IPStructure

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Thu, 10 Jul 2025 08:26:00 GMT
Server: EduAPI/3.0

{
  "code": "ELEE1157",
  "name": "Network Routing Management",
  "credits": 15,
  "module_leader": "Seb Blair BEng(H) PGCAP MIET MIHEEM FHEA"
}
```


Download as a PDF

What is IPv4?

- **IPv4**: Internet Protocol version 4 is a core protocol that defines IP addresses used in networking.

- **32-bit** address space, supporting up to **4.3 billion addresses**.

IPv4 Address Structure

- IPv4 addresses are written as four **octets** separated by periods: `192.168.1.1`
- Each octet is an **8-bit binary number** (total 32 bits).
- Each octet ranges from `0-255` in decimal.

```
IPv4 Address Example: 11000000.10101000.00000001.00000001
```

IPv4 Address Classes

IPv4 addresses are divided into five classes:

Class	Range	Usage	Networks	Hosts
A	1.0.0.0 - 126.0.0.0	Large networks	16777216	16777214
B	128.0.0.0 - 191.255.0.0	Medium networks	65534	65534
C	192.0.0.0 - 223.255.255.0	Small networks	2097152	254
D	224.0.0.0 - 239.255.255.255	Multicasting		
E	240.0.0.0 - 255.255.255.255	Experimental		

Reserved IP Ranges

IPv4 has reserved addresses for private networks and special purposes.

Range	Description
10.0.0.0 - 10.255.255.255	Class A private
172.16.0.0 - 172.31.255.255	Class B private
192.168.0.0 - 192.168.255.255	Class C private
127.0.0.0 - 127.255.255.255	Loopback (localhost)

Subnetting in IPv4

What is Subnetting?

- Dividing a large network into smaller, manageable subnetworks.
- Uses a **subnet mask** to identify network and host portions.

Example:

- IP: `192.168.1.0/24`
- Subnet Mask: `255.255.255.0`

```
Network Portion | Host Portion
192.168.1       | .0 - .255
```

CIDR Notation

CIDR (Classless Inter-Domain Routing)

- Uses `slash` notation to indicate the subnet mask length (e.g., `/24`).
- `192.168.1.0/24` means the first 24 bits are network bits.

Examples of CIDR:

- `/8` (Class A) -> 255.0.0.0
- `/16` (Class B) -> 255.255.0.0
- `/24` (Class C) -> 255.255.255.0

Example: Calculating Subnets P1

Given Network: whose IP is range is `192.168.1.0/24` create four subnets.

|1. To create four subnets, we need to borrow bits from the host portion

- **Subnet Mask**:
  - `/26` -> 255.255.255.192 or `11111111.11111111.11111111.11000000`

2. Calculate Number of Subnets and Hosts per Subnet:
   - With `/26`, the first 26 bits are the network portion, and the remaining 6 bits are for hosts.
   - Number of subnets created: $2^2 = 4$ (since we borrowed two bits)
   - Hosts per subnet: $2^6 = 64 $ addresses per subnet (62 usable)

Example: Calculating Subnets P2

3. **Determine the Subnet Ranges::**

- Starting with 192.168.1.0, each subnet has 64 addresses, incrementing by 64 for each subsequent subnet.

Subnet ID	Range	Description
`192.168.1.0/26`	`192.168.1.0` - `192.168.1.63`	First subnet
`192.168.1.64/26`	`192.168.1.64` - `192.168.1.127`	Second subnet
`192.168.1.128/26`	`192.168.1.128` - `192.168.1.191`	Third subnet
`192.168.1.192/26`	`192.168.1.192` - `192.168.1.255`	Fourth subnet

Why IPv6?

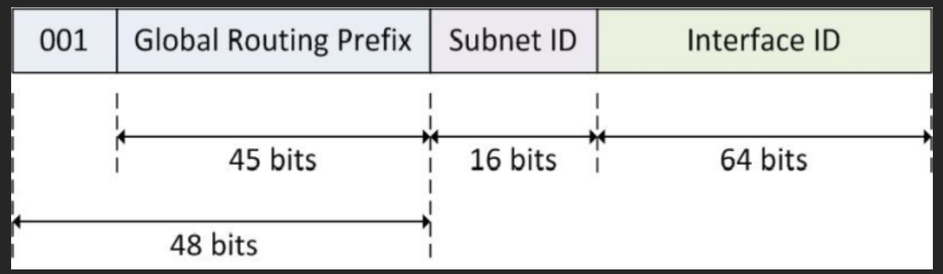
IPv4 Address Exhaustion

- IPv4: 32-bit address space (~4.3 billion addresses).
- Increasing number of internet-connected devices.
- IPv6: 128-bit address space
- ~340 undecillion addresses (enough for future needs).

Enhanced features:

- Auto-configuration.
- Improved security (IPSec).
- Simplified header format.

IPv6 Address Format

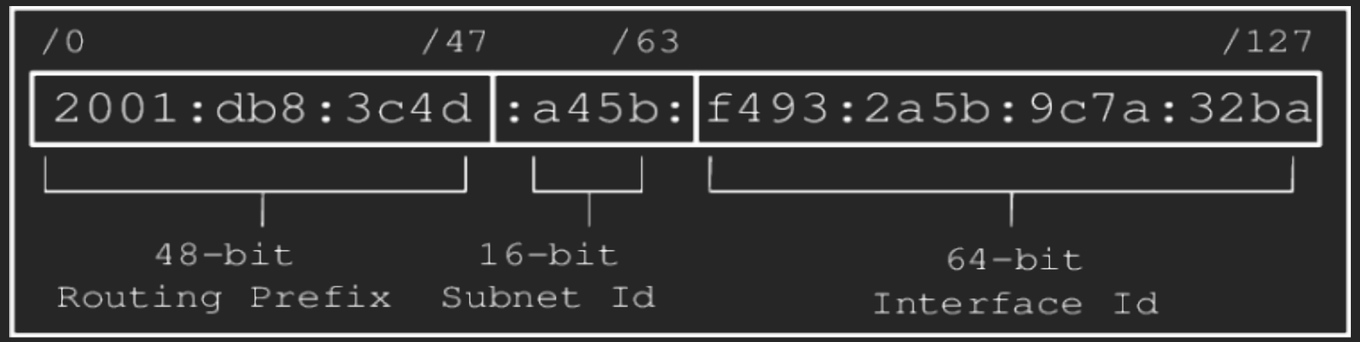


- Length: 128 bits (8 groups of 16 bits each).
- Hexadecimal notation.
- Groups separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

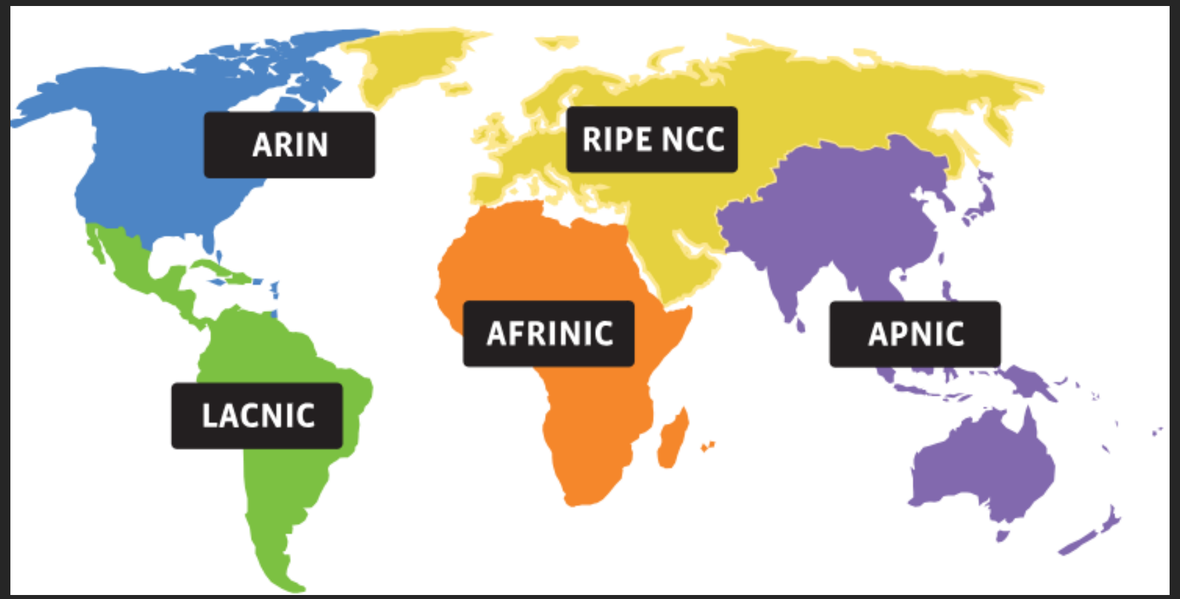
Simplification Rules:

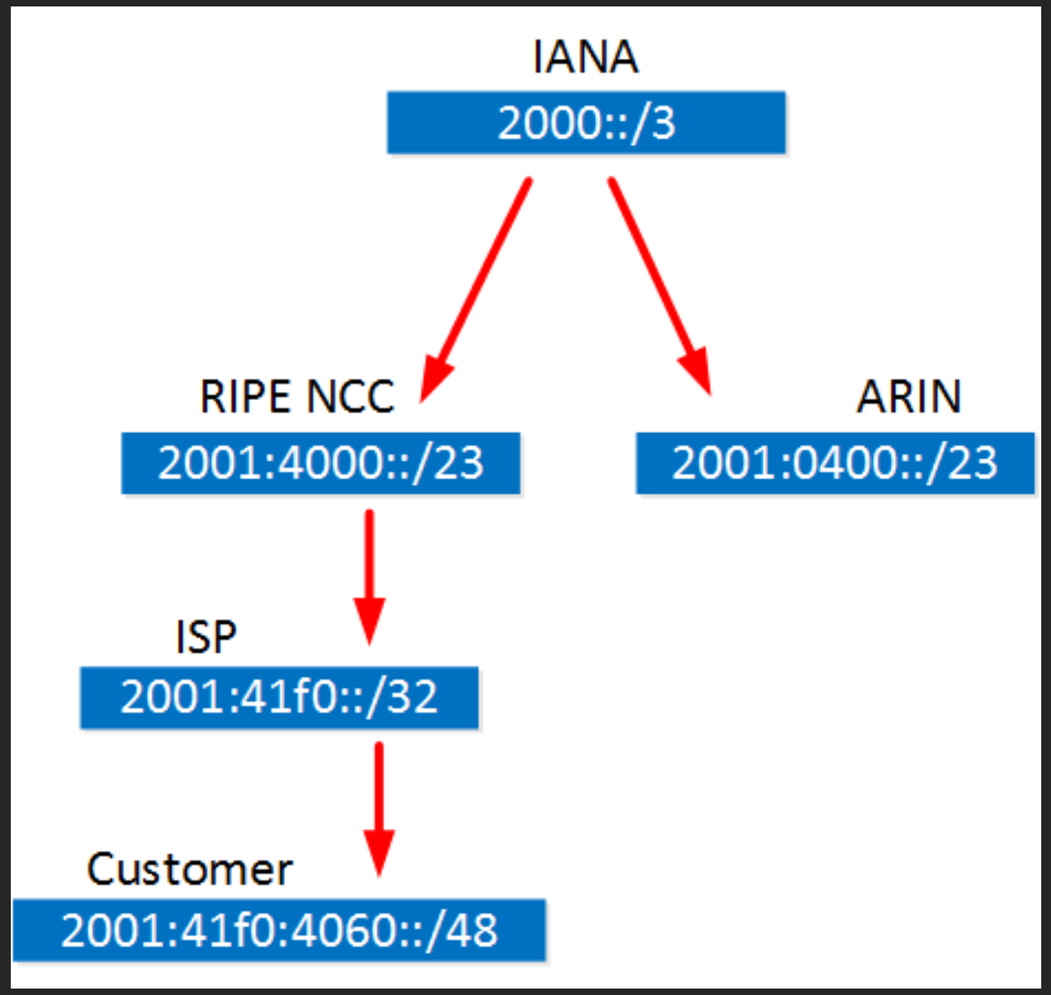
- Remove leading zeros (e.g., 0010 -> 10).
- Replace consecutive zero groups with :: (only once per address).

Anatomy of IPv6 address format: Example



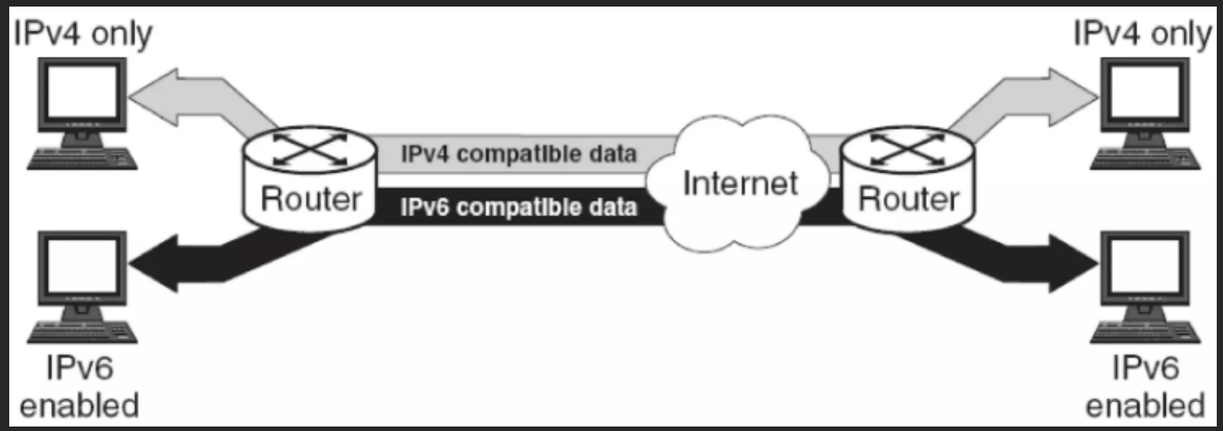
Internet Assigned Numbers Authority (IANA)





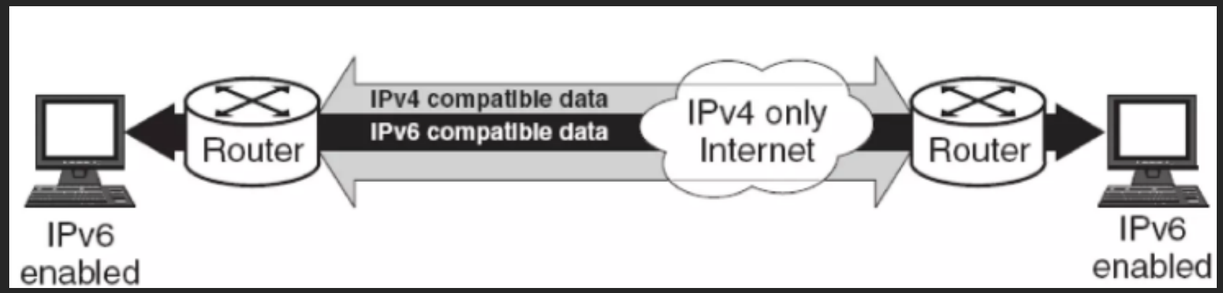
IPv4 to IPv6 Transition

- **Dual Stack**
  - Devices run both IPv4 and IPv6 simultaneously.
  - Allows gradual transition.



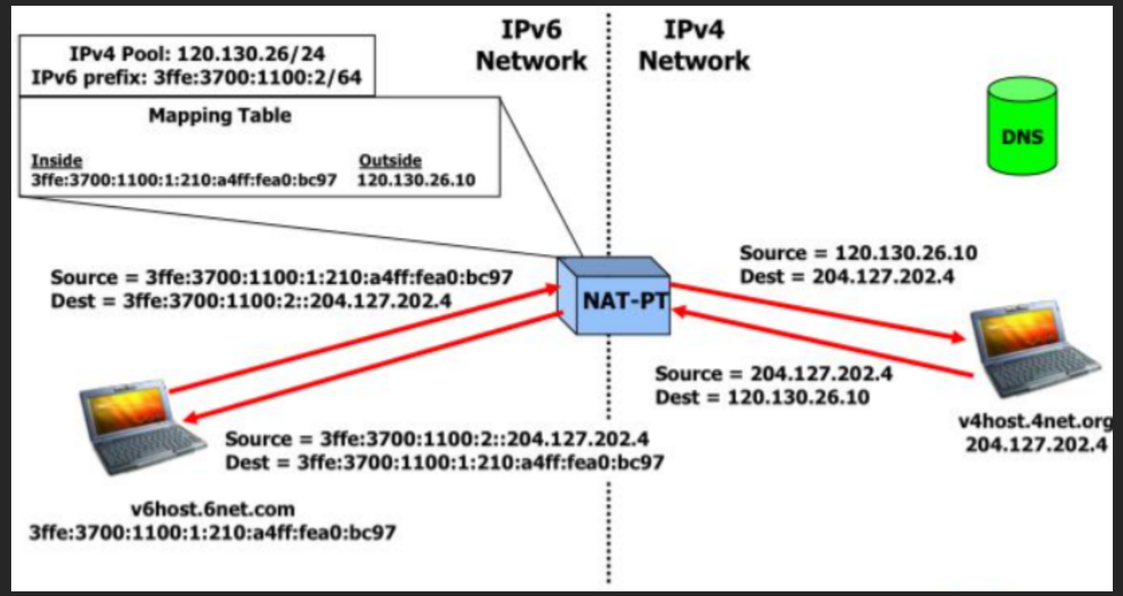
IPv4 to IPv6 Transition

- **Tunneling**
  - Encapsulates IPv6 packets within IPv4 headers.
  - **Examples**:
    - 6to4: Automatically assigns an IPv6 prefix to IPv4.
    - Teredo: Tunnels IPv6 over IPv4 using NAT.



IPv4 to IPv6 Transition

- **Translation**
  - **NAT64:**
    - Maps IPv6 addresses to IPv4 and vice versa.
    - Enables IPv6-only devices to communicate with IPv4 devices.



IPv4 to IPv6

Step 1: Understand the IPv4 Address

- Example IPv4: `192.168.0.1`
- Convert to binary:
  - 192 -> `11000000`
  - 168 -> `10101000`
  - 0 -> `00000000`
  - 1 -> `00000001`
- Binary IPv4: `11000000.10101000.00000000.00000001`

Step 2: Map IPv4 to IPv6 Format

- IPv6 has special prefixes for IPv4-mapped addresses:
  - `::ffff:0:0/96`
- Append the 32-bit binary IPv4 to the `::ffff:` prefix.

Step 3: Convert Binary IPv4 to Hexadecimal

- IPv4 in binary: `11000000.10101000.00000000.00000001`
- Group into 4-bit chunks:
  - `1100 0000 1010 1000 0000 0000 0000 0001`
- Convert each chunk to hexadecimal:
  - `1100` -> `C`
  - `0000` -> `0`
  - `1010` -> `A`
  - `1000` -> `8`
  - `0000` -> `0`
  - `0000` -> `0`
  - `0000` -> `0`
  - `0001` -> `1`
- Hexadecimal IPv4: `C0:A8:00:01`

Step 4: Construct the IPv6 Address

- Prefix: `::ffff:`
- Hexadecimal IPv4: `C0:A8:00:01`
- Combined: `::ffff:C0A8:01`

Examples for Practice

Example 1

Convert the IPv4 address `10.0.0.1` to IPv6.

▶ Solution

Example 2

Convert the IPv4 address `172.16.254.1` to IPv6.

▶ Solution

Cheat Sheet

IPv6 Header (320 bits)

Version (4)	Class (8)	Flow Label (20)
Payload length (16)	Next Header (8)	Hop Limit (8)
Source Address (128)		
Destination Address (128)		

MAC to EUI-64 Conversion

IPv6 Addresses

::/0	Default Route	
::/128	Unspecified address	RFC 4291
::1/128	Loopback address	RFC 4291
::ffff:0:0/96	IPv4-mapped address	RFC 4291
2001:1::1/128	Port-Control-Protocol Anycast	RFC 7723
2001:1::2/128	Traversal Using Relays around NAT anycast	RFC 8155
2001:db8::/32	Documentation Prefix	RFC 3849
2620:4f:8000::/48	AS112 DNS sinkhole servers	RFC 7534
fc00::/7	Unique-Local Addresses (ULA)	RFC 4193
fe80::/10	Link-Local Unicast	RFC 4291
fec0::/10	Site-Local Adresses (deprecated)	

Well Known Multicast Addresses

ff02::1	All nodes
ff02::2	All routers
ff02::5	All OSPF routers
ff02::6	All OSPF DRs
ff02::9	All RIP routers
ff02::a	All EIGRP Routers
ff02::d	All PIM routers
ff02::f	UPNP devices
ff02::11	All Homenet Nodes
ff02::12	VRRP
ff02::16	All MLDv2-capable routers
ff02::1a	All RPL Router (IoT)
ff02::fb	Multicast DNS IPv6
ff02::101	Network time (NTP)
ff02::1:2	All DHCP agents
ff02::1:3	LLMNR
ff02:0:0:0:0:1:ff00::/104	Solicited Node Address
ff02:0:0:0:0:2:ff00::/104	Node Information Query
ff05::1:3	All DHCP Server (site)
ff05::101	All NTP Server (site)

IPv6 Tunnel/Transition Addresses

64:ff9b::/96	IPv4-IPv6 Translation (NAT64)	RFC 6052
64:ff9b:1::/48	IPv4-IPv6 Translation (local)	RFC 8215
2001::/32	TEREDO Tunnel	RFC 4380
2002::/16	6to4 Tunnel (use deprecated)	RFC 3056

Linux Commands

ip -6 addr show	Display IPv6 addresses
ip -6 addr add <ipv6addr>/<prefixlen> dev <if>	Add a new IPv6 address to an interface
ip -6 addr del <ipv6addr>/<prefixlen> dev <if>	Remove an IPv6 address from an interface
ip -6 route show	List IPv6 routes
ip -6 route add <ipv6network>/<prefixlen> \ via <gateway> [dev <device>]	Add a route through a gateway
ip -6 neigh show	List the neighbors table
ping6 <address>	Test ICMPv6 connectivity

Protocols

TCP	6
UDP	17
RSVP	46
GRE	47
ICMPv6	58
IPIP (IP in IP)	94
PIM Multicast	103

Multicast Scopes

1	Node/ Interface-local
2	Link-local
3	Realm-local
4	Admin-local
5	Site-local
8	Organization -local
E	Global

Extension Header Order

1	IPv6 Header
2	Hop-by-Hop
3	Destination Option
4	Routing Header
5	Fragment Header
6	Authentication Header
7	ESP Header
8	Destination Option
9	Upper Layer Payload