices **wirelessly**.
- Many routers include **firewalls** to **block hackers** and **protect your data**.

6. Modem



- The modem is like a **translator** that converts signals from your **Internet Service Provider (ISP)** into data your computer can use—and vice versa.
- Example: It converts signals from a **telephone line** or **cable line** into **internet data**.
- A **router and modem** are often combined into **one device** these days.

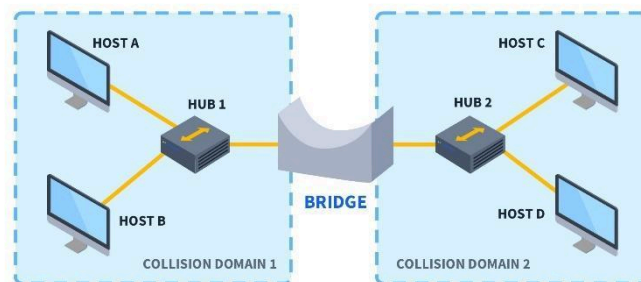
7. Server



- A server is like a **restaurant kitchen** where food (data) is prepared and sent out to customers (computers).
- It stores and manages data for websites, emails, and files.

- Examples:
 - **Web server** (shows websites).
 - **Mail server** (handles emails).
 - **File server** (stores files for sharing).
 - Servers run **24/7** to ensure the services are always available.
-

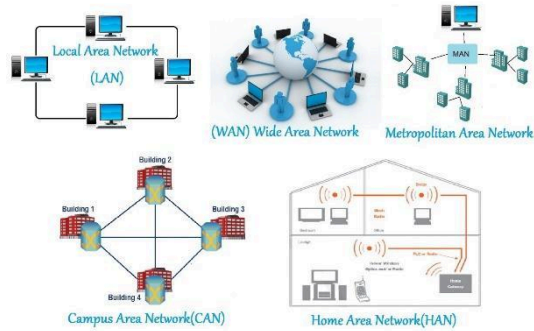
8. Bridge



- A bridge is like a **bridge over a river** that connects **two areas**.
 - It links **two separate networks** and allows them to **talk** to each other.
 - Used when you have **different types of networks** but want them to work as **one big network**.
 - Example: **Connecting wired and wireless networks** in a building.
-

Putting It All Together

1. Your **modem** connects to the **internet**.
2. Your **router** sends the internet to different devices at home using **Wi-Fi or Ethernet**.
3. A **switch** organizes communication between devices if you have many computers.
4. A **repeater** can boost signals to cover a **bigger area**.
5. Devices like **servers** store websites and files for you to access anytime.
6. **NICs** in every computer allow them to **join the network** and **communicate**.
7. A **bridge** can link networks if they need to work **together**.
8. A **hub** can connect devices simply, but modern networks mostly use **switches** instead.



1. LAN (Local Area Network)

- **Small network** in one place, like a **home, office, or school**.
 - Example: Computers in your office connected to share files or printers.
-

2. WLAN (Wireless Local Area Network)

- Same as **LAN**, but **wireless**—no cables needed.
 - Example: Wi-Fi at home or in a café.
-

3. CAN (Campus Area Network)

- **Bigger than LAN**, covering a **college campus or group of buildings**.
 - Example: All departments in a university connected together.
-

4. MAN (Metropolitan Area Network)

- Covers a **city or large area**.
 - Example: Internet service used across an entire **city**.
-

5. PAN (Personal Area Network)

- Very **small network** for personal devices.
 - Example: Connecting your **phone to Bluetooth headphones**.
-

6. SAN (Storage Area Network)

- A network used to **store and manage data**.
 - Example: Companies storing files or backups on **big servers**.
-

7. VPN (Virtual Private Network)

- Creates a **secure and private connection** over the internet.
 - Example: Accessing your **office network from home** securely.
-

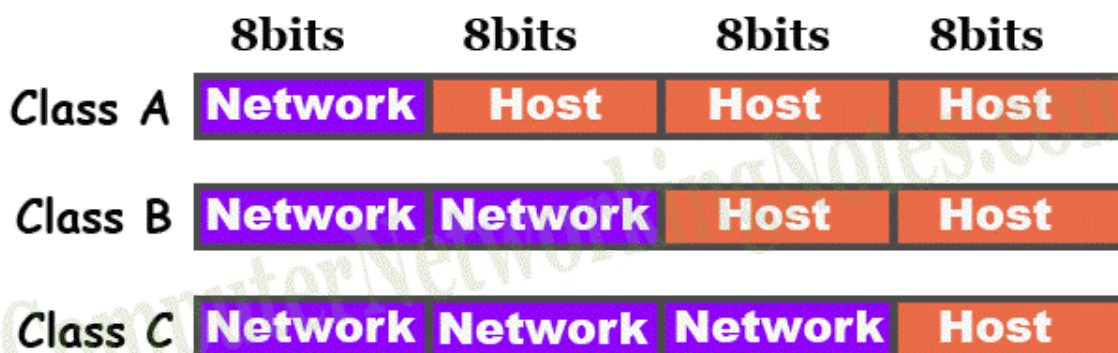
8. WAN (Wide Area Network)

- Covers a **large area**, like **countries or continents**.
- Example: The **Internet** is the largest WAN.

Network Terminologies

1. IP Address

CLASS A, B, C IP ADDRESSES EXPLAINED					
	IP Addresses	Range	Subnet Mask	Number of Available Networks	Number of Available Hosts per Network
Class A	0.0.0.0 to 127.255.255.255	0 - 127	255.0.0.0	126	16,777,214
Class B	128.0.0.0 to 191.255.255.255	128 - 191	255.255.0.0	16,384	65,534
Class C	192.0.0.0 to 223.255.255.255	192 - 223	255.255.255.0	2,097,152	254
Class D	224.0.0.0 to 239.255.255.255	224 - 239	255.255.255.255	Multicast Communication	Multicast Communication
Class E	240.0.0.0 to 255.255.255.255	240 - 255	255.255.255.255	Experimental Purposes	Experimental Purposes



- An **IP address** is like a **home address** for devices connected to the internet.
- It's a **unique number** that helps identify and locate devices.

- No **two devices** in the same network can have the **same IP address**—just like no two houses can have the **same address**.
- When your computer sends data, it attaches a **header** with both:
 - **Source IP address** (where data is coming from).
 - **Destination IP address** (where data is going).
- Example: **192.168.1.1**

CLASS	LEADING BITS	NET ID BITS	HOST ID BITS	NO. OF NETWORKS	ADDRESSES PER NETWORK	START ADDRESS	END ADDRESS
CLASS A	0	8	24	2^7 (128)	2^{24} (16,777,216)	0.0.0.0	127.255.255.255
CLASS B	10	16	16	2^{14} (16,384)	2^{16} (65,536)	128.0.0.0	191.255.255.255
CLASS C	110	24	8	2^{21} (2,097,152)	2^8 (256)	192.0.0.0	223.255.255.255
CLASS D	1110	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	224.0.0.0	239.255.255.255
CLASS E	1111	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	240.0.0.0	255.255.255.255

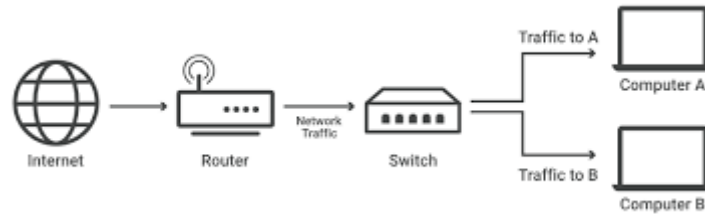
2. Nodes

- A **node** is any device connected to a network.
 - It could be a **computer, printer, modem, switch**, or any other device that can **send, receive, or store data**.
 - Think of it like **people in a meeting**—each person (node) can **communicate** with others.
 - Nodes have **unique IDs**, like **IP addresses**, so they can be identified in the network.
-

3. Routers

- A **router** works like a **traffic police officer** that guides data packets to their **destination**.
 - It connects **different networks** (like your home network and the internet).
 - Routers **analyze traffic**, decide the **best path** for data, and **forward packets** accordingly.
 - Sometimes, data may pass through **multiple routers** before reaching the final device.
 - Modern routers also provide **Wi-Fi** for wireless connections.
-

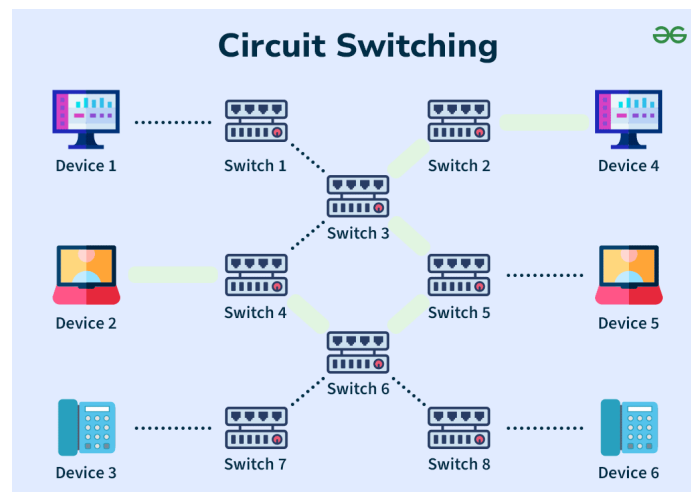
4. Switches



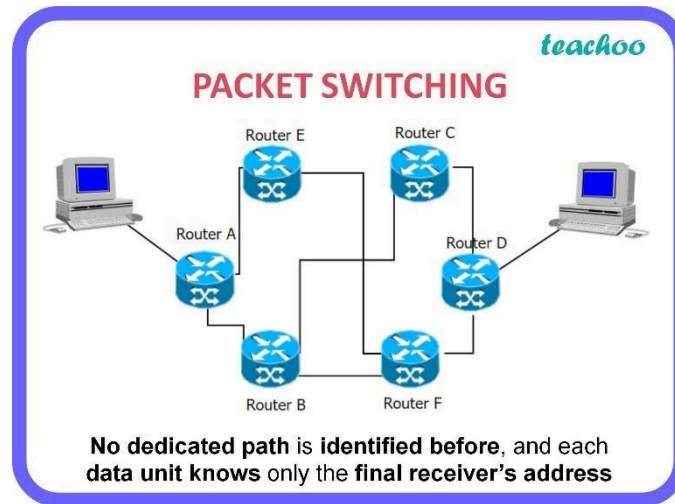
- A **switch** is like a **postman** inside a single network.
- It **connects devices** within the same network and **sends data** only to the **intended device** (instead of broadcasting it to all).
- **Difference from Router:**
 - **Router** connects **networks**.
 - **Switch** connects **devices** inside **one network**.

Switching Methods:

1. Circuit Switching:

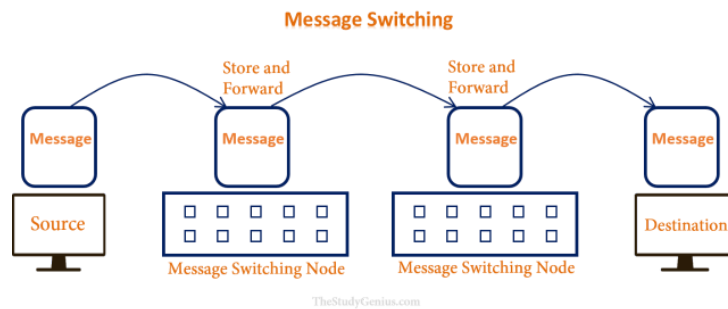


- Think of it as a **telephone call**—a **dedicated line** is set up between sender and receiver.
 - Once the call (data transfer) ends, the line is **freed** for others.
 - Best for **voice calls** or **video conferencing** where uninterrupted data flow is needed.
- #### 2. Packet Switching:



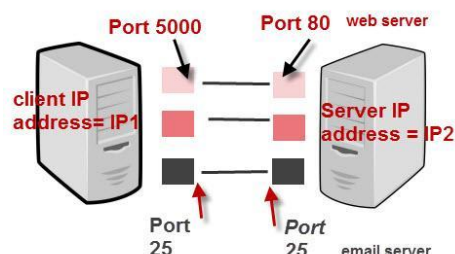
- Data is **broken into small packets** and sent individually.
- Each packet has its **source and destination address**.
- Example: Sending **emails** or browsing the **web**.

3. Message Switching:



- Complete **messages are stored and forwarded** step-by-step through different nodes.
- Slower but useful for **non-real-time applications** like **text messaging**.

4. Ports



IP Address + Port number = Socket

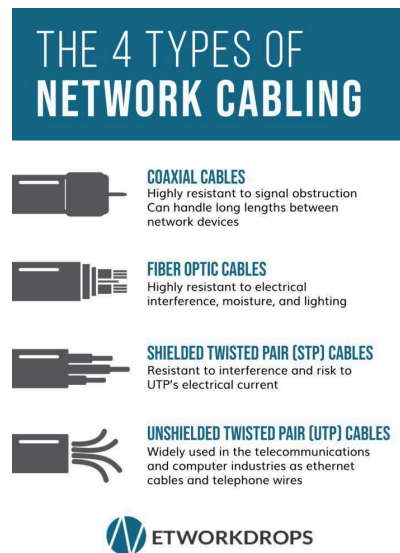
TCP/IP Ports And Sockets

- **Ports** are like **room numbers** in a hotel.

- While an **IP address** is like the **hotel address**, ports tell **which room (application/service)** the data should go to.
 - Example:
 - **Port 80** is used for **web browsing (HTTP)**.
 - **Port 25** is used for **emails (SMTP)**.
 - Each service or app uses a **specific port number** to handle its communication.
-

6. Network Cable Types

- **Network cables** are physical wires that **connect devices** in a network.



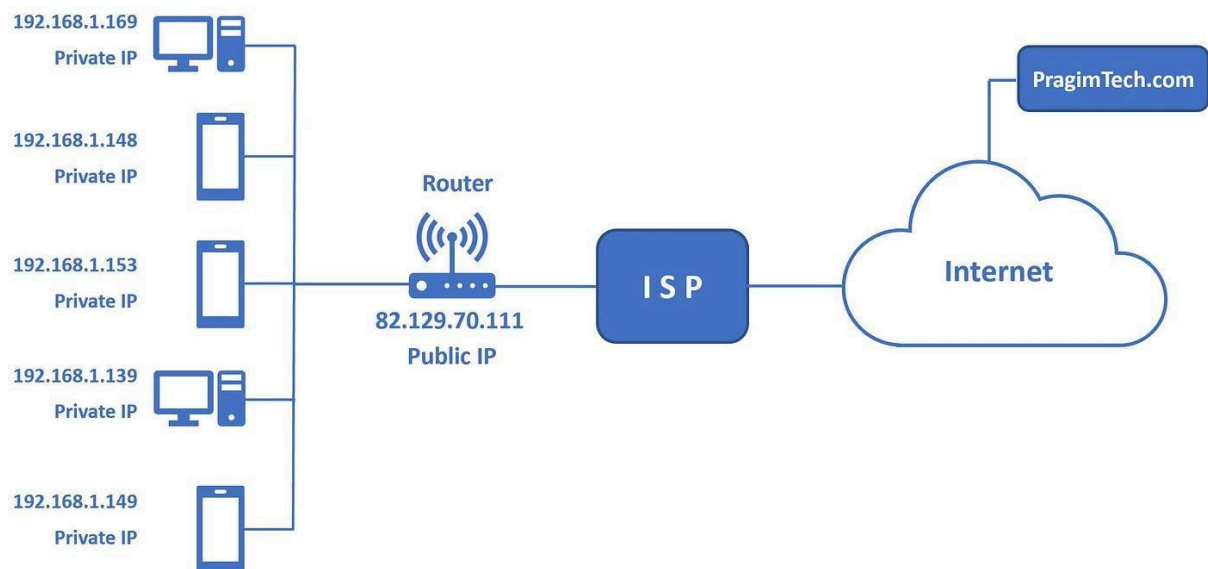
Types of Cables:

- Ethernet Cable (Twisted Pair):**
 - Most **common** for **home and office** networks.
 - Looks like a **telephone wire but thicker**.
 - Best for **short distances** and **local connections**.
- Coaxial Cable:**
 - Looks like the **cable TV wire**.
 - Used earlier for **internet connections** but less common now.
- Fiber Optic Cable:**
 - Uses **light signals** instead of electricity.
 - Super **fast** and supports **long distances** without losing signal.

- Mostly used in **big data centers** or for **high-speed internet**.

Putting It Together:

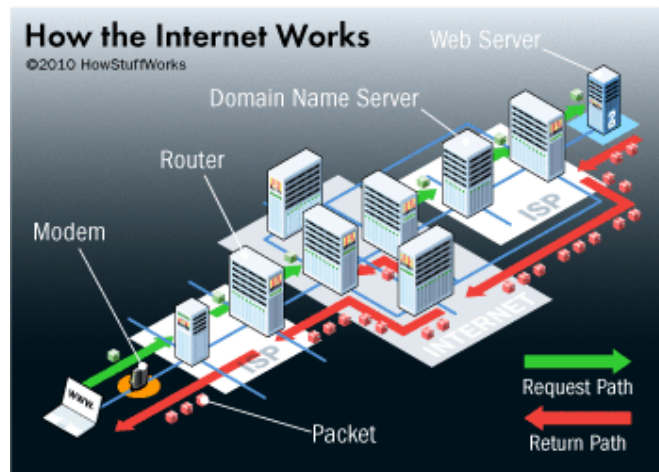
1. **IP addresses** identify where data needs to go.
2. **Nodes** are the devices (computers, printers) in the network.
3. **Routers** guide data between different networks and connect to the internet.
4. **Switches** manage communication within the same network and improve efficiency.
5. **Ports** organize data for specific applications or services.
6. **Cables** physically connect devices and help transfer data.



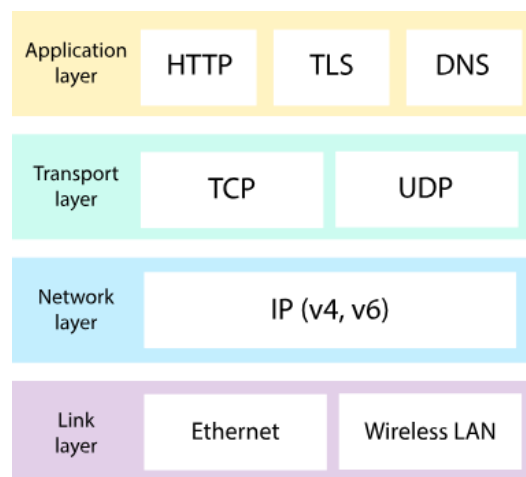
What is the Internet?

- The **Internet** is the **largest Wide Area Network (WAN)** that connects **billions of computers** all over the world.
- It allows computers, smartphones, and other devices to **communicate** and **share information** globally.

How Does It Work?



- The **Internet** works by following **rules** called **protocols** to send and receive data between devices.
- These protocols make sure that information reaches the **right device** without errors.
- Data travels in small pieces called **packets** through networks to reach its destination.



Important Protocols Used on the Internet

1. **HTTP (Hypertext Transfer Protocol):**
 - Allows websites to send and receive data.
 - Example: When you type a URL in your browser, it uses **HTTP** to load the web page.
2. **IP (Internet Protocol):**
 - Assigns a **unique address (IP address)** to each device, like a **home address**.
 - This helps in **identifying devices** and sending data to the **correct location**.
3. **TCP (Transmission Control Protocol):**
 - Breaks data into **small packets** for easy transfer.

- Ensures that packets are sent **accurately**, in the **right order**, and **without errors**.
- Example: Used for **emails** and **web browsing**.

4. UDP (User Datagram Protocol):

- Faster than TCP but does not check for **errors** or **packet order**.
- Example: Used for **video streaming** and **online gaming**, where **speed** is more important than **accuracy**.

5. FTP (File Transfer Protocol):

- Allows you to **upload** and **download files** between computers.
- Example: Used by **web developers** to update websites.

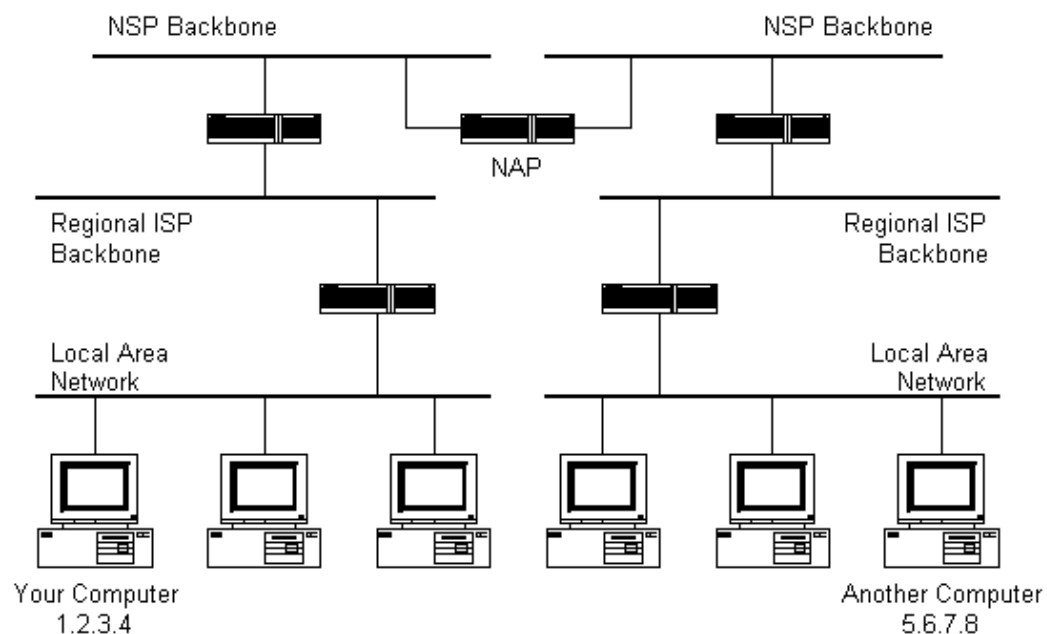
Who Manages the Internet?

1. ISPs (Internet Service Providers):

- Companies that **provide internet access** to homes and businesses.
- Example: Airtel, Jio, or BSNL in India.

2. NSPs (Network Service Providers):

- Larger organizations that **maintain the internet infrastructure**.
- They provide **global connections** to smaller ISPs.



How Does Data Travel Over the Internet?

- Data does not travel to **every device**—it only goes to the **requested destination**.
 - **Protocols** and **network devices** (like routers and switches) ensure that data packets:
 1. **Find the best path** to reach their destination.
 2. **Avoid delays** or getting **lost** in the network.
-

How Are Networks Created?

- Networks are made by **connecting devices** (computers, printers, etc.) called **nodes**.
 - Devices can be connected:
 1. **Wired (Cables):** Using **Ethernet cables** or **fiber optics**.
 2. **Wireless:** Using **Wi-Fi** or **Bluetooth**.
-

How Do Devices Communicate?

- Devices use **IP addresses** to identify each other.
 - When you send data:
 1. It is **broken into packets**.
 2. Each packet contains the **source** and **destination IP address**.
 3. The data is sent through **routers** and **switches** until it reaches the **right device**.
-

Role of Routers and Switches

1. **Routers:**
 - Like **traffic controllers**, they **direct data** between **different networks**.
 - Find the **best path** for data packets to reach their destination.
 2. **Switches:**
 - Connect **devices** inside a **single network** (like in an office).
 - They **send data** only to the **specific device** that needs it, not to all devices.
-

Example of How It Works:

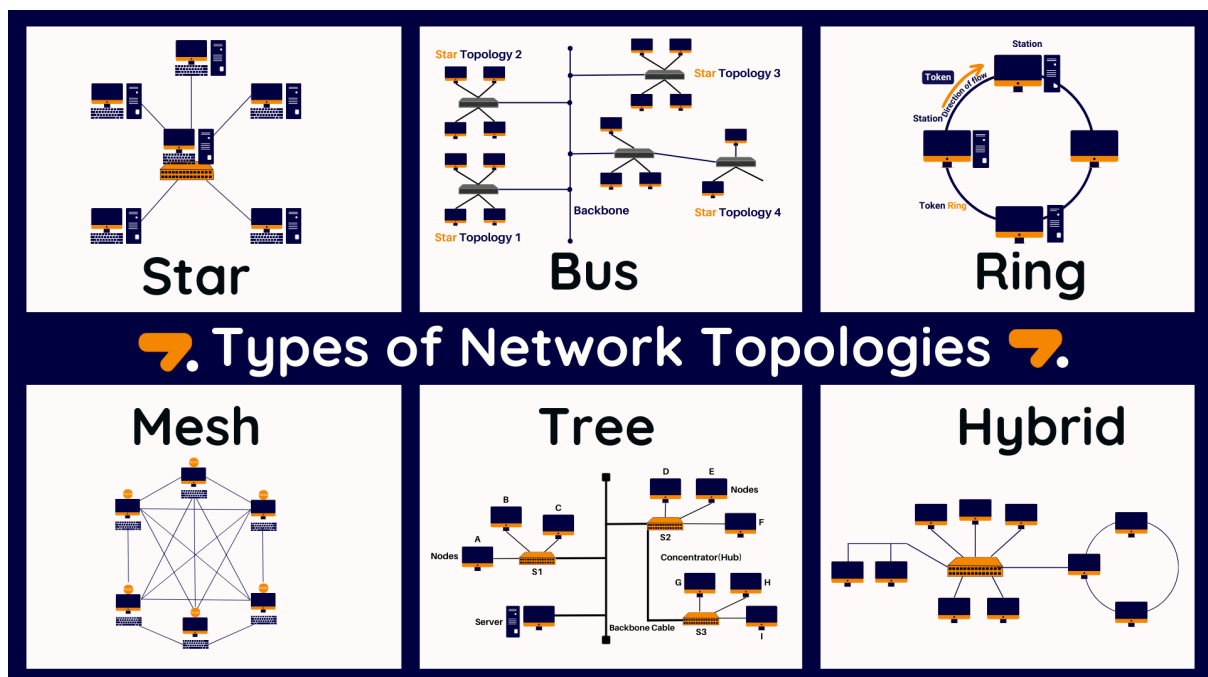
- Imagine you are **sending an email** to a friend.
 1. Your email is **broken into small packets**.
 2. Each packet has your **IP address** (sender) and your friend's **IP address** (receiver).
 3. **Routers** guide the packets through the **best path**.

4. **Switches** ensure packets stay **organized** inside networks.
5. Packets **reach your friend's device** and are **combined back** into the email.

Summary

- The **Internet** connects devices globally using **protocols** to manage communication.
- **Protocols** like **HTTP**, **TCP/IP**, and **FTP** ensure data is sent correctly.
- **Routers** handle **traffic between networks**, while **switches** manage devices **within a network**.
- Networks use **wired** (Ethernet, fiber optic) and **wireless** (Wi-Fi, Bluetooth) connections.
- **ISPs** and **NSPs** provide and maintain **internet infrastructure**.

Imagine you want to connect several computers or devices together so they can share information. The way you connect these devices is called "network topology." It's like setting up a system for passing notes between friends in different ways.



1. Bus Topology:

- Think of a single rope connecting all the devices in a straight line.
- When one device sends a message, all the devices can "hear" it, but only the one meant to get the message will respond.
- It's a simple system, but if the rope breaks, the whole network might stop working.

2. **Ring Topology:**

- Imagine the devices are sitting in a circle, and they pass messages one by one, like passing a note around the class.
- The message goes in one direction until it reaches the right device.
- If someone breaks the circle, the whole system can stop.

3. **Star Topology:**

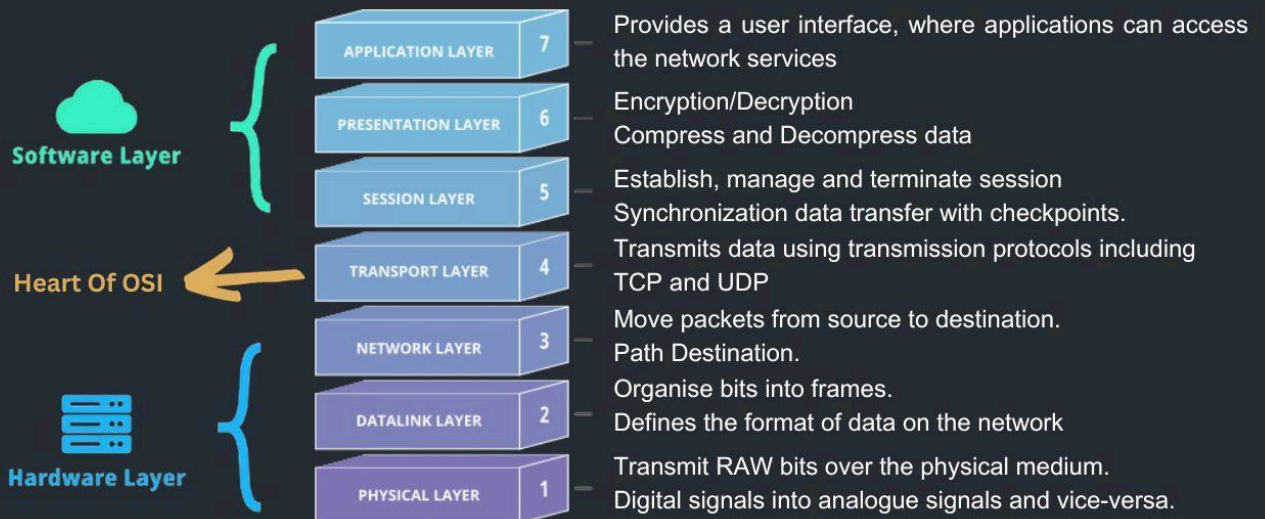
- Picture a central point, like a teacher, with everyone else sitting around them.
- All messages go to the teacher first, and then the teacher passes the message to the right person.
- It's easy to manage, but if the teacher (central point) is missing, no one can talk to each other.

4. **Mesh Topology:**

- Here, every device is connected directly to every other device, like every person in a group having a direct phone line to every other person.
- It's very reliable because if one connection fails, the others still work, but it's also more complicated and expensive to set up.
- There's a simpler version, where only some devices are connected to each other, to save money and make things easier.

⇒ **OSI Model**

7 Layers of the OSI Model



What is a Network Model?

When devices like computers or phones need to talk to each other, they need to follow certain rules and processes. These rules are organized into **models**. The **OSI Model** and **TCP/IP Model** are two such models that help understand how data travels over a network.

OSI Model (Open Systems Interconnection)

Think of the OSI model as a set of **7 layers** that explain how data is sent from one device to another. Each layer has a specific job, just like different parts of a factory where each person does a different task to build a product.

Here's a simple way to think about each layer:

1. Physical Layer (Layer 1)

- **Job:** This layer deals with the actual hardware, like cables and devices. It handles the physical connection between computers.
- **Example:** Ethernet cables, Wi-Fi signals, and even the light in fiber-optic cables.

2. Data Link Layer (Layer 2)

- **Job:** It makes sure the data you send is correctly packaged into chunks called frames, and that data is sent reliably between two devices.
- **Example:** Think of it like labeling packages so the right one goes to the right address. **Ethernet** is an example of a protocol here.

3. Network Layer (Layer 3)

- **Job:** This layer decides the best path for the data to travel across the network. It adds an address to the data, which is called an **IP Address** (like a home address for the data).
- **Example: IP (Internet Protocol)** is a common protocol used at this layer. It ensures the data finds its way to the correct destination.

4. Transport Layer (Layer 4)

- **Job:** This layer ensures that the data is sent properly and is complete. It checks if the data reaches the destination correctly and in the right order.
- **Example: TCP (Transmission Control Protocol)** makes sure data is delivered without errors, while **UDP (User Datagram Protocol)** is faster but doesn't check for errors.

5. Session Layer (Layer 5)

- **Job:** This layer manages the conversation between two devices. It opens, manages, and closes communication sessions.
- **Example:** It makes sure the devices know when to start and end a conversation.

6. Presentation Layer (Layer 6)

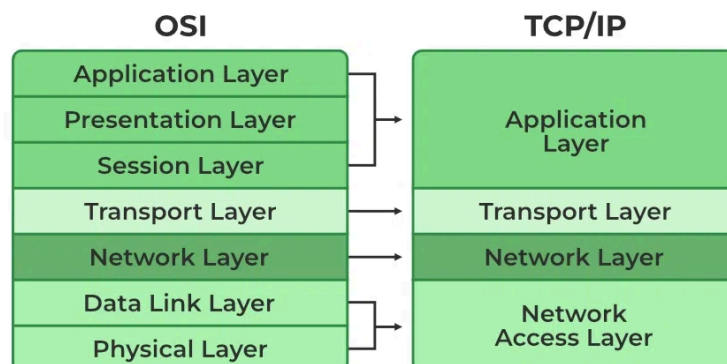
- **Job:** This layer translates the data into a format the application can understand. It's like translating languages.
- **Example:** If one computer speaks English and the other speaks French, this layer helps them understand each other.

7. Application Layer (Layer 7)

- **Job:** This is where the actual software applications work. It interacts directly with the user and provides the services the user needs, like browsing the web or sending emails.
- **Example: HTTP** (used for browsing websites), **FTP** (used for file transfers), and **SMTP** (used for sending emails).

TCP/IP Model

The **TCP/IP Model** is a simpler version of the OSI model. It's made up of **4 layers** and is the model that the Internet is based on. The layers are similar but combine some of the layers in the OSI model into fewer ones.



Here's a simple breakdown of the TCP/IP Model:

1. **Link Layer** (combines OSI's Physical and Data Link Layers)
 - Deals with hardware and how data is transmitted over a physical network.
 - Example: Ethernet, Wi-Fi.
2. **Internet Layer** (similar to OSI's Network Layer)
 - Handles routing and addresses data to get it to the right place.
 - Example: **IP (Internet Protocol)**.
3. **Transport Layer** (same as OSI's Transport Layer)
 - Ensures that data is transferred correctly.
 - Example: **TCP** for reliable data transfer, **UDP** for faster but less reliable transfer.
4. **Application Layer** (combines OSI's Session, Presentation, and Application Layers)
 - Deals with the actual applications and user interactions.
 - Example: **HTTP** (browsing websites), **FTP** (file transfer), **SMTP** (email).

Key Protocols in Networking

Protocols are like rules for communication. Here are some important ones:

8 Popular Network Protocols

blog.bytebytego.com

Protocol	How does It Work?	Use Cases
HTTP	<p>TCP Connection HTTP REQ HTTP RESP</p>	<p>Web Browsing</p>
HTTP/3 (QUIC)	<p>UDP Connection 1 2 3 4 5</p>	<p>IoT Virtual Reality</p>
HTTPS	<p>TCP Connection public key session key encrypted data</p>	<p>Web Browsing</p>
WebSocket	<p>HTTP Upgrade Full Duplex</p>	<p>Live Chat Real-Time Data Transmission</p>
TCP	<p>SYN SYN + ACK ACK</p>	<p>Web Browsing Email Protocols</p>
UDP	<p>REQUEST RESPONSE</p>	<p>Video Conferencing</p>
SMTP	<p>sender SMTP Server receiver</p>	<p>Sending/Receiving Emails</p>
FTP	<p>Control Channel Data Channel</p>	<p>Upload/Download Files</p>

1. TCP (Transmission Control Protocol)

- Ensures reliable data transfer. It checks if data is received correctly and requests retransmission if anything is missing.
- Example:** Used when sending emails or downloading files where accuracy is important.

2. UDP (User Datagram Protocol)

- A faster alternative to TCP, but it doesn't check if data is received correctly. It's okay if some data gets lost (used when speed is more important than accuracy).

- **Example:** Used for streaming videos or live games where speed matters more than perfection.

3. IP (Internet Protocol)

- Responsible for addressing and routing the data. Every device on the network has a unique IP address, just like every house has a unique address.
- **Example:** **IPv4** and **IPv6** are different versions of IP.

4. ICMP (Internet Control Message Protocol)

- Used for error messages and diagnostic functions. It helps devices communicate about issues in the network (like "Destination Unreachable").
- **Example:** When you type "ping" in your command prompt, you're using ICMP to check if a device is reachable.

5. ARP (Address Resolution Protocol)

- Helps find the physical address (MAC address) of a device on a network using its IP address. It's like looking up someone's phone number in a phone book using their name.
- **Example:** Used in local networks to map IP addresses to MAC addresses.

1. What is an IP Address?

An **IP Address (Internet Protocol Address)** is like the **home address** of your computer or device in a network. It helps to identify where your device is and allows it to send and receive data.

For example:

- If you want to send a letter to your friend, you need their **home address**.
- Similarly, computers need an **IP address** to send data to each other.

Two Versions of IP Addresses: IPv4 and IPv6

1. IPv4 (Internet Protocol Version 4)

- It is the older and most commonly used version.
- It looks like this: **192.168.1.1** (4 numbers separated by dots).
- Each number ranges from **0 to 255**.

Example:

192.168.0.1 – Used by Wi-Fi routers in homes.

Problem with IPv4:

- Limited addresses (about **4.3 billion**) and we are running out of them.

2. IPv6 (Internet Protocol Version 6)

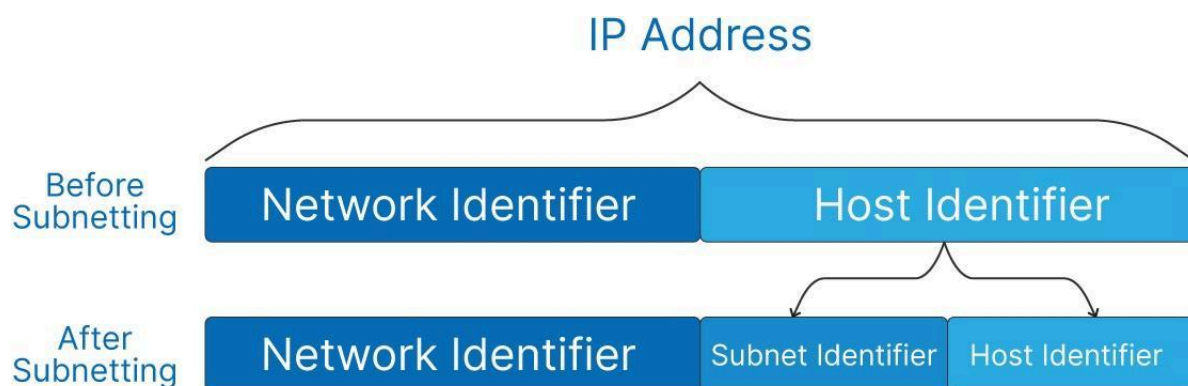
- Newer version that provides **a lot more addresses** (about **340 undecillion addresses**).
- It looks like this: **2001:0db8:85a3:0000:0000:8a2e:0370:7334** (longer and uses hexadecimal).

Why do we need IPv6?

- More devices (phones, laptops, IoT devices) need IP addresses, and IPv6 can handle this demand.

IPv4	vs.	IPv6
Deployed 1981		Deployed 1998
32-bit IP address		128-bit IP address
4.3 billion addresses		7.9×10^{28} addresses
Addresses must be reused and masked		Every device can have a unique address
Numeric dot-decimal notation		Alphanumeric hexadecimal notation
192.168.5.18		50b2:6400:0000:6c3a:b17d:0000:10a9 (Simplified - 50b2:6400::6c3a:b17d:0:10a9)
DHCP or manual configuration		Supports autoconfiguration

2. Subnetting and CIDR



What is Subnetting?

Subnetting means **dividing a big network into smaller networks**. This helps organize the network and improves security.

Example:

Imagine a big company with **1000 computers**. Instead of treating them all as one group, we can divide them into smaller groups like:

- Sales department: 250 computers
- HR department: 250 computers
- IT department: 500 computers

Each group can be a **subnet**.

Every IP Addresses in the Internet		Class	Classful IP Ranges	Subnet Mask for each Block	Number of Blocks	IP addresses per Block
0.0.0.0 /0	Unicast	A	0.0.0.0 - 127.255.255.255 0.0.0.0 /1	255.0.0.0 /8	128	16,777,216
		B	128.0.0.0 - 191.255.255.255 128.0.0.0 /2	255.255.0.0 /16	16,384	65,536
		C	192.0.0.0 - 223.255.255.255 192.0.0.0 /3	255.255.255.0 /24	2,097,152	256
	Multicast	D	224.0.0.0 - 239.255.255.255	n/a	n/a	n/a
	Reserved	E	240.0.0.0 - 255.255.255.255	n/a	n/a	n/a

CIDR (Classless Inter-Domain Routing)

CIDR is a way of writing IP addresses with a **prefix** that tells how many bits are used for the network and how many for devices.

Example:

192.168.1.0/24

- **/24** means the first 24 bits are for the **network**, and the remaining 8 bits are for **devices**.
- This allows **256 devices** in this subnet.

Benefits of Subnetting and CIDR:

1. Better use of IP addresses.
2. Improved security because different departments can have separate networks.
3. Reduces network traffic by dividing it into smaller groups.

Comparison Chart

BASIS FOR COMPARISON	CIDR	VLSM
Stands for	Classless Interdomain Routing	Variable Length Subnet Masking
Basic	Enable routers to group routes together	Facilitates in optimizing the available address space
Uses the concept of	Supernetting	Subnetting
Supported by	BGP and OSPF	RIPv2, OSPF, EIGRP, IS-IS and BGP

3. IP Address Basics

An **IP Address** is like a **house address** in a network.

Example: **192.168.1.1**

- It has **4 parts** separated by dots (e.g., **192.168.1.1**).
- Each part is called an **octet** and has **8 bits**.
- So, total = **32 bits** in IPv4.

Two Parts of an IP Address:

1. **Network Part** – Identifies the **network** (like the building name).
2. **Host Part** – Identifies a **device** inside the network (like a room number).

4. Subnet Mask

A **Subnet Mask** decides how many bits belong to the **network** and how many are for the **hosts**.

Example:

- IP Address: **192.168.1.1**
- Subnet Mask: **255.255.255.0**

255.255.255.0 means:

- **First 24 bits** are for the **network**.
- **Last 8 bits** are for the **hosts**.

5. Example of Subnetting

You have **192.168.1.0/24** (256 IPs) and want to split it into **4 subnets**.

1. **Original Network:**
 - 192.168.1.0/24
 - 256 IPs (0–255).
2. **Divide into 4 subnets:**
 - Each subnet needs **64 IPs** ($256 / 4 = 64$).
3. **New Subnets:**
 - Subnet 1: **192.168.1.0/26** (64 IPs).
 - Subnet 2: **192.168.1.64/26** (64 IPs).
 - Subnet 3: **192.168.1.128/26** (64 IPs).

- Subnet 4: **192.168.1.192/26** (64 IPs).

Note: The **/26** means **26 bits** for the **network** and **6 bits** for **hosts** ($2^6 = 64$ IPs).

6. What is CIDR?

CIDR (Classless Inter-Domain Routing) is a **simpler way** to write subnet masks.

Example:

Instead of writing **255.255.255.0**, we write **/24**.

How does this work?

- The **/24** means **24 bits** are used for the **network**.
- Remaining **8 bits** are for the **hosts**.

Key CIDR Values:

CIDR	Subnet Mask	Total IPs	Usable IPs
/24	255.255.255.0	256	254
/25	255.255.255.128	128	126
/26	255.255.255.192	64	62
/27	255.255.255.224	32	30
/28	255.255.255.240	16	14
/29	255.255.255.248	8	6
/30	255.255.255.252	4	2

7. Conversion Examples

Example 1: Find Number of IPs with /27

1. **/27** means **27 bits** for the **network**.
 2. Total bits = **32**, so **32 - 27 = 5 bits** for **hosts**.
 3. Number of IPs = $2^5 = 32$ **IPs**.
 4. **Usable IPs = 32 - 2 = 30** (because 2 IPs are reserved: 1 for Network ID and 1 for Broadcast).
-

Example 2: Find Subnet Mask for /28

1. **/28** means first **28 bits** are **1s** in binary.
2. Write in binary:

11111111.11111111.11111111.11110000

3. Convert to decimal: **255.255.255.240**.

8. Private vs Public IP Addresses

Private IPs:

- Used **inside** homes, offices, or companies.
- **Cannot** be used on the **internet** directly.
- Examples: **192.168.x.x**, **10.x.x.x**, **172.16.x.x** to **172.31.x.x**.

Public IPs:

- Used to connect to the **internet**.
- **Globally unique** addresses assigned by **ISPs**.
- Example: **203.0.113.5**.

9. Quick Tips

1. **Subnetting** helps split a large network into **smaller groups**.
2. **CIDR** notation (like **/24**) is just a **shortcut** to represent subnet masks.
3. **Private IPs** are for **local use**, and **public IPs** are for the **internet**.
4. Always subtract **2 IPs** (network + broadcast) to find **usable IPs**.

10. Final Example with Full Details

IP Address: 192.168.10.0/27

Step 1: Find Subnet Mask:

- **/27 = 255.255.255.224**.

Step 2: Find Number of IPs:

- **32 - 27 = 5 bits for hosts**.
- **2⁵ = 32 IPs** (Total).

Step 3: Usable IPs:

- **32 - 2 = 30 usable IPs**.

Step 4: Subnet Ranges:

- **192.168.10.0 - 192.168.10.31**
- **192.168.10.32 - 192.168.10.63**

- **192.168.10.64 - 192.168.10.95**
-

Summary

- **Subnetting** breaks big networks into smaller groups for efficiency.
- **CIDR** simplifies subnet masks (e.g., /24 means 255.255.255.0).
- **Private IPs** are for local networks, and **Public IPs** are for the internet.
- Always calculate **usable IPs** by subtracting **2** from the total.

3. Private vs Public IP Addresses

Private IP Address

- Used **inside homes, offices, and private networks**.
- Not visible on the internet.
- Examples:
 192.168.0.1 (used in Wi-Fi routers)
 10.0.0.1 (used in large networks)

Why use private IPs?

- Safe from hackers because they are **not directly exposed to the internet**.
- Saves IP addresses since private addresses can be reused in different networks.

Public IP Address

- Used **on the internet** to identify devices globally.
 - Examples: **8.8.8.8** (Google's DNS server).
 - These are **unique** and assigned by **ISPs (Internet Service Providers)**.
-

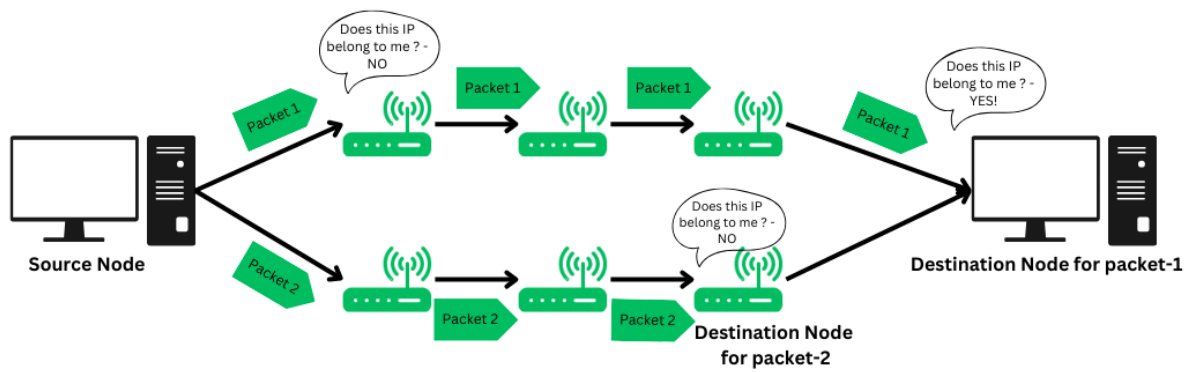
4. Routing

What is Routing?

Routing is the process of finding the **best path** for data to travel from one network to another.

Example:

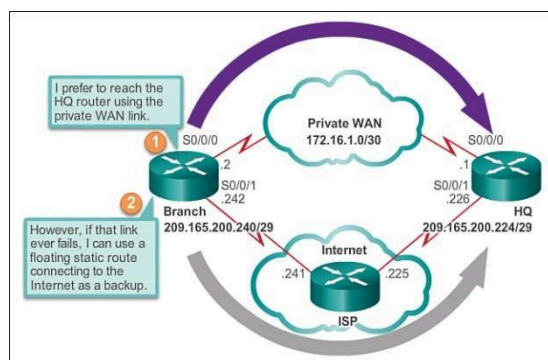
If you're sending a message to someone in another city, it goes through **post offices (routers)**, which decide the **shortest and fastest route** for delivery.



Static vs Dynamic Routing

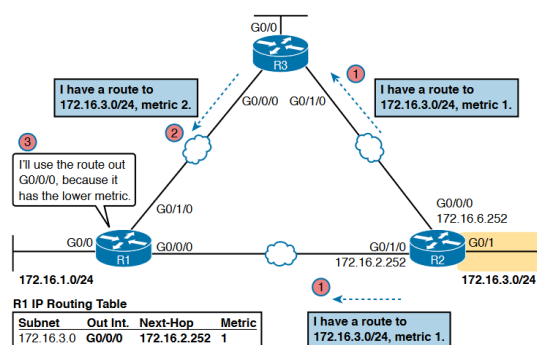
1. Static Routing:

- The routes are **manually set** by network administrators.
- It's **simple but not flexible** (if the network changes, routes must be updated manually).
- Used in **small networks**.

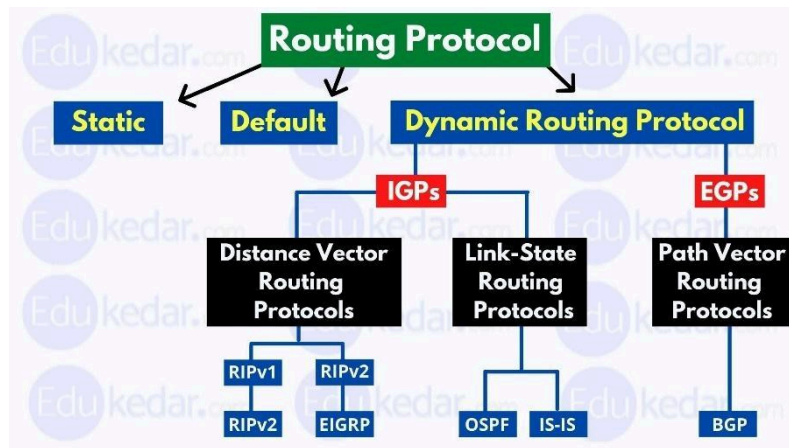


2. Dynamic Routing:

- The routes are **automatically updated** using **routing protocols**.
- **Flexible** and used in **large networks**.
- Routers exchange information to find the **best path**.



5. Routing Protocols



These are rules that routers follow to **find the best path** for data. Let's look at the most common ones:

The Various Routing Protocols					
Features	RIP v1	RIP v2	IGRP	OSPF	EIGRP
Classful / Classless	Classful	Classless	Classful	Classless	Classless
Metric	Hop	Hop	Composite (bw and delay)	Cost	Composite (bw and delay)
Periodic Advertisement	30 seconds	30 seconds	90 seconds	100,000/BW none	30 seconds
Advertising Address	255.255.255.255 (broadcast)	224.0.0.9 (multicast)	255.255.255.255 (broadcast)	224.0.0.5 224.0.0.6 (multicast)	224.0.0.10 (multicast)
Administrative Cost	120	120	100	110	Internal: 90 External: 170
Category	Distance Vector	Distance Vector	Distance Vector	Link State	Hybrid

1. RIP (Routing Information Protocol)

- Simple and easy to set up.
- Suitable for **small networks**.
- Limit: It only supports networks with **15 hops (steps)**, so it's not ideal for large networks.

2. OSPF (Open Shortest Path First)

- Used in **large networks**.
- Finds the **shortest path** using a map of the network.
- **Faster and smarter** than RIP because it updates only the changes, not the entire network.

3. EIGRP (Enhanced Interior Gateway Routing Protocol)

- Used mainly in **Cisco routers**.
- Faster than OSPF and can adapt quickly if a link fails.

- Supports both IPv4 and IPv6.

4. BGP (Border Gateway Protocol)

- Used on the **Internet** to connect **different networks** globally.
 - It's like a **postal service for the whole world** that manages routes between different regions and companies.
-

6. Default Gateway and Next-Hop Concepts

Default Gateway

- It's like the **door to the outside world** for your network.
- If your computer doesn't know where to send data, it sends it to the **default gateway** (usually a router).
- Example: Your Wi-Fi router is your default gateway to the internet.

Next-Hop

- It's the **next step** in the path that data takes to reach its destination.
 - Think of it as a **pit stop** before reaching the final destination.
 - Example:
If you're traveling from City A to City C via City B, City B is your **next-hop**.
-

Summary

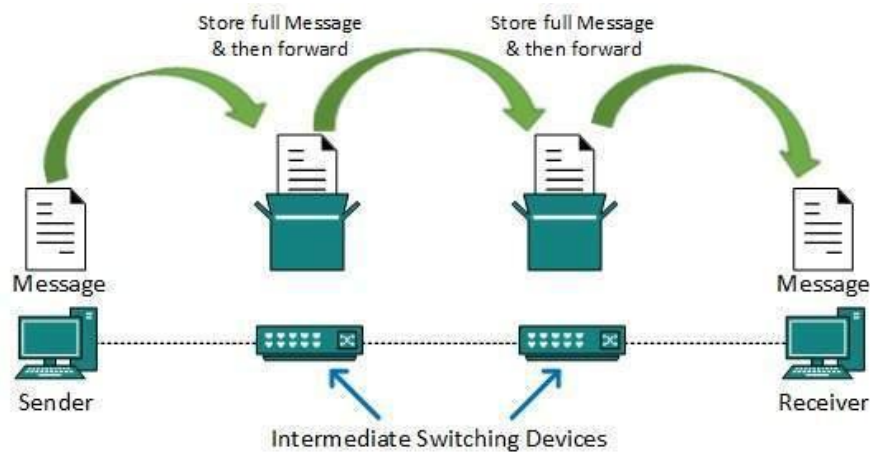
1. **IP Address** – Identifies devices on a network. IPv4 (older, shorter) and IPv6 (newer, longer) are used.
2. **Subnetting and CIDR** – Divide networks into smaller groups to organize and save addresses.
3. **Private vs Public IP** – Private IPs are used in local networks, and public IPs are used on the internet.
4. **Routing** – Finds the best path for data to travel. Static routing is manual, and dynamic routing is automatic.
5. **Routing Protocols** – RIP (small networks), OSPF (large networks), EIGRP (fast updates), BGP (internet routing).
6. **Default Gateway and Next-Hop** – Default gateway is the starting point for data leaving the network, and next-hop is the next stop in its journey.

1. What is Switching?

Switching is like a **traffic controller** for data in a network. It helps data packets (like letters) find their way from one device (computer) to another in the **local area network (LAN)**.

Imagine a **post office**:

- It takes letters from one house and delivers them to the right address within the same area.
- A **network switch** does the same thing but with **data** instead of letters.



2. VLANs (Virtual LANs)

What is a VLAN?

A **VLAN (Virtual LAN)** is like dividing one big room into **separate sections** using **partitions**.

Example:

- Imagine an office where employees work in different departments—Sales, HR, and IT.
- Instead of giving each department a separate physical network, we use **VLANs** to create **virtual networks**.
- Each department's devices can only talk to each other, even though they are physically connected to the **same switch**.

Why use VLANs?

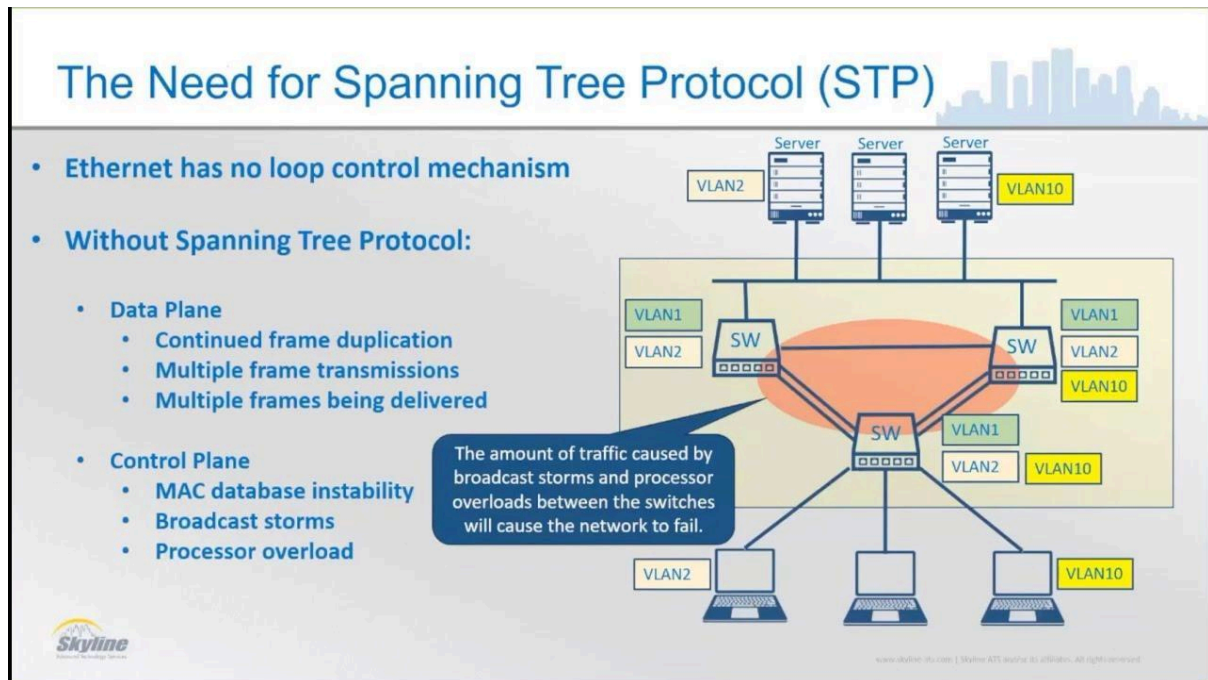
1. **Security** – Devices in one VLAN can't access devices in another VLAN unless allowed.
2. **Organization** – Easier to manage groups of devices.
3. **Efficiency** – Reduces unnecessary data traffic between groups.
4. **Cost-effective** – No need for separate hardware for each group.

Example:

- VLAN 10 – For Sales
- VLAN 20 – For HR
- VLAN 30 – For IT

Even though they use the same switch, these groups act like **separate networks**.

3. STP (Spanning Tree Protocol)



What is STP?

STP (Spanning Tree Protocol) prevents **loops** in a network.

What is a loop?

- Imagine you have **two roads** connecting your home to the market.
- If you go in circles between those two roads, you'll **never reach the market**.
- Similarly, if a network has **multiple paths**, data can keep going in circles (loops).

STP stops this problem by:

1. Detecting **loops**.
2. **Blocking extra paths** until they are needed (backup).

Why is STP important?

- Prevents **network crashes** caused by loops.
 - Ensures there's always a **backup path** if the main path fails.
-

4. MAC Address and ARP Table

What is a MAC Address?

A **MAC Address (Media Access Control)** is like a **fingerprint** for a device. It is **unique** and cannot change.

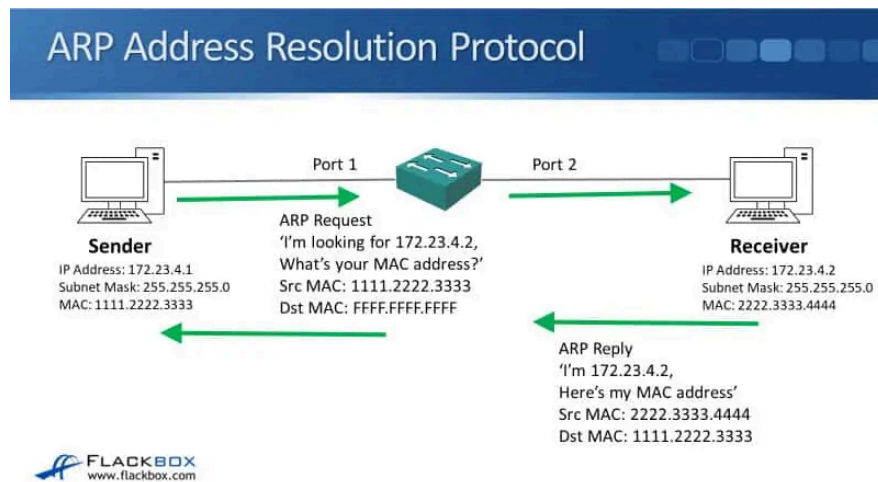
Example:

- Your **Wi-Fi card** has a MAC address like **00:1A:2B:3C:4D:5E**.

Purpose:

- Used to identify devices inside a **local network**.
- Switches use MAC addresses to deliver data to the **correct device**.

What is an ARP Table?



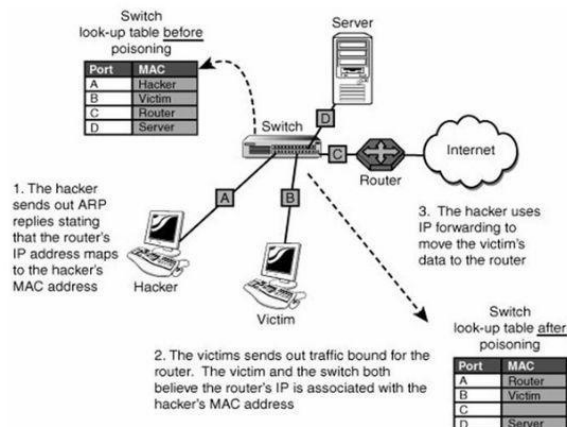
The **ARP (Address Resolution Protocol)** table is like a **contact list** for the network.

- It maps **IP addresses** (logical address) to **MAC addresses** (physical address).

Example:

- Your computer knows the IP address **192.168.1.5**, but it needs the **MAC address** to send data.
- ARP helps find this MAC address and stores it in the **ARP table** for future use.

The ARP Poisoning Process



5. DHCP (Dynamic Host Configuration Protocol)

What is DHCP?

DHCP automatically **assigns IP addresses** to devices on a network.

Imagine walking into a hotel:

- The receptionist gives you a **room number** (IP address).
- When you leave, the room is given to someone else.

DHCP does the same by assigning an IP address to your device **only while you're connected**.

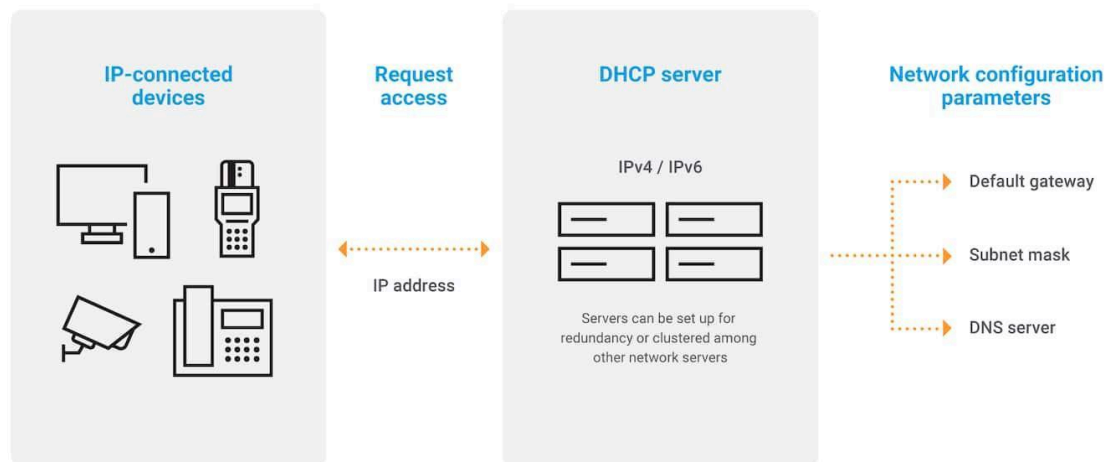
Purpose of DHCP:

1. Saves time – No need to assign IP addresses manually.
2. Prevents conflicts – Ensures two devices don't get the **same IP address**.
3. Dynamic – Frees up IP addresses when devices disconnect.

DHCP Process:

1. **Discover** – The device asks, "Can I have an IP address?"
2. **Offer** – DHCP server replies, "Sure! Here's one for you."
3. **Request** – The device says, "I'll take this IP address."
4. **Acknowledge** – DHCP server confirms, "It's yours now."

How does DHCP work?



6. DNS (Domain Name System)

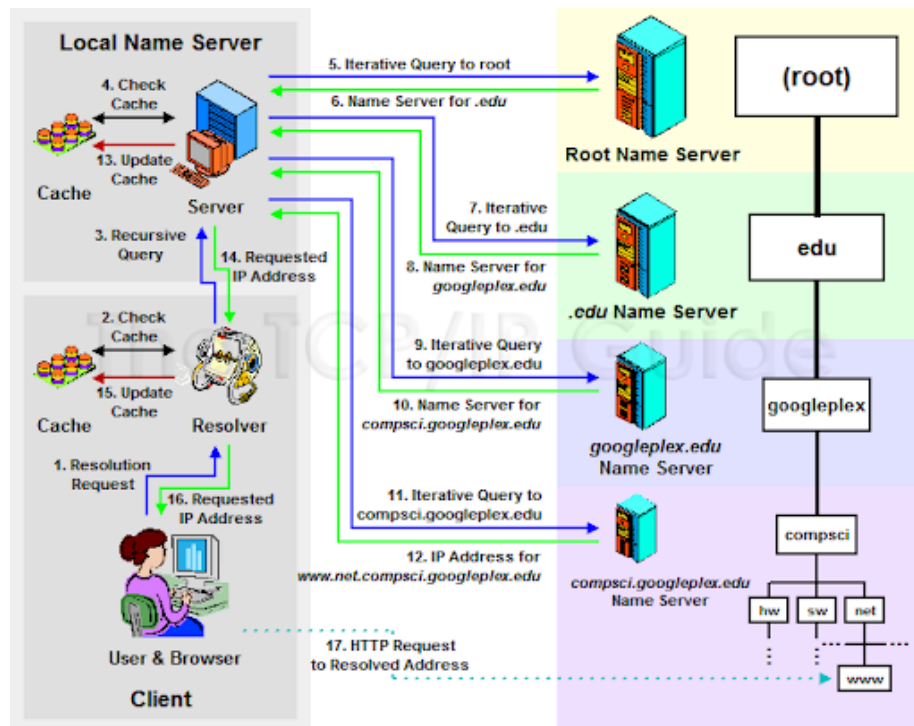
What is DNS?

DNS is like the **phonebook of the internet**.

- It converts **website names (domain names)** into **IP addresses** that computers understand.

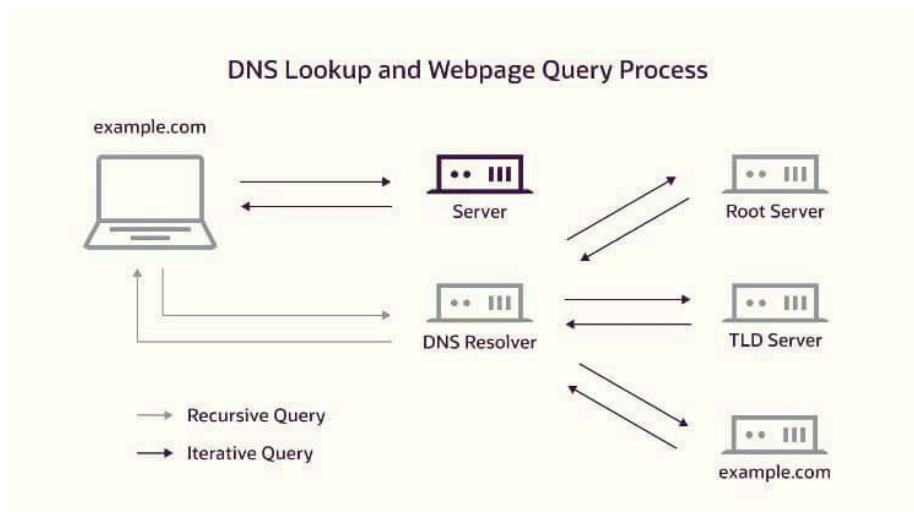
Example:

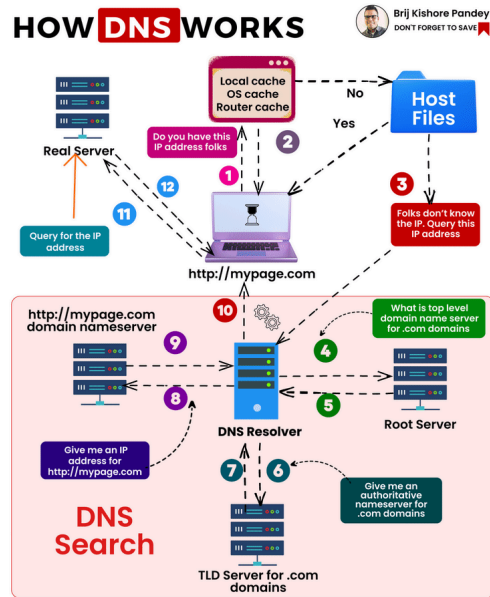
- You type www.google.com in your browser.
- DNS translates it to **142.250.182.14** (IP address).
- Your computer connects to **142.250.182.14** to load the website.



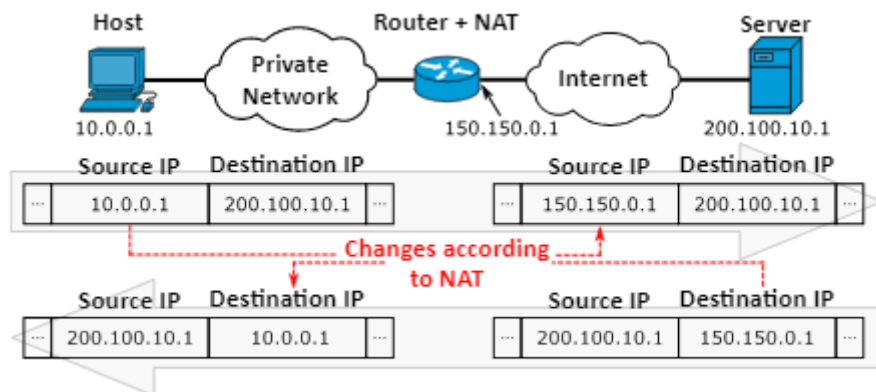
Why is DNS important?

- Humans use **names** (like google.com), but computers use **numbers** (IP addresses).
- DNS makes the internet **user-friendly**.





7. NAT (Network Address Translation)



What is NAT?

NAT acts like a **translator** between **private** and **public** IP addresses.

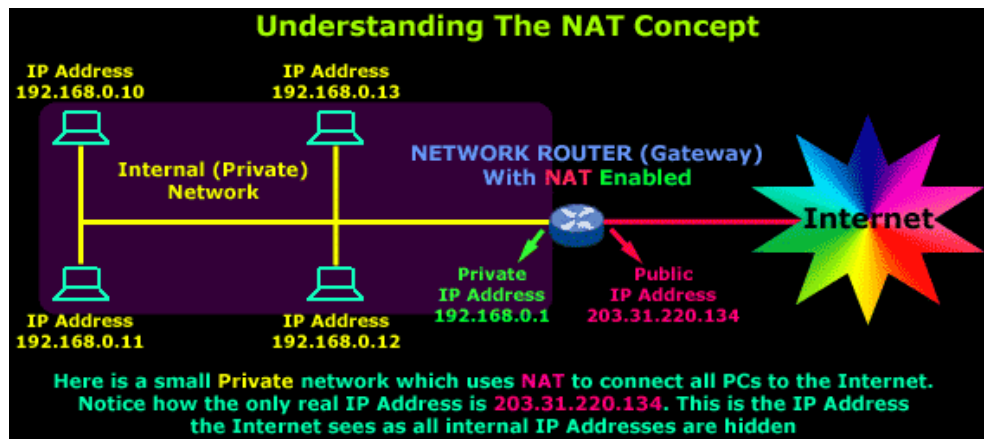
Example:

- Your home has **many devices** (phone, laptop, TV) with **private** IPs like **192.168.1.10**.
- But the **internet** only understands **public** IPs.
- NAT allows all your devices to share **one public IP** for internet access.

Types of NAT:

1. **Static NAT** – One private IP is permanently mapped to one public IP.
Example: Used for **web servers**.

2. **Dynamic NAT** – Private IPs are mapped to **any available public IP** from a pool.
Example: Used for **temporary connections**.
3. **PAT (Port Address Translation)** – Many private IPs share **one public IP** by assigning different **ports**.
Example: Most commonly used in **home networks**.

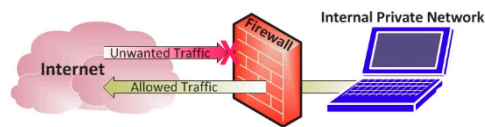


Summary

1. **Switching** – Moves data between devices in a network, like a traffic controller.
 2. **VLANs** – Create **virtual groups** inside a network for better security and organization.
 3. **STP** – Prevents **loops** in networks by allowing **backup paths**.
 4. **MAC and ARP** – MAC addresses are like **fingerprints**; ARP helps map IP to MAC addresses.
 5. **DHCP** – Automatically assigns IP addresses, making networking easier.
 6. **DNS** – Translates **website names** into IP addresses, making the internet simple for humans.
 7. **NAT** – Allows devices with **private IPs** to access the **internet** using a single **public IP**.
-

1. What is a Firewall?

Computer Firewalls



the first line of defense in network security

Think of a **firewall** like a **security guard** at the entrance of a building (which represents your computer or network). The guard's job is to **check everyone who tries to enter or exit** the building.

- In computer terms, a **firewall** is a **security system** that **controls traffic** coming into and going out of your computer or network.
- It does this by **examining data** packets (like little bundles of information) and deciding whether they should be allowed or blocked based on **predefined security rules**.

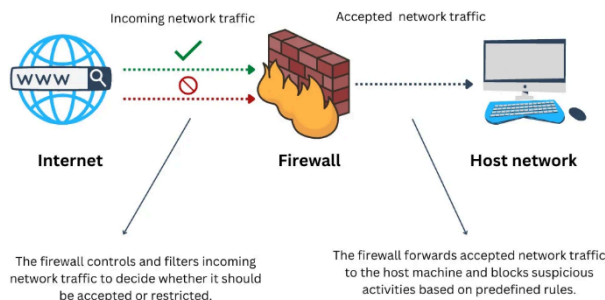
In simple terms: A firewall is a **barrier** that keeps **bad things (like hackers, malware, etc.) out**, while allowing **safe things** to come in and go out.

2. Why Do We Need Firewalls?

Firewalls help to:

1. **Block unauthorized access** – They stop hackers or malicious software from getting into your network or device.
2. **Control network traffic** – They decide which data can enter or leave the network.
3. **Protect sensitive information** – They prevent the leakage of private data from your computer to the outside world.

3. How Does a Firewall Work?



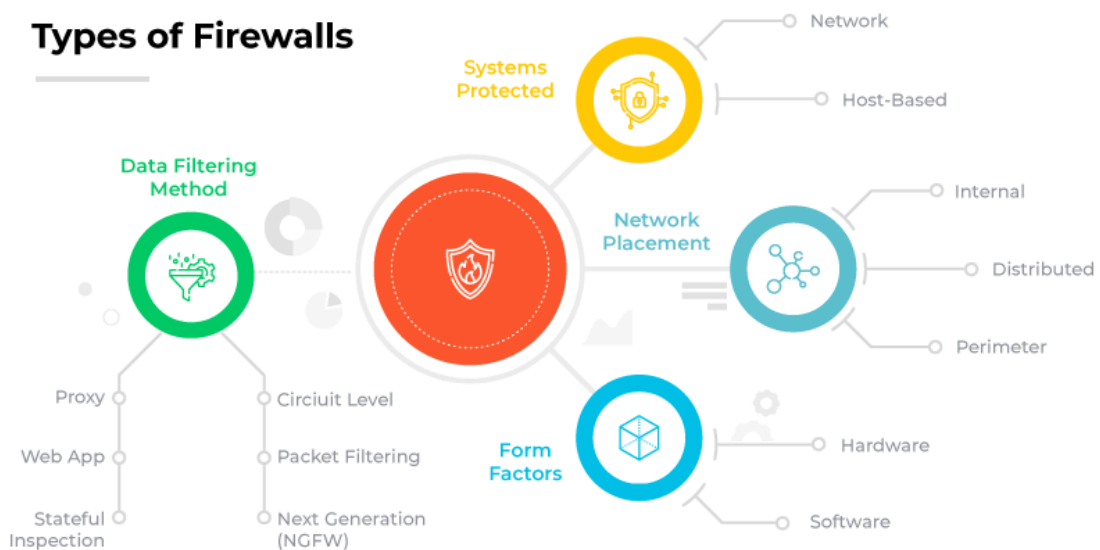
A firewall works by looking at the **data packets** traveling across your network. These packets contain:

- **Source IP address:** Where the data is coming from.
- **Destination IP address:** Where the data is going.
- **Ports:** The doors the data wants to pass through (just like a building has many doors).
- **Protocol:** The type of communication (like HTTP for websites, FTP for file transfers).

When a packet reaches the firewall, it checks the **rules** to decide:

- **Allow:** If the packet is from a trusted source and it's safe.
- **Block:** If the packet comes from a suspicious source or is trying to do something harmful.

4. Types of Firewalls



a. Stateless Firewall

A **stateless firewall** is like a **basic security guard** that checks every packet independently, without keeping track of what has happened before.

- It makes decisions based only on the **current packet's information** (like source, destination, and port).
- **Example:** If a packet is coming from an untrusted source, it will be blocked right away.

b. Stateful Firewall

A **stateful firewall** is more **intelligent**. It remembers the **history** of the connections.

- For example, if you open a website, the firewall **remembers** that you've made a request to that website. When a response comes back, the firewall allows it because it knows it's a part of your earlier request.
- It tracks the **state of connections**, such as whether they are new or established, which helps make smarter decisions.

Example:

- You start a video call with someone (this is the **state** of the connection).
- The firewall **remembers** that you've already made this connection and will allow the response from the other side to come through.

- 🏠 **Network firewalls** protect the entire network.
- 🏠 **Host-based firewalls** protect individual devices.
- 🏠 **Hardware firewalls** are physical devices placed at the network's entry point.
- 🏠 **Software firewalls** are programs installed on a device to monitor traffic.
- 🏠 **Internal firewalls** control communication within different parts of the network.
- 🏠 **Distributed firewalls** involve multiple firewalls working together across the network.
- 🏠 **Perimeter firewalls** are the first line of defense at the network boundary.
- 🏠 **Next-generation firewalls (NGFW)** offer advanced protection beyond basic filtering.
- 🏠 **Packet filtering firewalls** examine data packets to allow or block them.
- 🏠 **Circuit-level gateways** verify the establishment of connections.
- 🏠 **Web application firewalls (WAF)** protect websites from web-based attacks.
- 🏠 **Proxy firewalls** act as intermediaries between users and websites.
- 🏠 **Stateful inspection firewalls** track active connections for context-based decisions.

5. Common Firewall Rules

Firewall rules are like instructions that tell the firewall how to handle data. These rules are based on **security policies** and can be customized depending on what is considered safe or harmful.

Here are some common types of firewall rules:

1. Allow or Block Based on IP Address

- **Example:** You can create a rule to **block all traffic** from a certain IP address if you suspect that the source is malicious.

2. Allow or Block Based on Port

- **Ports** are like doors on your computer. Different services (like a web server or email service) use different doors (ports).
 - **HTTP** traffic (web browsing) uses port **80**.
 - **HTTPS** traffic (secure web browsing) uses port **443**.
 - **FTP** (File Transfer) uses port **21**.
- You can block or allow traffic coming through specific ports. For example, you can block port **80** to stop people from accessing a website.

3. Allow or Block Based on Protocol

- A **protocol** is the **method of communication** (like how people speak different languages to communicate). Some common protocols are **HTTP**, **FTP**, **SMTP** (for email), and **DNS** (for domain name system lookups).
- You can set rules to allow or block certain protocols. For example, you could block **FTP** if you're not using it on your network.

4. Allow or Block Based on Time

- You can create rules to **allow or block traffic only during certain hours**. For example, you might allow remote access to your network only during business hours and block it after hours for security.

6. Common Firewall Configurations

1. Default Deny (Most Common)

- This means the firewall **blocks everything by default**, and only allows traffic that is explicitly permitted by the rules.

2. Default Allow

- This means the firewall **allows everything by default**, and blocks only what is explicitly not allowed (usually considered **less secure**).

7. Additional Features of Firewalls

1. Network Address Translation (NAT)

- A firewall can **hide the internal addresses** of your network using **NAT**. It uses one public IP address for the outside world and translates requests into private addresses for internal communication.
- This helps in hiding your internal network from external attacks.

2. VPN Support (Virtual Private Network)

- Firewalls can also support **VPNs**, which are like **encrypted tunnels**. They allow users to securely access the network remotely, even from outside, as if they were directly connected to the internal network.
-

8. Where Are Firewalls Used?

Firewalls can be used in various places:

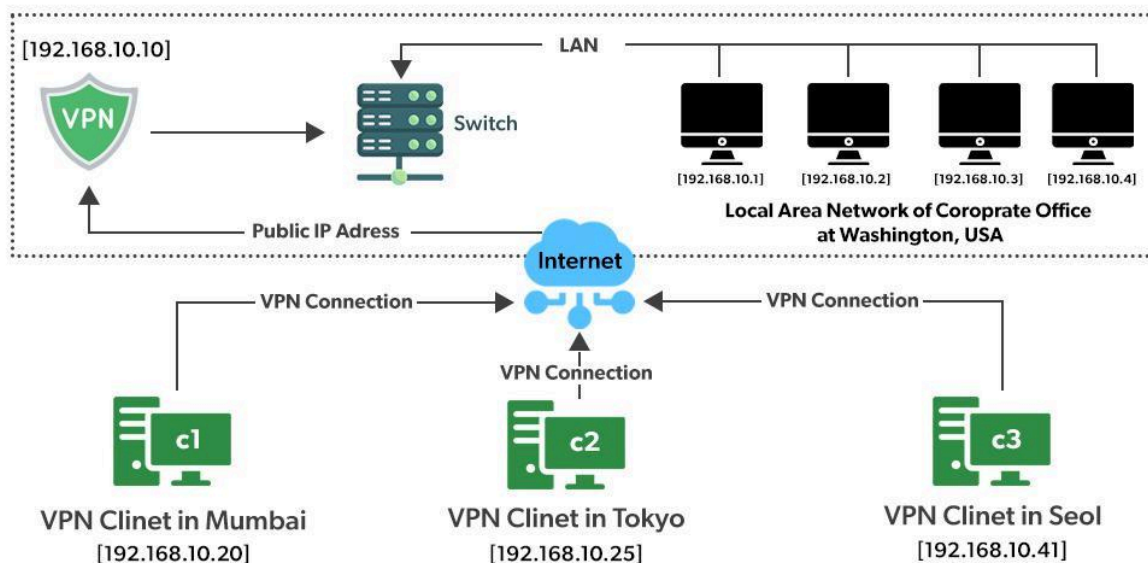
1. **On your personal computer** – To block unwanted traffic from the internet and protect against malware or hacking attempts.
 2. **At the network perimeter** – Protecting an entire network (like a company network) from external threats.
 3. **Cloud-based firewalls** – Protecting cloud resources, such as data stored in the cloud, from unauthorized access.
-

9. Conclusion:

Firewalls are essential for network security, acting as **gatekeepers** that protect your devices and data from harmful access.

- **Stateful firewalls** are more advanced and can keep track of communication, while **stateless firewalls** are simpler and check each packet independently.
- Firewalls use **rules** to decide what is safe and what should be blocked.

1. VPN (Virtual Private Network)



What is a VPN?

Imagine you are sitting in a **coffee shop** and using the public Wi-Fi. Anyone on that Wi-Fi could potentially **spy** on what you're doing, like stealing your passwords or credit card details.

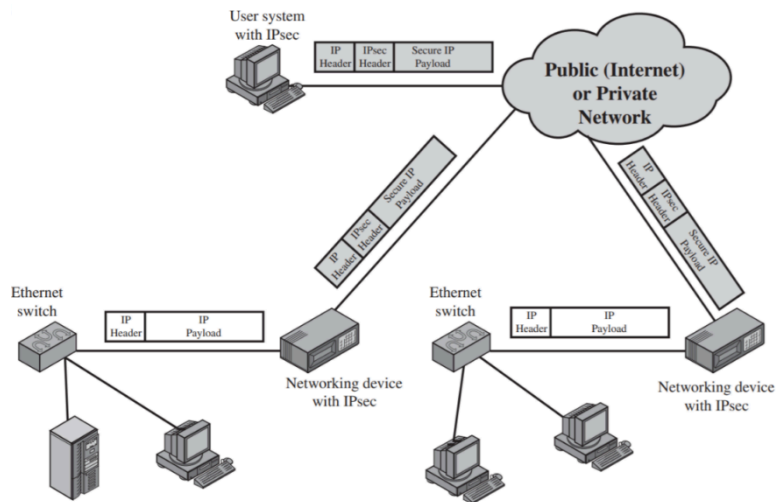
A **VPN (Virtual Private Network)** acts like a **secure tunnel** between your device (laptop, phone) and the internet. It **hides your data** from others by **encrypting it**—turning it into unreadable code that only you and the VPN server can understand.

It's like you're **sending secret messages** in a locked box that no one can open while traveling through a dangerous area.

Why use a VPN?

- **Privacy:** Hides your online activities from hackers, internet providers, or governments.
- **Security:** Protects sensitive data like passwords and bank details when using public Wi-Fi.
- **Access Restricted Content:** Allows you to access websites or streaming services that may be blocked in your region.
- **Remote Access:** Employees can securely connect to their company's network from home or while traveling.

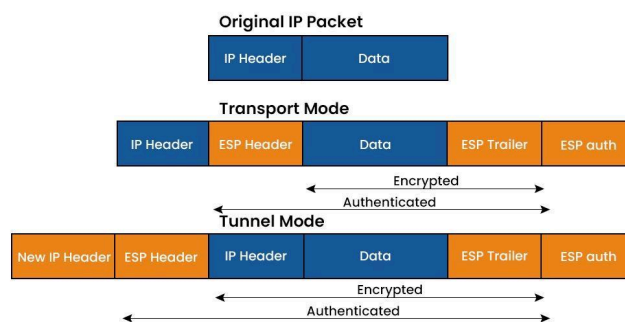
2. IPSec (Internet Protocol Security)



What is IPsec?

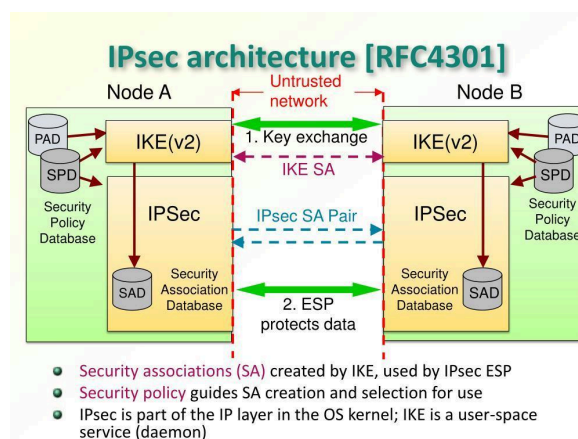
IPsec is a **set of rules and technologies** that make the data traveling over a network **safe and secure**.

It works like adding **locks and seals** to the boxes (data packets) you send over the internet.



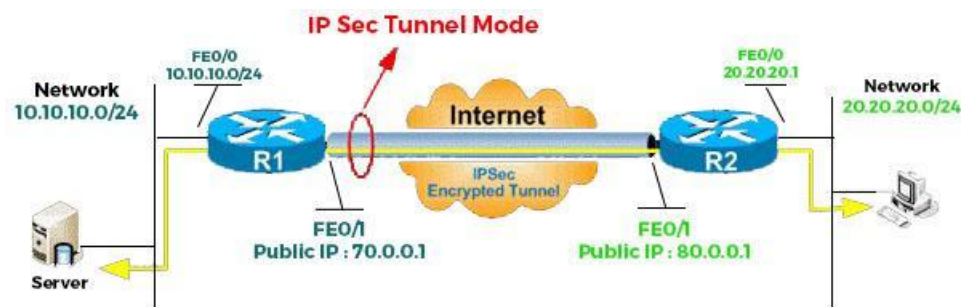
How does it secure data?

1. **Encryption:** It scrambles data so that even if someone intercepts it, they can't understand it.
2. **Authentication:** It verifies that the data comes from a trusted source and hasn't been tampered with.
3. **Integrity Checks:** Ensures the data isn't altered while traveling.

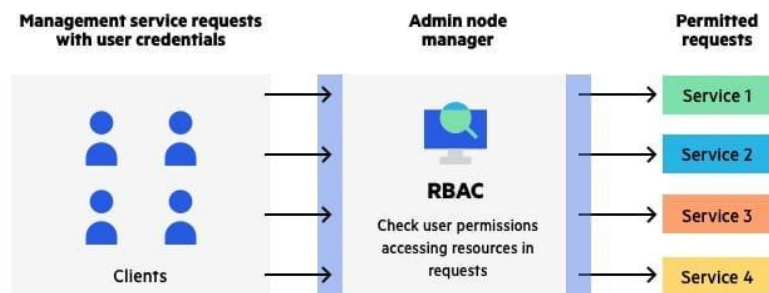


Where is IPSec used?

- **VPNs** often use IPSec to secure the data between your device and the VPN server.
- **Remote work connections** to safely access company resources.



3. ACL (Access Control List)



What is an ACL?

An **Access Control List (ACL)** is like a **security guard** standing at the **gate** of a network. It **checks ID cards** before letting anyone in or out.

- It decides **who is allowed** to access the network and **what they're allowed to do**.
- It can block or allow traffic based on rules like **IP address**, **protocol**, or **port numbers**.

How does it work?

- **Example:** Imagine you have a list at the entrance of a club, and only people whose names are on the list can enter. Similarly, ACLs allow or block traffic based on pre-defined rules.

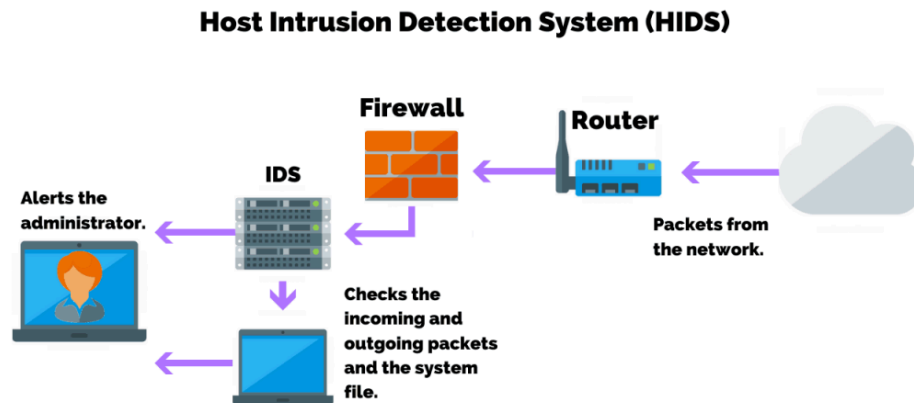
Types of ACLs:

1. **Standard ACL:** Filters based on **source IP address** (e.g., block all traffic from 192.168.1.1).
2. **Extended ACL:** Filters based on **source, destination, port numbers, and protocol** (e.g., allow only HTTP traffic from a specific IP).

4. IDS (Intrusion Detection System)

What is IDS?

An **Intrusion Detection System (IDS)** is like a **security camera** that **watches for suspicious activity** in your network and **alerts you** if something looks wrong.



It **does not block the attack**; it only **notifies** you when something unusual happens.

How does it work?

- It **monitors network traffic** and **compares it** to a list of known threats (like signatures of viruses or hacking patterns).
- If it sees something suspicious, it **raises an alarm** so that network administrators can investigate and stop the attack.

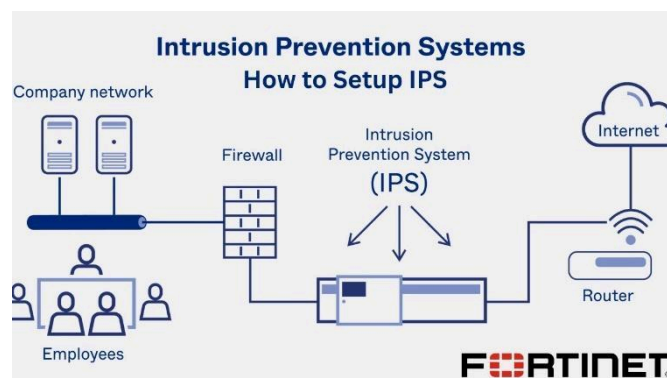
Types of IDS:

1. **Network-based IDS (NIDS):** Watches traffic flowing through the network.
2. **Host-based IDS (HIDS):** Monitors activity on a specific device, like a server or computer.

5. IPS (Intrusion Prevention System)

What is IPS?

An **Intrusion Prevention System (IPS)** is like an **active security guard** who not only **detects an attack** but also **stops it immediately**.



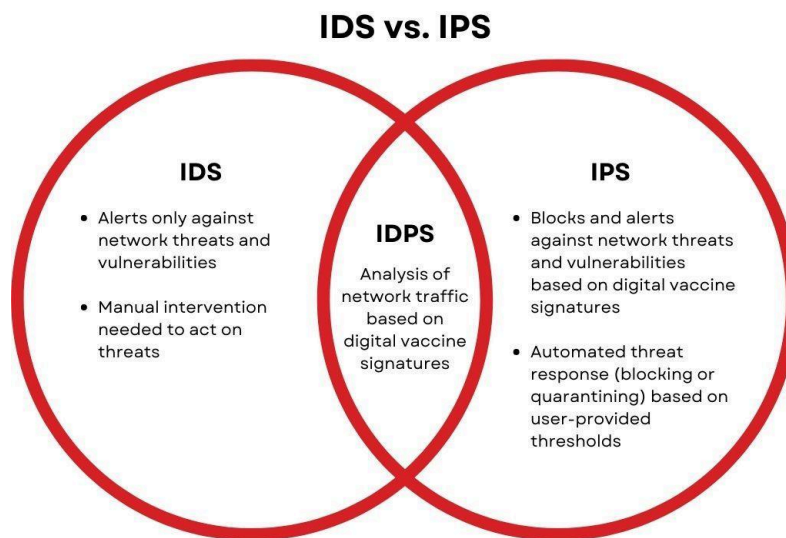
It can **block malicious traffic** before it causes any harm.

How does it work?

- It analyses incoming traffic in **real-time** and **blocks suspicious activities** automatically.
- Unlike IDS, which only **alerts**, IPS **takes action** to protect the network.

Example:

- If a hacker tries to break into your network by guessing passwords repeatedly, an **IPS can block** their IP address to prevent further attempts.



1. Command Line Utilities

a) ping

What is ping?

- Imagine you're **calling your friend** to check if they're available and how long it takes for them to answer.
- **Ping** works the same way—it's used to check if a device (computer, server, or website) is **reachable** and how long it takes for data to **travel back and forth**.

Command Example:

```
ping google.com
```

Output:

```
Reply from 142.250.185.14: bytes=32 time=24ms TTL=117
```

How to read it?

- **Reply from 142.250.185.14** → The device is reachable, and this is its **IP address**.

- **time=24ms** → It took **24 milliseconds** for the data to go to Google and come back.
- **TTL=117** → Time-to-Live, how many hops (routers) the packet can pass through before being discarded.

Why use it?

- To check if a website or server is **working**.
 - To see if your **internet connection** is working properly.
-

b) traceroute (Linux/Mac) / tracert (Windows)

What is traceroute?

- Imagine you're **tracking a package** to see **all the stops** it makes before reaching its destination.
- **Traceroute** shows the **path data takes** through routers to reach a website or server.

Command Example:

tracert google.com (Windows)

traceroute google.com (Linux/Mac)

Output:

```
1 192.168.1.1 1 ms
2 10.0.0.1 3 ms
3 93.184.216.34 24 ms
```

How to read it?

- Each line shows a **hop (router)** the data passed through.
- The **IP addresses** and **time (ms)** tell you how fast each hop responded.

Why use it?

- To **troubleshoot slow networks** by finding out where the delay happens.
 - To **identify network failures** between your device and a server.
-

c) ipconfig (Windows) / ifconfig (Linux/Mac)

What is ipconfig/ifconfig?

- Think of this as asking, **"What's my address?"**
- **ipconfig** and **ifconfig** show your **IP address, gateway, subnet mask**, and other network settings.

Command Example (Windows):

ipconfig

Command Example (Linux/Mac):

ifconfig

Output:

IPv4 Address: 192.168.1.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.1.1

How to read it?

- **IPv4 Address** → Your device's address on the network.
- **Subnet Mask** → Helps define your network range.
- **Default Gateway** → The router that connects you to the internet.

Why use it?

- To find your **IP address** and check network settings.
 - To **release or renew** your IP address when facing network problems.
-

d) nslookup

What is nslookup?

- Imagine asking a **phonebook** for your friend's phone number.
- **nslookup** finds the **IP address** of a **domain name** (like google.com).

Command Example:

nslookup google.com

Output:

Name: google.com

Address: 142.250.185.14

How to read it?

- **Name** → Domain name you asked for.
- **Address** → The IP address of that domain.

Why use it?

- To **find the IP address** of a website.
 - To **troubleshoot DNS problems** (when websites don't load).
-

e) netstat

What is netstat?

- Think of it as **checking who is talking to you** and **what ports they're using**.
- **netstat** shows **active connections** on your computer, like web browsers or apps connecting to the internet.

Command Example:

netstat -an

Output:

Copy code

```
TCP 192.168.1.10:5678 93.184.216.34:443 ESTABLISHED
```

How to read it?

- **TCP/UDP** → Protocol used for communication.
- **Local Address** → Your computer's IP and port number.
- **Foreign Address** → The remote computer it's connected to.
- **State** → Shows if the connection is **ESTABLISHED** (active) or **CLOSED**.

Why use it?

- To **check open ports** and **active connections**.
 - To **detect suspicious connections** or malware.
-

2. Wireshark Basics for Packet Analysis

What is Wireshark?

- Imagine **listening to all the conversations** happening in a crowded room.
- **Wireshark** is a tool that **captures and analyzes network traffic**, showing all the **data packets** traveling in and out of your device.

What is a Packet?

A **packet** is like a **letter** containing data sent across the network. Each packet has:

1. **Header** → Tells where it's coming from and going to.
2. **Payload** → Contains the actual data being sent.

Why use Wireshark?

- **Troubleshooting Problems:** Find out if data is being sent properly.
- **Monitoring Traffic:** See what kind of data is flowing in and out.
- **Detecting Threats:** Check for hacking attempts or suspicious activities.

Steps in Wireshark:

1. **Start Capturing Traffic:** Select your network interface (Wi-Fi or Ethernet).
2. **Filter Traffic:** Use filters like **http** to only see web traffic.
3. **Analyze Packets:** Click on a packet to see its details—like source and destination addresses.

Example Filters:

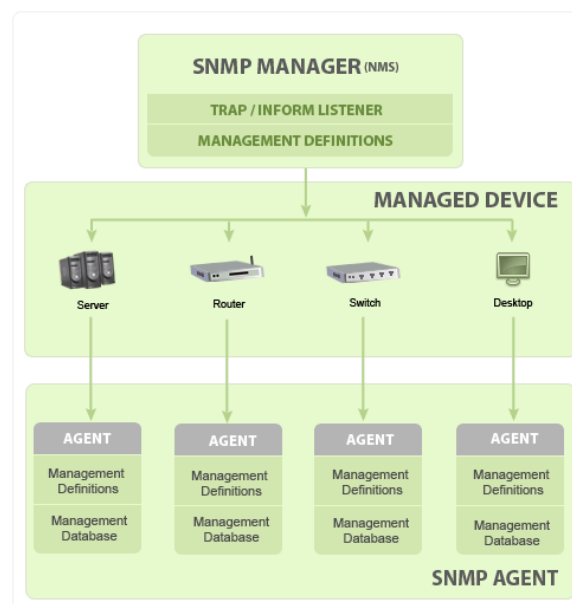
- **ip.addr == 192.168.1.1** → Show traffic to/from a specific IP.
- **tcp.port == 80** → Show web traffic only.

1. What is Network Monitoring?

Imagine you're managing a **large network** (computers, servers, routers, switches, etc.).

- **Problem:** You need to make sure **everything is working properly**, and if something breaks, you need to **fix it quickly**.
- **Solution:** Use **Network Monitoring Tools** to **track performance**, **detect issues**, and **alert you** if there's a problem.

2. SNMP (Simple Network Management Protocol)



What is SNMP?

- **SNMP** is a **language** that devices (like routers, switches, and servers) use to **talk to each other** and **share information**.
- It's used to **monitor and manage devices** in a network.

How does SNMP work?

Think of SNMP like a **doctor and patient**:

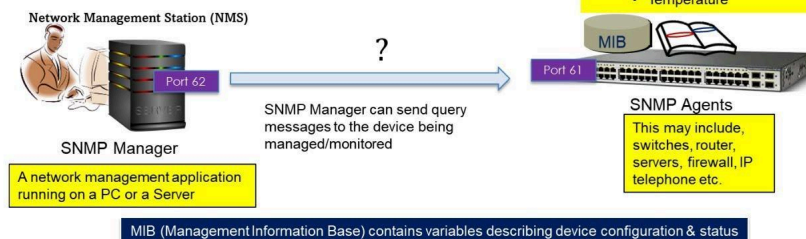
- The **doctor (manager)** asks the **patient (device)** about their **health (data)**.
- The **patient** gives back **information**, such as CPU usage, memory status, or errors.

SNMP Simple Network Management Protocol

An application layer protocol used for management and monitoring of devices on network

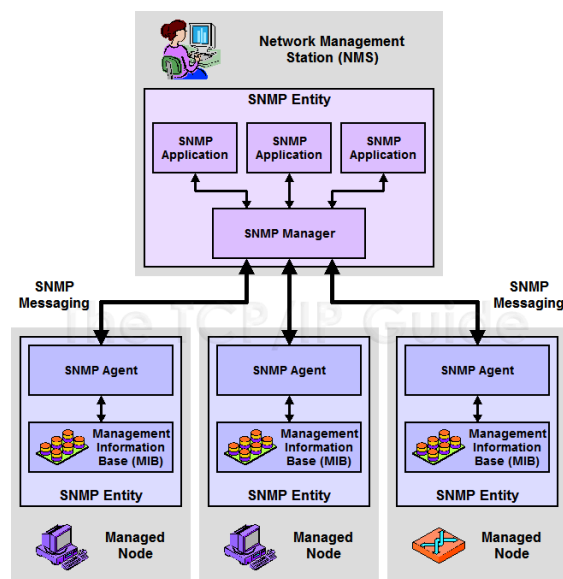
SNMP is used to monitor network performance and to troubleshoot issues

SNMP provides a framework consisting of the following components



MIB (Management Information Base) contains variables describing device configuration & status

Key Components of SNMP



1. SNMP Manager (Monitoring Tool):

- The **central system** that asks for and collects data from devices.
- Example: **Nagios, SolarWinds** (monitoring tools).

2. SNMP Agents (Devices):

- Installed on each **device** (router, switch, printer, etc.) to **respond to requests** from the manager.

3. MIB (Management Information Base):

- A **dictionary** that defines what **data** the device can share.
- Example: CPU usage, memory, temperature, network errors.

4. **OID (Object Identifier):**

- Each piece of information has a **unique ID number** for identification.
- Example: **1.3.6.1.2.1.1.3** might represent the **uptime** of a device.

5. **SNMP Traps:**

- **Alerts** sent automatically by devices when something goes wrong.
- Example: If a router fails, it immediately **sends a trap** to the SNMP Manager.

How SNMP Collects Data:

1. **Polling:** The SNMP Manager **asks devices** for their status regularly.
2. **Traps:** Devices **send alerts** when there's an issue without waiting for a request.

Real-Life Example:

- You monitor **100 devices** in your network.
 - One **router overheats** and **sends a trap** to your SNMP Manager.
 - The Manager **alerts you immediately** so you can fix it.
-

3. Tools for Network Monitoring

a) Nagios (Open Source Tool)

What is Nagios?

- **Nagios** is like a **security guard** that **watches your network 24/7**.
- It checks if devices are **up and running** and **alerts you** if something is wrong.

Key Features:

1. **Monitoring Devices:** Checks **servers, routers, and applications** for performance and errors.
2. **Alerts:** Sends **email or SMS alerts** when issues occur.
3. **Customizable:** Allows adding **plugins** for additional monitoring features.
4. **Historical Data:** Tracks performance over time to **analyze trends**.

Example Use Case:

- Monitor a **web server** to check if it's **down** or **slow**.
 - Alert the team if **CPU usage exceeds 80%**.
-

b) SolarWinds (Commercial Tool)

What is SolarWinds?

- **SolarWinds** is a **professional monitoring tool** used by **large companies**.

- It provides **detailed reports** and **automated troubleshooting tools**.

Key Features:

1. **Real-Time Monitoring:** Tracks **bandwidth usage**, **CPU load**, and **device health** in **real-time**.
2. **Network Maps:** Shows a **visual diagram** of your network to help you understand the connections.
3. **Reporting and Alerts:** Generates **reports** and sends **alerts** for performance issues.
4. **Traffic Analysis:** Tracks **who is using the most bandwidth** and **what they're doing**.

Example Use Case:

- Monitor the **entire data center** to see if any **router or server fails** and identify **network slowdowns**.
-

4. Why Use Monitoring Tools?

Benefits of Monitoring Tools:

1. **Detect Problems Quickly:**
 - If a server goes down or a router stops working, you get an **instant alert**.
 2. **Plan for Growth:**
 - Analyze traffic and performance to **upgrade hardware** before it gets overloaded.
 3. **Improve Security:**
 - Detect **unauthorized devices** or **suspicious activity** in the network.
 4. **Save Time and Money:**
 - Prevent **downtime** and **network failures**, saving repair costs.
-

5. Example Scenarios for Monitoring Tools

Scenario 1: Server Down Issue

- **Problem:** The company website is **not loading**.
 - **Solution:** Use **Nagios** to check if the server is **reachable** and if the **CPU is overloaded**.
 - **Action:** Restart the server or increase resources to fix the issue.
-

Scenario 2: Bandwidth Monitoring

- **Problem:** The network is **slow** because someone is **downloading large files**.
- **Solution:** Use **SolarWinds** to track **bandwidth usage** and identify which device is consuming it.

- **Action:** Block or limit their usage to free up bandwidth.
-

Scenario 3: Security Issue

- **Problem:** Suspicious activity is detected, and someone is **accessing a restricted area**.
 - **Solution:** Use SNMP Traps to get **alerts** about unauthorized access.
 - **Action:** Block the user and investigate the activity.
-

6. Summary of Tools and SNMP

Tool/Concept	Purpose	Example Use Case
SNMP	Collects and monitors data from devices in a network.	Check the CPU usage or temperature of a router.
Nagios (Open Source)	Monitors networks, devices, and applications for performance and alerts.	Track if a server is down or disk space is running out.
SolarWinds (Commercial)	Advanced monitoring tool for large networks, offering real-time analysis and reporting.	Monitor bandwidth usage and find the most active devices consuming traffic.

7. Final Thoughts

- **SNMP** is like a **universal language** for devices to share data.
- Tools like **Nagios** and **SolarWinds** act as **network guardians**—always watching for issues and keeping things running smoothly.