UNIT-4

IP version 6 (IPv6): IP version 6 (IPv6) is the latest version of IP. IP version 4 (IPv4) is currently used in intranets and private networks, as well as the Internet. IPv6 is the successor to IPv4, and is based for the most part on IPv4.
IPv4 has been widely deployed and used to network the Internet today. With the rapid growth of the Internet, enhancements to IPv4 are needed to support the influx of new subscribers, Internet-enabled devices, and applications. IPv6 is designed to enable the global expansion of the Internet.

IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security.

Advantages of IPv6:

1. **Expanded addressing capabilities**—IPv6 provides a larger address space. IPv6 addresses consist of 128 bits, while IPv4 addresses consist of 32 bits. 128-bit addressing increases the address space.
2. **Header format simplification**—IPv6 packet header format is designed to be efficient. IPv6 standardizes the size of the packet header to 40 bytes, divided into 8 fields.
3. **Improved support for extensions and options**—Extension headers carry Internet-layer information and have a standard size and structure.
4. **Flow labeling capability**—Flow labels provide consistent handling of packets belonging to the same flow.
5. **Improved privacy and security**—IPv6 supports extensions for authentication and data integrity, which enhance privacy and security.

IPv6 Addressing: IPv6 uses a 128-bit addressing model. This creates a much larger address space than IPv4 addresses, which are made up of 32 bits. IPv6 addresses also contain a scope field that categorizes what types of applications are suitable for the address. IPv6 does not support broadcast addresses, but instead uses multicast addresses to serve this role. In addition, IPv6 also defines a new type of address called *anycast*.

Address Representation: IPv6 addresses consist of 8 groups of 16-bit hexadecimal values separated by colons (:). The IPv6 address format is as follows:

```
aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa
```

aaaa is a 16-bit hexadecimal value, and **a** is a 4-bit hexadecimal value. Following is an example of an actual IPv6 address:

```
3FFE:0000:0000:0001:0200:F8FF:FE75:50DF
```

You can omit the leading zeros, as shown:

```
3FFE:0:0:1:200:F8FF:FE75:50DF
```

You can compress 16-bit groups of zeros to the notation **::** (two colons), as shown here, but only once per address:

```
3FFE::1:200:F8FF:FE75:50DF
```

Address Types: There are three types of IPv6 addresses:

1. Unicast—For a single interface.

2. Multicast—For a set of interfaces on the same physical medium. A packet is sent to all of the interfaces associated with the address.

3. Anycast—For a set of interfaces on different physical mediums. A packet is sent to only one of the interfaces associated with this address, not to all the interfaces.

Address Scope: IPv6 addresses have *scope*, which identifies the application suitable for the address. Unicast and multicast addresses support scoping.

Unicast addresses support two types of scope:

- *Global* scope
- *Local* scope.

There are two types of local scope:

- *link-local* addresses
- *site-local* addresses.

Link-local unicast addresses are used within a single network link. The first ten bits of the prefix identify the address as a link-local address. Link-local addresses cannot be used outside a network link.

Site-local unicast addresses are used within a site or intranet. A site consists of multiple network links, and site-local addresses identify nodes inside the intranet. Site-local addresses cannot be used outside the site.

Multicast addresses support 16 different types of scope, including node, link, site, organization, and global scope. A 4-bit field in the prefix identifies the scope.

Address Structure:

Unicast addresses identify a single interface. The address consists of n bits for the prefix, and 128 – n bits for the interface ID.

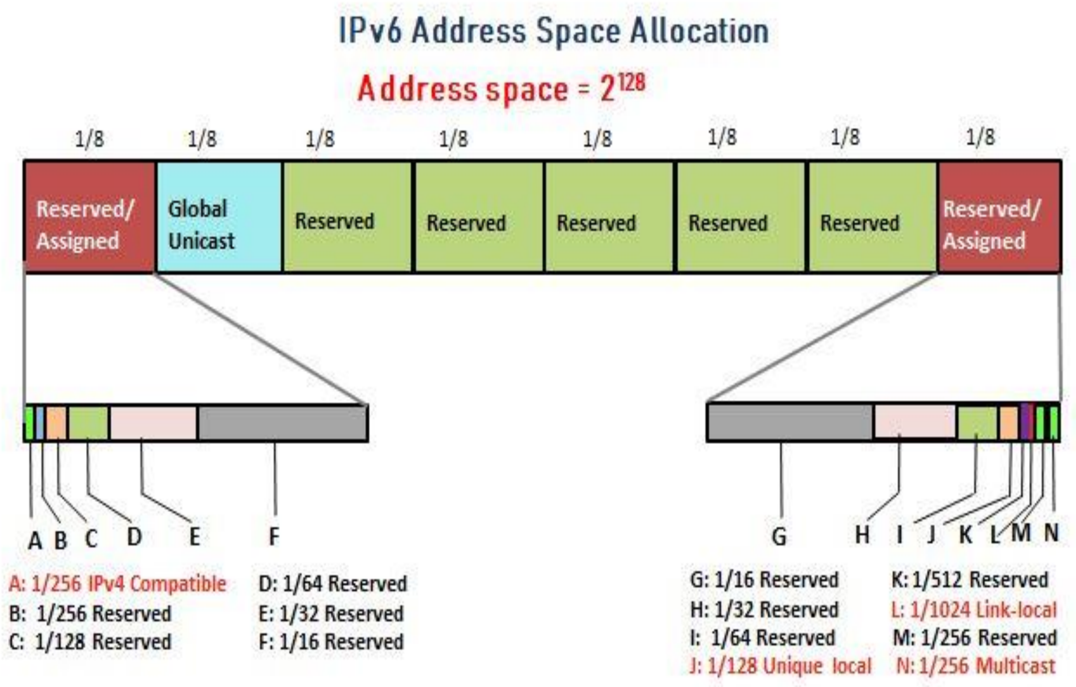
Multicast addresses identify a set of interfaces. The address is made up of the first 8 bits of all ones, a 4-bit flags field, a 4-bit scope field, and a 112-bit group ID:

11111111 | flags | scope | group ID

The first octet of ones identifies the address as a multicast address. The flags field identifies whether the multicast address is a well-known address or a transient multicast address. The scope field identifies the scope of the multicast address. The 112-bit group ID identifies the multicast group.

Similar to multicast addresses, anycast addresses identify a set of interfaces. However, packets are sent to only one of the interfaces, not to all interfaces. Anycast addresses are allocated from the normal unicast address space and cannot be distinguished from a unicast address in format.

IPv6 Address Space Allocation: The address space of IPv6 is divided into several blocks of varying size and each block is allocated for special purpose. Most of the blocks are still unassigned and have been left aside for future use. To better understand the allocation and the location of each block in address space, we first divide the whole address space into eight equal ranges. This division does not show the block allocation, but we believe it shows where each actual block is located (Figure).



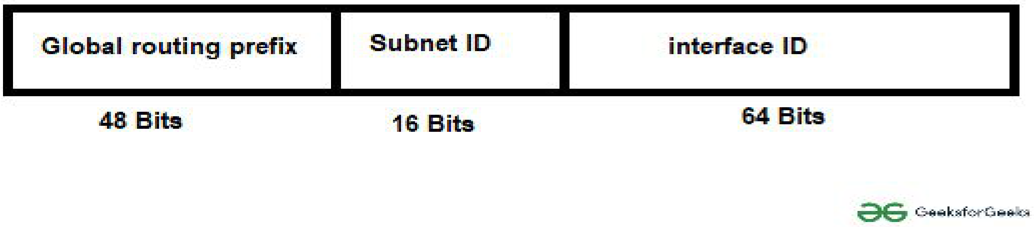
Each section is one-eighth of the whole address space (2125 addresses). The first section contains six variable-size blocks; three of these blocks are reserved and three unassigned. The second section is considered one single block and is used for global unicast addresses. The next five sections are unassigned addresses. The last section is divided into eight blocks. Some of these blocks are still unassigned and some are reserved for special purposes. The figure shows that more than five-eighths of the address space is still unassigned. Only one-eighth of the address space is used for unicast communication between the users.

Important IPv6 prefixs

Prefix	**Designation**	**Description**	**Equivalent IPv4 Address**
::/128	**Unspecified**	All zeros, this address can only be used by a host as a source address unless it learns its self-address.	0.0.0.0
::1/128	**Loopback**	This address is used by a host to talk to itself in IPv6 format.	127.0.0.1
Embedded IPv4 addresses ::/96 Example ::129.122.123.22	**IPv4-compatible IPv6 address**	It consists of 96 bits of zero followed by an IPv4 address. This address is useful when an IPv6 host sends a message to another IPv6 host through an intermediate IPv4 network.	No Equivalent in IPv4
Embedded IPv4 addresses ::ffff/96 Example ::ffff:168.128.2.5	**IPv4-mapped IPv6 address**	This address consists of 80 bits of zero, 16 bits of one, followed by IPv4 address. It is used when an IPv6 host wants to communicate with an IPv4 host. The packet travels through an IPv6 network to an Ipv4 destination host.	No Equivalent in IPv4
fe80::/10 Example: fe80::200:5aee:feaa:20a2 First 10 bits-(**1111111010**)	**Link-Local Addresses**	The first 10 bits are reserved as shown, 54 bits subnet ID is all zeros, 64 bits interface identifier. Link-local is the unicast address used for the single link for the purpose like automatic address configuration and network discovery protocol, etc. These addresses do not need to be globally unique datagram using link-local addresses are not forwarded by the router. But all IPv6 nodes must have this address even if there is a routable address assigned to it. The IPv6 node may have more than one address.	169.254.0.0/16
fc00::/7 Example: fdf8:f53b:82e4::53	**Unique Local Addresses**	Equivalent to the IPv4 private addresses, these addresses are reserved to be used by home or	10.0.0.0/8 172.16.0.0/12 198.168.0.0/16

Prefix	Designation	Description	Equivalent IPv4 Address
First 7 bits-**1111-110** 8th bit- **0 or 1**	**(ULAs)**	enterprises within itself, and not in the public domain. Packets with Unique local addresses as the source or destination are routed by the routers within the private organization but not in Internet routers.	
2000::/3 First 3 bits – **001**	**Global unicast address**	Global Unicast Addresses are similar to public addresses in IPv4. They are routable globally by Internet routers. The details of Global unicast Address is discussed in the next section.	There is no single block equivalent address in IPv4.
ff00::/8 Example: ff01:0:0:0:0:0:0:2 First 8 bits-**11111111**	**Multicast**	The multicast addresses are only used as the destination address, never as a source address. The use of a multicast address is to define a group rather than a single host. The first 8 bits are 11111111; next 8 bits consists of 4 bits of flags and 4 bits of scope. The flag field defines whether the group is permanent or transient. The third field is the scope field, which specifies the scope parameters of the group address. The various scopes are as shown in the following figure.	224.0.0.0/4

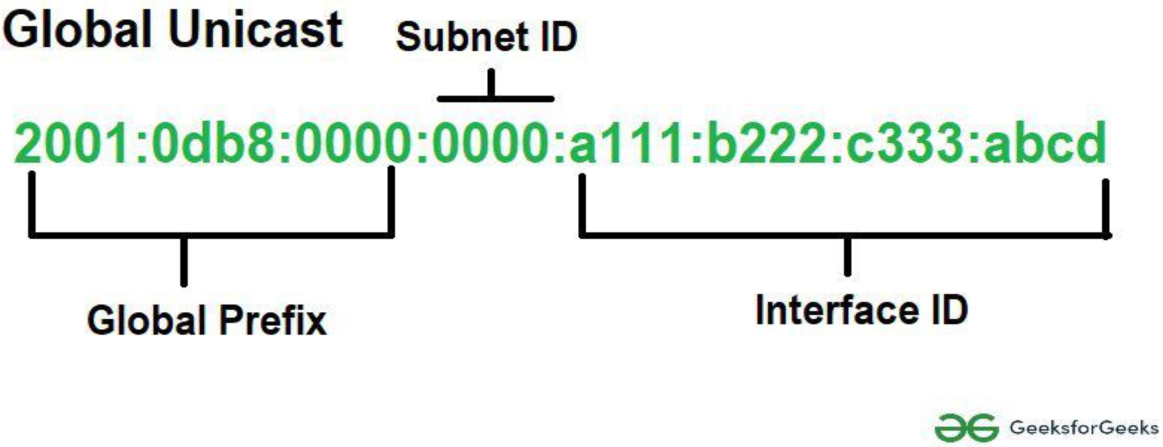
Global Unicast Address: Global Unicast Address is equivalent to IPv4 public address. Global Unicast Addresses in IPv6 are globally identifiable and uniquely addressable.

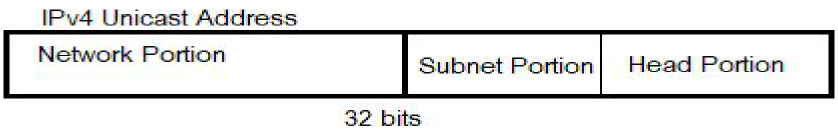


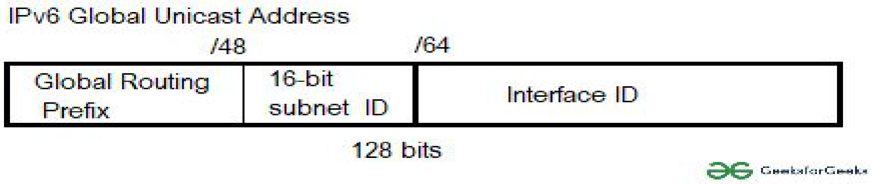
The Most significant 48-bits are designated as global routing prefix which is assigned to a specific automatic system. The three most significant bits of the global routing prefix are always set to 001.

Global Unicast Address (GUA):

- 2000::/3 (First hextet: 2000::/3 to 3FFF::/3).
- Globally unique and routable.
- Similar to public IPv4 addresses.
- 2001:db8::/32 – RFC 2839 and RFC 6890 reserve this range of addresses for documentation.







- 64-bit Interface ID = 18 quintillion (18,446,744,073,709,551,616) devices/subnet .
- 16 bit subnet ID (initially recommended) = 65,536 subnets

IPv6 Global Unicast Address Format Fields:

1. Global routing prefix: global routing prefix is the portion of the address that is assigned by the provider such as an ISP to a customer or site. The most significant 48-bits are assigned as a Global routing prefix which is assigned to a specific autonomous system.

2. Subnet ID: The subnet ID is the portion between the global routing prefix and the interface ID.

3. Interface ID: The Interface ID is equal to the host part of an IPv4 address, It is must recommend that in most cases /64 subnets must be used which creates a 64-bit ID.

Autoconfiguration: One of the interesting features of IPv6 addressing is the autoconfiguration of hosts. In IPv6, DHCP protocol can be used to allocate an IPv6 address to a host, but a host can also configure itself.
When a host in IPv6 joins a network, it can configure itself using the following process:
1. The host first creates a link local address for itself. This is by taking the 10-bit link local prefix (1111 1110 10), adding 54 zeros, and adding the 64-bit interface identifier, which any host knows how to generate it from its interface card. The result is a 128-bit link local address.

2. The host then tests to see if this link local address is unique and not used by other hosts. Since the 64-bit interface identifier is supposed to be unique, the link local address generated is unique with a high probability. However, to be sure, the host sends a neighbor solicitation message and waits for neighbor advertisement message. If any host in the subnet is using this link local address, the process fails and the host

cannot autoconfigure itself; it needs to use other means such as DHCP protocol for this purpose.

3. If the uniqueness of the link local address is passed, the host stores this address as its link-local address (for private communication), but it still needs a global unicast address. The host then sends a router solicitation message to a local router. If there is a router running on the network, the host receives a router advertisement message that includes the global unicast prefix and the subnet prefix that the host needs to add to its interface identifier to generate its global unicast address. If the router cannot help the host with the configuration, it informs the host in the router advertisement message (by setting a fl-flag). The host then needs to use other means for configuration**.**

Renumbering: To allow sites to change the service provider, renumbering of the address prefix (n) was built into IPv6 addressing. Each site is given a prefix by the service provider to which it is connected. If the site changes the provider, the address prefix needs to be changed. A router to which the site is connected can advertise a new prefix and let the site use the old prefix for a short time before disabling it. In other words, during the transition period, a site has two prefixes. The main problem in using the renumbering mechanism is the support of the DNS, which needs to propagate the new addressing associated with a domain name. A new protocol for DNS, called Next Generation DNS, is under study to provide support for this mechanism.

IPv6 Routing Protocols: Like IPv4, IPv6 also supports routing protocols that enable routers to exchange information about connected networks.

Routing protocols can be divided in two categories:

- **Interior Routing Protocol**: Protocols in this category are used within an autonomous system or organization to distribute routes among all routers inside its boundary. Examples: RIP, OSPF.

- **Exterior Routing Protocol**: An Exterior Routing Protocol distributes routing information between two different autonomous systems or organization. Examples: BGP.

There exist two forms of routing protocols:

- **Distance Vector Routing Protocol**: A router running distance vector protocol advertises its connected routes and learns new routes from its neighbors. The routing cost to reach a destination is calculated by means of hops between the source and destination. A router generally relies on its neighbor for best path selection, also known as “routing-by-rumors”. RIP and BGP are Distance Vector Protocols.
- **Link-State Routing Protocol**: This protocol acknowledges the state of a Link and advertises to its neighbors. Information about new links is learnt from peer routers. After all the routing information has been converged, the Link-State Routing Protocol uses its own algorithm to calculate the best path to all available links. OSPF and IS-IS are link state routing protocols and both of them use Dijkstra’s Shortest Path First algorithm.

IPv6 supports the following routing protocols:

1. RIPng (RIP New Generation)

2. OSPFv3

3. EIGRP for IPv6

4. IS-IS for IPv6

5. MP-BGP4 (Multiprotocol BGP-4)

RIPng: RIPng stands for Routing Information Protocol Next Generation. It is the **Next Generation IP, IPv6** available next level protocol of RIPv2. This is an Interior Routing Protocol and is a Distance Vector Protocol. RIPng has been upgraded to support IPv6.

RIP Next Generation has below similar characteristic as RIPv2:

- Distance vector protocol
- Uses "Hop count" as metric
- The default administrative Distance is 120,
- Uses Split Horizonin, Poison Reverse
- Uses periodic (30 seconds) and triggered

RIPng is almost the same as RIPv2. But there are some additional differences because of the supported IP version (IPv6). These differences one by one:

i) **IPv6 Support:** RIPv2 is the protocol that is used with IPv4. With this new version of RIP, **RIP Next Generation**, IPv6 addresses are supported. The neighbourship is established with the IPv6 addresses.

Here, with the support of the IPv6 addresses, some of the commands are added. So, similar but different commands are used for RIP Next Generation.

ii) **No Network But Interface:** RIPng is established over interfaces. If an interface needed to be in the **RIP Next Generation** network, RIPng network is added under this interface. In RIPv2, we use network command to add subnets (networks) to the RIP network. But with **RIP Next Generation**, like other IPv6 routing protocols, interfaces individually used as member of **RIP Next Generation** network and RIPng membership command is added under interface configuration.

iii) **UDP Port 521:** RIPv2 uses UDP port 520. This port changes a little with **RIP Next Generation**. RIPng uses UDP port 521.

OSPFv3: OSPFv3 adds support for IPv6 in the Open Shortest Path First (OSPF) routing protocol, as detailed in RFC 2740OSPFv3 is the IPv6-capable version of the OSPF routing protocol although the foundation remains the same as in IPv4 and OSPFv2. It is an Interior Routing Protocol which is modified to support IPv6. This is a Link-State Protocol and uses Djikrasta's Shortest Path First algorithm to calculate best path to all destinations.

OSPFv3 also supports the same interface types, including broadcast, point-to-point, point-to-multipoint, NBMA, and virtual links.

Similarities between OSPFv2 and OSPFv3:

- Both are link-state Interior Gateway Protocol (IGP) routing protocols
- Both use a 2-level hierarchy with Area 0.0.0.0 at the core
- Both use Area Border Routers (ABRs) and Autonomous System Boundary Routers (ASBRs)
- Both use the Shortest Path First (SPF) calculation within each area utilizing Edsger Dijkstra's SPF algorithm
- Both use metrics that are based on interface bandwidth (or manual configuration)
- Both have 5 common protocol packet types: Hello, Database description (DBD), Link-state request (LSR), Link-state update (LSU), Link-state acknowledgment (LSA)
- They use similar interface types: Broadcast, P2P, P2MP, NBMA, and Virtual-Links
- They have the same LSA flooding and aging timers

Differences between OSPFv2 and OSPFv3:

- OSPFv3 introduces new LSA types
- OSPFv3 has different packet format
- OSPFv3 uses different flooding scope bits (U/S2/S1)
- OSPFv3 adjacencies are formed over link-local IPv6 communications
- OSPFv3 runs per-link rather than per-subnet
- OSPFv3 supports multiple instances on a single link, Interfaces can have multiple IPv6 addresses
- OSPFv3 uses multicast addresses FF02::5 (all OSPF routers), FF02::6 (all OSPF DRs)
- OSPFv3 Neighbor Authentication done with IPsec (AH)
- OSPFv2 Router ID (RID) must be manually configured, still a 32-bit number

How OSPFv3 Works: Think of a link as being an interface on a networking device. A link-state protocol makes its routing decisions based on the states of the links that connect source and destination machines. The state of a link is a description of that interface and its relationship to its neighboring networking devices. The interface information includes the IPv6 prefix of the interface, the network mask, the type of network it is connected to, the devices connected to that network, and so on. This information is propagated in various type of link-state advertisements (LSAs).
A device's collection of LSA data is stored in a link-state database. The contents of the database, when subjected to the Dijkstra algorithm, result in the creation of the OSPF routing table. The difference between the database and the routing table is that the database contains a complete collection of raw data; the routing table contains a list of shortest paths to known destinations via specific device interface ports.
Most configuration and operational commands function essentially the same as in OSPFv2:

All OSPFv3 operational and configuration commands include the identifier ospf3 in place of the familiar ospf option. For example, show ospf database in OSPFv2 becomes show ospf3 database in OSPFv3.
OSPFv3 Router IDs, Area IDs, and LSA link-state IDs remain at the OSPFv2 IPv4 size of 32 bits.
Link-local scope—The OSPFv3 packet is flooded to the members of a link.
Area scope—The OSPFv3 packet is flooded to all members of an OSPFv3 area.
AS scope—The OSPFv3 packet is flooded to all members of an AS.

Authentication has been removed from the OSPFv3 protocol itself and relies on the authentication header (AH) and Encapsulating Security Payload (ESP) portions of the IP Security (IPsec) protocol for all authentication tasks in IPv6.

BGPv4: BGPv4 stands for Border Gateway Protocol version 4. BGP is the only open standard Exterior Gateway Protocol available. BGP is a Distance Vector protocol which takes Autonomous System as calculation metric, instead of the number of routers as Hop. BGPv4 is an upgrade of BGP to support IPv6 routing.

BGP-4 is an extension of BGP-3 (BGP version 3), and it is a major feature that it supports CIDR (Classless Inter-Domain Routing). BGP-4 has the following features.

- **Using TCP:** In order to have reliability, the route information is exchanged using TCP connection. As long as no error occurs, the connection will not be closed and the same connection will be used permanently. One connection is required for one peer.
- **Advertising only differences**: On routing protocols such as RIP, the router periodically advertises all routes. However, on BGP-4, it advertises only the difference when the route changed. This means that if there is no change in the route, the route information will not flow. However, using keepalive makes enable to periodically flow messages that do not contain the route information.
- **Controlling the route with the Path Vector method**: A route is represented by a sequence of AS numbers called Path. By looking at the Path, you can find useful information such as detecting the loops and comparing the length of

routes. In BGP-4, the element of the information representing the route is called **Path Attribute**.

EIGRP: “**EIGRP for IPv6**” as the enhancements to **EIGRP** that is done for **IPv6 support**. Basically, Enhanced Interior Gateway Routing Protocol is the same protocol but it has some additional properties that provides IPv6 availability.

Key characteristics of IPv6 Routing Protocol**.**

EIGRP for IPv6 Runs on Links Instead Networks: In Enhanced Interior Gateway Routing Protocol, networks are added to the EIGRP process. But with EIGRP for IPv6, interfaces are added to the EIGRP process. In other words, there is no “network” command in EIGRP. Instead, a per-interface based association is done.

Being under the same subnet is not important in **EIGRP neighbourship**. Even if they are in different subnets, a neighbourship can be established between these nodes. For Enhanced Interior Gateway Routing Protocol neighbourship, link-local addresses are used.

EIGRP Commands Evolve: The configuration commands are also changing with the new enhanced version of Enhanced Interior Gateway Routing Protocol. Here, IPv6 commands are coming for both **EIGRP** configuration and verification.

So, for the global IPv6 enable, we use “**IPv6 unicast-routing**” command. This is required to enable IPv6 addresses globally. To start the process, we use “**ipv6 router eigrp process-name**” command.

Multicast Address of EIGRP: Enhanced Interior Gateway Routing Protocol **uses multicast address of** 224.0.0.10**. I**n the new version, this is also changed to its IPv6 version. IPv6 EIGRP uses IPv6 version of this address. This address **is** FF02:0:0:0:0:0:0:A (FF02::A) **.** EIGRP for IPv6 has a lot in common with EIGRP for IPv4 and some differences.

Commonalities:

- Uses layer 3 header protocol type of 88
- Uses successor and feasible successor log
- Uses DUAL
- Uses triggered updates
- Uses composite metric, default using bandwidth & delay.
- A metric of 2^32 – 1 = Infinity

Differences:

- Uses the neighbor’s link-local address as the next-hop IP address

- EIGRP for IPv6 cannot do auto-summarization due to no classful netwoks in v6.
- Neighbors are not required to be in the same subnet
- FF02::A is used as the multicast update address

Protocols Changed to Support IPv6

- **ICMPv6**: Internet Control Message Protocol version 6 is an upgraded implementation of ICMP to accommodate IPv6 requirements. This protocol is used for diagnostic functions, error and information message, statistical purposes. ICMPv6's Neighbor Discovery Protocol replaces ARP and helps discover neighbor and routers on the link.

- **DHCPv6**: Dynamic Host Configuration Protocol version 6 is an implementation of DHCP. IPv6 enabled hosts do not require any DHCPv6 Server to acquire IP address as they can be auto-configured. Neither do they need DHCPv6 to locate DNS server because DNS can be discovered and configured via ICMPv6 Neighbor Discovery Protocol. Yet DHCPv6 Server can be used to provide these information.

- **DNS**: There has been no new version of DNS but it is now equipped with extensions to provide support for querying IPv6 addresses. A new AAAA (quad-A) record has been added to reply IPv6 query messages. Now the DNS can reply with both IP versions (4 & 6) without any change in the query format.

IPV4 to IPV6 Tunneling : The basic idea behind tunneling methods is that IPv6 will be tunneled over an existing IPv4 network. A number of different tunneling methods are available and can be selected based on the requirements of the situation.

Tunneling provides a way to use an existing **IPv4** routing infrastructure to carry **IPv6** traffic.

The key to a successful **IPv6** transition is compatibility with the existing installed base of **IPv4** hosts and routers. Maintaining compatibility with **IPv4** while deploying **IPv6** streamlines the task of transitioning the Internet to **IPv6**. While the **IPv6** infrastructure is being deployed, the existing **IPv4** routing infrastructure can remain functional, and can be used to carry **IPv6** traffic.

IPv6 or **IPv4** hosts and routers can tunnel **IPv6** datagrams over regions of **IPv4** routing topology by encapsulating them within **IPv4** packets. Tunneling can be used in a variety of ways:

Item	Description
Router-to-Router	**IPv6** or **IPv4** routers interconnected by an **IPv4** infrastructure can tunnel **IPv6** packets between themselves. In this case, the tunnel spans one segment of the end-to-end path that the **IPv6** packet takes.

Item	**Description**
Host-to-Router	**IPv6** or **IPv4** hosts can tunnel **IPv6** packets to an intermediary **IPv6** or **IPv4** router that is reachable through an **IPv4** infrastructure. This type of tunnel spans the first segment of the packet's end-to-end path.
Host-to-Host	**IPv6** or **IPv4** hosts that are interconnected by an **IPv4** infrastructure can tunnel **IPv6** packets between themselves. In this case, the tunnel spans the entire end-to-end path that the packet takes.
Router-to-Host	**IPv6**/**IPv4** routers can tunnel **IPv6** packets to their final destination **IPv6** or **IPv4** host. This tunnel spans only the last segment of the end-to-end path.

Tunneling techniques are usually classified according to the mechanism by which the encapsulating node determines the address of the node at the end of the tunnel. In router-to-router or host-to-router methods, the **IPv6** packet is being tunneled to a router. In host-to-host or router-to-host methods, the **IPv6** packet is tunneled all the way to its final destination.

The entry node of the tunnel (the encapsulating node) creates an encapsulating **IPv4** header and transmits the encapsulated packet. The exit node of the tunnel (the decapsulating node) receives the encapsulated packet, removes the **IPv4** header, updates the **IPv6** header, and processes the received **IPv6** packet. However, the encapsulating node needs to maintain soft state information for each tunnel, such as the maximum transmission unit (MTU) of the tunnel, to process **IPv6** packets forwarded into the tunnel.

There are two types of tunnels in **IPv6**:

Automatic tunnels

Automatic tunnels are configured by using **IPv4** address information embedded in an **IPv6** address – the **IPv6** address of the destination host includes information about which **IPv4** address the packet should be tunneled to.

Configured tunnels

Configured tunnels must be configured manually. These tunnels are used when using **IPv6** addresses that do not have any embedded **IPv4** information. The **IPv6** and **IPv4** addresses of the endpoints of the tunnel must be specified.

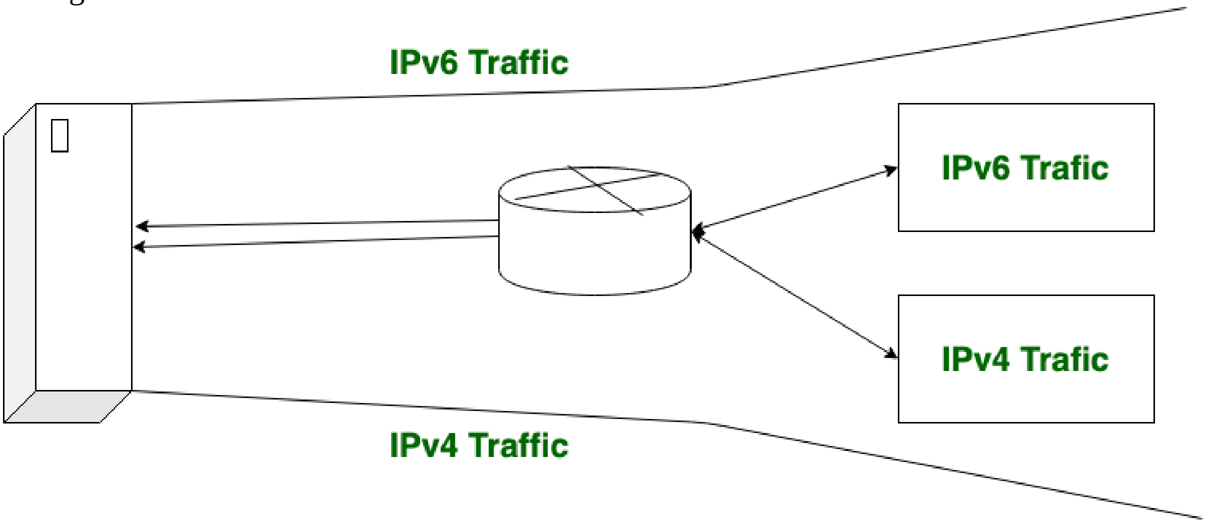
Transition from IPv4 to IPv6 address: The idea behind translation is that at a boundary router between an IPv4 and an IPv6 network a translation process maps an IPv4 address to an IPv6 address (or vice versa).

Various organization is currently working with IPv4 technology and in one day we can’t switch directly from IPv4 to IPv6. Instead of only using IPv6, we use combination of both and transition means not replacing IPv4 but co-existing of both.

Complete transition from IPv4 to IPv6 might not be possible because IPv6 is not backward compatible. This results in a situation where either a site is on IPv6 or it is not. It is unlike implementation of other new technologies where the newer one is backward compatible so the older system can still work with the newer version without any additional changes.

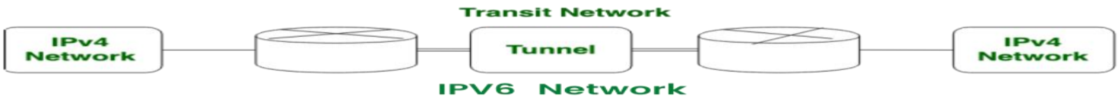
To overcome this short-coming, we have a few technologies that can be used to ensure slow and smooth transition from IPv4 to IPv6.

1. Dual-Stack Routers:
In dual-stack router, A router's interface is attached with IPv4 and IPv6 addresses configured are used in order to transition from IPv4 to IPv6.



In this above diagram, A given server with both IPv4 and IPv6 addresses configured can communicate with all hosts of IPv4 and IPv6 via dual-stack router (DSR). The dual stack router (DSR) gives the path for all the hosts to communicate with the server without changing their IP addresses.

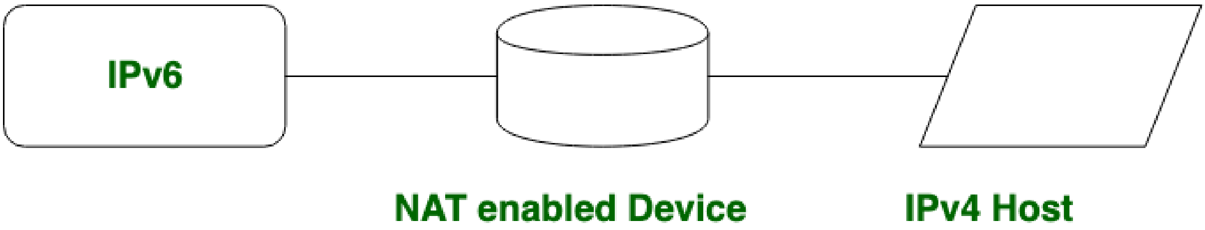
2. Tunneling:
Tunneling is used as a medium to communicate the transit network with the different IP versions.



In this above diagram, the different IP versions such as IPv4 and IPv6 are present. The IPv4 networks can communicate with the transit or intermediate network on IPv6 with the help of the Tunnel. It's also possible that the IPv6 network can also communicate with IPv4 networks with the help of a Tunnel.

3. NAT Protocol Translation: With the help of the NAT Protocol Translation technique, the IPv4 and IPv6 networks can also communicate with each other which do not understand the address of different IP version.

Generally, an IP version doesn't understand the address of different IP version, for the solution of this problem we use NAT-PT device which removes the header of first (sender) IP version address and add the second (receiver) IP version address so that the Receiver IP version address understand that the request is sent by the same IP version, and its vice-versa is also possible.



In the above diagram, an IPv4 address communicates with the IPv6 address via a NAT-PT device to communicate easily. In this situation, the IPv6 address understands that the request is sent by the same IP version (IPv6) and it responds.

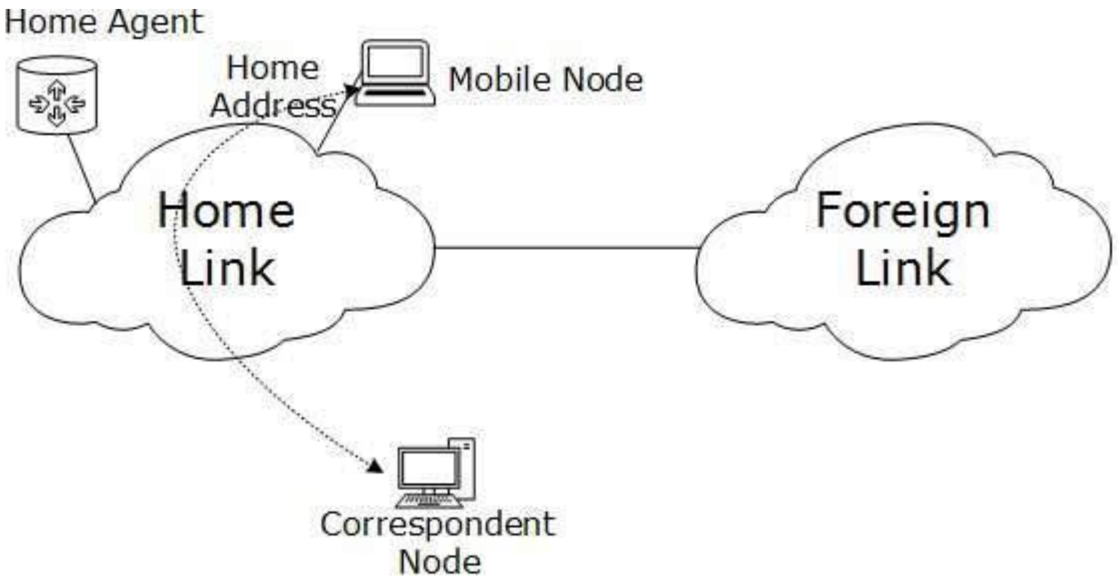
IPV6 mobility: When a host is connected to a link or network, it acquires an IP address and all communication take place using that IP address on that link. As soon as, the same host changes its physical location, that is, moves into another area / subnet / network / link, its IP address changes accordingly, and all the communication taking place on the host using old IP address, goes down.

IPv6 mobility provides a mechanism for the host to roam around different links without losing any communication/connection and its IP address.

Multiple entities are involved in this technology:

- **Mobile Node**: The device that needs IPv6 mobility.
- **Home Link**: This link is configured with the home subnet prefix and this is where the Mobile IPv6 device gets its Home Address.
- **Home Address**: This is the address which the Mobile Node acquires from the Home Link. This is the permanent address of the Mobile Node. If the Mobile Node remains in the same Home Link, the communication among various entities take place as usual.
- **Home Agent**: This is a router that acts as a registrar for Mobile Nodes. Home Agent is connected to Home Link and maintains information about all Mobile Nodes, their Home Addresses, and their present IP addresses.
- **Foreign Link**: Any other Link that is not Mobile Node's Home Link.
- **Care-of Address**: When a Mobile Node gets attached to a Foreign Link, it acquires a new IP address of that Foreign Link's subnet. Home Agent maintains the information of both Home Address and Care-of Address. Multiple Care-of addresses can be assigned to a Mobile Node, but at any instance, only one Care-of Address has binding with the Home Address.
- **Correspondent Node**: Any IPv6 enabled device that intends to have communication with Mobile Node.

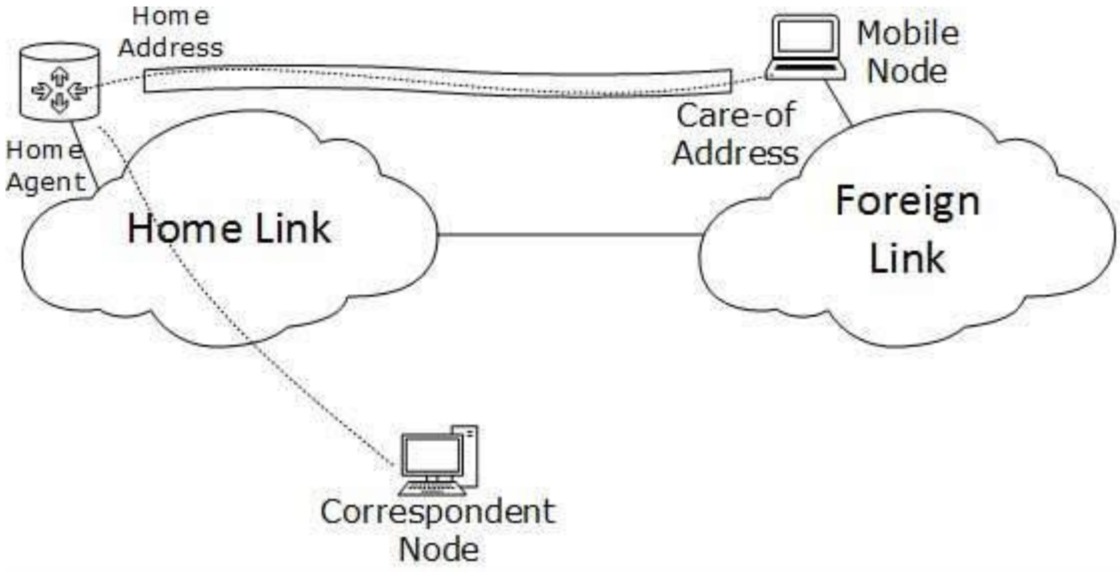
Mobility Operation: When Mobile Node stays in its Home Link, all communications take place on its Home Address as shown below:



[*Image: Mobile Node connected to Home Link*]

When a Mobile Node leaves its Home Link and is connected to some Foreign Link, the Mobility feature of IPv6 comes into play. After getting connected to a Foreign Link, the Mobile Node acquires an IPv6 address from the Foreign Link. This address is called Care-of Address. The Mobile Node sends a binding request to its Home Agent with the new Care-of Address. The Home Agent binds the Mobile Node's Home Address with the Care-of Address, establishing a Tunnel between both.

Whenever a Correspondent Node tries to establish connection with the Mobile Node (on its Home Address), the Home Agent intercepts the packet and forwards to Mobile Node's Care-of Address over the Tunnel which was already established.



[*Image: Mobile Node connected to Foreign Link*]

Route Optimization: When a Correspondent Node initiates a communication by sending packets to Mobile the Node on the Home Address, these packets are tunneled to the Mobile Node by the Home Agent. In Route Optimization mode, when the Mobile Node receives a packet from the Correspondent Node, it does not forward replies to the Home Agent. Rather, it sends its packet directly to the Correspondent Node using Home Address as Source Address. This mode is optional and not used by default.