

Module – 5

NETWORK LAYER

5.1 IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

Address Space

- An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2^N because each bit can have two different values (0 or 1) and N bits can have 2^N values.
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion).

Notations

There are two prevalent notations to show an IPv4 address: binary notation and dotted decimal notation.

1) Binary Notation:

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. The following is an example of an IPv4 address in binary notation:

01110101 10010101 00011101 00000010

2) Dotted-Decimal Notation:

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted decimal notation of the above address:

117.149.29.2

Example:

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

Solution:

- a. 129.11.11.239
- b. 193.131.27.255

Classful Addressing

IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing. In classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.

	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10			
Class C	110			
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Classes and Blocks

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size.

<i>Class</i>	<i>Number of Blocks</i>	<i>Block Size</i>	<i>Application</i>
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

- Class A addresses were designed for large organizations with a large number of attached hosts or routers.
- Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.

- Class C addresses were designed for small organizations with a small number of attached hosts or routers.
- Class D addresses were designed for multicasting.
- Class E addresses were reserved for future use.
- In classful addressing, a large part of the available addresses were wasted.

Netid and Hostid

- In classful addressing, an IP address in class A, B, or C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address.
- In class A, one byte defines the netid and three bytes define the hostid.
- In class B, two bytes define the netid and two bytes define the hostid.
- In class C, three bytes define the netid and one byte defines the hostid.

Mask

Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous 1s followed by contiguous as.

<i>Class</i>	<i>Binary</i>	<i>Dotted-Decimal</i>	<i>CIDR</i>
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.

Subnetting: If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors.

Supernetting: In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork or a supernet.

Address Depletion

The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations. One solution that has alleviated the problem is the idea of classless addressing.

Classless Addressing

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

Address Blocks

In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:

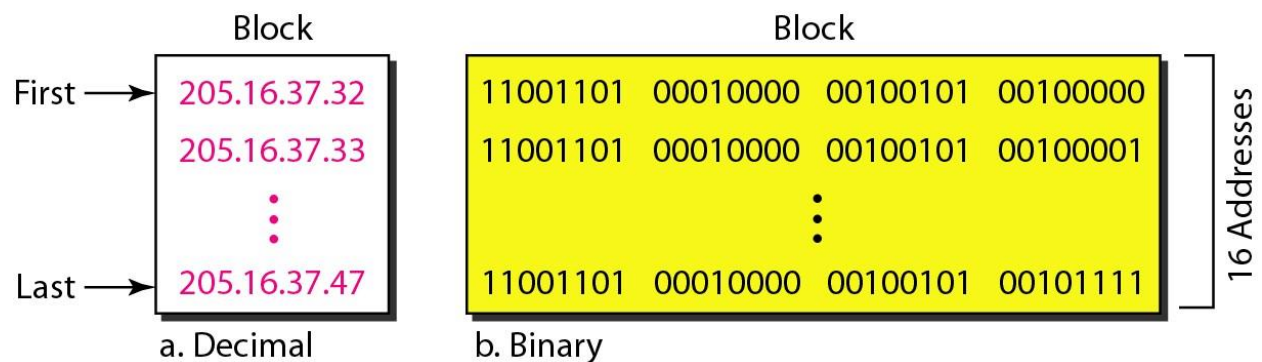
1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (1, 2, 4, 8, ...).
3. The first address must be evenly divisible by the number of addresses.

Mask

- In classless addressing the mask for a block can take any value from 0 to 32. It is very convenient to give just the value of n preceded by a slash (CIDR notation).
- In IPv4 addressing, a block of addresses can be defined as $x.y.z.t/n$ in which $x.y.z.t$ defines one of the addresses and the $/n$ defines the mask.
- The first address in the block can be found by setting the rightmost $32 - n$ bits to 0s.
- The last address in the block can be found by setting the rightmost $32 - n$ bits to 1s.
- The number of addresses in the block can be found by using the formula 2^{32-n}

Network Addresses

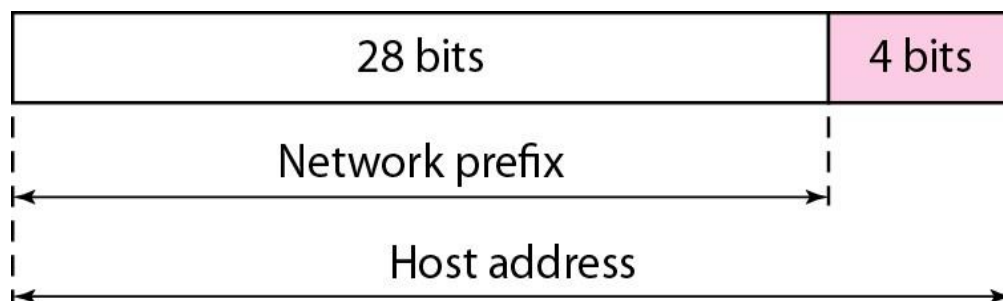
When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet. The first address in the class, however, is normally (not always) treated as a special address. The first address is called the network address and defines the organization network. It defines the organization itself to the rest of the world.



Hierarchy

Two-Level Hierarchy: No Subnetting

- An IP address can define only two levels of hierarchy when not subnetted.
- The n leftmost bits of the address $x.y.z.t/n$ define the network (organization network); the $32 - n$ rightmost bits define the particular host (computer or router) to the network.
- The part of the address that defines the network is called the prefix; the part that defines the host is called the suffix.



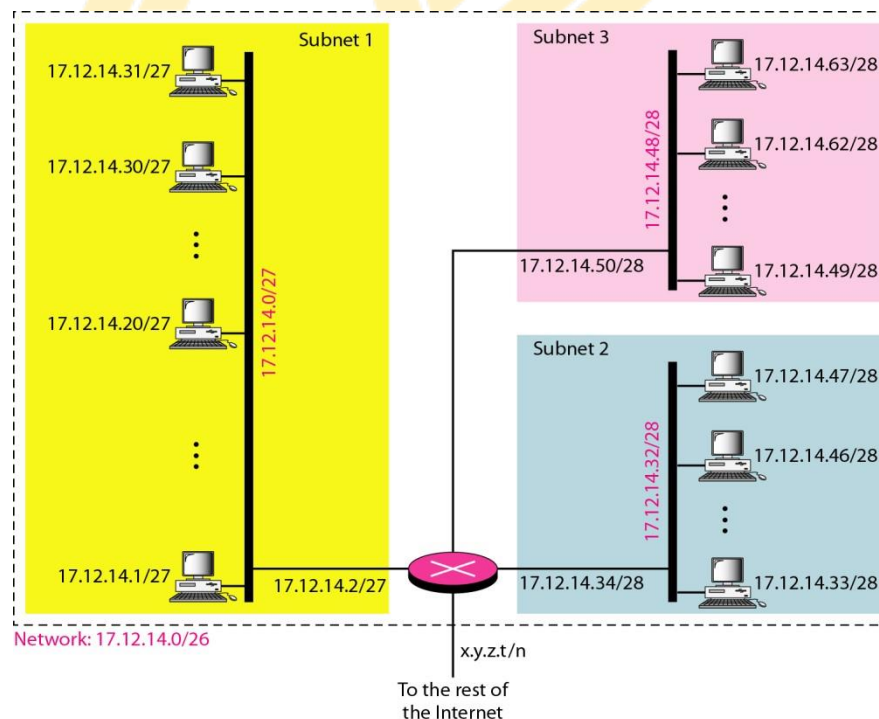
Three-Levels of Hierarchy: Subnetting

- An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets. The rest of the world still sees the organization as one entity; however, internally there are several subnets.
- All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets.
- The organization, however, needs to create small sub blocks of addresses, each assigned to specific subnets. The organization has its own mask; each subnet must also have its own.

Example:

Suppose an organization is given the block 17.12.40.0/26, which contains 64 addresses. The organization has three offices and needs to divide the addresses into three sub blocks of 32, 16, and 16 addresses. We can find the new masks by using the following arguments:

1. Suppose the mask for the first subnet is n_1 , then 2^{32-n_1} must be 32, which means that $n_1 = 27$.
2. Suppose the mask for the second subnet is n_2 , then 2^{32-n_2} must be 16, which means that $n_2 = 28$.
3. Suppose the mask for the third subnet is n_3 , then 2^{32-n_3} must be 16, which means that $n_3 = 28$.



- a. In subnet 1, the address 17.12.14.29/27 can give us the subnet address if we use the mask /27 because

Host: 00010001 00001100 00001110 00011101

Mask: /27

Subnet: 00010001 00001100 00001110 00000000 (17.12.14.0)

- b. In subnet 2, the address 17.12.14.45/28 can give us the subnet address if we use the mask /28 because

Host: 00010001 00001100 00001110 00101101

Mask: /28

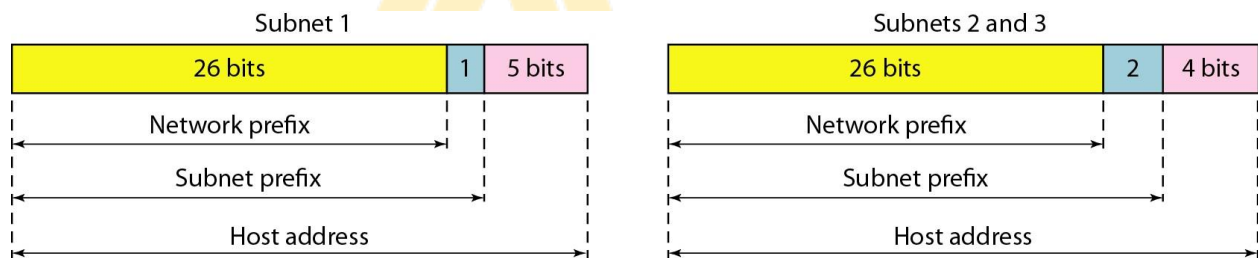
Subnet: 00010001 00001100 00001110 00100000 (17.12.14.32)

- c. In subnet 3, the address 17.12.14.50/28 can give us the subnet address if we use the mask /28 because

Host: 00010001 00001100 00001110 00110010

Mask: /28

Subnet: 00010001 00001100 00001110 00110000 (17.12.14.48)



More Levels of Hierarchy

The structure of classless addressing does not restrict the number of hierarchical levels. An organization can divide the granted block of addresses into sub blocks. Each sub block can in turn be divided into smaller sub blocks.

Network Address Translation (NAT)

- The number of home users and small businesses that want to use the Internet is ever increasing.
- In the beginning, a user was connected to the Internet with a dial-up line. An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed.
- But the situation is different today. Home users and small businesses can be connected by an ADSL line or cable modem. In addition, many are not happy with one address; many have

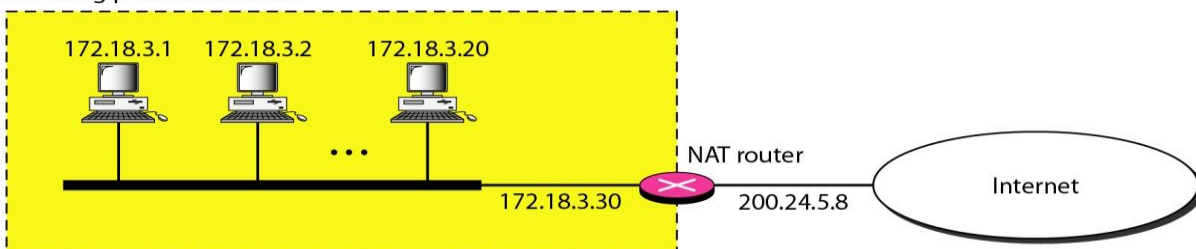
created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem.

- A quick solution to this problem is called network address translation (NAT).
- NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.
- To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses.

<i>Range</i>			<i>Total</i>
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

- Any organization can use an address out of this set without permission from the Internet authorities. Everyone knows that these reserved addresses are for private networks. They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these addresses as the destination address.
- The site must have only one single connection to the global Internet through a router that runs the NAT software.

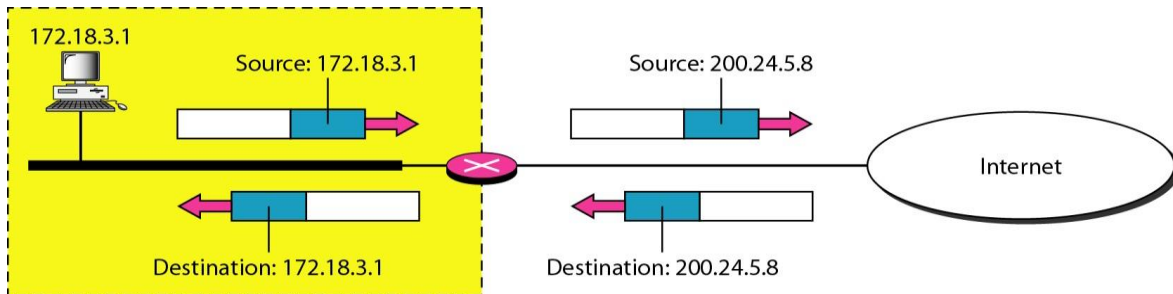
Site using private addresses



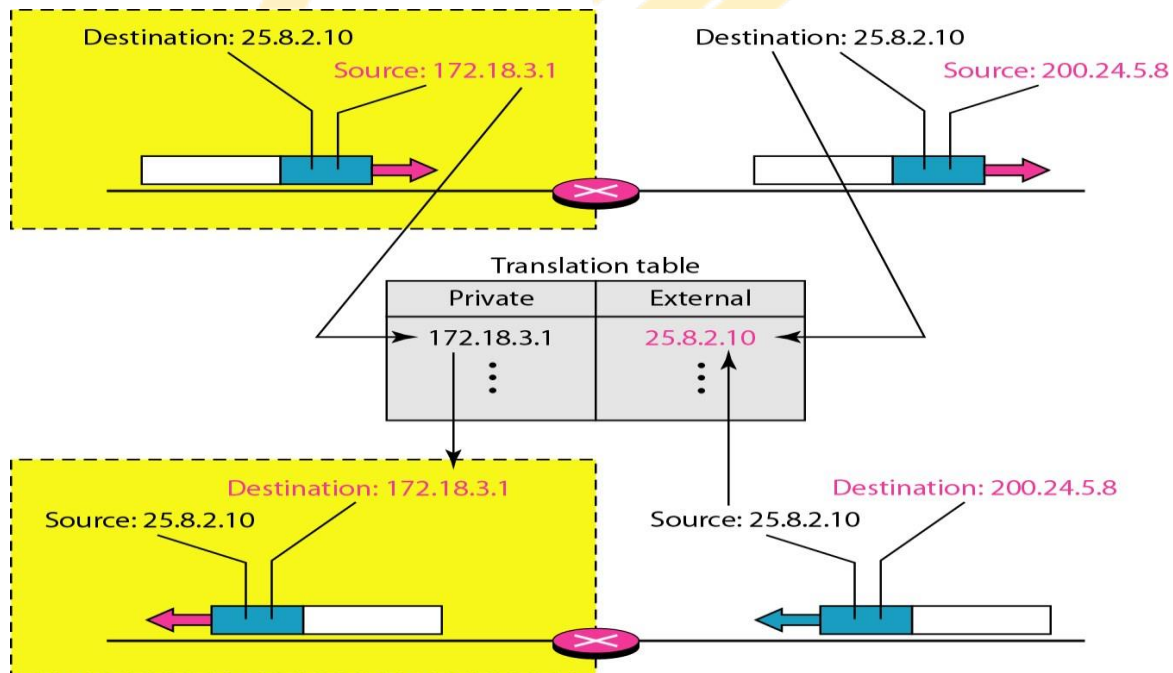
In the above figure, the router that connects the network to the global address uses one private address and one global address. The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

Address Translation

- All the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address.



- A **translation table** has only two columns: the private' address and the external address (destination address of the packet). When the router translates the source address of the outgoing packet, it also makes note of the destination address-where the packet is going. When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.

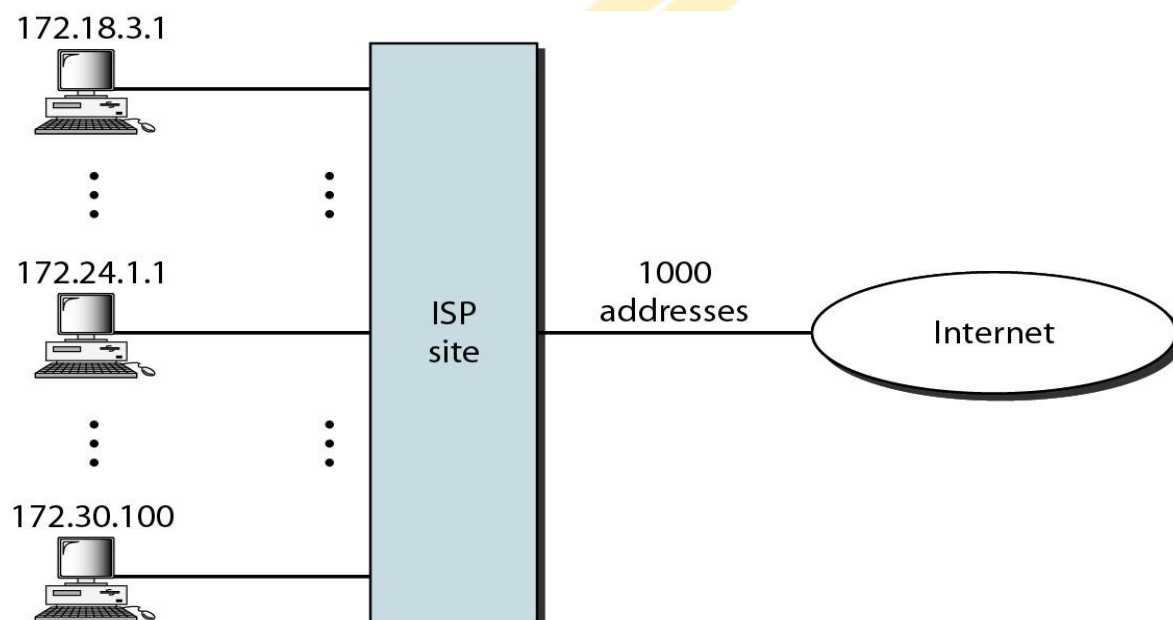


- **Using a Pool of IP Addresses** Since the NAT router has only one global address, only one private network host can access the same external host. To remove this restriction, the NAT router uses a pool of global addresses.
- **Using Both IP Addresses and Port Numbers** To allow a many-to-many relationship between private-network hosts and external server programs, we need more information in the translation table.

<i>Private Address</i>	<i>Private Port</i>	<i>External Address</i>	<i>External Port</i>	<i>Transport Protocol</i>
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
...

NAT and ISP

An ISP that serves dial-up customers can use NAT technology to conserve addresses. For example, suppose an ISP is granted 1000 addresses, but has 100,000 customers. Each of the customers is assigned a private network address. The ISP translates each of the 100,000 source addresses in outgoing packets to one of the 1000 global addresses; it translates the global destination address in incoming packets to the corresponding private address.

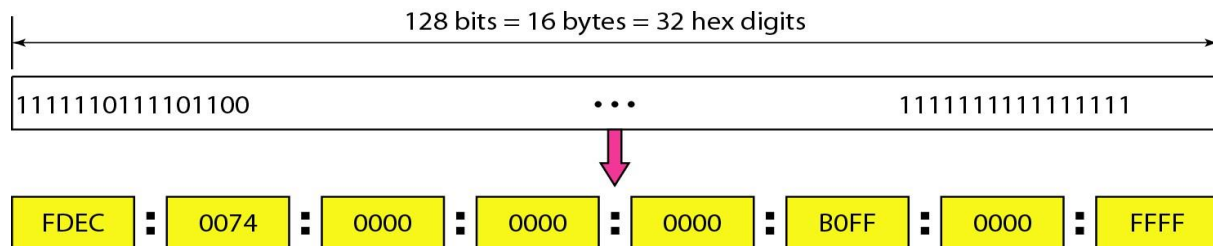


5.2 IPv6 Addresses

An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

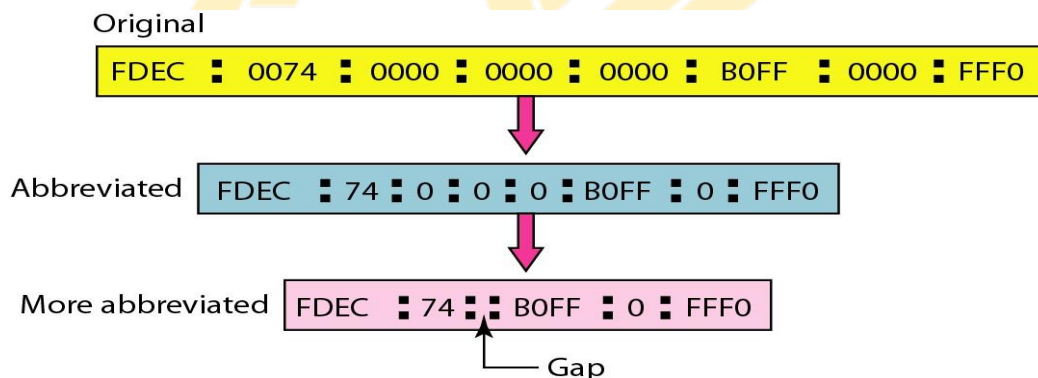
Hexadecimal Colon Notation

To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.



Abbreviation

Although the IP address, even in hexadecimal format, is very long, many of the digits are Zeros. In this case, we can abbreviate the address.



Address Space

IPv6 has a much larger address space; 2^{128} addresses are available. The designers of IPv6 divided the address into several categories. A few leftmost bits, called the *type prefix*, in each address define its category. The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code.

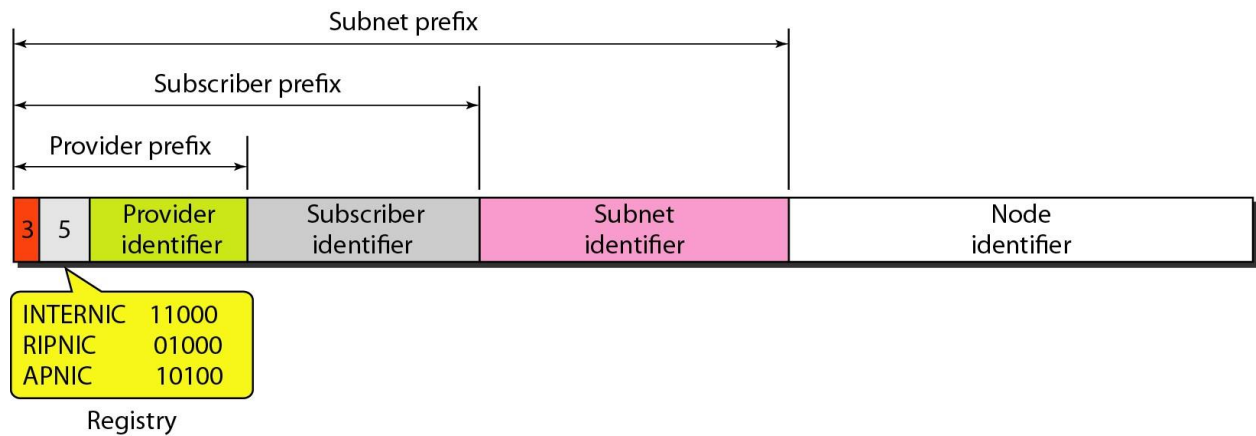
<i>Type Prefix</i>	<i>Type</i>	<i>Fraction</i>
0000 0000	Reserved	1/256
0000 0001	Unassigned	1/256
0000 001	ISO network addresses	1/128
0000 010	IPX (Novell) network addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Reserved	1/16
001	Reserved	1/8
010	Provider-based unicast addresses	1/8



<i>Type Prefix</i>	<i>Type</i>	<i>Fraction</i>
011	Unassigned	1/8
100	Geographic-based unicast addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link local addresses	1/1024
1111 1110 11	Site local addresses	1/1024
1111 1111	Multicast addresses	1/256

Unicast Addresses

A **unicast address** defines a single computer. The packet sent to a unicast address must be delivered to that specific computer. IPv6 defines two types of unicast addresses: geographically based and provider-based.

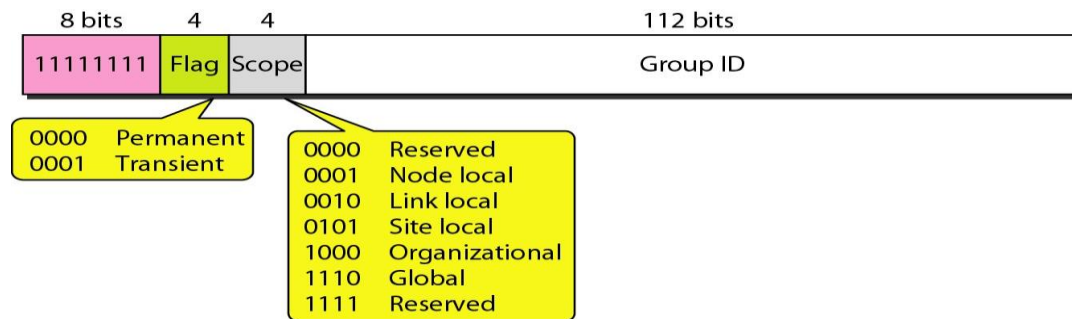


Fields for the provider-based address are as follows:

- **Type identifier.** This 3-bit field defines the address as a provider-based address.
- **Registry identifier.** This 5-bit field indicates the agency that has registered the address. Currently three registry centers have been defined. INTERNIC (code 11000) is the center for North America; RIPNIC (code 01000) is the center for European registration; and APNIC (code 10100) is for Asian and Pacific countries.
- **Provider identifier.** This variable-length field identifies the provider for Internet access (such as an ISP).
- **Subscriber identifier.** When an organization subscribes to the Internet through a provider, it is assigned a subscriber identification.
- **Subnet identifier.** Each subscriber can have many different subnetworks, and each subnetwork can have an identifier. The subnet identifier defines a specific subnetwork under the territory of the subscriber.
- **Node identifier.** The last field defines the identity of the node connected to a subnet.

Multicast Addresses

Multicast addresses are used to define a group of hosts instead of just one. A packet sent to a multicast address must be delivered to each member of the group.



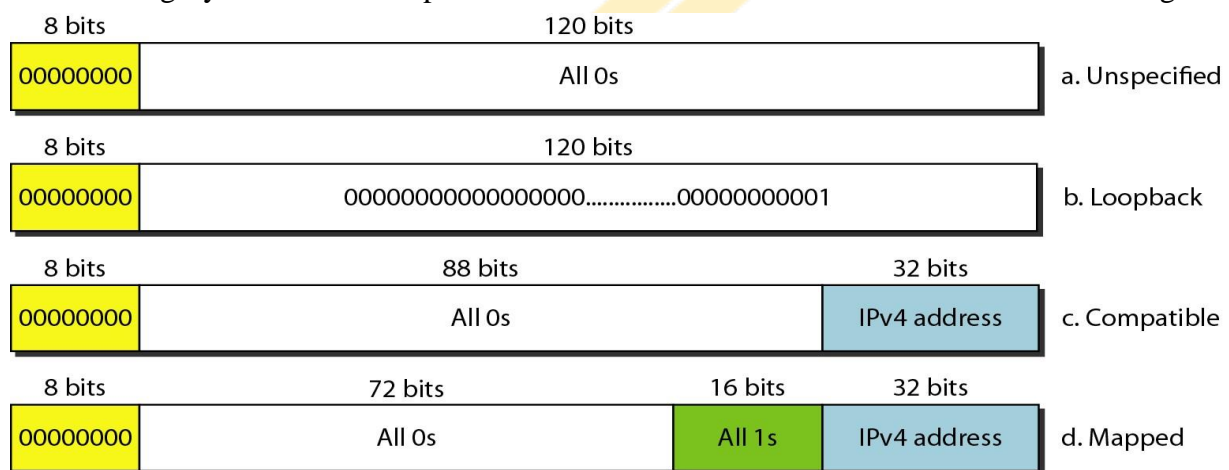
- The second field is a flag that defines the group address as either permanent or transient. A permanent group address is defined by the Internet authorities and can be accessed at all times. A transient group address, on the other hand, is used only temporarily.
- The third field defines the scope of the group address.

Anycast Addresses

An anycast address, like a multicast address, also defines a group of nodes. However, a packet destined for an anycast address is delivered to only one of the members of the anycast group, the nearest one (the one with the shortest route).

Reserved Addresses

Another category in the address space is the reserved address. These addresses start with eight 0s

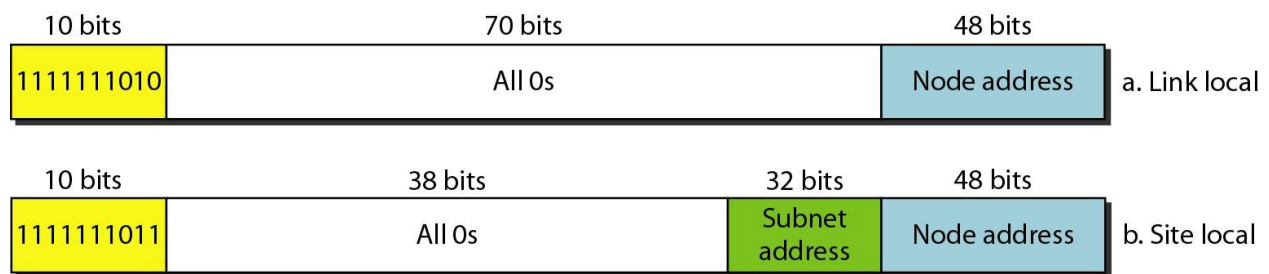


- An **unspecified address** is used when a host does not know its own address and sends an inquiry to find its address.
- A **loopback address** is used by a host to test itself without going into the network.

- A **compatible address** is used during the transition from IPv4 to IPv6. It is used when a computer using IPv6 wants to send a message to another computer using IPv6, but the message needs to pass through a part of the network that still operates in IPv4.
- A **mapped address** is also used during transition. However, it is used when a computer that has migrated to IPv6 wants to send a packet to a computer still using IPv4.

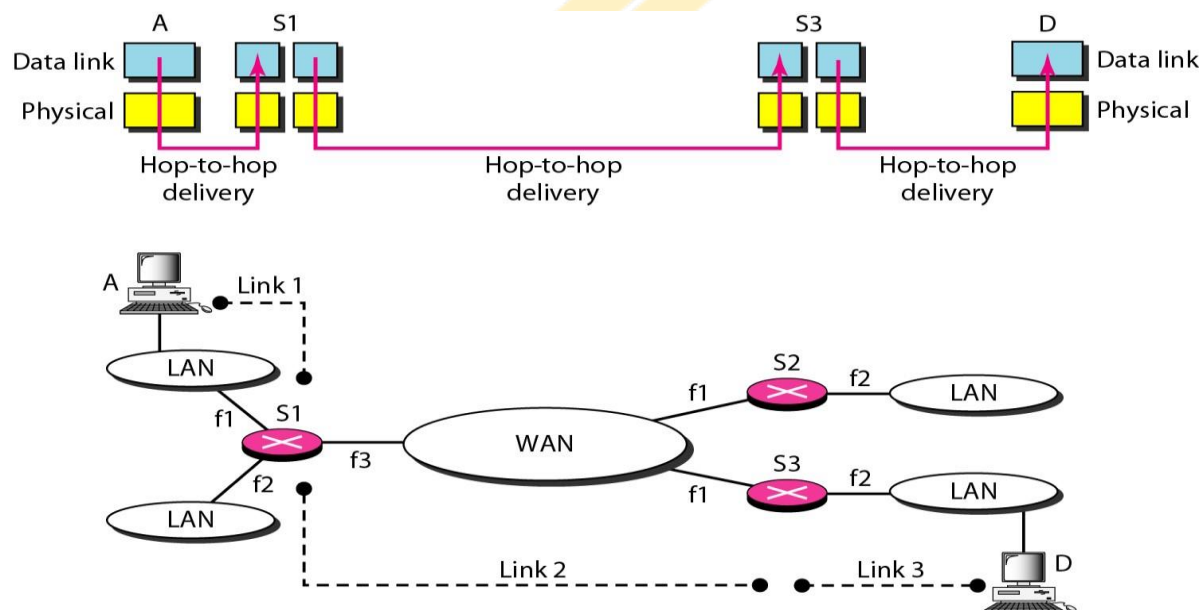
Local Addresses

These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet.



5.3 Internetworking

The physical and data link layers of a network operate locally. These two layers are jointly responsible for data delivery on the network from one node to the next.



This internetwork is made of five networks: four LANs and one WAN.

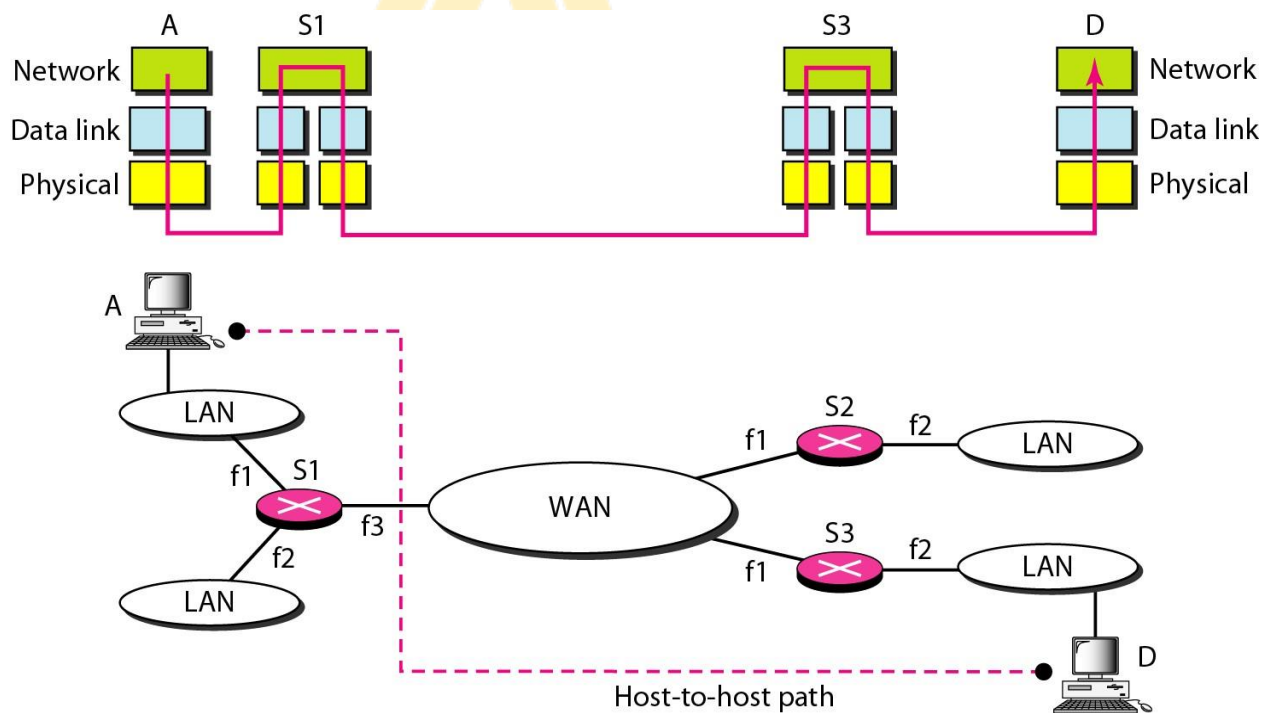
If host A needs to send a data packet to host D, the packet needs to go first from A to R1 (a switch or router), then from R1 to R3, and finally from R3 to host D.

In each link, two physical and two data link layers are involved. However, there is a big problem here. When data arrive at interface f1 of R1, There is no provision in the data link (or physical) layer to help R1 make the right decision. The frame does not carry any routing information either. The frame contains the MAC address of A as the source and the MAC address of R1 as the destination.

Need for Network Layer

To solve the problem of delivery through several links, the network layer (or the internetwork layer, as it is sometimes called) was designed.

The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches.

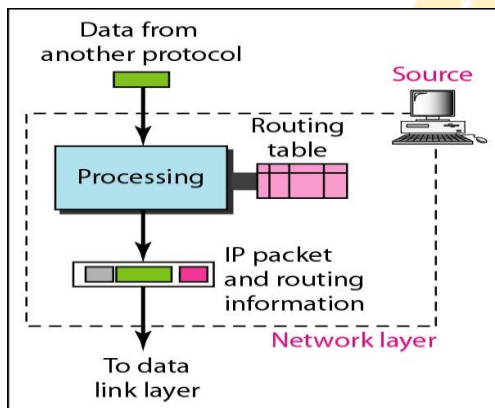


The network layer at the source is responsible for creating a packet from the data coming from another protocol (such as a transport layer protocol or a routing protocol). The header of the packet contains, among other information, the logical addresses of the source and destination. The network layer is responsible for checking its routing table to find the routing information (such as

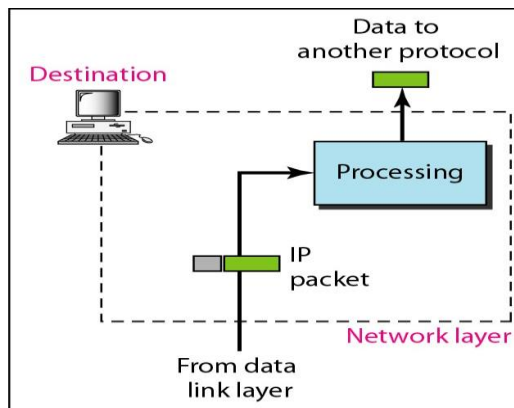
the outgoing interface of the packet or the physical address of the next node). If the packet is too large, the packet is fragmented.

The network layer at the switch or router is responsible for routing the packet. When a packet arrives, the router or switch consults its routing table and finds the interface from which the packet must be sent. The packet, after some changes in the header, with the routing information is passed to the data link layer again.

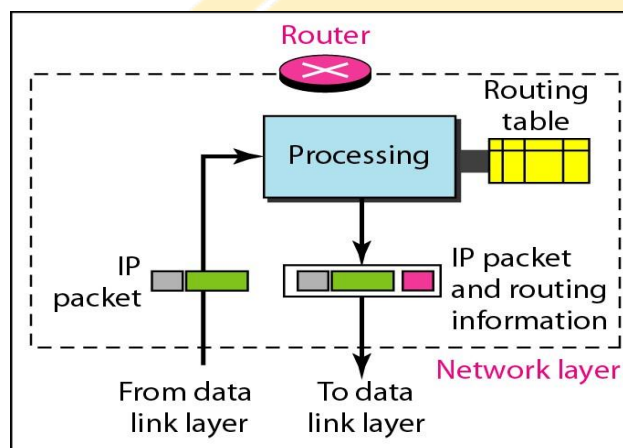
The network layer at the destination is responsible for address verification; it makes sure that the destination address on the packet is the same as the address of the host. If the packet is a fragment, the network layer waits until all fragments have arrived, and then reassembles them and delivers the reassembled packet to the transport layer.



a. Network layer at source



b. Network layer at destination



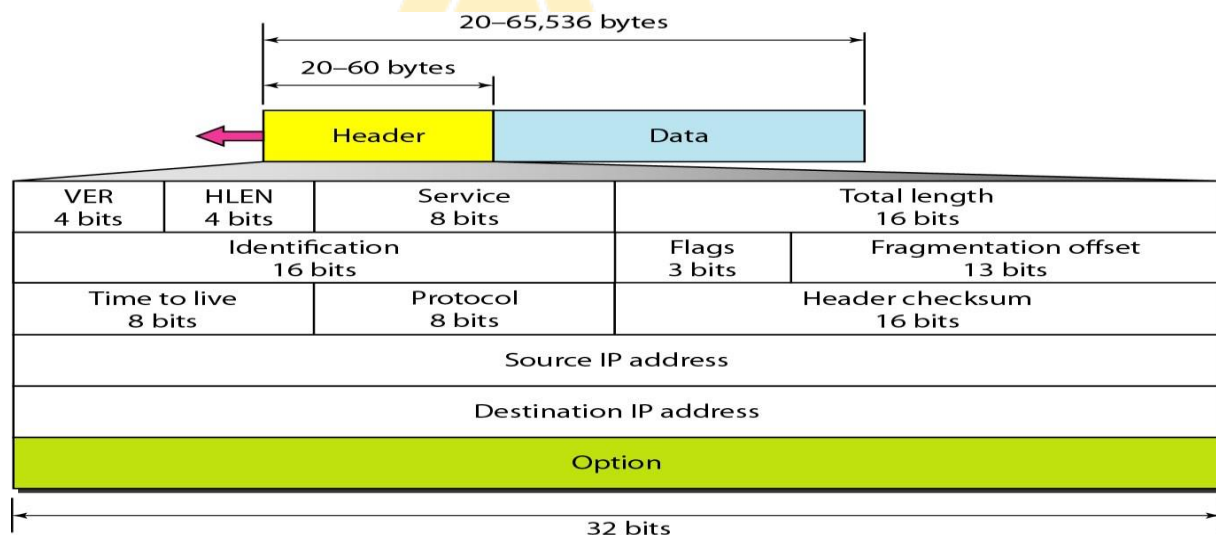
c. Network layer at a router

5.4 IPv4

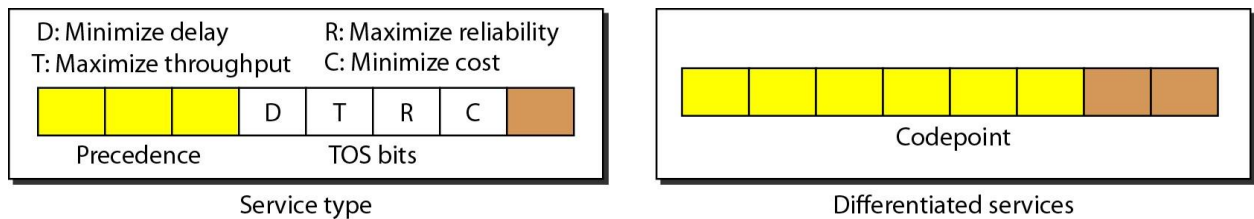
The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol—a best-effort delivery service. The term *best-effort* means that IPv4 provides no error control or flow control. IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. If reliability is important, IPv4 must be paired with a reliable protocol such as TCP.

Datagram

Packets in the IPv4 layer are called datagrams. A datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery.



- **Version:** Version no. of Internet Protocol used (e.g. IPv4).
- **HLEN:** This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 ($5 \times 4 = 20$). When the option field is at its maximum size, the value of this field is 15 ($15 \times 4 = 60$).
- **Service:** This field, previously called service type, is now called differentiated services.



Service Type:

In this interpretation, the first 3 bits are called precedence bits ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion.. The next 4 bits are called type of service (TOS) bits, and the last bit is not used.

<i>TOS Bits</i>	<i>Description</i>
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Application programs can request a specific type of service. The defaults for some applications are:

<i>Protocol</i>	<i>TOS Bits</i>	<i>Description</i>
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Differentiated Services

In this interpretation, the first 6 bits make up the codepoint subfield, and the last 2 bits are not used. The codepoint subfield can be used in two different ways.

- a. When the 3 rightmost bits are 0s, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation.
- b. When the 3 rightmost bits are not all 0s, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities.

- **Total Length:** Length of entire IP Packet (including IP header and IP Payload).

$\text{Length of data} = \text{total length} - \text{header length}$

- **Identification:** If IP packet is fragmented during the transmission, all the fragments contain same identification number to identify original IP packet they belong to.
- **Flags:** As required by the network resources, if IP Packet is too large to handle, these 'flags' tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to '0'.
- **Fragment Offset:** This offset tells the exact position of the fragment in the original IP Packet.
- **Time to Live:** To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Protocol:** Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol.

<i>Value</i>	<i>Protocol</i>
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

- **Header Checksum:** This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.

Example:

4	5	0	28			
1			0	0		
4	17	0				↑
10.12.14.5						
12.6.7.9						

4, 5, and 0	→	4	5	0	0
28	→	0	0	1	C
1	→	0	0	0	1
0 and 0	→	0	0	0	0
4 and 17	→	0	4	1	1
0	→	0	0	0	0
10.12	→	0	A	0	C
14.5	→	0	E	0	5
12.6	→	0	C	0	6
7.9	→	0	7	0	9
Sum	→	7	4	4	E
Checksum	→	8	B	B	1

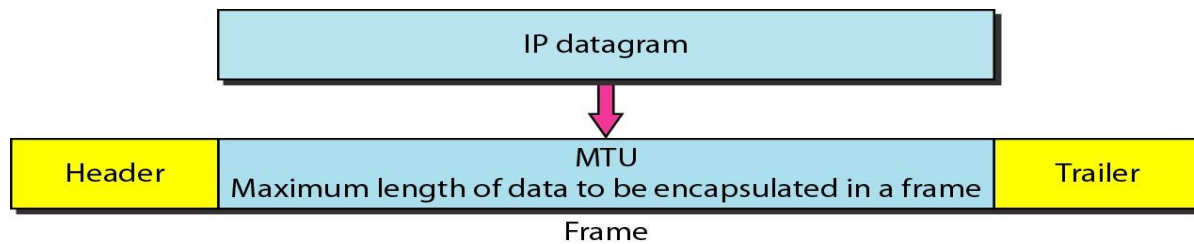
- **Source Address:** 32-bit IPv4 address of the Sender (or source) of the packet.
- **Destination Address:** 32-bit IPv4 address of the Receiver (or destination) of the packet.

Fragmentation

A datagram can travel through different networks. Each router decapsulates the IPv4 datagram from the frame it receives, processes it, and then encapsulates it in another frame. The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled. The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.

Maximum Transfer Unit (MTU)

Each data link layer protocol has its own frame format. One of the fields defined in the format is the maximum size of the data field. The value of the MTU depends on the physical network protocol.



<i>Protocol</i>	<i>MTU</i>
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

To make the IPv4 protocol independent of the physical network, the designers decided to make the maximum length of the IPv4 datagram equal to 65,535 bytes.

When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but with some changed. A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU. In other words, a datagram can be fragmented several times before it reaches the final destination.

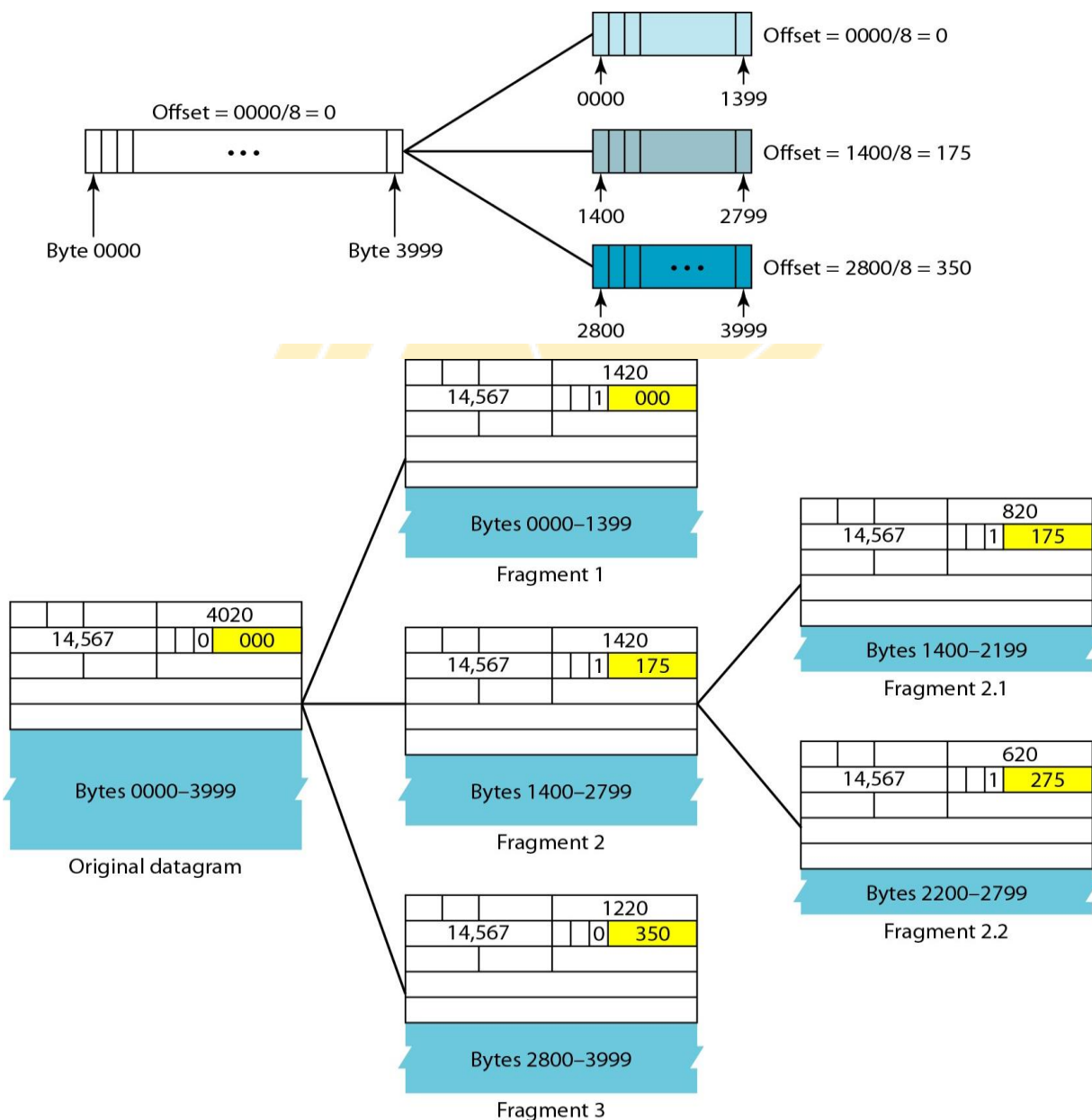
Fields Related to Fragmentation

- **Identification.** This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host. To guarantee uniqueness, the IPv4 protocol uses a counter to label the datagrams. The counter is initialized to a positive number. When the IPv4 protocol sends a datagram, it copies the current value of the counter to the identification field and increments the counter by 1.
- **Flags.** This is a 3-bit field. The first bit is reserved. The second bit is called the *do not fragment* bit. If its value is 1, the machine must not fragment the datagram. If its value is 0,

the datagram can be fragmented if necessary. The third bit is called the *more fragment* bit. If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one. If its value is 0, it means this is the last or only fragment

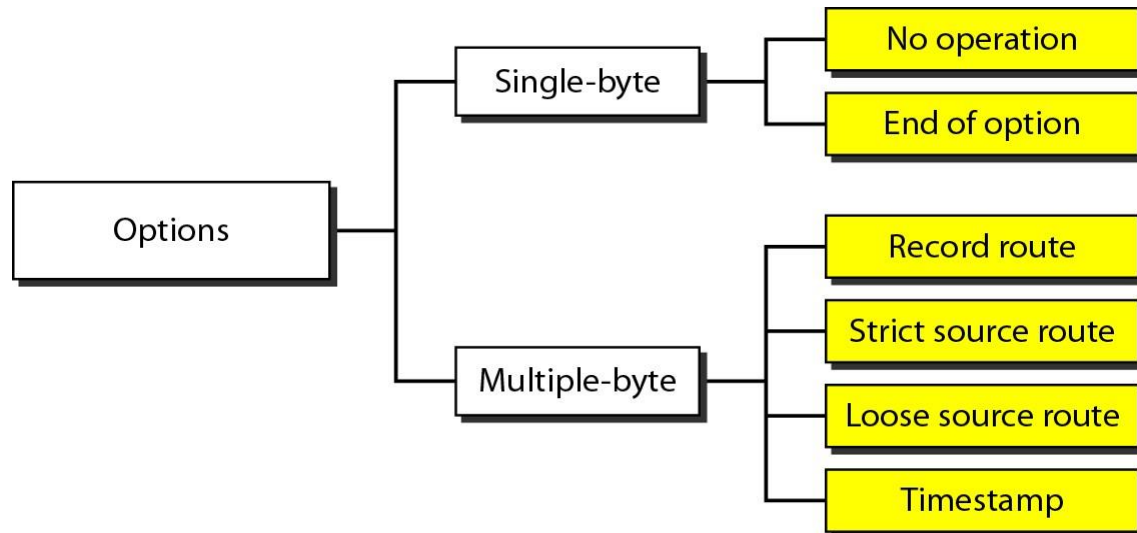


- **Fragmentation offset.** This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes.



Options:

This is optional field, which is used if the value of HEL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.



➤ *No Operation*

A **no-operation option** is a 1-byte option used as a filler between options.

➤ *End of Option*

An end-of-option option is a 1-byte option used for padding at the end of the option field. It, however, can only be used as the last option.

➤ *Record Route*

A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.

➤ *Strict Source Route*

A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Dictation of a route by the source can be useful for several purposes. The sender can choose a route with a specific type of service, such as minimum delay or maximum throughput. Alternatively, it may choose a route that is safer or more reliable for the sender's purpose.

➤ *Loose Source Route*

A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.

➤ *Timestamp*

A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet.

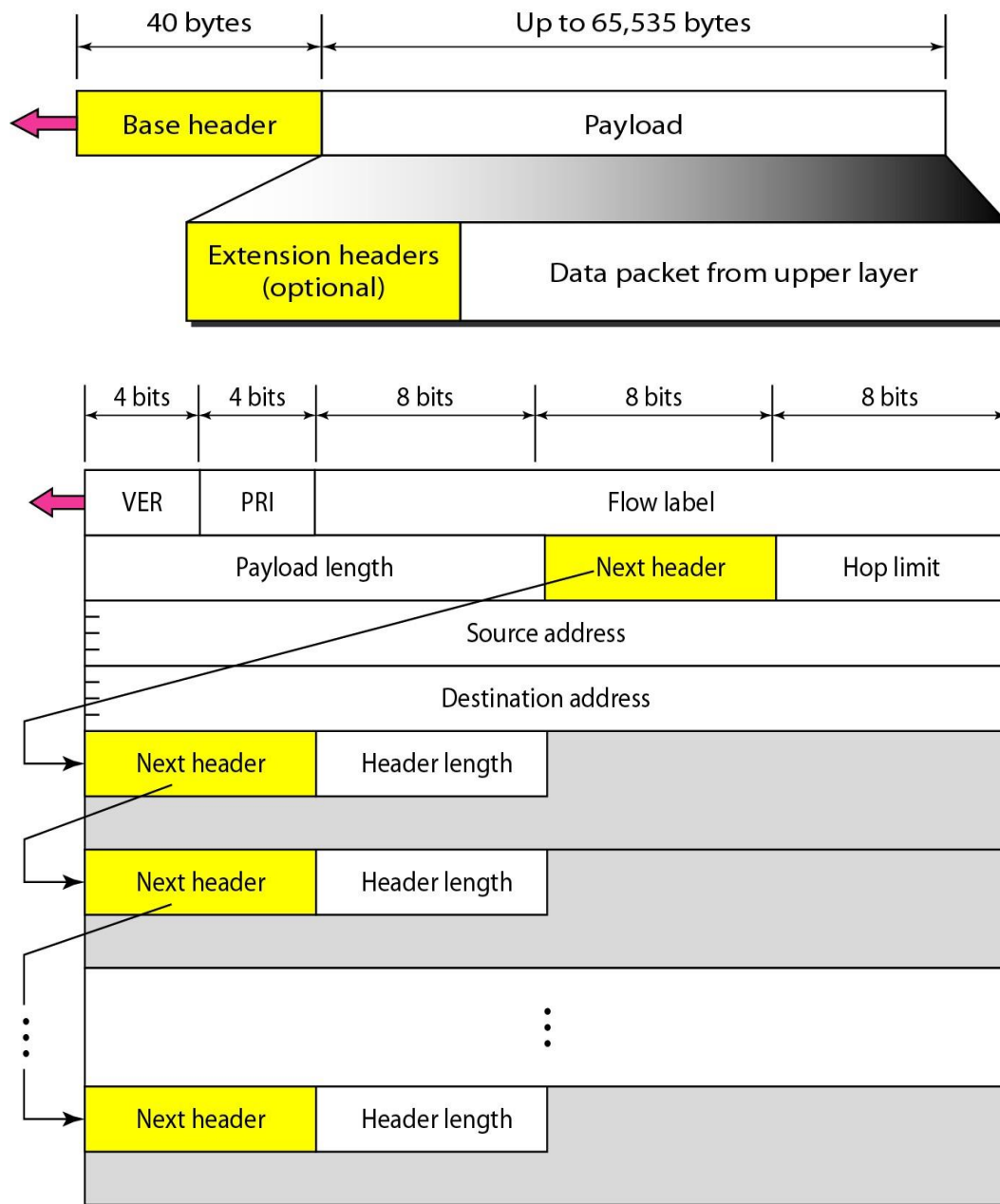
5.5 IPv6

Advantages

- **Larger address space.** An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge increase in the address space.
- **Better header format.** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- **New options.** IPv6 has new options to allow for additional functionalities.
- **Allowance for extension.** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
- **Support for resource allocation.** In IPv6, the type-of-service field has been removed, but a mechanism has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- **Support for more security.** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

Packet Format

Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts: optional extension headers and data from an upper layer. The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.



- **Version.** This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- **Priority.** The 4-bit priority field defines the priority of the packet with respect to traffic congestion.
- **Flow label.** The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.
- **Payload length.** The 2-byte payload length field defines the length of the IP datagram excluding the base header.

- **Next header.** The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field.

<i>Code</i>	<i>Next Header</i>
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

- **Hop limit.** This 8-bit hop limit field serves the same purpose as the TTL field in IPv4.
- **Source address.** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
- **Destination address.** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram.

Priority

The priority field of the IPv6 packet defines the priority of each packet with respect to other packets from the same source. IPv6 divides traffic into two broad categories: congestion-controlled and noncongestion-controlled.

Congestion-Controlled Traffic:

- If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion-controlled traffic.

- In congestion-controlled traffic, it is understood that packets may arrive delayed, lost, or out of order. Congestion-controlled data are assigned priorities from 0 to 7. A priority of 0 is the lowest; a priority of 7 is the highest.

<i>Priority</i>	<i>Meaning</i>
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

- **No specific traffic.** A priority of 0 is assigned to a packet when the process does not define a priority.
- **Background data.** This group (priority 1) defines data that are usually delivered in the background. Delivery of the news is a good example.
- **Unattended data traffic.** If the user is not waiting (attending) for the data to be received, the packet will be given a priority of 2. E-mail belongs to this group.
- **Attended bulk data traffic.** A protocol that transfers data while the user is waiting (attending) to receive the data (possibly with delay) is given a priority of 4. FTP and HTTP belong to this group.
- **Interactive traffic.** Protocols such as TELNET that need user interaction are assigned the second-highest priority (6) in this group.
- **Control traffic.** Control traffic is given the highest priority (7). Routing protocols such as OSPF and RIP and management protocols such as SNMP have this priority.

Noncongestion-Controlled Traffic: This refers to a type of traffic that expects minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. Real-time audio and video are examples of this type of traffic.

<i>Priority</i>	<i>Meaning</i>
8	Data with greatest redundancy
...	...
15	Data with least redundancy

Flow Label

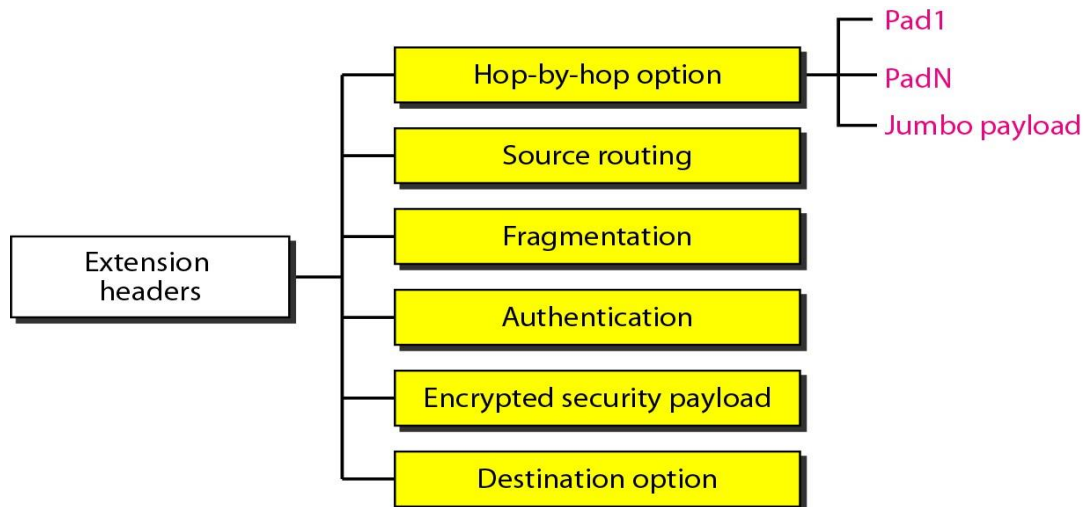
- A sequence of packets, sent from a particular source to a particular destination, that needs special handling by routers is called a *flow* of packets. The combination of the source address and the value of the *flow label* uniquely defines a flow of packets.
- To a router, a flow is a sequence of packets that share the same characteristics, such as traveling the same path, using the same resources, having the same kind of security, and so on.
- A router that supports the handling of flow labels has a flow label table.
- The table has an entry for each active flow label; each entry defines the services required by the corresponding flow label.
- When the router receives a packet, it consults its flow label table to find the corresponding entry for the flow label value defined in the packet. It then provides the packet with the services mentioned in the entry.

Comparison Between IPv4 and IPv6 Headers

<i>Comparison</i>
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

Extension Headers

The length of the base header is fixed at 40 bytes. However, to give greater functionality to the IP datagram, the base header can be followed by up to six extension headers.



□ *Hop-by-Hop Option*

The hop-by-hop option is used when the source needs to pass information to all routers visited by the datagram. The Pad 1 option is 1 byte long and is designed for alignment purposes. Pad N is similar in concept to Pad 1. The jumbo payload option is used to define a payload longer than 65,535 bytes.

□ *Source Routing*

The source routing extension header combines the concepts of the strict source route and the loose source route options of IPv4.

□ *Fragmentation*

In IPv6, only the original source can fragment. A source must use a path MTU discovery technique to find the smallest MTU supported by any network on the path. The source then fragments using this knowledge.

□ *Authentication*

The authentication extension header has a dual purpose: it validates the message sender and ensures the integrity of data.

□ *Encrypted Security Payload*

The encrypted security payload (ESP) is an extension that provides confidentiality and guards against eavesdropping.

□ Destination Option

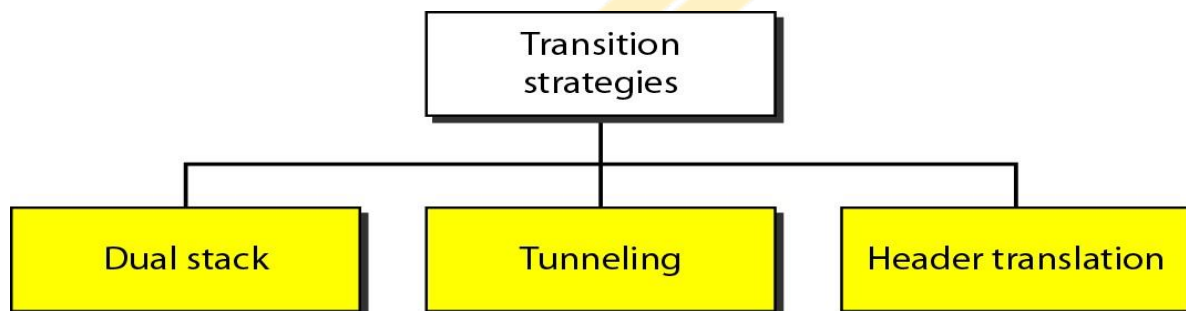
The destination option is used when the source needs to pass information to the destination only. Intermediate routers are not permitted access to this information.

Comparison between IPv4 options and IPv6 extension headers

Comparison
1. The no-operation and end-of-option options in IPv4 are replaced by Pad1 and PadN options in IPv6.
2. The record route option is not implemented in IPv6 because it was not used.
3. The timestamp option is not implemented because it was not used.
4. The source route option is called the source route extension header in IPv6.
5. The fragmentation fields in the base header section of IPv4 have moved to the fragmentation extension header in IPv6.
6. The authentication extension header is new in IPv6.
7. The encrypted security payload extension header is new in IPv6.

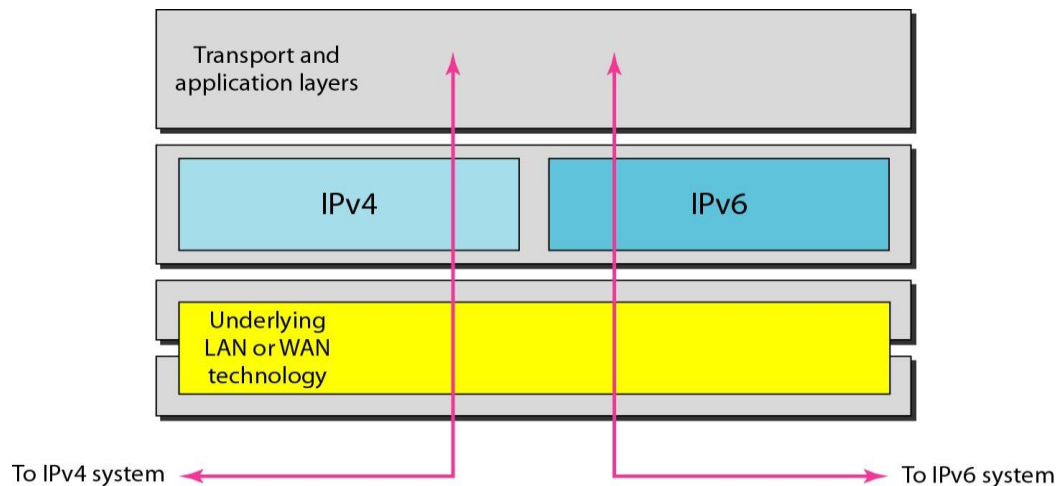
5.6 TRANSITION FROM IPv4 TO IPv6

Three strategies have been defined to help the transition.



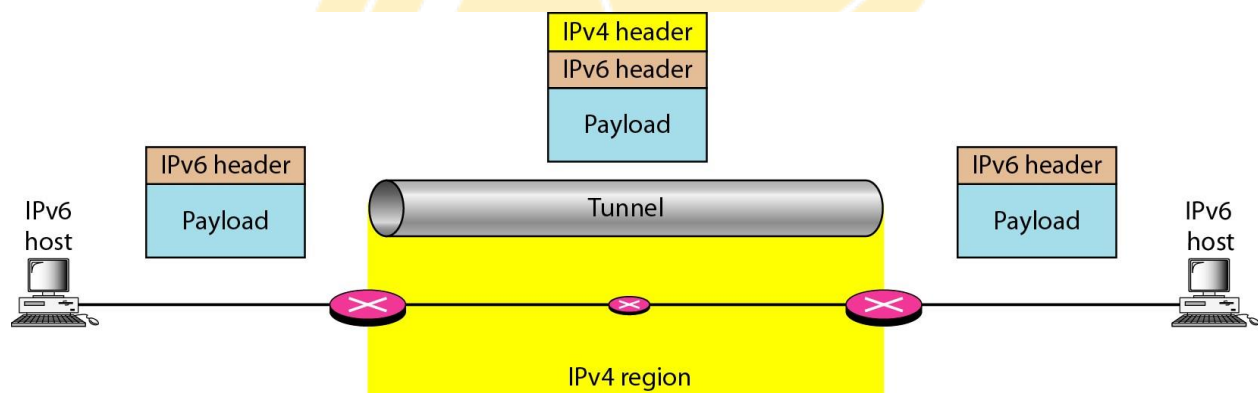
Dual Stack

It is recommended that all hosts, before migrating completely to version 6, have a dual stack of protocols. In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.



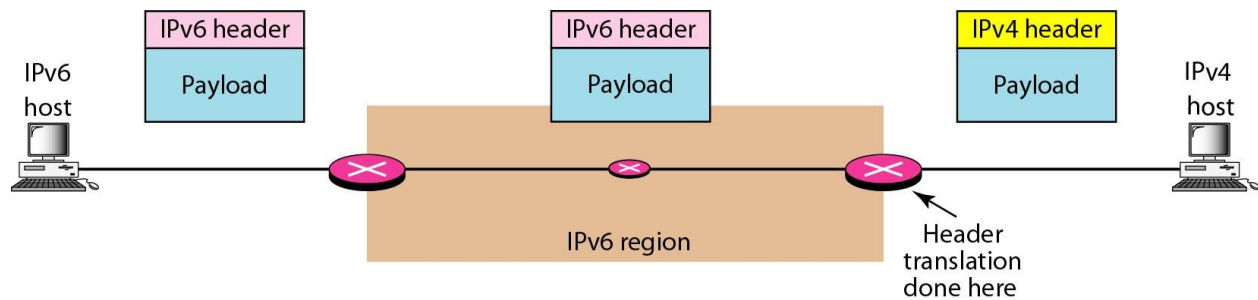
Tunneling

Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region.



Header Translation

The sender wants to use IPv6, but the receiver does not understand IPv6. Tunneling does not work in this situation because the packet must be in the IPv4 format to be understood by the receiver. In this case, the header format must be totally changed through header translation. The header of the IPv6 packet is converted to an IPv4 header.



Header Translation Procedure

1. The IPv6 mapped address is changed to an IPv4 address by extracting the rightmost 32 bits.
2. The value of the IPv6 priority field is discarded.
3. The type of service field in IPv4 is set to zero.
4. The checksum for IPv4 is calculated and inserted in the corresponding field.
5. The IPv6 flow label is ignored.
6. Compatible extension headers are converted to options and inserted in the IPv4 header. Some may have to be dropped.
7. The length of IPv4 header is calculated and inserted into the corresponding field.
8. The total length of the IPv4 packet is calculated and inserted in the corresponding field.