Internet Protocol Version 6 (IPv6)

- **I**P version 6 (IPv6) is a new version of the Internet Protocol, designed as a successor to IP version 4 (IPv4).

Longer Address Fields

- Length of the address field is extended from 32 bits to 128 bits.
- Also provides a hierarchical structure in the address format.

Header Format Simplification

- Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv6 header.
- Fields such as checksum, IHL, Identification, flags and Fragment Offset have been dropped.

Improved Support for Extensions and Options

- Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.
- Options in Ipv6 are specified in optional **Extension Headers**.

Flow Labeling Capability

- A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service.
-
- This is how **Quality of Service** (QoS) is implemented.

Fragmentation at Source Only

- Routers will not fragment packets; fragmentation must be done at the source.

Large Packets

- Supports payloads that are larger than 64 Kbytes. Referred to as **Jumbo** payloads.

Authentication and Privacy Capabilities

- Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.

No Checksum Field

- Error detection is left to higher and lower level protocols in order to reduce processing time.
- Ethernet, TCP, and UDP all perform their own error detection anyway.

Ipv6 Header Format:

Version

- 4-bit Internet Protocol version number = 6.

Traffic Class

- 4-bit priority value that is used to provide differentiated service.

Flow Label

- 24-bit flow label that is used to specify a QoS requested by the source.

Payload Length

- 16-bit unsigned integer. Length of payload i.e., the rest of the packet following the IPv6 header, in octets.
- If zero, indicates that the payload length is carried in a Jumbo Payload hop-by-hop option.

Next Header (8-bit selector)

- Identifies the type of header immediately following the IPv6 header.
- Similar to the options field in the IPv4 header, but more flexible and efficient.

Hop Limit (8-bit unsigned integer)

- Decremented by 1 by each node that forwards the packet.
- The packet is discarded if Hop Limit is decremented to zero.

Source/Destination Address

- 128-bit address of the source and destination of the packet.
- Overcomes the 32-bit limitation imposed by Ipv4.

Expanded Addressing Space and Increased Addressing Flexibility

- IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses.
- IPv6 defines three types of addresses and these are described next.

Unicast Addresses

- This scheme contains the entire functionality of IPv4's three main addressing classes (A, B, and C).
- These are structured in several ways as shown by the diagram.

Provider-Based Global Unicast Address

- Provides for a global addressing scheme. There are five main fields as described below.

- **Registry ID**
  -
  - Identifies the registration authority, which assigns the provider portion of the address.

- **Provider ID**
  -
  - The Internet Service Provider, which assigns the subscriber portion of the address.

- **Subscriber ID**
    -
    - Specifies the particular subscriber attached to the provider portion of the address.

- **Subnet ID**
    -
    - Specifies a particular subnet (a group of hosts) within the whole subscriber network.

- **Node ID**
    -
    - Specifies a single host interface among the group of host interfaces specified by the subnet prefix.

Link-Local Addresses

- These are used for addressing on a single link or subnet.
- Cannot be used as part of the global addressing scheme.
- The idea here is to enable a host to construct an address that will work on a subnet without having to worry about the global uniqueness of the address.
- This is also very useful for autoconfiguration.

Site-Local Addresses

- These are designed for local use but formatted in such a way as to allow eventual integration into the global address scheme.

Embedded IPv4

- IPv6 needs to be deployed incrementally in such a way that hosts and routers that can only understand IPv4 can continue to function for as long as possible.
- This scheme addresses the key issue of the transition of IPv4 to IPv6. That is, it provides a means for IPv4 addresses to exist within the IPv6 scheme.
- An embedded IPv4 address consists of a 32-bit IPv4 address in the lower-order 32 bits prefixed by either 96 zeroes or prefixed by 80 zeros followed by 16 ones.

Loopback Address

- This is the Unicast address 0:0:0:0:0:0:0:1 can be used by a host to send an IPv6 packet to itself.
- This is used for testing purposes. The packet remains within the host and cannot be sent into the network.

Anycast Addresses

- A new type of address called an "anycast address" is defined, used to send a packet to any one of a group of nodes via a single address.
- An anycast address is assigned to a set of interfaces, and packets sent to that address will go to the nearest set of interfaces as determined by the routing protocols.

Multicast Addresses

- An IPv6 multicast address is an identifier for a group of nodes and has the following format:

8 bits	4 bits	4 bits	112 bits
11111111	flgs	scop	group ID

- The motivation for having this type of address is to allow applications to send a packet to a group of destination hosts simultaneously.
- Hosts can join or leave this multicast group at will, without affecting other members of the set.
- A host can also belong to more than one multicast group.
- The fields have the following meanings:

  11111111 at the start of the address identifies the address as being a multicast address.

  flgs is a set of 4 flags.

  The high-order 3 flags are reserved, and must be initialized to 0.

  The value 0 for the low-order flag indicates a permanently-assigned ("well-known") multicast address, assigned by the global Internet numbering authority.

  The value 1 for the low-order flag indicates a non-permanently-assigned ("transient") multicast address.

- The scalability of multicast routing is improved by adding a "**scope**" field to multicast addresses.

- The **4-bit scope** value can be used to expand or limit the scope of the multicast group. The values are:

0	Reserved
1	Node-local scope
2	Link-local scope
3, 4	(unassigned)
5	Site-local scope
6, 7	(unassigned)
8	Organization-local scope
9 - D	(unassigned)
E	Global scope
F	Reserved

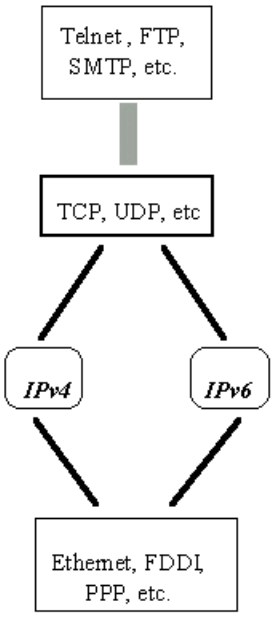
- **group ID** identifies the multicast group.

Ipv4 Compatibility and Coexistence

- Ipv6 needs to be deployed incrementally in such a way that hosts and routers that only understand IPv4 can continue to function for as long as possible.
- There are two mechanisms defined to facilitate the compatibility between IPv4 and IPv6.

Dual-Stack Operation

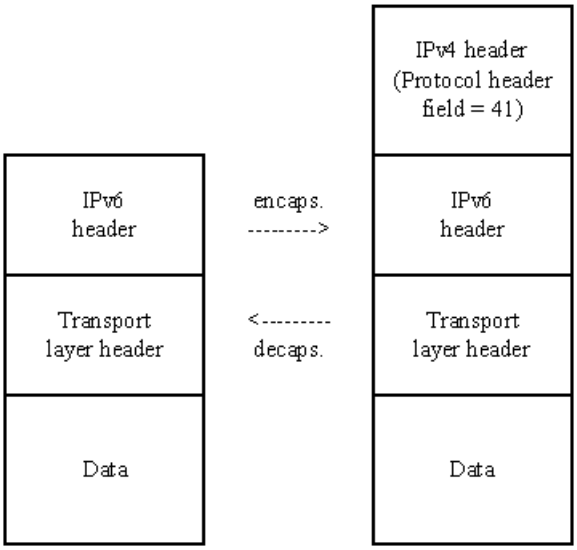
- The most straightforward way for IPv6 nodes to remain compatible with IPv4-only nodes is by providing a complete IPv4 implementation.
- Such nodes are called IPv6/IPv4 nodes, and have the ability to send and receive both
- The node uses the Version field in the packet to decide which stack should process an arriving packet.
- The protocol layering in IPv6/IPv4 dual nodes is represented as follows:



- Note that the Dual IP layer technique may or may not be used in conjunction with the IPv6-over-IPv4 tunneling techniques, which are described next.

Tunneling

- IPv6 packets can be carried across segments of an IPv4-complete topology by using the IPv6-over-IPv4 tunneling technique.



- An IPv6/IPv4 node that has IPv4 reachability to another IPv6/IPv4 node may send IPv6 packets to that node by encapsulating them within IPv4 packets as shown below.

- In order for this technique to work, both nodes must be assigned IPv4-compatible IPv6 addresses.
- This is necessary because the low-order 32-bits of those addresses are used as source and destination addresses of the encapsulating IPv4 packet.
- Two types of tunneling are used. "**Automatic tunnels**" are used to deliver IPv6 packets all the way to their end destinations.
- "**Configured tunnels**" are used to deliver IPv6 packets to an intermediary IPv6/IPv4 router.
- Both types of tunneling make use of the IPv4 address embedded in IPv4-compatible IPv6 addresses.
- In **automatic tunneling**, the tunnel endpoint address is taken from the IPv4 address embedded in the IPv6 destination address. No additional configuration information is needed because the destination address is carried in the IPv6 packet being tunneled.
- In **configured tunneling**, the tunnel endpoint address is that of an intermediate IPv6/IPv4 router. That address must be configured using configuration information from of a routing table entry on a host, or neighbor configuration information on a router.

Address Autoconfiguration

- A very important goal for IPv6 is to provide support for a host to automatically configure one or more interfaces, thus providing "plug-and-play" operation.

- This feature would allow a user to attach a host to the subnet and have IPv6 addresses attached automatically to its network interface card.

- There are three models of address assignment.

Local Scope Model

- Designed for use on a network without routers.
- The IPv6 address can be the MAC address or an address assigned using another algorithm.

Stateless Server Model

- A very simple technique requiring minimal system administration support.
- A new device sends a configuration request packet to a local well-known multicast address. The request includes a unique ID such as its MAC address.
- An address server will receive the request and constructs an IPv6 address for the device based on its knowledge of the network and sends it back to the device.

Stateful Server Model

- Requires greater administrative system control since it retains address-assignment information.
- The new device sends a configuration request packet containing a unique ID such as its MAC address.
- The address server will use this ID to look up information in a database and create a new IPv6 address.
- The new address is given a life-time value, after which the host must request an IPv6 address again.
- Using this model addresses in a subnet can be reconfigured quite easily.

Improved Header Options

- In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the **IPv6 header** and the **upper- layer header** in a packet.

- A full implementation of IPv6 includes implementation of the following extension headers:

    - Hop-by-Hop Options (Code 0)
    - Routing (Code 43)
    - Fragment (Code 44)
    - Authentication Header (Code 51)
    - Encapsulating Security Payload (Code 52)
    - Destination Options (Code 60)

Hop-by-Hop Options Header

- The Hop-by-Hop Options header is used to carry optional information that must be examined by every node along a packet's delivery path.

- The header consists of the following fields:

- **Next Header**

    - 8-bit selector. Identifies the type of header immediately following the Hop-by-Hop Options header. Uses the same values as the IPv4 Protocol field.

- **Header Extension Length**

    - 8-bit unsigned integer. Length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets.

- **Options**
    - Variable-length field, of length such that the complete **Hop-by-Hop** Options header is integer multiple of 8 octets long.
    - The **Option Type** identifiers are internally encoded such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type:

      00 - skip over this option and continue processing the header.
      01 - discard the packet.
      10 - discard the packet and, regardless of whether or not the packet's destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.
      11 - discard the packet and, only if the packet's Destination Address was not a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

Large Packet

- "Jumbo Payload" option is used to send IPv6 packets with payloads longer than 65,535 octets.
- This is very useful for video conferencing applications that require large packets.

Routing Header

- The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination.
- The Routing header is identified by a Next Header value of 43 in the immediately preceding header, and has the following format:
- **Next Header**
    - 8-bit selector. Identifies the type of header immediately following the Routing header.
- **Routing Type**
    - 8-bit unsigned integer. Set to zero.
- **Segment Left**
    - 8-bit selector. Number of route segments (addresses) in the routing header. Maximum = 23.
    - Each router decrements this value by 1 until it reaches its destination.

- **Reserved**
    - 8 bit selector. For future use.
- **Strict/Loose Bit Mask**
    - 24-bit bit-map, numbered 0 to 23, left-to-right. Indicates, for each segment of the route, whether or not the next destination address must be a neighbor of the preceding address.
    - 1 means strict (must be a neighbor).
    - 0 means loose (need not be a neighbor).

<u>Fragment Header</u>

- The Fragment header is used by an IPv6 source to send packets larger than would fit in the path MTU to their destinations. (Note: unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path).
- The Fragment header is identified by a Next Header value of 44 in the immediately preceding header, and has the following format:
- **Next Header**
    - 8-bit selector. Identifies the initial header type of the Fragmentable Part of the original packet.
- **Reserved**
    - 8-bit reserved field. Initialized to zero for transmission; ignored on reception.
- **Fragment Offset**
    - 13-bit unsigned integer. The offset, in 8-octet units, of the data following this header, relative to the start of the Fragmentable Part of the original packet.
- **Res**
    - 2-bit reserved field. Initialized to zero for transmission; ignored on reception.
- **M flag**
    - 1 = more fragments; 0 = last fragment.

- **Identification**
    - 32 bits. Uniquely identifies the original packet. For every packet that is to be fragmented, the source node generates an Identification value.
    - The Identification must be different than that of any other fragmented packet sent recently with the same Source Address and Destination Address.

Destination Options Header

- The Destination Options header is used to carry optional information that need be examined only by a packet's destination node(s).
- The Destination Options header is identified by a Next Header value of 60 in the immediately preceding header, and has the following format:
- **Next Header**
    - 8-bit selector. Identifies the type of header immediately following the Destination Options header.
- **Header Extension Length**
    - 8-bit unsigned integer. Length of the Destination Options header in 8-octet units, not including the first 8 octets.
- **Options**
    - Variable-length field, of length such that the complete Destination Options header is an integer multiple of 8 octets long.
    - Contains one or more encoded options, as described earlier.