Understanding IPv4 and IPv6 Addressing

Chapter-4

What is IANA?

The **Internet Assigned Numbers Authority (IANA)** is a global organization responsible for coordinating and managing key elements of the Internet's addressing system.
IANA operates under ICANN (Internet Corporation for Assigned Names and Numbers).

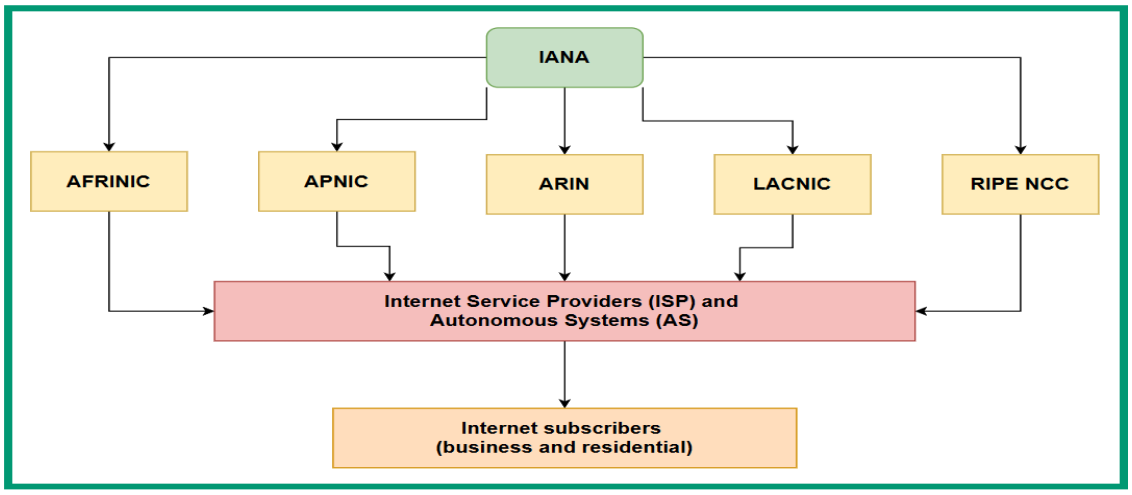
Main Responsibilities of IANA

1. **IPv4 and IPv6 address allocation**
    - IANA manages the entire global pool of public IPv4 and IPv6 addresses.
    - It allocates large address blocks to the world's Regional Internet Registries (RIRs).
2. **Autonomous System Number (ASN) allocation**
    - ASNs are used by organizations that run routers participating in BGP.
    - IANA assigns ASN blocks to RIRs.
3. **Protocol assignments**
    - Manages protocol numbers, port numbers, and other identifiers (e.g., TCP/UDP ports, ICMP types).
4. **DNS Root Zone Management**
    - Maintains the world's root DNS servers.
    - Ensures proper functioning of TLDs (.com, .org, .in, etc.).

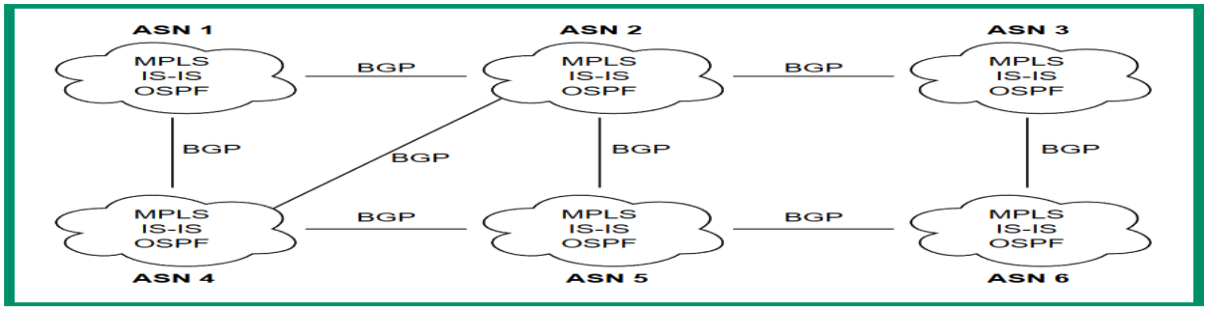
To manage global address allocation efficiently, IANA does **not** distribute IP addresses directly to ISPs or users.
Instead, it delegates this responsibility to **five Regional Internet Registries (RIRs**

The following are the **five RIRs** and their **responsible** geolocations:
• **African Network Information Center (AFRINIC)**: Supports the continent of Africa
• **Asia-Pacific Network Information Centre (APNIC):** Supports regions of Asia and the Pacific
• **American Registry for Internet Numbers (ARIN):** Supports regions of Canada, the USA, and parts of the Caribbean
• **Latin America and Caribbean Network Information Centre (LACNIC):** Supports Latin America and parts of the Caribbean regions
• **Reseaux IP Europeens Network Coordination Centre (RIPE NCC):** Supports Europe, the Middle East, and Central Asia

- Each RIR is responsible for distributing IPv4 and IPv6 address blocks to ISPs and any Autonomous System (AS) within their responsible geolocation.



- An AS is simply any organization that is responsible for managing a large number of internet routing networks such as an ISP within a country or region.
- IANA assigns AS numbers to the various RIRs around the world. Then, the RIRs allocate these AS numbers to network operators such as ISPs.
- The ISPs use their AS numbers to share their network routing prefixes with other ISPs using the Border Gateway Protocol (BGP).



Public versus private address spaces

IANA created two separate IPv4 address spaces to help reduce the depletion of IPv4 addresses around the world. These IPv4 address spaces are known as the public address space and the private address space.

Class	Range	Default Subnet Mask
A	0.0.0.1 - 127.255.255.255	255.0.0.0
B	128.0.0.1 - 191.255.255.255	255.255.0.0
C	192.0.0.1 - 223.255.255.255	255.255.255.0
D	224.0.0.1 - 239.255.255.255	N/A
E	240.0.0.1 - 255.255.255.255	

Classes A, B and C can be assigned to devices that are directly connected to the internet. The Class D address range is used for multicast communication between applications and services that operate on devices within a network, and the Class E address range is reserved for experimental uses.

The **subnet mask** is used to determine the network and host portion of an IP address, and the total number of IP addresses and usable IP addresses within a network.

Classful IPv4 address

Class	Range	Default Subnet Mask	Number of Networks	Number of Usable IPv4 Addresses
A	0.0.0.1 - 127.255.255.255	255.0.0.0	126	16,777,214
B	128.0.0.1 - 191.255.255.255	255.255.0.0	16,384	65,534
C	192.0.0.1 - 223.255.255.255	255.255.255.0	2,097,152	254

Demerits Classful IP address

Classful IP addressing wastes a large number of addresses because network sizes are fixed (Class A, B, C) and cannot be customized. It lacks flexibility, does not support CIDR or route summarization, and leads to inefficient routing and faster IPv4 exhaustion.

Difference between Classful and Classless addressing

Feature	Classful Addressing	Classless Addressing (CIDR)
Address Format	Divided into fixed classes (A, B, C)	Uses prefix length notation (/n)
Subnet Mask	Default mask only; cannot vary	Flexible masks
Network Size	Fixed sizes → wasteful	Adjustable to exact requirement
Address Utilization	Huge wastage (e.g., Class B = 65,000 hosts)	Highly efficient; minimal wastage
IPv4 Conservation	Poor	Excellent; slows depletion

Private IP Address space

Class	Range	Default Subnet Mask
A	10.0.0.1 - 10.255.255.255	255.0.0.0
B	172.16.0.1 - 172.31.255.255	255.255.0.0
C	192.168.0.1 - 192.168.255.255	255.255.255.0

Difference between Public and private ipv4 address

Public IP Address	Private IP Address
Assigned by ISP or IANA and is **globally unique** on the Internet.	Used inside local networks; **not globally unique** and can be reused in different networks.
Routable on the Internet (accessible from anywhere).	**Not routable on the Internet**; routers automatically block them.
Required for devices that need **direct Internet access** (web servers, email servers, gateways).	Used by internal devices like PCs, laptops, printers, switches, Wi-Fi devices.
More limited in number; can be **expensive** to obtain.	Free to use; no need for ISP or IANA permission.
Exposed to the Internet → higher security risks if not protected.	Safer by default because outside networks cannot reach them.
Examples: **8.8.8.8**, **203.0.113.5**	Examples: **10.x.x.x**, **172.16–172.31.x.x**, **192.168.x.x**

Public IP addresses are globally unique addresses assigned by ISPs and are routable on the Internet. **Private IP addresses** are used inside local networks, are not routable on the Internet, and can be reused by multiple organizations. Private IPs require **NAT** to access the Internet.

Network Address Translation (NAT)

When a residential user subscribes to an Internet service, the Internet Service Provider (ISP) installs a **modem** (or modem-router) in the customer's home. This device performs **multiple functions**:

The modem provides the interface between the customer's **private LAN** and the ISP's **public network**.

- Inside the home → devices use **private IPv4 addresses** (like 192.168.1.x).
- ISP network and Internet → use **public IPv4 addresses**.
- The modem is **preconfigured by the ISP** to perform **NAT**.
  NAT converts **private IP addresses** into a **single public IP address** assigned by the ISP.
- **Why is NAT needed?**
- Because private IP addresses **cannot travel across the Internet**.
  Only **public** addresses are routable on the Internet.

The diagram shows how a router connects:

- A **private IPv4 network** (inside)
- A **public IPv4 network** (outside / Internet)

The router performs **NAT** to translate private addresses into public addresses.

Inside Addresses (Private Network)

- PC1 has private IP: **192.168.1.10**
- Inside network: **192.168.1.0/24**
- Router inside interface (Gi0/1): **192.168.1.1**

✅ These addresses are **not routable on the Internet**.

Outside Addresses (Public Network)

- Router outside interface (Gi0/0): **209.65.1.2/28**
- Public server: **209.65.1.10**
- Outside network: **209.65.1.0/28**

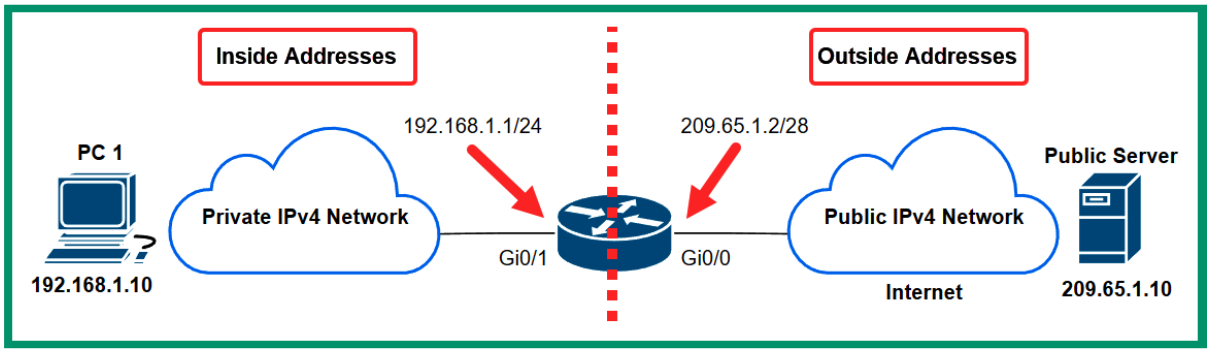
These are **publicly routable addresses** on the Internet.

Role of the Router (Middle Device)

The router sits between:

- **Inside network (private)**
- **Outside network (public)**

It performs **NAT** so that private devices (192.168.1.x) can communicate with public servers on the Internet.



The diagram shows NAT operation where a private IPv4 network (192.168.1.x) connects to the public Internet through a router. The router has an inside interface (192.168.1.1) and an outside public interface (209.65.1.2). When a private host sends traffic to a public server, the router translates the private IP address into its public IP, allowing communication with external networks.

Advantages of NAT

1. **Conserves Public IPv4 Addresses**
   NAT allows organizations to use private IPv4 addresses internally and only one public IPv4 address on their router. This helps reduce the use of scarce public IPv4 addresses.
2. **Hides the Internal Network**
   Hundreds or thousands of private devices can share a single public IP. NAT hides the internal network structure from the Internet, adding a basic security layer.
3. **Flexible Internal Addressing**
   Organizations can use any private IPv4 address ranges without coordinating with ISPs. The NAT-enabled router handles all translations between private and public IPs.

Disadvantages of NAT

1. **Performance Overhead**
   The router must translate private IPs to public IPs for every packet. This translation process adds delay and may slightly reduce network performance.
2. **Issues with IPsec VPNs**
   NAT changes the IP address in the packet header. IPsec relies on the original IP address for authentication, so NAT can break IPsec VPN connections.

3. **Loss of End-to-End Connectivity**
   NAT hides the true source address. The receiving device cannot see the real sender's IP, making end-to-end communication less transparent and sometimes problematic.

Types of NAT

1. Static NAT
2. Dynamic NAT
3. Port Address Translation

Static NAT

Static NAT creates a permanent one-to-one mapping between a private inside IP address and a public IP address. This allows an internal device, such as a web server, to be accessible from the Internet using a fixed public IP.

When using Static NAT, the one-to-one mapping remains as-is on the router or modem until it's modified by the network professional. Keep in mind that Static NAT mapping will forward all traffic types between a public IP address to a private IP address.

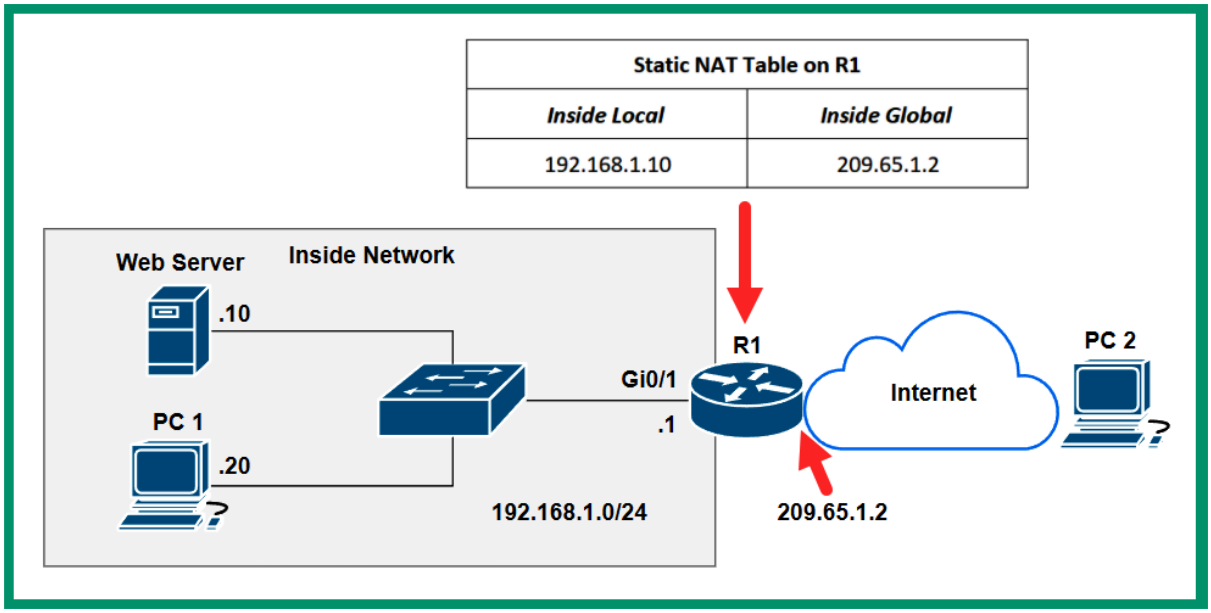


Diagram shows:

- Inside network: **192.168.1.0/24**
  - **Web server** = **192.168.1.10**
  - **PC1** = **192.168.1.20**
- Router R1: inside interface **192.168.1.1**, outside/public address **209.65.1.2**
- **Static NAT** entry on R1:

- **Inside local** = `192.168.1.10` → **Inside global** = `209.65.1.2`

Dynamic NAT

Dynamic Network Address Translation (Dynamic NAT) is a type of NAT where **multiple private (inside local) IPv4 addresses** are mapped to **a pool of public (inside global) IPv4 addresses (many-to-many mapping)**.
The mapping is **not permanent**; it is created **dynamically** whenever an internal device initiates communication to the internet.

Dynamic NAT uses a **first-come, first-served** approach:

- Hosts take a public IP only when they need it.
- If all public IPs are already in use, new requests **must wait**.

✓ 2. How Dynamic NAT Works

1. A router (NAT device) is configured with:
    - A group of **inside private IP addresses**
    - A **pool of public IP addresses**
2. When an internal host tries to access the internet:
    - The router assigns the host **one available public IP** from the pool.
    - This mapping remains active only for the duration of the session.
3. When the session ends:
    - The public IP is **returned to the pool** and becomes available again.

Example of Dynamic NAT

Network Setup

- **Inside private network:**
    - PC1 → 192.168.1.10
    - PC2 → 192.168.1.11
    - PC3 → 192.168.1.12
- **Public NAT pool:**
    - 209.65.1.2
    - 209.65.1.3
    - 209.65.1.4

This means the router has **3 public IPv4 addresses** available for NAT.

Step-by-Step Dynamic NAT Operation

Step 1: PC1 sends traffic to the Internet

- PC1 (192.168.1.10) initiates a request.
- Router assigns the **first available** public IP: **209.65.1.2**

- NAT table entry is created:

Inside Local	Inside Global
192.168.1.10	209.65.1.2

Step 2: PC2 sends traffic to the Internet

- PC2 (192.168.1.11) initiates a connection.
- Router assigns the next available public IP: **209.65.1.3**

Inside Local	Inside Global
192.168.1.10	209.65.1.2
192.168.1.11	209.65.1.3

Step 3: PC3 sends traffic to the Internet

- PC3 (192.168.1.12) wants to access the Internet.
- Router assigns the third public IP: **209.65.1.4**

Inside Local	Inside Global
192.168.1.10	209.65.1.2
192.168.1.11	209.65.1.3
192.168.1.12	209.65.1.4

Step 4: A 4th device tries to access the Internet

- Assume PC4 (192.168.1.13) now makes a request.
- **All 3 public IPs are already in use.**
- The NAT pool is **exhausted**.

✅ The router **cannot** assign a public IP
❌ PC4 **cannot access the Internet** until one NAT entry times out.

What happens after a session ends?

- When PC1 stops browsing, the mapping **192.168.1.10 → 209.65.1.2** is removed.
- Now **209.65.1.2 becomes free**.
- The next device (PC4) can use this public IP.

7. Advantages

- More flexible than Static NAT
- Useful when internal hosts do not need a constant public address
- Provides unique public IPs for connections that require it

8. Disadvantages

- Limited by the number of public IPs in the pool
- Cannot support many internal devices simultaneously
- Not efficient for large networks (PAT is preferred)

Port Address Translation (PAT)

Port Address Translation (PAT), also known as **NAT overload**, is a type of Network Address Translation where **many private IPv4 addresses share a single public IPv4 address**.
It is the most commonly used NAT method in homes and organizations.

Purpose

- To conserve public IPv4 addresses
- To allow **multiple devices** inside the network to access the Internet using **one public IP**
- To support communication for hundreds or thousands of devices through port-based identification

Why PAT Needs Port Numbers

Static and Dynamic NAT cannot translate multiple devices to a single public IP because they translate only the IP address.

PAT uses:

- Source IP
- Source Port
- Destination IP
- Destination Port

This 4-tuple creates a **unique NAT entry**, enabling thousands of devices to share one public address.

Example of Port Address Translation (PAT)

Network Setup

Inside Private IPs:

- PC1 → **192.168.1.10**
- PC2 → **192.168.1.11**
- PC3 → **192.168.1.12**

Public IP available:

- Only **one** public IP: **209.65.1.2**

PAT allows **all three PCs** to share this **single public IP**.
PAT does this by translating:

- Private IP address
  -
  - TCP/UDP port number
    → to unique **port numbers** on the public IP.

✅ Step-by-Step PAT Operation

✅ Step 1: PC1 sends traffic

PC1 (192.168.1.10) opens a web connection to an internet server.

- Source IP = **192.168.1.10**
- Source Port = **1025** (random high port selected by OS)
- Destination IP = **93.184.216.34** (example.com)
- Destination Port = **80** (HTTP)

PAT Translation:

192.168.1.10:1025 → 209.65.1.2:30001

The router rewrites the packet to use public IP + unique port **30001**.

A NAT table entry is created.

✅ Step 2: PC2 sends traffic

PC2 (192.168.1.11) also connects to the same or different website.

- Source IP = **192.168.1.11**
- Source Port = **1044**

PAT Translation:

192.168.1.11:1044 → 209.65.1.2:30002

Another unique port (**30002**) is assigned.

✅ Step 3: PC3 sends traffic

PC3 (192.168.1.12) initiates a connection.

- Source Port = **1050**

PAT Translation:

192.168.1.12:1050 → 209.65.1.2:30003

Again, PAT uses another unique port on the same public IP.

✅ PAT NAT Table Example

Inside Local (Private)	Inside Global (Public)
192.168.1.10:1025	209.65.1.2:30001
192.168.1.11:1044	209.65.1.2:30002
192.168.1.12:1050	209.65.1.2:30003

All 3 devices share **one public IP**

Pv6 Address Format

1. IPv6 Address Length

- An **IPv6 address is 128 bits long**.
- IPv4 was only **32 bits**, which limited available addresses.
- IPv6 provides $\mathbf{2^{128} \approx 3.4 \times 10^{38}}$ possible addresses — enough for every device for many generations.

2. Representation of IPv6 Addresses

IPv6 addresses are written in **hexadecimal notation**, instead of the dotted-decimal format used in IPv4.

Hexadecimal digits allowed:

```
0 1 2 3 4 5 6 7 8 9 A B C D E F
```

- Hex digits represent values from **0 to 15**.
- Letters (A–F) are **not case-sensitive**.

IPv6 Address Structure

An IPv6 address has:

- **8 hextets**
- Each hextet = **16 bits**
- Total = **8 × 16 = 128 bits**

Example IPv6 address:

```
2001:0DB8:0000:1111:0000:0000:0000:0200
```

Each hextet is separated by a **colon (:)**.

Rules for Shortening (Compressing) IPv6 Addresses

✅ Rule 1: Remove Leading Zeros

You can remove **leading zeros inside any hextet**.

Example:

```
0DB8 → DB8
0000 → 0
0200 → 200
```

✅ Rule 2: Use Double Colon (::) for Consecutive Zero Hextets

Whenever **two or more consecutive hextets** are all zeros, you can replace them with **::**.

✅ **:: can appear only once in an IPv6 address.**

✅ Router differentiates sessions using **unique port numbers**

Example Compression

Original IPv6 Address

```
2001:0DB8:0000:1111:0000:0000:0000:0200
```

Step 1 — Remove leading zeros

```
2001:DB8:0:1111:0:0:0:200
```

Step 2 — Compress consecutive zero hextets

There are three consecutive zeros:

```
0:0:0
```

So they become:

```
::
```

✅ Final Simplified IPv6 Address

```
2001:DB8:0:1111::200
```

Types of IPv4 and IPv6 Addresses

1. Automatic Private IP Addressing (APIPA) – IPv4 Link-Local

What is APIPA?

Automatic Private IP Addressing (APIPA) is a feature built into many operating systems (especially Microsoft Windows) that automatically assigns an IP address to a client **when it cannot obtain one from a DHCP server**.

APIPA addresses are **self-assigned**, meaning the device configures its own IP address without any external server.

When Does a Device Use APIPA?

A client device may fail to receive an IP address from a DHCP server because:

- ✅ It cannot communicate with the DHCP server
- ✅ It is configured with a static IP (so DHCP is bypassed)
- ✅ The DHCP server is offline, down, or not present
- ✅ Network connectivity issues prevent DHCP responses

When this happens, the device shifts into **self-configuration mode** (APIPA) to allow basic limited communiction.

APIPA Address Range

When APIPA is used, the client automatically assigns itself an IPv4 address from:

✅ 169.254.0.1 to 169.254.255.254

With the default subnet mask:

✅ 255.255.0.0 (/16)

This makes the entire 169.254.0.0/16 block available for local communication.

Purpose of APIPA

- Allows devices to still communicate **locally** on the same network segment
- Helps maintain limited functionality until DHCP service is restored
- Useful for troubleshooting DHCP failures

However:

❌ APIPA addresses **cannot access the Internet**
❌ APIPA addresses are valid **only within the local broadcast domain**

- **APIPA** → Term mainly used in **Windows** systems
- **Link-Local** → Term preferred in **Linux**, macOS, and standards documents (IETF)

Both refer to the same IPv4 range: **169.254.0.0/16**

Extended Unique Identifier (EUI-64)

EUI-64 (Extended Unique Identifier–64) is a method used in IPv6 networks to **generate the 64-bit Interface ID** of an IPv6 address **from the device's 48-bit MAC address.**

It is commonly used when IPv6 Stateless Address Autoconfiguration (**SLAAC**) is enabled.

Role of EUI-64 in SLAAC

When SLAAC is used on an IPv6 network:

- The **router** provides the first **64 bits (network prefix)** in its **Router Advertisement (RA)** message.
- The **client** must generate the remaining **64 bits (Interface ID)**.
- The client uses its **MAC address** and applies the **EUI-64 process** to form this Interface ID.

This results in a complete **128-bit Global Unicast IPv6 address**.

Why EUI-64 is Needed?

- The MAC address is **48 bits**, but IPv6 Interface IDs require **64 bits**.
- EUI-64 expands the MAC address to 64 bits by inserting **FFFE** in the middle and flipping the **7th bit (Universal/Local bit)**.

Given

- **IPv6 Prefix:** 2001:DB8:0:1111::/64
- **MAC Address:** FC:99:47:75:CE:E0

(You wrote 001:DB8, but IPv6 documentation always uses **2001:DB8** as the example prefix, so the correct prefix is used.)

Step 1: Split the MAC address into two halves

MAC:

FC:99:47 : 75:CE:E0

- First half: **FC 99 47**
- Second half: **75 CE E0**

Step 2: Insert FFFE in the middle

Insert **FFFE** after the first 3 bytes:

FC:99:47:FF:FE:75:CE:E0

Step 3: Flip the 7th bit of the first byte

First byte = **FC**

Convert FC to binary:

FC = 1111 1100

The **U/L bit** = **bit 1** (second least significant bit):

- Current bit1 = **0**
- Flip it → becomes **1**

New binary:

1111 1100 → 1111 1110

Binary **1111 1110** = **FE**

Updated first byte: **FE**

Step 4: Final EUI-64 Interface ID

Replace FC → FE:

FE:99:47:FF:FE:75:CE:E0

Step 5: Combine Prefix + Interface ID

Prefix (first 64 bits):

2001:DB8:0:1111

Interface ID (last 64 bits):

FE99:47FF:FE75:CEE0

Final IPv6 Address (EUI-64):

2001:DB8:0:1111:FE99:47FF:FE75:CEE0

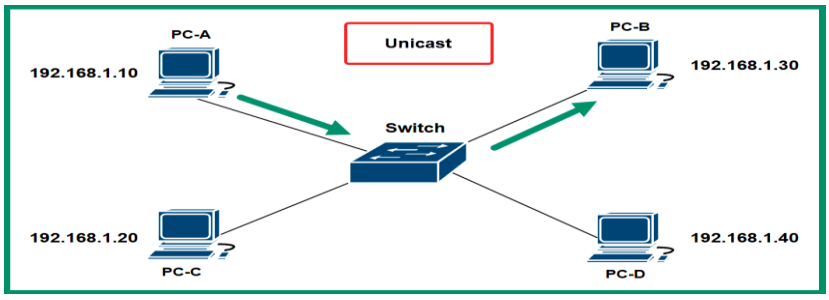
Final Answer

IPv6 Address using EUI-64:

2001:DB8:0:1111:FE99:47FF:FE75:CEE0

Unicast

Unicast is a type of communication in which data is sent **from one sender to one specific receiver**. It's a **one-to-one** communication model used in both IPv4 and IPv6 networks.



Type	Description	Prefix
Global Unicast Address (GUA)	Globally unique, routable on the Internet	`2000::/3`

Multicast

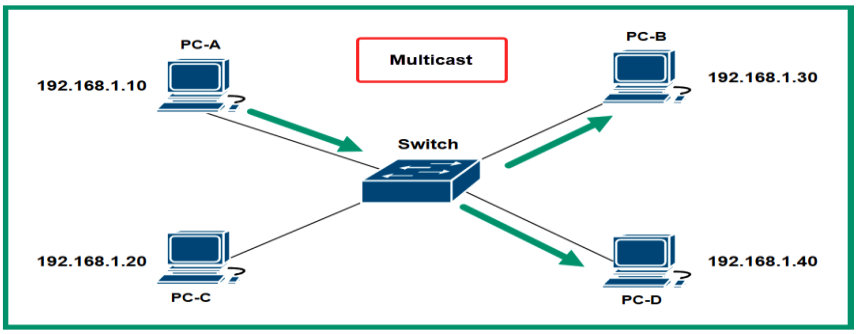
Multicast is a communication method where data is sent **from one sender to multiple specific receivers** — that is, **one-to-many** communication.

Multicast sends data **only to a group of devices** that have expressed interest in receiving it.

Both **IPv4** and **IPv6** support multicast communication.

The following is a list of IPv6 multicast addresses and their designated groups:

• FF02::1/12: All IPv6-enabled nodes on a network
• FF02::2/12: All IPv6-enabled routers on a network
• FF02::5/12: All routers that are running the Open Shortest Path First (OSPF) dynamic routing protocol
• FF02::A/12: All routers that are running the Enhanced Interior Gateway Routing Protocol(EIGRP) dynamic routing protocol



Broadcast

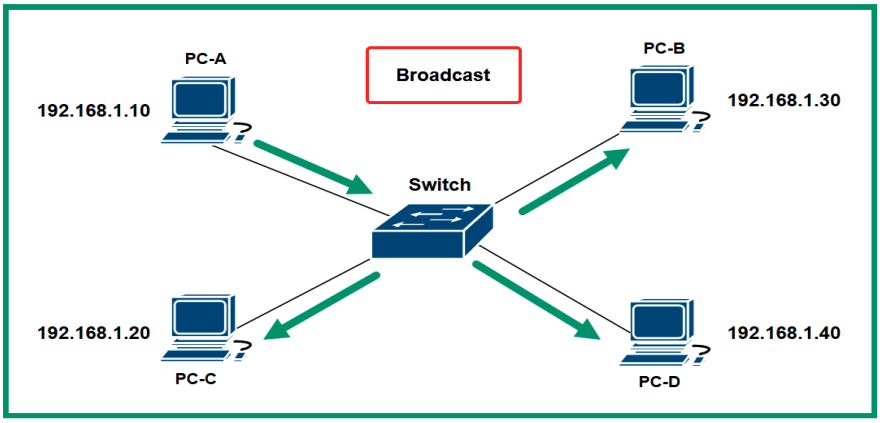
Broadcast communication allows **one-to-all** transmission within a single IPv4 network.
It means a single message sent by one device is received by **all devices** in the same subnet.

Broadcast is supported **only in IPv4**.

Example

If the network ID is `192.168.1.0` with a subnet mask of `255.255.255.0`:

Component	Value
Network ID	192.168.1.0
Subnet Mask	255.255.255.0
Broadcast Address	**192.168.1.255**
Usable Host Range	192.168.1.1 – 192.168.1.254



Anycast

Anycast is a communication method used **only in IPv6** (not in IPv4).
It provides **one-to-nearest** communication between a sender and the **closest** device (in routing terms) that shares the same anycast address. • Multiple devices (e.g., web servers or DNS servers) are configured with the **same IPv6 address**.

- Each of these devices is located in different geographic or network locations.
- When a client sends a packet to that IPv6 address:
  - Routers use the **shortest path** (based on routing metrics) to deliver the packet.
  - The packet reaches the **nearest** device configured with that anycast address.

Difference between Unicast,, Multicast, Broadcast, and Anycast

Type	Pattern	Meaning	IPv4 Support	IPv6 Support	Example
Unicast	One → One	One sender to one specific receiver	✓ Yes	✓ Yes	PC → Server
Multicast	One → Many	One sender to a selected group	✓ Yes	✓ Yes	IPTV, OSPF
Broadcast	One → All	One sender to all devices on the LAN	✓ Yes	✗ No	DHCP Discover
Anycast	One → Nearest	One sender to nearest node in a group	✓ Limited	✓ Yes	DNS, CDN

Delving into IPv6 concepts

IPv4 and IPv6 addresses exist in **different address spaces** and are **not directly compatible**. This means that an IPv4-only device cannot communicate natively with an IPv6-only device because their packet formats and addressing schemes are different.

To allow both IPv4 and IPv6 networks to **coexist and interoperate** during the transition to full IPv6 adoption, several **transition technologies** have been developed.

Tunneling

Tunneling is a technique that allows data from one IP version to travel through a network that uses another IP version.

One of the major concerns about having two different versions of IP on the internet or within an organization is that these two versions cannot communicate with each other natively. Simply put, if a device such as a computer is configured with an IPv6 address, it will not be able to communicate with devices on an IPv4 network.

Tunneling allows IPv6 packets to be transported over an IPv4 network. The forwarding router is responsible for encapsulating the IPv6 packet inside an IPv4 packet before sending it over the IPv4 network.
This type of tunneling is referred to as 6to4 tunneling. 4to6 tunneling, on the other hand, allows an IPv4 packet to be encapsulated within an IPv6 packet so that it can be transported over an IPv6 network

It works by **encapsulating** (wrapping) one type of IP packet inside another:

- The outer packet belongs to the **current network type** (IPv4 or IPv6).
- The inner packet is the **original message** (IPv6 or IPv4).

Type	Description	Used When
6to4 Tunneling	Encapsulates an **IPv6 packet inside an IPv4 packet**	When IPv6 data needs to travel over an **IPv4 network**
4to6 Tunneling	Encapsulates an **IPv4 packet inside an IPv6 packet**	When IPv4 data needs to travel over an **IPv6 network**

Example

- A company uses IPv6 internally but connects to the internet (which is mostly IPv4).
- The router **wraps the IPv6 packet inside an IPv4 packet** before sending it across the IPv4 network.
- At the destination, the IPv4 layer is removed, and the **original IPv6 packet** is delivered.

Dual Stacking

Dual stacking is another method that allows **IPv4 and IPv6 to work together** on the same network.

How It Works

- In a **dual-stack setup**, a device (like a computer or router) is **configured with both IPv4 and IPv6 addresses** on the **same network interface (NIC)**.
- This allows the device to **communicate on both IPv4 and IPv6 networks**.

Operation

- When sending data to an **IPv4 device**, it uses its **IPv4 address**.
- When sending data to an **IPv6 device**, it uses its **IPv6 address**.
- The device automatically chooses which protocol to use based on the **destination address**.

Translation (NAT64)

◆ Overview

NAT64 (Network Address Translation 64) is a technology that allows **IPv6 devices to communicate with IPv4 devices**.

◆ How It Works

- Just like traditional **NAT** converts **private IPv4 addresses** into **public IPv4 addresses**,
  NAT64 converts **IPv6 addresses** into **IPv4 addresses**, and vice versa.
- It is usually **configured on routers** or **gateway devices**.
- This translation allows data to pass smoothly between **IPv6-only** and **IPv4-only** networks.

◆ Example

- A computer with only an **IPv6 address** wants to reach a website that has only an **IPv4 address**.
- The **NAT64 router** translates the IPv6 packet into an IPv4 packet so the communication can happen.

Router Advertisement (RA)

In an **IPv6 network**, devices can **automatically get an IPv6 Global Unicast Address (GUA)** using special **ICMPv6 messages**.

When a new device connects:

1. It sends a **Router Solicitation (RS)** message asking if any IPv6 router is available.
2. The **IPv6 router** replies with a **Router Advertisement (RA)** message.

Purpose of the RA Message

The **RA message** tells the device how to configure its IPv6 address and provides important network details.

◆ InformationProvided by RA

- ✓ **Network prefix** and **prefix length** (for address formation)
- ✓ **Default gateway IPv6 address**
- ✓ **DNS server addresses** and **domain name**

Address Configuration Methods Indicated in RA

The RA message also tells the client **how to get its IPv6 address** using one of these methods:

Method	Description
SLAAC	Device creates its own IPv6 address using the router's prefix
Stateful DHCPv6	Device gets full IPv6 address and info from DHCPv6 server

Stateless Address Autoconfiguration (SLAAC)

IPv6 Address Configuration Methods

SLAAC allows a device to automatically create its own **Global Unicast Address (GUA)** without using a DHCPv6 server.

How SLAAC Works

1. The **client** sends a **Router Solicitation (RS)** message to find an IPv6 router.
2. The **router** replies with a **Router Advertisement (RA)** message containing:
    - The **network prefix**
    - The **prefix length**
3. The client uses the **EUI-64 process** to form the Interface ID:
    - Converts its **48-bit MAC address** into a **64-bit Interface ID**.
    - Combines it with the router's **64-bit network prefix** → forms a **128-bit IPv6 address**.

Thus, the device generates its own **GUA IPv6 address**.

SLAAC with Stateless DHCPv6

In this setup, the **RA message** tells the client to:

1. Use **SLAAC** to create its own IPv6 address.
2. Use the **router's Link-Local address** as the **default gateway**.
3. Contact a **stateless DHCPv6 server** to get:
    - **DNS server addresses**
    - **Domain name**

The **DHCPv6 server does NOT assign the IPv6 address** — only additional network info.

SLAAC	Stateful DHCPv6
Device configures itself.	DHCPv6 server assigns the address.
Less administrative overhead.	Full control and address tracking by admin.
Router gives prefix; host builds address.	DHCPv6 server gives complete info.
Lightweight and automatic.	Centralized and managed.
Ex- Small LAN or home IPv6 setup	Ex-Large organization needing central control