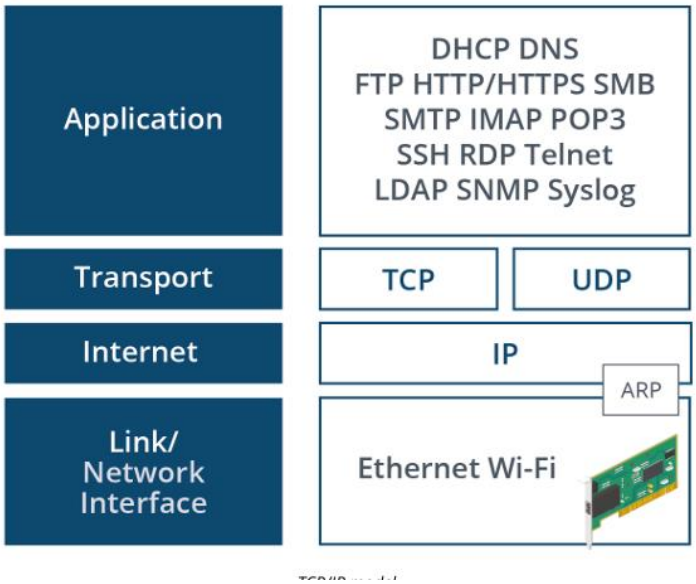
# Topic 5B: Use Basic TCP/IP Concepts:

## TCP/IP:

A **protocol** is like a set of traffic rules for data, making sure computers and devices can communicate properly over a network. Just like drivers follow stop signs and speed limits, computers follow protocols to send and receive information in an organized way.

Most networks use the **TCP/IP protocol suite**, which is a group of protocols working together, like different departments in a company handling specific tasks. Data is sent through different **layers**, where each layer adds its own **header (label)**, like how a package gets a shipping label, tracking number, and address before it reaches its destination. This process is called **encapsulation**, ensuring data moves smoothly between devices and networks.

There are 4 Layers under the TCP/IP model. They are described below:




## Link or Network Interface Layer:

The **Link (or Network Interface) Layer** is like the foundation of a road system—it's responsible for moving data **within** a local network (not across the internet). Instead of using IP addresses like a router, it uses **MAC addresses** to identify devices.

- **Example:** Think of a Wi-Fi network in a coffee shop. When you send a file to a nearby printer, your data stays inside that Wi-Fi network and travels using **frames** (small data packets) with MAC addresses.

This layer supports **Ethernet, Wi-Fi, DSL, and cable modems**, making sure data is delivered inside the local network before higher layers handle internet communication.

## Internet Layer:

Internet Layer is the second layer in the TCP/IP protocol layer. It provides packet addressing and routing within the network of networks. The **Internet Protocol (IP)** is like the postal system for the internet—it assigns addresses to devices and helps route data between different networks. Any device that connects to an IP network, like a **PC, phone, or server**, is called an **end system host**.

For data to travel between networks, it must pass through a **router**, just like mail needs a post office to reach another city. However, devices on a local network (like Wi-Fi or Ethernet) use **MAC addresses** instead of IP addresses. To match an **IP address to a MAC address**, networks use **Address Resolution Protocol (ARP)**, which acts like a directory lookup.

IP sends data using a **best effort** approach, meaning it **doesn't guarantee** delivery—some packets might get **lost, arrive late, or out of order**. That's why higher-layer protocols, like **TCP**, help ensure data is received correctly.

## Transport Layer:

The **Transport Layer** is like a traffic manager, ensuring data flows smoothly between applications on different devices. While the **Network Layer (IP)** handles addressing and routing, the Transport Layer makes sure multiple applications can communicate at the same time without mixing up their data.
There are two main **Transport Layer protocols**:

- **TCP (Transmission Control Protocol)**: Like certified mail, it ensures all packets arrive in the correct order and requests missing ones if lost. It's used for important data like emails, web pages, and file downloads, where missing or incorrect data can cause errors.
- **UDP (User Datagram Protocol)**: Like a regular text message, it sends data quickly but doesn't check if everything arrives properly. It's used for real-time applications like video calls and online gaming, where a slight glitch or delay is better than waiting for missing data to be resent.

🔹 **Example:** If you're watching a live sports stream, **UDP** allows the video to keep playing smoothly even if a few frames are lost. But if you're downloading a file, **TCP** ensures you get every bit of data correctly, so the file isn't corrupted.

## Application Layer:

The **Application Layer** is like the front-end of the internet—it provides the actual services users interact with, such as **web browsing, email, and file transfers**. Unlike lower layers, which focus on moving data, the Application Layer defines **how data is used** by different programs.
There are many **Application Layer protocols**, each serving a specific purpose:

- **HTTP/HTTPS** – Loads websites in browsers.
- **SMTP, POP3, IMAP** – Manages email sending and receiving.
- **FTP, SFTP** – Transfers files between devices.

Each protocol uses a specific **TCP or UDP port** to communicate between a **client (user's device)** and a **server (host providing the service)**.

🔹 **Example:** When you open a website, your browser (client) requests the page using **HTTP/HTTPS** over **port 80 or 443**, and the web server responds with the webpage content.
TCP/IP, originally developed by the **U.S. Department of Defense**, is now an **open standard**, meaning anyone can contribute to its development. The **Internet Engineering Task Force (IETF)** oversees improvements, publishing official internet standards as **Request for Comments (RFCs)** at **rfc-editor.org**.

# IPv4 Addressing:

An **IPv4 address** is a **unique identifier** for a device on a network, allowing it to send and receive data. IPv4 addresses are **32-bit numbers**, but since long binary numbers are hard to read, they are written in **dotted decimal notation** (four numbers separated by dots).

🔹 **Example:** The binary **11000000 10101000 00000000 00000001** is converted to **192.168.0.1**, which is much easier to remember and use. **Octet** means each 8 bits that are grouped together.
Each **section (octet)** can range from **0 to 255**. Means if all the bits in a octet are set to 1, then the number obtained is 255 and if all the bits are set to 0, then the number obtained is 0. Hence, IPv4 addresses can be from **0.0.0.0 to 255.255.255.255**, some are **reserved for special purposes**, like private networks or broadcasting.
IPv4 is the most widely used addressing system, but due to the **limited number of available addresses**, **IPv6** (a newer and longer address system) is being adopted.

# Network Prefixes:

An IPv4 address is like an apartment address. It provides two key of information:

a. Network ID: This is like the apartment address. All residents (devices) must share the same main address.
b. Host ID: This is like the apartment number. It identifies a specific resident (device) within the building (network).

Let's take an example:
**IP Address:** 192.168.0.1
**Subnet Mask:** 255.255.255.0 (or /24 in slash notation)

Here, the subnet mask helps separate the network and host portions of the IP address. Meaning, in the IP address, the first **three** numbers (192.168.0) represent the **Network ID** (building address). And the last number (**1**) represents the Host ID.

Imagine you live in an apartment complex named **"Sunset Towers"** at **"192 Sunset Blvd."**

- All apartments in that building share the same **street address** (Network ID).
- Each apartment has a **unique number** (Host ID), like **Apt 1, Apt 2, Apt 3**.

Similarly, in a **192.168.0.0/24** network:

- **192.168.0.0** is like "Sunset Towers."
- **192.168.0.1** could be a laptop.
- **192.168.0.2** could be a printer.
- **192.168.0.100** could be a smartphone.

The subnet mask **255.255.255.0** (or /24) means that all devices in the **192.168.0.x** range belong to the **same network**.

**What Does the Subnet Mask Do?**

A Subnet Mask is like a filter that separates an IP address into two parts: **Network ID** and **Host ID.** The subnet mask acts like a **filter** that determines which part of the IP address belongs to the **network** and which part belongs to the **host**.

- A **/24 subnet mask** means **the first 24 bits are the Network ID** (192.168.0).
- The **remaining 8 bits are for Host IDs** (0-255).

So, in a **/24 network**, you can have up to **254 usable devices** (since .0 is reserved for the network itself, and .255 is used for broadcasting).

# IPv4 Forwarding:

When a computer (host) **wants to send data** over an IPv4 network, it needs to determine if the destination is on the **same network** or a **different network**. It does this by comparing the **source and destination IP addresses** using the **subnet mask**.

---

**Case 1: Same Network (Local Delivery)**

📌 **Example from the first image:**

- **Source IP:** 192.168.0.100
- **Destination IP:** 192.168.0.200
- **Subnet Mask:** 255.255.255.0 (/24)

✓ **What happens?**

- The subnet mask tells the device that only the **last octet (0-255)** can change within the same network.
- Since **both IPs have the same first three octets (192.168.0)**, they **belong to the same network** (192.168.0.0/24).

- The computer sends the data **directly** to the destination using **ARP** (Address Resolution Protocol) to find the MAC address.

🔹 **Real-life Example:**

Imagine you're in an office and need to send a file to a co-worker sitting next to you. Since they are in the **same office (network)**, you just **hand them the file directly** (no need for a courier).

---

**Case 2: Different Network (Needs a Router)**

📌 **Example from the second image:**

- **Source IP:** 192.168.0.100
- **Destination IP:** 192.168.1.100
- **Subnet Mask:** 255.255.255.0 (/24)

❌ **What happens?**

- The **first three octets don't match** (192.168.0 ≠ 192.168.1), meaning the **destination is on a different network**.
- The computer **cannot send the data directly** because the destination is **outside its subnet**.
- Instead, it **forwards the data to a router (default gateway)**, which will route it to the correct network.

🔹 **Real-life Example:**

Imagine you want to send a letter to a friend in another city. Since they are in a **different area (network)**, you **can't deliver it yourself**—you take it to the **post office (router)**, and they **send it to the right place**.

---

**What is a Default Gateway?**

A **default gateway** is the **router's IP address** that the computer sends data to **when the destination is on a different network**.

- If a device **cannot find the destination locally**, it **forwards** the packet to the default gateway (router).
- The router then **routes** the packet to the correct network.

✓ **Example:**

- Your home router's IP is **192.168.0.1**.
- Your PC's IP is **192.168.0.100**.
- If you try to access **8.8.8.8 (Google DNS)**, your PC **forwards the request to the router**, which sends it to the internet.

# Public and Private Addressing:

Imagine you live in an **apartment building**. Your building has a **street address (public IP)** that everyone in the world can see, but inside the building, each apartment has its own **room number (private IP)** that is only used within the building.

Similarly, in networking:

- A **Public IP address** is like your **building's street address**—it's unique and allows anyone (like visitors or mail carriers) to find and communicate with you. Public IPs are assigned by **Internet Service Providers (ISPs)** and are used to access the internet.
- A **Private IP address** is like your **room number**—only people **inside** the building (your local network) know it. Private IPs are used within homes, offices, and businesses and cannot be accessed directly from the internet.

**Real-Life Example:**

Let's say you have **Wi-Fi at home**:

- Your **router** is connected to the internet and has a **Public IP (e.g., 203.0.113.10)** assigned by your ISP.
- Inside your home, each device (laptop, phone, printer) gets a **Private IP (e.g., 192.168.1.2, 192.168.1.3, etc.)** from the router.

- When your phone wants to visit www.google.com, it sends the request to your **router**, which then forwards it using the **Public IP** to communicate with Google's servers.

Since private IP addresses are **not directly reachable from the internet**, your router uses **Network Address Translation (NAT)** to manage the communication between your private devices and the public internet. This helps keep your devices **secure** from direct attacks and saves IPv4 addresses by allowing multiple devices to share a single public IP.

## Private Adress Ranges:

In IPv4, certain **IP address ranges** are reserved for use in **private networks** like homes, offices, and businesses. These **private IPs cannot** be used to access the public internet directly. Instead, they work **only within a local network (LAN)** and require a router with **Network Address Translation (NAT)** to communicate with the internet.
The **three private address ranges**, defined in **RFC 1918**, are:

1. **10.0.0.0 to 10.255.255.255** → Used in **large** networks (big companies, ISPs).
2. **172.16.0.0 to 172.31.255.255** → Used in **medium-sized** networks (universities, enterprises).
3. **192.168.0.0 to 192.168.255.255** → Used in **small** networks (homes, small offices, Wi-Fi routers).

◆ **Real-Life Example:**

Imagine your **Wi-Fi router at home** assigns devices **private IPs** like **192.168.1.10** (your phone) and **192.168.1.20** (your laptop). These devices can talk to each other inside the home but **need the router (which has a public IP) to reach the internet**.
Private IPs help **organize networks, improve security**, and **reduce the demand for public IP addresses**.

## Address Classes and Default Subnet Masks:

In the early days of **IPv4**, IP addresses were divided into **classes (A, B, and C)** to organize networks based on their size. Each class had a **default subnet mask**, which determined how many devices (hosts) could be in the network.
Here's how the **three main classes** work:

| Class | Dotted Decimal Mask | Network Prefix | Binary Mask | Used For |
|---|---|---|---|---|
| A | 255.0.0.0 | /8 | 11111111 00000000 00000000 00000000 | Very **large** networks (big companies, ISPs) |
| B | 255.255.0.0 | /16 | 11111111 11111111 00000000 00000000 | **Medium-sized** networks (universities, enterprises) |
| C | 255.255.255.0 | /24 | 11111111 11111111 11111111 00000000 | **Small** networks (homes, small offices) |

Imagine you are organizing **homes in a city**:

- **Class A** is like assigning one street name to an **entire country**! (e.g., "Main St" for all U.S. addresses). There's a **huge** number of houses (devices) under it.
- **Class B** is like assigning one street name per **state**—less than Class A, but still a lot of houses.
- **Class C** is like naming a street **in a small neighborhood**, where only a limited number of houses exist.

Similarly, in networking:

- **Class A** allows **16 million+ devices** per network.
- **Class B** allows **65,000+ devices** per network.
- **Class C** allows **254 devices**, making it ideal for home and office networks.

For example, Microsoft has **offices worldwide** with **millions of devices** (computers, servers, printers, phones, etc.). If they were to use **Class C (which only allows 254 devices per network)**, they would **run out of addresses quickly**. Instead, large organizations like Microsoft, Google, and Amazon use **Class A IP ranges** (e.g., **10.0.0.0/8**), which can support **over 16 million devices** in a single network. Whereas, A **small business** (like a local coffee shop or office) might only have **a few dozen devices**, such as **computers, printers, and security cameras**. They don't need millions of addresses, so they typically use **Class C IP ranges** (e.g., **192.168.1.0/24**), which support **up to 254 devices**—more than enough for their needs.

## Internet Access Using Private Addressing:

Devices with **private IP addresses** (like **192.168.x.x, 10.x.x.x, or 172.16.x.x**) **cannot access the internet directly** because these addresses are **not routable** on the public internet. To solve this, the network must use a method to **translate private IPs into public IPs** before sending data online.
There are **two main ways** to do this:

**1. Network Address Translation (NAT) – Used by Routers**

**✓ How it works:**

- A **router with a public IP** act as a middleman.
- When a device (e.g., a laptop with **192.168.1.10**) sends a request to a website, the router **replaces** its private IP with the router's **public IP** (e.g., **203.0.113.10**) before forwarding it to the internet.
- When the response comes back, the router knows which device requested it and sends it back to the right private IP.

**2. Proxy Server – Used in Businesses**

**✓ How it works:**

- Instead of sending requests directly to the internet, devices send them to a **proxy server**.
- The proxy server **fetches the requested website** and sends it back to the user.
- This can also provide **extra security and filtering** (e.g., blocking certain websites).

## IPv4 Host Address Configuration:

Every device (host) on an **IPv4 network** needs an **IP address** and a **subnet mask** to communicate with other devices. However, for full functionality, additional settings like a **default gateway** and **DNS servers** are also required.

---

**1. Required Settings for IPv4 Configuration**

**✓ IP Address:** A unique address assigned to a device (e.g., **192.168.0.2**).

**✓ Subnet Mask:** Defines the network and host portion of the IP (e.g., **255.255.255.0** or **/24**).

- The **first address (192.168.0.0)** is the **network ID** (identifies the whole network).
- The **last address (192.168.0.255)** is the **broadcast address** (used to send messages to all devices in the network).
- **Valid host addresses: 192.168.0.1 to 192.168.0.254** (usable for devices).

---

**2. Additional Important Settings**

**✓ Default Gateway:** The router's IP address (**192.168.0.1**) that forwards packets to other networks (like the internet).

**✓ DNS Server:** Translates website names into IP addresses.

- **Preferred DNS:** Often set to the router's IP (**192.168.0.1**) or an external service.
- **Alternative DNS:** A backup in case the primary fails (e.g., **8.8.8.8** – Google's DNS).

Without a **default gateway**, the device can **only communicate within the local network** but **not access the internet**. Without **DNS servers**, a user would need to enter IP addresses instead of website names (e.g., typing **172.217.1.142** instead of www.google.com).

---

**3. Static vs. Dynamic Configuration**

- **Static IP (Manual Entry):** The user manually enters all settings (as shown in the image).
- **Dynamic IP (DHCP):** The router automatically assigns an IP address and other settings, making it easier for most users.

## Static vs Dynamic Host Address Configuration:

There are two main ways to assign IP addresses to devices on a network: **Static** and **Dynamic (DHCP)**.

---

**1. Static IP Addressing (Manual Configuration)**

✓ **How it works:**

- An **administrator manually** enters the **IP address, subnet mask, gateway, and DNS settings** for each device.
- If the device moves to another network, the IP must be **changed manually**.
- The admin must **track all assigned IPs** to avoid duplication.

✓ **Best for:**

- **Routers, servers, printers**, and other devices that need a **fixed** IP.

✓ **Example:**

A company **web server** must always be reachable at **192.168.1.10** so that employees can access it reliably.

❌ **Downside:**

- **Time-consuming and error-prone** in large networks.

---

**2. Dynamic IP Addressing (DHCP - Automatic Configuration)**

✓ **How it works:**

- A **DHCP server (like the router in the image)** automatically assigns IP addresses to devices when they connect.
- The **Subnet Mask, Default Gateway, and DNS** are also assigned automatically.
- The **IP address lease expires** after a set time (e.g., **1440 minutes = 24 hours** in the image).

✓ **Best for:**

- **Home networks, offices, and large networks** where devices frequently connect/disconnect.

✓ **Example:**

When you connect your phone to **Wi-Fi**, the **router assigns it an IP** (e.g., **192.168.0.101**) automatically.

✅ **Benefit:**

- **Easier to manage**—no need to manually configure each device.

---

**3. What Happens if DHCP Fails? (APIPA / Link-Local Addressing)**

- If a computer **can't reach the DHCP server**, it **assigns itself a random IP** in the range **169.254.x.x** (APIPA).
- APIPA **only allows communication with other devices using APIPA** but **cannot access the internet**.

✓ **Example:**

If your **Wi-Fi router crashes**, your computer might get an **APIPA address (169.254.1.10)**, meaning it **can't access the internet** but may still talk to other local devices.

# Automatic Private IP Addressing (APIPA):

APIPA (Automatic Private IP Addressing) is a **failover mechanism** used when a computer **cannot get an IP address** from a **DHCP server**. Instead of having no IP, the computer assigns itself a **random IP** from the range **169.254.0.1 to 169.254.255.254**.

---

- **How APIPA Works**
  1. A device connects to the network and requests an IP from a **DHCP server**.
  2. If the DHCP **doesn't respond** (due to failure, misconfiguration, or no DHCP available), the device **self-assigns** an IP in the **169.254.x.x** range.
  3. The device can **communicate with other devices** using APIPA on the same network **but cannot access the internet**.

# Dynamic Host Configuration Protocol:

**DHCP (Dynamic Host Configuration Protocol)** is a **network service** that **automatically assigns IP addresses** to devices on a network. Instead of manually setting up an IP address, subnet mask, gateway, and DNS on each device, **DHCP does it automatically**, making networking **faster and easier**.

**How DHCP Works (Step-by-Step Process)**

1. **Device Connects** – When a device (like a laptop or phone) joins a network, it **asks for an IP address**.

2.**DHCP Server Responds** – A **DHCP server** (usually the router) checks its available IP addresses and **assigns one**.

3. **Device Receives IP** – The device gets an **IP address, subnet mask, default gateway, and DNS settings**.

4. I**P Lease Time** – The IP is assigned for a limited time (e.g., **24 hours**) and is renewed when needed.

# SOHO Router Configuration:

A **SOHO (Small Office/Home Office) router** connects your local network (LAN) to the internet through your **Internet Service Provider (ISP)**. It has **two main interfaces**:

1️⃣**Public Interface (WAN/Internet)** – Connects to the ISP and gets a **public IP address** (e.g., **203.0.113.1**) so the router can access the internet.

2️⃣**Private Interface (LAN)** – Assigns **private IP addresses** (e.g., **192.168.0.1**) to local devices like laptops and phones, acting as a **default gateway** for them.

---

- **Steps to Configure a SOHO Router**

**✓ 1. Connect to the Router**

- Plug a computer into one of the **RJ45 Ethernet ports** or join the **Wi-Fi network** using the default credentials (found on the router's sticker).

**✓ 2. Access the Web Interface**

- Open a web browser and enter the router's **management IP** (e.g., http://192.168.0.1 or http://192.168.1.1).
- Some routers also use a domain name like http://www.routerlogin.com.

**✓ 3. Log in as Administrator**

- Use the **default username and password** (usually printed on the router).
- The first time you log in, **change the admin password** to something strong (at least 12 characters).

**✓ 4. Configure Internet Settings**

- Most routers automatically **obtain a public IP** from the ISP via **DHCP**.
- If the ISP provides a **static IP**, enter the details manually.

- Some ISPs require **PPPoE login credentials**, which you enter in the settings.

✓ **5. Configure Wi-Fi & Security**

- Set up **Wi-Fi network names (SSID)** and strong **passwords**.
- Disable **WPS** for better security.
- Enable **firewall and encryption (WPA3 or WPA2-PSK).**

✓ **6. Check Network Status & Logs**

- View **line status and system logs** to troubleshoot connection issues.
- If internet issues occur, contact the ISP and provide details from the router logs.

# IPv6 Addressing:

The pool of available IPv4 public addresses is not very large, compared to the number of devices that need to connect to the Internet. While private addressing and NAT provides a workable solution, IP version 6 (IPv6) is intended to replace IPv4 completely, at some point. An IPv6 address is a 128-bit number and so can express exponentially more address values than the 32-bit number used in IPv4.

## IPv6 Notation:

IPv6 addresses are written in **hexadecimal (base-16) format**, making them **easier to read and type** compared to long binary numbers. Since **one hex digit** represents **4 bits (a nibble)**, an IPv6 **128-bit address** is divided into **eight 16-bit groups**, separated by colons (:).

### 🔹 Example of an IPv6 Address

makefile — Copy — Edit

```
2001:0db8:0000:0000:0abc:0000:def0:1234
```

Each **group** (like `2001`, `0db8`, etc.) represents **16 bits** of the full **128-bit address**.

### 🔹 Shortening IPv6 Addresses (Rules)

To make IPv6 addresses **shorter and easier to type**, we can apply two rules:

**1️⃣ Remove Leading Zeros**

- `0db8` → `db8`
- `0000` → `0`

So, the address becomes:

ruby — Copy — Edit

```
2001:db8:0:0:abc:0:def0:1234
```

**2️⃣ Use Double Colon (`::`) for Consecutive Zeros**

- A **single sequence of zeros** (`0:0`) can be replaced by `::`
- But **this can only be done once** in an address

So, our final shortened IPv6 address is:

ruby — Copy — Edit

```
2001:db8::abc:0:def0:1234
```

## IPv6 Network Prefixes:

In **IPv6**, an address is divided into **two main parts**, each **64 bits long**:

1. **Network ID (First 64 Bits)** – Identifies the network, similar to how a **street name** identifies a neighborhood.
2. **Interface ID (Last 64 Bits)** – Identifies a specific device (interface), similar to how a **house number** identifies a home.

---

🔹 **No Need for a Subnet Mask in IPv6**

Unlike **IPv4**, where subnet masks define network and host portions, **IPv6 has a fixed 64-bit network size**, so subnet masks are **not needed**. Instead, **prefix notation** (e.g., /64) is used to indicate how many bits belong to the network.

---

🔹 **Understanding IPv6 Prefixes (/nn Notation)**

- **/32** → Used by ISPs to allocate large network blocks.
- **/48** → Assigned to customers, allowing them to create **65,536 subnets**.
- **/64** → The most common prefix used for LANs and home networks.

For example:

- If an ISP assigns a business a **/48 prefix**, it can divide that into **65,536 /64 subnets**, each capable of handling **trillions of devices**.
- A home network is usually given a **/64 prefix**, meaning all devices share the first **64 bits as the network ID**.

## Global and Link-Local Addressing:

Unlike **IPv4**, where a device usually has **one IP per interface**, **IPv6 allows multiple addresses per interface**. The two most important types are **Global Addresses** and **Link-Local Addresses**.

---

🔹 **1. Global IPv6 Address (Public, Internet-Accessible)**

✓ Equivalent to a **public IPv4 address**—used for communication **over the internet**.

✓ Uniquely assigned to devices worldwide by ISPs.

✓ **Starts with:** 2xxx:: or 3xxx::

✓ Example: 2001:db8::1

🔹 **Real-Life Example:**

A **web server** with a global IPv6 address (2001:db8::1) can be accessed from anywhere in the world, just like a public IPv4 address (e.g., 203.0.113.10).

---

🔹 **2. Link-Local IPv6 Address (Local Network Only)**

✓ Used only **inside a local network**, not routable on the internet.

✓ Every IPv6-enabled device **must have a link-local address**.

✓ **Automatically assigned** to an interface (like APIPA 169.254.x.x in IPv4).

✓ **Starts with:** fe80::

✓ Example: fe80::1a2b:3c4d:5e6f:7g8h

🔹 **Real-Life Example:**

Imagine a **printer** on a home network that has a **link-local address** (fe80::1234). Your computer can talk to it **without needing an internet connection**.

In **IPv6**, devices automatically obtain their IP addresses using **Stateless Address Auto Configuration (SLAAC)** instead of relying on a **DHCP server**, as in IPv4. When a device connects to a network, it first assigns itself a **link-**

**local address** (fe80::/10). It then uses **Neighbor Discovery (ND)** to find a router, which provides the network's **global IPv6 prefix**. The device then creates its own **global IPv6 address** based on this prefix, allowing it to communicate over the internet. Unlike IPv4, IPv6 does not require manual configuration of a **default gateway** since ND automatically discovers the router. ND also replaces **ARP** by helping devices find each other's MAC addresses within the network. This makes IPv6 networks **more efficient, automated, and scalable** compared to IPv4.

## Dual Stack:

**Dual Stack** is a networking approach that allows devices to run both **IPv4 and IPv6 simultaneously** to support a smooth transition from IPv4 to IPv6. Since completely replacing IPv4 is challenging, most **hosts, routers, and networks** are configured to use both protocols at the same time. When a device connects to another, it will **first try to use IPv6** and, if the destination does not support it, will **fall back to IPv4**. This ensures compatibility with older networks while allowing newer IPv6-enabled systems to communicate efficiently. Dual Stack is commonly used by **ISPs, enterprises, and data centers** to maintain seamless internet access while gradually shifting towards full IPv6 adoption.

## Some Questions and Answers:

**1. A host is configured with the IP address 172.16.1.100 in the 172.16.1.0/16 network. What value should be entered as the subnet mask?**

✅ **Answer:** The subnet mask should be **255.255.0.0**. A **/16 prefix** means that the first **16 bits** are set to **1** in binary (11111111 11111111 00000000 00000000), which converts to **255.255.0.0** in dotted decimal format.

---

**2. You are setting up a printer to use static IPv4 addressing. What type of value is expected in the default gateway field?**

✅ **Answer:** The **IPv4 address of the local router interface**, entered in **dotted decimal format** (e.g., 192.168.1.1). The printer uses the router as a gateway to communicate with other networks, including the internet.

---

**3. Another technician has scribbled some notes about IPv4 addresses used in various networks associated with support tickets. One of them is assigned to the WAN interface of a SOHO router that requires troubleshooting. Which of these addresses must it be?**

**Options:**

- **52.165.16.254**
- **192.168.100.52**
- **169.254.1.121**
- **172.30.100.32**
- **224.100.100.1**

✅ **Answer:** The WAN interface must use a **valid public IPv4 address**, so the correct choice is **52.165.16.254**.

✓ **Explanation:**

- **192.168.100.52** and **172.30.100.32** are **private IPs** (Class C and Class B, respectively).
- **169.254.1.121** is an **APIPA (self-assigned) address**, used when DHCP fails.
- **224.100.100.1** is a **Class D multicast address**, not a valid public IP.

---

**4. True or False? A SOHO router can be configured to provide an IPv4 address configuration to hosts without further administrator attention.**

✅ **Answer: True.** This is done through the **Dynamic Host Configuration Protocol (DHCP)**, which automatically assigns IP addresses, subnet masks, default gateways, and DNS settings to connected devices.

---

**5. True or False? A valid IPv6 configuration does not require a subnet mask.**

✅ **Answer: True.** IPv6 **does not use subnet masks** like IPv4. Instead, it uses **prefix notation** (e.g., /64) to determine the network portion of an address. The last **64 bits** always represent the **host ID**, making subnet masks unnecessary.

---