UNIT-4

Network Layer

⊛. Introduction & Functions:

Network layer selects and manages the best logical path for data transfer between nodes. It manages device addressing, tracks the location of devices on the network and determines the best way to move data. This layer contains hardware devices such as routers, bridges, firewalls and switches. Internet Protocol and Netware IPX/SPX are the most common protocols associated with the network layer.

Functions:

→ It translates logical network address into physical address, concerned with circuit, message or packet switching.
→ Routers and gateways operate in this layer.
→ Breaks larger packets into small packets.
→ Connection services are provided including network layer flow control, network layer error control and packet sequence control.

⊛. IPv4 Addressing:

IPv4 addresses are unique. They are unique in the sense that each address defines one and only one connection to the Internet. Two devices on the internet can never have the same address at the same time. The IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet. IPv4 addresses are 32-bit numbers that are typically displayed in dotted decimal notation. A 32-bit address contains two primary parts: the network prefix and the host number. IP addresses are assigned by a central numbering authority called the Internet Assigned Numbers Authority (IANA). The IPv4 contains the following sections:

1) IPv4 Classfull Addressing:

In classful addressing the address space is divided into five classes: A, B, C, D and E. Each ~~part~~ class occupies some part of the address space. We can find the class of an address when address is given in binary notation or dotted-decimal notation. If the address is

given in binary notation, the first few bits can immediately tell us the class of address. If the address is given in decimal-dotted notation, the first byte defines the class.

Class A → Addresses use only the first byte (octet) to specify the network prefix, leaving 3 bytes to define individual host numbers.

Class B → Addresses use the first 2 bytes to specify the network prefix, leaving 2 bytes to define host address.

Class C → Addresses use the first 3 bytes to specify the network prefix, leaving only the last byte to identify hosts.

In binary format, with an x (can have 0 or 1 value) representing each bit in the host number, the three address classes can be represented as follows:-

(first byte) (second byte) (third byte) (fourth byte)

```
00000000  xxxxxxxx  xxxxxxxx  xxxxxxxx  (Class A)
00000000  00000000  xxxxxxxx  xxxxxxxx  (Class B)
00000000  00000000  00000000  xxxxxxxx  (Class C)
```

Each bit x represents a power of 2 indicating how many host numbers can be created for a particular network prefix. Class A have 2^{24} possible host numbers, class B have 2^{16} and class C have 2^8.

⊛ Finding the classes in binary and dotted decimal notation.

Classes	Binary notation (first byte)	Dotted-decimal notation (first byte)	Range.
class A	0	0-127	0.0.0.0 – 172.255.255.255
class B	10	1128-19111	128.0.0.0 – 191.x
class C	110	1192-22311	192.x – 223.x
class D	1110	1224-23911	224.x – 239.x
class E	1111	1240-25511	240.x – 255.x

⊛ Examples: Find the class of each address.

a). 00000001 00001011 00001011 11101111
→ The first bit is 0. This is a class A address.

b). 11000001 10000011 00011011 11111111
→ The first 2 bits are 1 & 3rd bit is 0. This is class C address.

c). 14.23.120.8
→ The first byte is 14 (between 0 and 127); the class is A.

d) 252.5.15.111
→ The first byte is 252 (between 240 and 255); the class is E.

⊛. Rules for assigning Host ID:

Host ID's are used to identify a host within a network. The host ID are assigned based on following rules:

→ The Host ID must be unique.
→ Host ID in which all bits are set to 0 cannot be assigned because it is used to represent network ID of the IP address.
→ Host ID in which all bits are set to 1 cannot be assigned because it is used for broadcasting address to all hosts.

⊛ Rules for assigning Network ID:

The network ID is assigned based on following rules;-

→ Network ID cannot start with 127, because it is reserved for internal loop-back functions.
→ Network ID with all bits set to 0 cannot be assigned because it is used to denote specific host on local network.
→ Network ID with all bits set to 1 cannot be assigned because it is ~~used~~ reserved for an IP broadcast address.

⊛ Table (Representing host ID & Network ID):

Class	Network ID bit	Host ID bit
Class A	8	24
Class B	16	16
Class C	24	8
Class D	Not Defined	Not Defined.
Class E	Not Defined	Not Defined.

⊛ IPv4 Subnetting / Supernetting:

Subnetting → Subnetting is the procedure to divide the network into sub-networks. In subnetting, Network addresses bits are increased and the mask bits are moved towards right. It is implemented via variable length subnet masking. In subnetting, address depletion is reduced or removed. Each of the subnets has its own specific address.

A subnet address is created by borrowing bits from the host field and designating them as subnet field. The number of bits borrowed ~~from~~ varies and is specified by the subnet mask.

Why use subnetting?

When a network becomes too big with too much traffic, performance can begin to suffer. Breaking the network into smaller parts can help to increase network performance upto its original performance. A subnet allows routers to choose the right destination for packets. Subnetting can also improve network security.

How to create subnets?

To create a subnet, ~~we~~ we will start by fulfilling following three steps:

i) Determine the number of required network IDs.
   - → One for each LAN subnet.
   - → One for each wide area network connection.

ii) Determine the number of required host IDs per subnet.
   - → One for each TCP/IP host.
   - → One for each router interface.

iii) Finally create the following:
   - → A unique subnet mask for entire network.
   - → A unique subnet ID for each physical segment.
   - → A range of host IDs for each subnet.

Subnet Mask

A subnet mask is a 32-bit value that allows the device that is receiving IP packets to distinguish the network IP portion of the IP address. The 32-bit subnet mask is composed of 1s and 0s, where 1s represent positions that refer to the network subnet addresses.

Class	Format	Default Subnet Mask
A	network.host.host.host	255.0.0.0
B	network.network.host.host	255.255.0.0
C	network.network.network.host	255.255.255.0

Table: Default Subnet Mask

Q. If 200.100.10.66/26 is IPv4 address then answer the following questions:

a) Is this a host, network or broadcast address?
b) What is the subnet mask in dotted decimal?
c) What is the network address?
d) What is the broadcast address?
e) What is the first usable host address?
f) What is the last usable host address?
g) How many usable hosts are in the network?
h) What is the next available network address?

Solution :-

IP address: 200.100.10.66/26

Subnet Mask: 11111111 11111111 11111111 11000000

= 255.255.255.192

Total subnets = $2^2 = 4$

Total hosts = $2^6 = 64$

Usable hosts = $2^6 - 2 = 64 - 2 = 62$

Valid Subnets (4^{th} octet) = 256 - 192 = 64.

Subnet (Network IP)	Usable IP Pool (first host to last host)		Broadcast IP
	First host	Last host	
200.100.10.0	200.100.10.1	200.100.10.62	200.100.10.63
200.100.10.64	200.100.10.65	200.100.10.126	200.100.10.127
200.100.10.128	200.100.10.129	200.100.10.190	200.100.10.191
200.100.10.192	200.100.10.193	200.100.10.254	200.100.10.255

aAns: It is host address

bAns: 255.255.255.192

cAns: 200.100.10.64/26

dAns: 200.100.10.127/26

eAns: 200.100.10.65/26.

fAns: 200.100.10.126/26

gAns: 62

hAns: 200.100.10.128/26

Classless Inter-Domain Routing (CIDR):-

It is basically the method that Internet service providers (ISPs) use to allocate a number of addresses. They provide address in a certain block size. We will receive a block of address from ISP like this: 192.168.10.32/28. This is telling us what our subnet mask is. The slash notation (/) means how many bits are turned on (1s). The maximum could only be /32 because a byte is 8 bits and there are 4 bytes in an IP address: (4×8 = 32). But we have to keep at least 2 bits for host bits so the largest subnet mask available relevent to the Cisco exams objectives can only be a /30.

Class A → /8 to /15
Class B → /16 to /23
Class C → /24 to /30.

@ Subnetting Class C Addressing:

Binary	Decimal Dotted	CIDR
00000000	255.255.255.0	/24
10000000	255.255.255.128	/25
11000000	255.255.255.192	/26
11100000	255.255.255.224	/27
11110000	255.255.255.240	/28
11111000	255.255.255.248	/29
11111100	255.255.255.252	/30

Table: Class C subnet masks.

i) No. of subnets → 2^x, where x is the number of masked bits (1s).

ii) No. of hosts per subnet → $2^y - 2$, where y is the number of unmasked bits (0s). [No. of usable hosts $2^y - 2$ & No. of hosts = 2^y]

iii) **Block size = 256 − subnet mask**

iv) Broadcast address → The number right before the next subnet.

v) Valid hosts → Numbers between the subnets, omitting the all-0s and all-1s. For example: If 64 is the subnet number and 127 is the broadcast address, then 65–126 is the valid host range. It is group of numbers between the subnet address and the broadcast address.

(b) Subnetting Class B Addresses:

Binary (3rd and 4th octet)	Decimal Dotted	CIDR
00000000 00000000	255.255.0.0	/16
10000000 00000000	255.255.128.0	/17
⋮	⋮	⋮
11111111 00000000	255.255.255.0	/24
11111111 10000000	255.255.255.128	/25
⋮	⋮	⋮
11111111 11111100	255.255.255.252	/30

Table: Class B subnet masks.

The process of subnetting a class B is pretty much the same as it is for a class C, except that we have more host bits and we start in the third octet. Use the same subnet numbers for the third octet with class B that we used for the fourth octet with class C, but add zero to the network portion and a 255 to the broadcast section in the fourth octet.

Example: Let we take same example of subnetting class C which we discussed before and now here we only discuss differ ones only neglecting same ones.

Solution:

Subnet Mask: 11111111 11111111 10

Q. If 172.16.0.0/17 is IPv4 address then answer the following questions:
(Same questions as in before question a to h).

Solution:-

IP address: 172.16.0.0/17

Subnet Mask: 11111111 11111111 10000000 00000000
= 255.255.128.0

Total Subnets = $2^1 = 2$

Total hosts = $2^{15} = 32768$

Usable hosts = $2^{15} - 2 = 32768 - 2 = 32766$

Valid Subnets (3rd octet) = $256 - 128 = 128$

Valid Subnets (4th octet) = $256 - 0 = 256$

Subnet (Network IP)	Usable IP Pool (First host to last host)		Broadcast IP
	First host	Last host	
172.16.0.0	172.16.0.1	172.16.127.254	172.16.127.255
172.16.128.0	172.16.128.1	172.16.255.254	172.16.255.255

aAns:- It is network address

bAns:- 255.255.128.0

cAns:- 172.16.0.0/17

dAns:- 172.16.127.255/17

eAns:- ~~172.16.127.254/17~~ 172.16.0.1/17

fAns:- 172.16.127.254/17

gAns:- 32766

hAns:- 172.16.128.0/17

ⓒ Subnetting Class A Address:-

Binary (2nd, 3rd and 4th octet)	Subnet Mask	CIDR value
00000000 00000000 00000000	255.0.0.0	/8
10000000 00000000 00000000	255.128.0.0	/9
⋮	⋮	⋮
11111111 11111111 10000000	255.255.255.128	/25
⋮	⋮	⋮
11111111 11111111 11111100	255.255.255.252	/30

Table: Class A subnet mask

Subnetting class A is also similar as class B or class C subnet. We must leave at least 2 bits for defining hosts. Again we have more host bits and we just use the same subnet numbers we used with class B and class C but we start using these numbers in the second octet.

Q. If 10.1.0.0/9 is IPv4 address then find all network address, broadcast address, usable host, Total subnets, total hosts, valid subnets.

Solution:- IP address: 10.1.0.0/9

Subnet Mask: 11111111 10000000 00000000 00000000
= 255.128.0.0

Total Subnets = $2^1 = 2$

Total hosts = $2^{23} = 8388608$

Usable hosts = $2^{23} - 2 = 8388606$

Valid Subnets (2nd octet) = 256 − 128 = 128

Valid Subnets (3rd octet) = 256 − 0 = 256

Valid Subnets (4th octet) = 256 − 0 = 256

Subnet (Network IP)	Usable IP Pool (first host to last host)		Broadcast IP
	First host	Last host	
10.0.0.0	10.0.0.1	10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1	10.255.255.254	10.255.255.255

Supernetting:

Supernetting is the procedure to combine the small networks into larger space. In subnetting, Network addresses' bits are increased, but in supernetting, Host addresses bits are increased. It is implemented via variable-length subnet masking.

Why supernetting?

The routing table contains the entry of a subnet mask for every network. If there are lots of small networks then size of the routing table increases. When the router has a big routing table then it takes a lot of time for the router to process the routing table. Supernetting is used to reduce the size of the IP routing table to improve network routing efficiency.

How does supernetting work?

All the networks are not suitable for supernetting. For any network to be supernetted it should follow three rules:

i) Contiguous → All the networks should be contiguous.

ii) Same size → All the networks should be of the same size and also a power of 2 i.e, 2^n.

iii) Divisibility → The first network ID should be divisible by the size of the block.

Note: If a binary number is divided by 2^n then last n bits are the remainder.

(Not more imp)

Example: Suppose we have four small networks with network ID as 201.1.0.0, 201.1.1.0, 201.1.2.0, 201.1.3.0. Check if this can be supernetted (or aggregated) or not.

Solution:

i) Contiguous → As we can see that all the four networks are class C networks. The range of the first network is from 201.1.0.0 to 201.1.0.255. The range of the second network start from 201.1.1.0. If we add 1 to the last IP address of the first network we get the starting IP address of the second network. Similarly we can check that all the networks are contiguous.

ii) Same Size → All the networks are of class C. Each network has 2^8 i.e, 256 hosts.

iii) Divisibility → The first IP address should be divisible by the total size of the networks. The total size of the network is 4×2^8 i.e, 2^{10}. The last 10 bits are the remainder if we divide the first IP address by 2^{10}. In order that they are divisible, the last ten bits should be 0.

First IP address binary representation: 11001001.00000001.00000000.
00000000

The last 10 bits are zero. Hence it is divisible by the size of the network. Hence all three conditions are satisfied.

2. Classless Addressing:

Classless addressing is a concept of addressing the IPv4 addresses which was adopted after the failure of classful addressing. The classful addressing leads to wastage of addresses as it assigns a fixed-size block of addresses to the customer. But, the classless addressing assigns a block of addresses to the customer according to its requirement which prevents the wastage of addresses. The classless IPv4 addressing does not divide the address space into classes like classful addressing. It provides a variable-length of blocks which have a range of addresses according to the need of users.

CIDR Notation:

Like in classful addressing, the address was divided into two parts Network ID and Host ID. Network ID define the address of the network and host id defines the host address in the corresponding network. The Network ID and Host ID part would vary with the classes.

The classless addressing also divides the IPv4 addresses into two parts referred to as 'prefix' and 'suffix'. Prefix defines the Network ID whereas suffix defines the host address in the corresponding network. The length of prefix(n) is added to the last of address separated by a slash. It is known as CIDR notation.

Properties: (OR restrictions on classless address blocks).

→ Addresses in a block must be in contiguous form.
→ The number of address in a block must be the power of 2. i.e, 2,4,8,16,...
→ The first address must be evenly divisible by the number of addresses.

Representation:

Let IP address be 192.168.10.1/28. (no. of mask bit)

Here we have 28 bits. So, we need to put 28 bits out of 32 bits as 1 and rest of bits as 0. will give us the mask for the Ip address block.

11111111.11111111.11111111.11100000
255. 255. 255. 240

Mask is: 255.255.255.240

Q. If 205.16.37.39/28 be the address. Now, find the first address, last address and the number of addresses.

Solution:-

Address (In Binary): 11001101 00010000 00100101 00100111
Mask of given address: 11111111 11111111 11111111 11110000

Now, The first address can be found by AND operation of given address with the mask.

Address: 11001101 00010000 00100101 00100111
Mask : 11111111 11111111 11111111 11110000

First Address: 11001101 00010000 00100101 00100000

The last address can be found by OR operation of given address with complement of the mask. Complement of number can be found by changing each 1 to 0 and each 0 to 1.

Address: 11001101 00010000 00100101 00100111
Mask complement: 00000000 00000000 00000000 00001111
Last Address: 11001101 00010000 00100101 00101111.

The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

Mask complement: 00000000 00000000 00000000 00001111
Number of addresses: 15+1 = 16

⊛ IPv6 Addressing and Its Features:

Internet protocol version 6 is a new addressing protocol designed to incorporate all the possible requirements of future Internet known to us as Internet version 2. This protocol works on Network Layer (ie, layer-3). It is a 128-bit alphanumeric string that identifies an endpoint device in the Internet Protocol Version 6 (IPv6) addressing scheme. It is arranged in eight groups, each of which is 16 bits. Each group is expressed as four hexadecimal digits and the groups are seperated by colours.

Example: FE80:CD00:0000:0CDE:1257:0000:211E:729C

The address can be shortened, because the addressing scheme allows the omission of any leading zero, as well as sequences consisting only of zeros. Here's the short version:

FE80:CD00:0:CDE:1257:0:211E:729C.

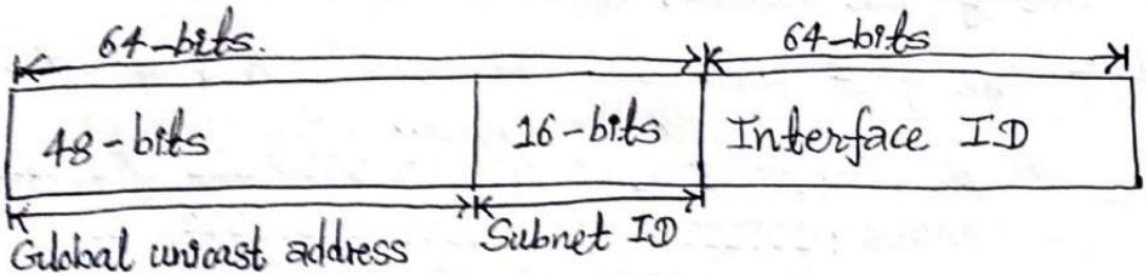


Fig. IPv6 address structure.

Features:

i) **Larger Address Space:-** In contrast to IPv4, IPv6 uses 4 times more bits to address a device on the Internet. This much of extra bits can provide approximately 3.4×1034 different combination of addresses. According to an estimate, 1564 addresses can be allocated to every square meter of this earth.

ii) **Simplified Header:-** IPv6's header has been simplified by moving all unnecessary information and options to the end of the IPv6 header. IPv6 header is only twice as bigger than IPv4 provided the fact that IPv6 address is four times longer.

iii) **End-to-end Connectivity:-** Every system has now unique IP address and can traverse through the Internet without using any translating components.

iv) **Auto-configuration:-** IPv6 supports both stateful and stateless auto configuration mode of its host devices. This way, absence of a DHCP server does not put a halt on inter segment communication.

v) **Mobility:-** IPv6 was designed keeping mobility in mind. This feature enables hosts (such as mobile phone) to roam around in different geographical area and remain connected with the same IP address.

vi) **No Broadcast:-** IPv6 does not have any broadcast support any more. It uses multicast to communicate with multiple hosts.

⊛. IPv4 and IPv6 Datagram Formats:

IPv4 Datagram Formats:- Packets in the network (internet) 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.

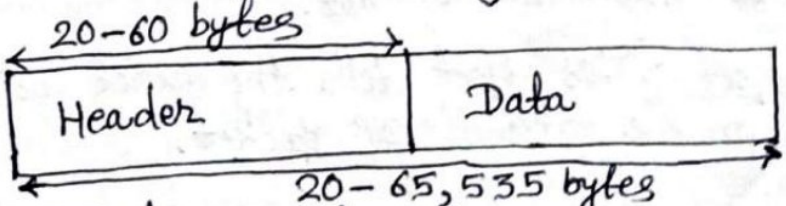


fig. IP Datagram for IPv4

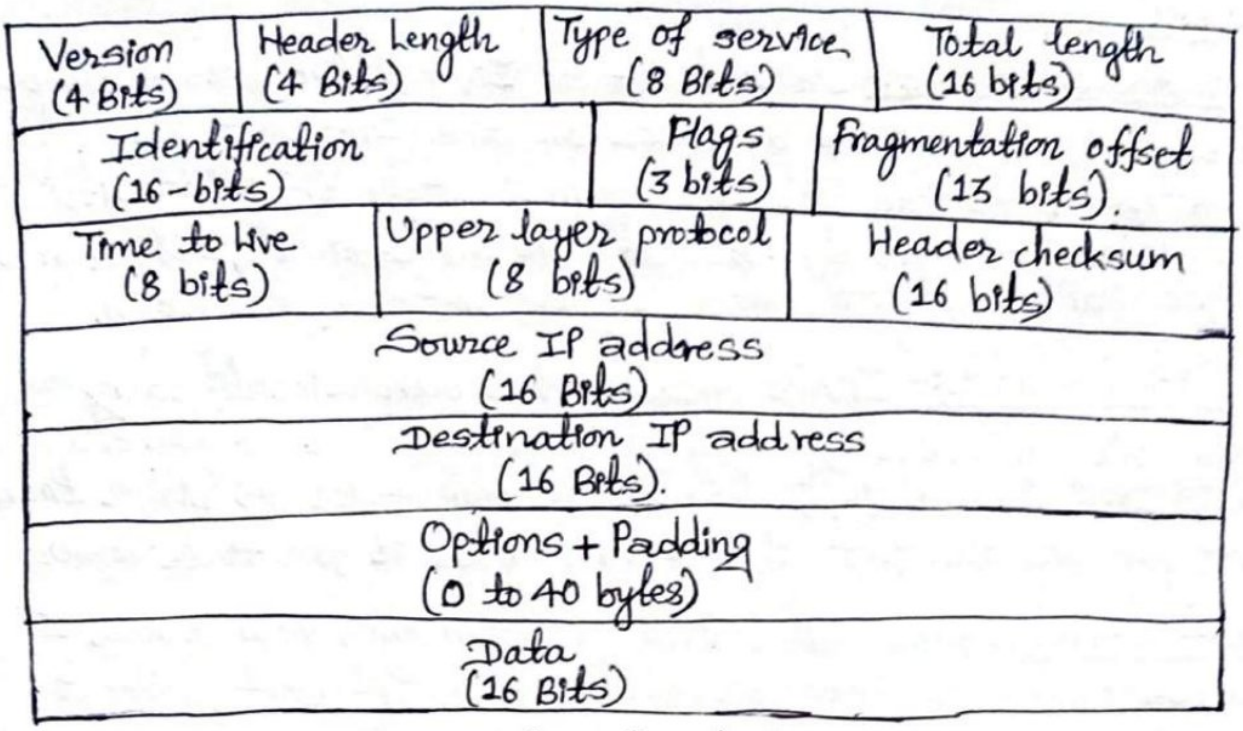


Fig. Header format of IPv4.

Version Number → It specifies the IP protocol version of the datagram. By looking at the version number the router can determine how to interpret the remainder of the IP diagram.

Header Length → It determines where in the IP datagram the data actually begins. Most IP datagrams do not contain options, so the typical IP datagram has a 20-byte header.

Type of service → The type of service bits were included in the IPv4 header to allow different types of IP datagrams to be distinguished from each other.

Datagram Length (Total Length) → Since this field is 16-bits long, the theoretical maximum size of the IP datagram is 65,535 bytes. However, datagrams are rarely larger than 1500 bytes.

Length of data = total length - 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 → If IP Packet is too large to handle, these flags tells if they can be fragmented or not.

Fragmentation 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 network how many routers this packet can cross.

Protocol → This field value indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed.

Header checksum → This field is used to keep checksum value of entire header.

Source address → This 32-bit field define the address of the sender (or source) of the packet.

Destination address → This 32-bit field define the address of the receiver (or destination) of the packet.

Options → This is optional field, which is used if the value of IHL is greater than 5.

Data → It contains the transport layer segment (TCP or UDP) to be delivered to destination.

IPv6 Datagram Formats:

IPv6 Datagram format has a much ~~larger~~ simpler packet header compared with IPv4, by including only the information needed for forwarding the IP datagram. IPv6 header allows the routers to process the IPv6 datagram packets more efficiently.

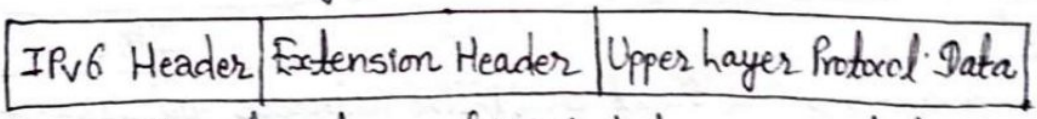


Fig. Structure of IPv6 datagram packet.

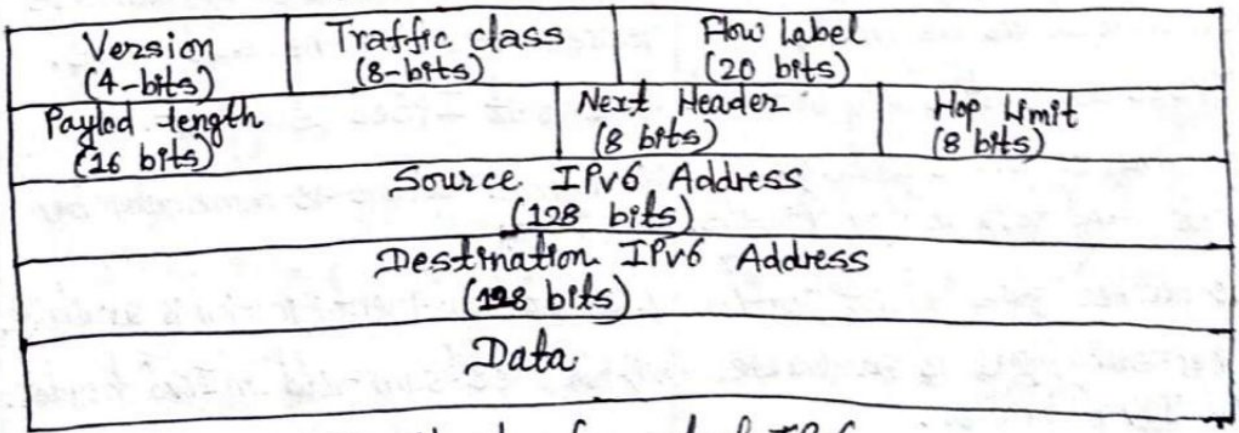


Fig. Header format of IPv6.

Version → It represents the version of Internet Protocol i.e, 0110.

Traffic Class → Among 8 bits, the most significant 6 bits are used for type of service and least significant 2 bits are used for Explicit Congestion Notification (ECN).

Flow label → This label is used to maintain the sequential flow of the packets belonging to a communication. This field also helps to avoid re-ordering of data packets.

Payload length → This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper layer data.

Next header → This field identifies the protocol to which the contents of this datagram will be delivered.

Hop limit → This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4.

Source Address → The field indicates the address of originator of the packet.

Destination Address → It provides the address of intended recipient of the packet.

Data → This is payload portion which will be removed from the IP datagram and passed on to the protocol specified in the next header field when the datagram reaches its destination.

Comparision of IPv4 and IPv6 Addressing:-

IPv4	IPv6
i) IPv4 addresses are 32-bit length.	i) IPv6 addresses are 128-bit length.
ii) IPv4 addresses are binary numbers represented in decimals.	ii) IPv6 addresses are binary numbers represented in hexadecimals.
iii) IPSec support is only optional.	iii) Inbuit IPSec support.
iv) Fragmentation is done by sender and forwarding routers.	iv) Fragmentation is done only by sender.
v) No packet flow identification.	v) Packet flow identification is available.
vi) Checksum field is available in IPv4 header.	vi) No checksum field in IPv6 header.
vii) It can generate 4.29 X 109 addresses.	vii) It can generate 3.4 X 1038 addresses.

Network Address Translation (NAT):

The number of home users and small businesses that want to use the internet in the past were connected to the Internet with a dial-up line. Which means that connection was for specific period of time. But the situation is different today. Home users and small businesses can be connected by an ADSL line or cable modem. Many are not happy with one address. 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 can use the small set.

To seperate 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 below:

Range	Total
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. They are unique inside the organization, but they are not unique globally.

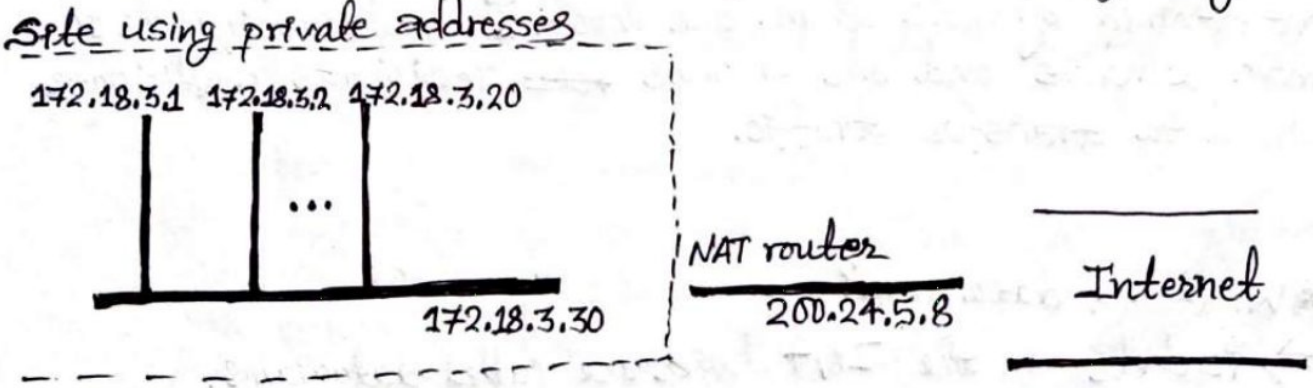


Fig: A NAT Implementation.

⊛. Example Addresses: Unicast, Multicast and Broadcast

Unicast Addressing Mode:- In this mode, data is sent only to one destined host. The destination field contains 32-bit IP address of the destination host. Here the client sends data to the targeted server. This type of information transfer is useful when there is a participation of single sender and single recipient. It is one-to-one transmission.

one to all transmission

Broadcast Addressing Mode:- It is classified into two types:

1) Limited Broadcasting → This method is used when we have to send stream of packets to all the devices over the network. It will append 255.255.255.255 (all the 32 bits of IP address set to 1) called as limited broadcast address in the destination address of the datagram header.

2) Direct Broadcasting → This is useful when a device in one network wants to transfer packet stream to all the devices over the other network. This is achieved by translating all the Host ID part bits of the destination address to 1, referred as Direct Broadcast Address in the datagram header for information transfer.

Multicast Addressing Mode:-

This mode is the mix of unicast and broadcast addressing mode. In this packet, the destination address contains a special address which starts with 224.x.x.x and can be entertained by more than one host. In multicasting, one or more senders and one or more recipients participate in data transfer traffic.

⊛ What is datagram?

⇒ Packets in the IPv4 layer are called datagrams.

Packets → Small segment of a larger message are packets. Data sent over computer networks such as internet is divided into packets.

* Routing

Routing is the process of selecting a path for traffic in a network or between or across multiple networks. Routing is broadly performed in many types of networks, such as the public switched telephone network (PSTN) and computer networks, such as the Internet. Routing is the path that network data or a packet takes to reach it's destination on a network. The routing process usually directs forwarding on the basis of routing tables. Routing tables maintain a record of the routes to various network destinations.

⊛. Types of Routing:-

@ Static vs Dynamic:

Static Routing → Static routing is a process in which we have to manually add routes in routing table. Static routing is used when we have very few devices to configure and when we know the routes which probably never change. Static routing does not handle failures in external networks well because any route configured manually must be updated or reconfigured manually to fix or repair lost connectivity.

Advantages:

→ Easily implemented in a small network.
→ No ~~other~~ overheads are produced on router CPU.
→ Secure because the routes are managed statically.
→ Bandwidth usage is not required between routers.

Disadvantages:

→ Unsuitable for large networks.
→ Large networks increase configuration complexity and time consumption.
→ Link failure can hinder traffic rerouting.
→ The administrator must be extra careful while configuring the routes.

Dynamic Routing → Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and routes to reach it. Automatic adjustment will be made to reach the network destination if one route goes down.

Advantages:

→ Suitable for all topologies.
→ Network size doesn't affect the router operations.
→ Topologies are adapted automatically to reroute the traffic.

Disadvantages:

→ Initially it could be complicated to implement.
→ Routes rely on current topologies.
→ The broadcasting and multicasting of routing updates makes it less secure.

(b) Unicast vs. Multicast:

Unicast → It is the simplest form of routing because destination is already known. In unicast there is only one sender and only one receiver. When we want to send data to multiple people then unicast will waste lots of bandwith. It does not perform well while streaming medias. An example of unicast is surfing web.

Multicast → In multicast routing data is sent only nodes which wants to receive the packets. In multicast there is only one sender but multiple receivers. When we want to send data to multiple people then multicast will utilize the bandwidth more efficiently. It does not perform well across large networks. An example of multicast is stock exchange.

(c) Link State vs Distance vector:

→ In link state, bandwidth required is more due to flooding and sending of large state packets. But in distance vector, bandwidth required is less due to local sharing, no flooding and sending of small state packets.

→ Link state make use of Dijkstra's algorithm while distance vector make use of Bellman Ford algorithm.
→ Link state routing has more traffic while distance vector routing has less.
→ Link state routing has faster coverages while distance vector routing has ~~low~~ slower.
→ Link state routing has difficult configuration while distance vector routing has easy configuration.
→ Link state routing has hierarchical structure while distance vector routing doesn't have hierarchical structure.

d) Interior vs Exterior:-

Interior routing protocols are designed for use within a contained network of limited size ~~whee~~ whereas exterior routing protocols are designed to link multiple networks together. Nearly all routing protocols are interior routing protocols. Only BGP is commonly used as an exterior routing protocol. Interior gateway protocol (IGP) used to refer to ~~inter ro~~ interior routing protocols and exterior gateway protocol (EGP) used to refer to exterier routing protocols.

⊛ Path Computation Algorithms:

Path computation algorithms are algorithms that helps to compute shorted path from a source to destination among several paths. Bellman Ford and Dijkstra algorithm are two main path computation algorithms.

a) Bellman Ford Algorithm:-

The Bellman-Ford algorithm is an algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted graph. It is slower than Dijkstra's algorithm but more versatile, as it is capable of handling graphs in which some of the edge weights are negative numbers.

Similar to Dijkstra's algorithm this algorithm also initializes the source node to 0 and other nodes to infinity.

Then we go on relaxing all the edges repeatedly for n-1 times. Where n is the number of vertices.

What is Relaxation?

Ans:- If (u,v) is an edge between two vertices. then,

$$\text{if}\left(d[u] + W(u,v) < d[v]\right)$$

then,

$$d[v] = d[u] + W(u,v)$$

min cost weight recently calculated

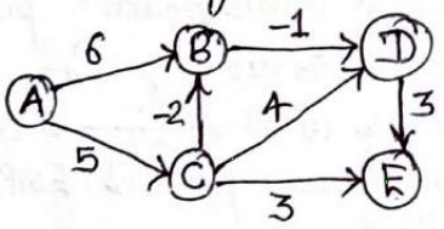
where,

d[u] = weight at source vertex

d[v] = weight at destination vertex.

W(u,v) = weight of edge from source to destination.

Example:- Find the shortest path of following graph by using Bellman Ford algorithm.



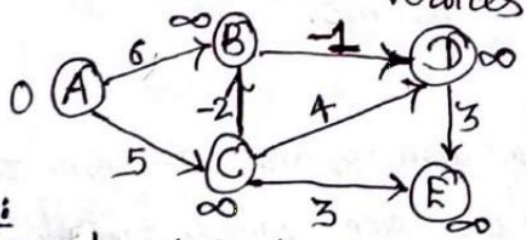
Solution:-

Let source vertex be A then we write all the edges in any order as: (A,B), (A,C), (C,B), (C,D), (C,E), (B,D), (D,E).

not compulsory for easier calculation

Since there are 5 vertices i.e, n=5, so no. of iterations for solving problem is (n-1) = (5-1) = 4.

Now we initialize the vertices and redraw the figure as;



i.e,

A	B	C	D	E
0	∞	∞	∞	∞

Iteration-1:

Now For edge AB: $d[A] + W(A,B) < d[B]$

or, $0 + 6 < \infty$ [True]

Then, $d[B] = d[A] + W(A,B)$

$= 0 + 6$

$= 6$

formula
if(source + edge < destination)
then destination = source + edge

Similarly for edge AC:

$d[A] + W(A,C) < d[C]$ i.e, $0 + 5 < \infty$ [True]

So, $d[C] = d[A] + W(A,C)$ or $d(C) = 0 + 5 = 5$

For edge CB:

always put recently calculated value

$d[C]+w(C,B) < d[B]$

$= 5+(-2) < 6$ [True]

So, $d[B] = d[C]+w(C,B)$

$= 5+(-2)$

$= 3$

For edge CD

$d[C]+w(C,D) < d[D]$

$= 5+4 < \infty$ [True]

So, $d[D] = 9$

For edge CE

$d[C]+w(C,E) < d[E]$

$= 5+3 < \infty$ [True]

So, $d[E] = 8$

For edge BD

$d[B]+w(B,D) < d[D]$

$= 3+(-1) < 9$ [True]

$= 2 < 9$

So $d[D] = 2$

For edge DE

$d[D]+w(D,E) < d[E]$

$= 2+3 < \infty$ [True]

So, $d[E] = 5$

Now, we continue other iterations using table as follows:-

Iterations.	A	B	C	D	E
Initialization	0	∞	∞	∞	∞
1st Iteration		3	5	2	5
2nd Iteration		3	5	2	5
3rd Iteration		3	5	2	5
4th Iteration		3	5	2	5

since $0+6 < 3$ [False] not true, so, we copy same previous value

lowest values from calculated values are put here →.

Here we get all values same as above so we can stop calculation here, but computer algorithm will go upto n-1 iterations.

We may get same values like this in any iteration according to question, Maybe 3rd iteration or, 4th or any other last iteration. Not necessary to get in second iteration.

Hence, shortest distances for respective vertices are as follows:-

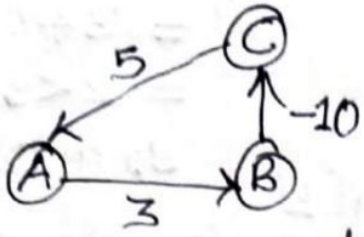
A = 0
B = 3
C = 5
D = 2
E = 5

Time Complexity of Bellman ford Algorithm: If E is the no. of edges then time complexity is $O(E(V-1))$. Since we iterate V-1 times. Or we can write it as $O(E.V)$. If we take E and V both as n, we can write it as $O(n^2)$. In case of complete graph we have $\frac{n(n-1)}{2}$ no of edges. So, in this case time complexity will be $O[\frac{n(n-1)}{2}(n-1)]$.

Drawback of Bellman Ford Algorithm:

The main drawback of Bellman Ford algorithm is that, if we have negative weight cycle in graph then we can not get the correct solution.

For example:



Here, $5+3-10=-2$ (which is negative weight cycle).

b) Dijkstra's Algorithm:-

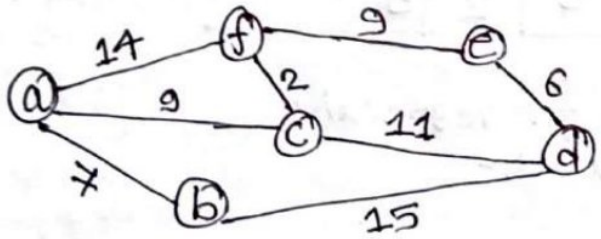
This is another approach of getting single source shortest paths. In this algorithm it is assumed that there is no negative weighted edge. This algorithm also starts with initializing source vertex to 0 and remaning other vertex to infinity. Then we start finding shortest path for each vertex using formula as;

if $(d[u]+w(u,v)<d[v])$ then,

$$d[v]=d[u]+w(u,v).$$

where, u = source vertex.
v = destination vertex.

Example:- Find the shortest paths from source node to all other vertices using Dijkstras algorithm.



Solution:- Let source vertex be a, so we initialize initially source vertex 'a' with weight 0 and ∞ to all other remaining vertices. Now, we construct table for faster calculations. Calculations are same as Bellman Ford Algorithm only difference is that we use n-1 iterations there but no. of iterations are not fixed in this method.

once the vertex is selected no need to relax next time

Vertex Selected ↓	a	b	c	d	e	f
a	[0]	∞	∞	∞	∞	∞
b		[7] ←	9	∞	∞	14
c			[9]	22	∞	14
f				20	∞	[11]
d				[20]	20	
e					[20]	

[ ] → shows selected vertex. which is min in row.

calculations done in rough as;

~~d[b]>d~~

d[a]+w(a,b) < d[b]

0+7 < ∞ [True]

∴ d[b]=7.

Similarly for others.

Hence, the shortest paths with weights for different vertex are as follows;

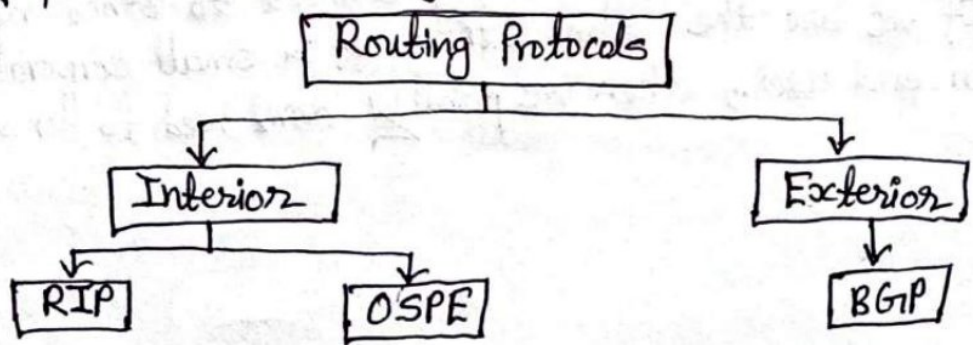
Shortest path from a to b = {a, b} with weight 7.
" " " a to c = {a, c} with weight 9.
" " " a to d = {a, c, d} with weight 20.
" " " a to e = {a, c, f, e} with weight 20.
" " " a to f = {a, c, f} with weight 11.

Disadvantage:- This algorithm ~~[redacted]~~ may not work for negative weighted edge having graph.

Time complexity:- $O(V^2)$ where, V is no. of vertices in the graph.

⊛ Routing Protocols:-

The routing protocol specifies how routers communicate, to select the routes for data transfer. Different types of routing protocols are as follows:-



a) Routing Information Protocol (RIP):

In RIP distance vector routing protocol is used for data transmission. The maximum number of Hop in RIP is 15. Mechanism like split horizon, holdown etc. are used to prevent from incorrect or wrong routing information. RIP is a dynamic protocol used to find the best route from source to destination over a network. Compared to other routing protocols RIP is poor and limited sized network. RIP v1 (version 1), RIP v2 (version 2) and RIPng (next generation) are the types of routing information protocol (RIP).

~~b) Border Gateway Protocol (BGP):~~

b) Open Shortest Path First (OSPF):

It is the link-state routing protocol which is used to find the best path between the source and the destination. OSPF is developed by Internet Engineering Task Force (IETF) as one of the Interior Gateway Protocol (IGP). It is a network layer protocol which works on the protocol number 89 and used AD value 110. OSPF uses multicast address 224.0.0.5 for normal communication and 224.0.0.6 for update to designated router.

c) Border Gateway Protocol (BGP):

BGP are the core routing protocol of the internet and responsible to maintain a table of Internet protocol networks which authorize network reaching capability between AS. BGP is expressed as path vector protocol. BGP is relevant to network administrators of large organizations which connect to two or more ISPs, as well as to ISPs who connect to other network providers. If we are the administrator of a small corporate network, or an end user, then we probably don't need to know about BGP.

Comparision of OSPF and BGP

OSPF	BGP
i) OSPF is an internal gateway protocol.	i) BGP is an external gateway protocol.
ii) OSPF is comparatively easy to implement.	ii) BGP is comparatively complex to implement.
iii) Port number 89 is used.	iii) Port number 179 is used.
iv) IP protocol is used.	iv) TCP protocol is used.
v) OSPF is mainly used on smaller scale networks that are centrally administered.	v) The BGP protocol is mainly used on very large-scale networks, like the internet.
vi) Dijkstra's algorithm is suitable to implement OSPF routing protocal.	vi) Best path algorithm is suitable to implement BGP routing protocol.

Overview of IPv4 to IPv6 Transition Mechanisms:-

In modern devices both versions IPv4 and IPv6 exist today simultaneously. Following are the some methods that can be used when transitioning a network from IPv4 to IPv6.

a) Dual Stack:- The process of running both IPv4 and IPv6 on the same devices is called dual stack. It is the simplest method to run IPv6 on all of the devices that are currently running IPv4. It is easy to implement, however IPv6 is not supported on all of the IPv4 devices, in these situations other methods must be considered.

b) Tunneling:- The process of transporting IPv6 traffic through an IPv4 network transparently is called tunneling. In this method a packet is encapsulated into a wrapper then enables its transport from a source to destination where it is decapsulated and retransmitted. The following list shows the different available tunneling methods:-

i) Manual IPv6 tunnels
ii) 6 to 4 tunnels
iii) IPv6 rapid deployment

iv) IPv4 compatible tunnels.
v) Generic Routing Encapsulation (GRE) IPv6 tunnels.

c) Translation:- The process of converting IPv6 traffic to IPv4 traffic for transport and vice versa is called translation. When using translation, the traffic is not encapsulated but is converted to the destination type. There are two methods of translation:

i) Network Address Translation → This method enables the ability to either statically or dynamically configure a translation of a IPv4 network address into an IPv6 network address and vice versa.

ii) NAT 64 → NAT 64 offers both stateless and stateful option when deploying.

⊛. Overview of ICMP/ICMP v6:-

ICMP stands for Internet Control Message Protocol which depends on internet to provide an error control. Since IP does not have a inbuilt mechanism for sending error and control messages. ICMP is used for reporting errors and management queries. It is a supporting protocol and is used by network devices like routers for sending the error messages and operations information.

e.g: The requested service is not available.

ICMP v6 is the version 6 of ICMP which plays far more important role in the operation of IPv6. ICMP v6 is used for several purposes beyond simple error reporting and signaling. It is used for:

→ Neighbour Discovery
→ Router Discovery
→ Managing hand-offs in Mobile IPv6.

⊛ Security Concepts: Firewall & Router Access Control

Firewall → A Firewall is a network security device that monitors incoming and outgoing network traffic and permits or blocks data packets based on a set of security rules. Its purpose is to establish a barrier between our internal network and incomming traffic from external sources (such as the internet) in order to block malicious traffic like viruses and hackers.

Firewalls carefully analyze incoming traffic based on pre-established rules and filter traffic comming from unsecured or suspicious sources to prevent attacks. Firewalls guard traffic at a computer's entry points called ports where information is exchanged with external devices.

Firewalls can either be software or hardware, though it's best to have both. A software firewall is a program installed on each computer and regulates traffic through port numbers and applications, while a physical firewall is a piece of equipment installed between network and gateway.

Firewalls can be divided into several different categories based on their general structure and method of operation. Following are some types:

- → Packet filtering firewalls
- → Circuit-level gateways
- → Stateful inspection firewalls
- → Application level gateways
- → Software firewalls
- → Hardware firewalls etc.