Chapter 4
Network Layer

Dr. Nilesh Patil

DJSCE

Unit	Description	Duration	CO
4	**Network layer**: Network Layer design issues, Communication Primitives: Unicast, Multicast, Broadcast. IPv4 Addressing (classful and classless), Subnetting, Supernetting design problems, IPv4 Protocol, Network Address Translation (NAT), IPv6 **Routing algorithms**: Shortest Path (Dijkstra’s), Link state routing, Distance Vector Routing **Routing Protocols**: ARP, RARP, ICMP, IGMP, RIP, OSPF **Congestion control algorithms**: Open loop congestion control, Closed loop congestion control, QoS parameters, Token & Leaky bucket algorithms.	07	CO3

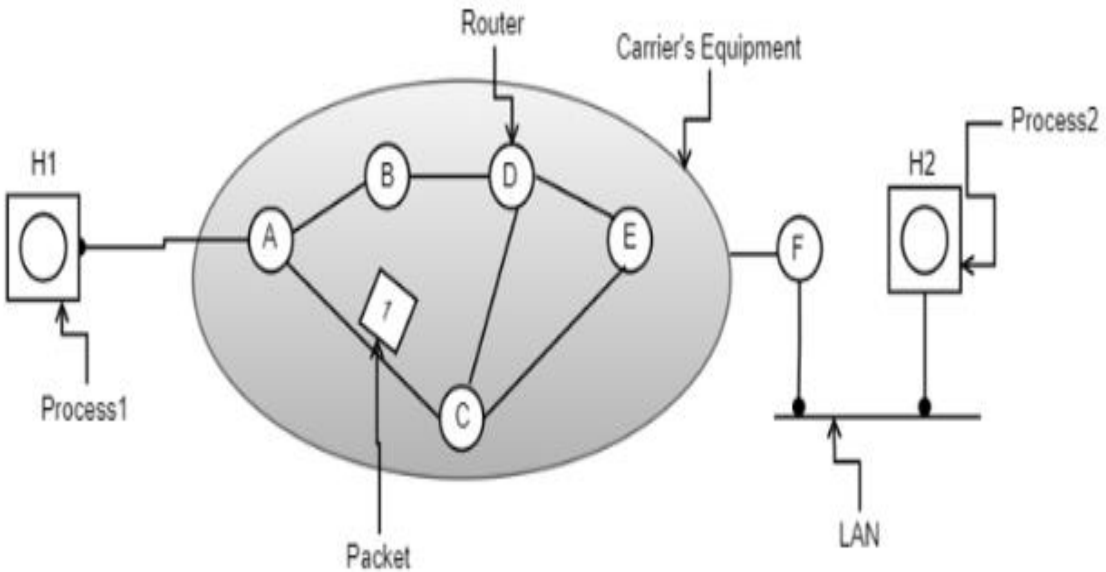
Network Layer Design Issues

- Network layer is majorly focused on getting packets from the source to the destination, routing, error handling and congestion control.
- It is the lowest layer that deals with end-to-end transmission.
- The network layer comes with some design issues described as follows:

1. Store and Forward packet switching
2. Services provided to Transport Layer (Connectionless, Connection-Oriented)
3. Implementation of Connectionless Service
4. Implementation of Connection Oriented service (Circuit Switched Connection, Virtual Circuit Switched Connection)

Store and Forward packet switching

- H1 has a direct connection with carrier router ‘A’, while H2 is connected to carrier router ‘F’ on a LAN connection.
- One of the carrier router ‘F’, is pointed outside the carrier’s equipment as it does not come under the carrier, whereas considered as protocols, software, and construction.
- This switching network performs as transmission of data happens when the host (H1) with a packet transfers it to the nearby router through LAN (or) point-to-point connection to the carrier.
- The carrier stores the packet until it completely arrives thus confirms the checksum.
- Then after, the packet is transmitted over the path until H2 is reached.



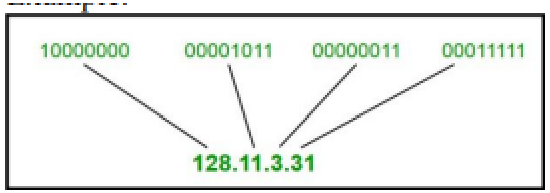
Circuit – Switching	**Packet – Switching**
Circuit switching is a method that is used when a dedicated channel or circuit needs to be established.	Packet switching is a method of grouping data which is transmitted over a digital network into packets.
Circuit switching connections are classified into two categories half-duplex or full-duplex.	Packet Switching is a connectionless network switching method.
You need to establish a dedicated path between the source and the destination before the transfer of data commences.	You do not need to establish a dedicated path from the source to the destination.
It was initially designed for voice transfer.	It was initially designed for data transfer.
It is implemented at Physical Layer.	It is implemented at Network Layer.
In-Circuit switching, data is processed and transmitted at the source only.	In packet switching, data is processed and transmitted, not only at the source but also at the destination.
Its initial cost is low.	Packet switching demands high installation costs.
The protocols for delivery are simpler.	It requires complex protocols for delivery.
Charging happens per minute.	Charging happens per packet.
Each packet follows the same route.	Each packet does not follow the same route.
It does not store and forward transmission.	It does store and forward transmission.
It is an inflexible method because once a path is set, all parts of a transmission follow the same path.	It is a flexible method because the route is created for each packet to travel to the destination.
The message is received in the order, which is sent from the source.	In, packet switching message are received out of order, which is assembled at the destination.
Reserve the entire Bandwidth in advance.	Never reserves the Bandwidth.
You can achieve Circuit switching using two technologies 1) Time or 2) Space Division Switching.	Packet Switching has Datagram Virtual Circuit Approach.

Key	Virtual Circuits	Datagram Networks
Definition	Virtual Circuit is a connection-oriented service in which there is an implementation of resources like buffers, CPU, bandwidth, etc., used by virtual circuit for a data transfer session.	Datagram networks are a type of connectionless service where no such resources are required for data transmission.
Path	In Virtual circuits, as all the resources and bandwidth get reserved before the transmission, the path which is utilized or followed by first data packet would get fixed and all other data packets will use the same path and consume same resources.	In a Datagram network, the path is not fixed as data packets are free to decide the path on any intermediate router on the go by dynamically changing routing tables on routers.
Header	As there is same path followed by all the data packets, a common and same header is being used by all the packets.	Different headers with information of other data packet is being used in Datagram network.
Complexity	Virtual Circuit is less complex as compared to that of Datagram network.	Datagram network are more complex as compared to Virtual circuit.
Reliability	Due to fixed path and assurance of fixed resources, Virtual Circuits are more reliable for data transmission as compared to Datagram network.	Datagram networks, due to their dynamic resource allocation and dynamic path, are more errorprone and less reliable than Virtual circuits.
Example and Cost	Virtual circuits are costlier in installation and maintenance. They are widely used by ATM (Asynchronous Transfer Mode) Network, which is used for the Telephone calls.	Datagram networks are cheaper as compared to the Virtual Circuits. They are mainly used by IP network, which is used for Data services like Internet.

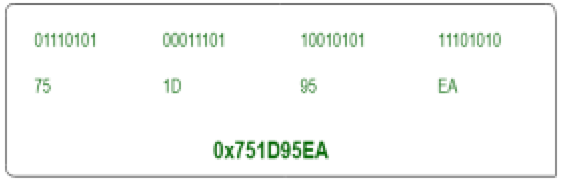
Communication Primitives

- Data is transported over a network by three simple methods i.e. Unicast, Broadcast, and Multicast.
- Unicast: From one source to one destination i.e. One-to-One. Example: Telephone call
- Broadcast: From one source to all possible destinations i.e. One-to-All Example: Cable Television
- Multicast: From one source to multiple destinations stating an interest in receiving the traffic i.e. One-to-Many. Example: Email

IPv4 Addressing



Dotted Decimal Notation



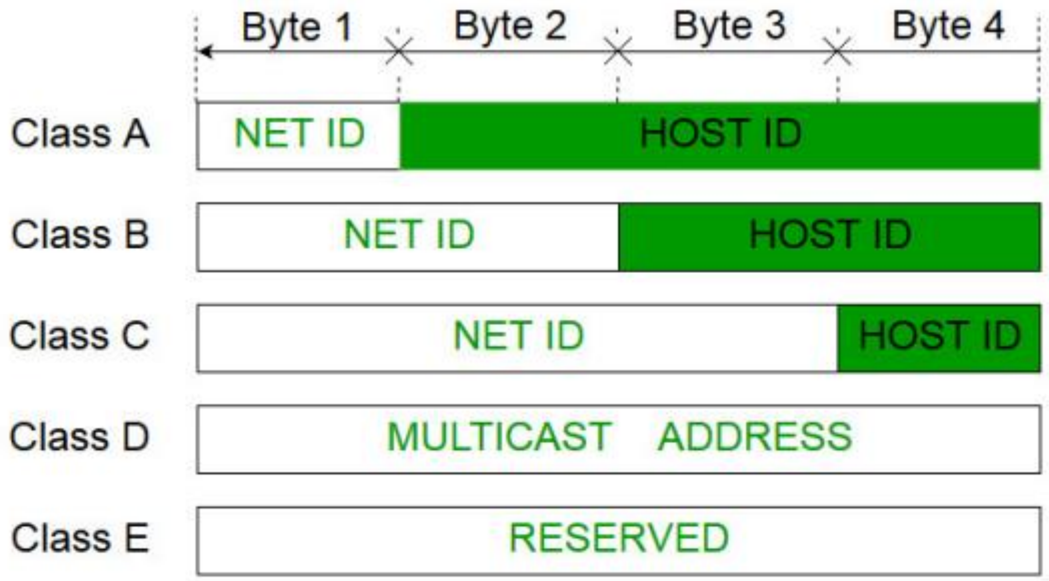
Hexadecimal Notation

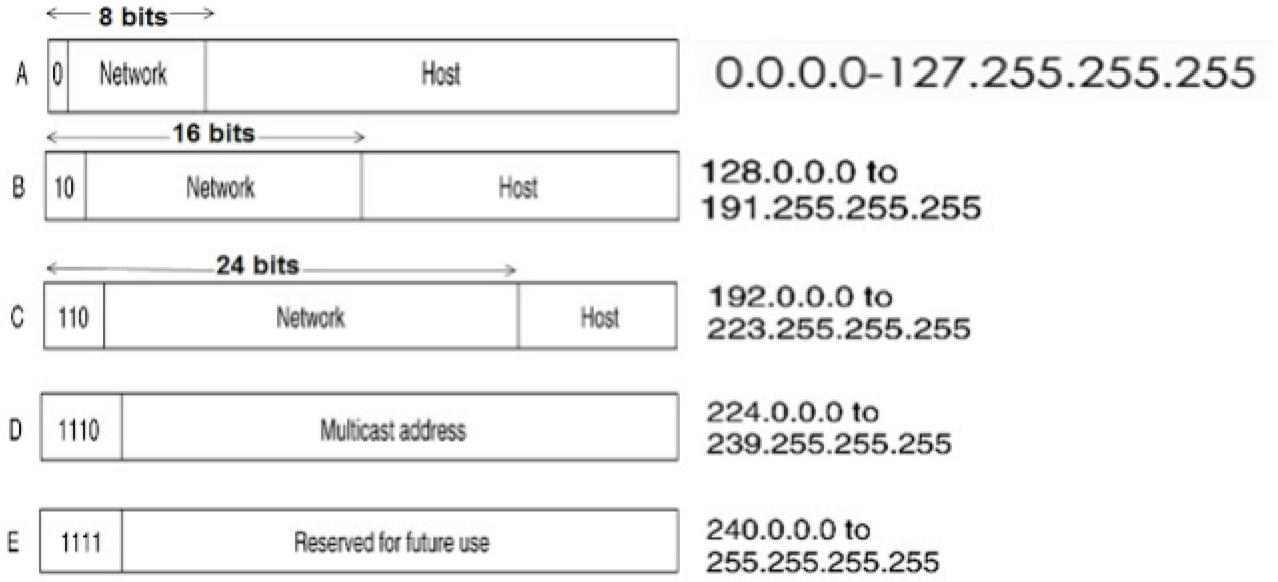
- The IPv4 address is a 32-bit number that uniquely identifies a network interface on a system.
- An IP address can be written in three notations; dotted-decimal, binary and hexadecimal.
- Among these types, dotted-decimal is the most popular and frequently used method for writing an IP address.
- An IPv4 address written in decimal digits is divided into four 8-bit fields that are separated by periods. Each 8-bit field represents a byte of the IPv4 address.
- Some points to be noted about dotted decimal notation:

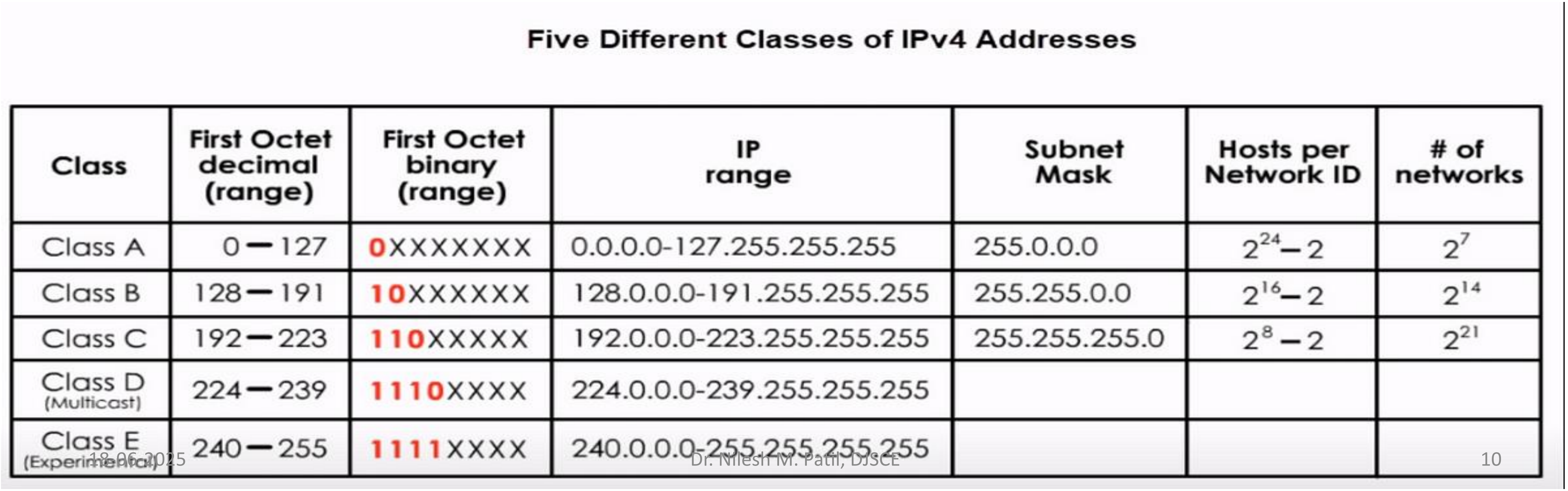
1. The value of any segment (byte) is between 0 and 255 (both included).
2. There are no zeroes preceding the value in any segment (054 is wrong, 54 is correct).

Classfull IP Addressing

- The 32 bit IP address is divided into five sub-classes.
- These are:
  - ➢ Class A
  - ➢ Class B
  - ➢ Class C
  - ➢ Class D
  - ➢ Class E
- Each of these classes has a valid range of IP addresses.
- Classes D and E are reserved for multicast and experimental purposes respectively.
- IPv4 address is divided into two parts:
  - ➢ Network ID
  - ➢ Host ID
- The class of IP address is used to determine the bits used for network ID and host ID and the number of total networks and hosts possible in that particular class.
- Each ISP or network administrator assigns IP address to each device that is connected to its network.
- IP addresses are globally managed by Internet Assigned Numbers Authority(IANA) and regional Internet registries(RIR).
- While finding the total number of host IP addresses, two IP addresses are not counted and are therefore, decreased from the total count because the first IP address of any network is the network number and whereas the last IP address is reserved for broadcast IP.







Five Different Classes of IPv4 Addresses

Class	First Octet decimal (range)	First Octet binary (range)	IP range	Subnet Mask	Hosts per Network ID	# of networks
Class A	0 – 127	0XXXXXXX	0.0.0.0-127.255.255.255	255.0.0.0	$2^{24}-2$	2^{7}
Class B	128 – 191	10XXXXXX	128.0.0.0-191.255.255.255	255.255.0.0	$2^{16}-2$	2^{14}
Class C	192 – 223	110XXXXX	192.0.0.0-223.255.255.255	255.255.255.0	$2^{8}-2$	2^{21}
Class D (Multicast)	224 – 239	1110XXXX	224.0.0.0-239.255.255.255			
Class E (Experimental)	240 – 255	1111XXXX	240.0.0.0-255.255.255.255			

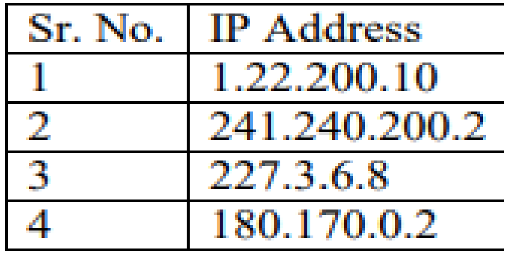
Range of special IP addresses

- 169.254.0.0 – 169.254.0.16: Link local addresses
- 127.0.0.0 – 127.0.0.8: Loop-back addresses
- 0.0.0.0 – 0.0.0.8: used to communicate within the current network.

Problems with Classfull Addressing

- The problem with this Classfull addressing method is that millions of class A address are wasted, many of the class B address are wasted, whereas, number of addresses available in class C is so small that it cannot cater the needs of organizations.
- Class D addresses are used for multicast routing and are therefore available as a single block only.
- Class E addresses are reserved.
- Since there are these problems, Classfull networking was replaced by Classless Inter-Domain Routing (CIDR) in 1993.

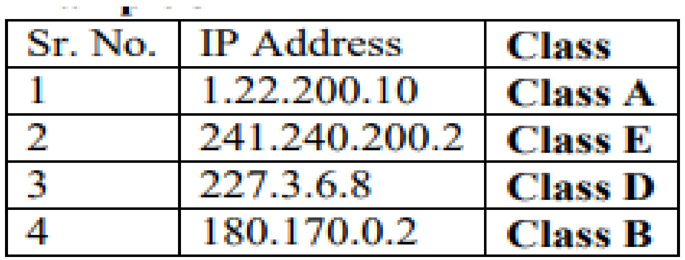
Find the class of the following IP address.



Sr. No.	IP Address
1	1.22.200.10
2	241.240.200.2
3	227.3.6.8
4	180.170.0.2

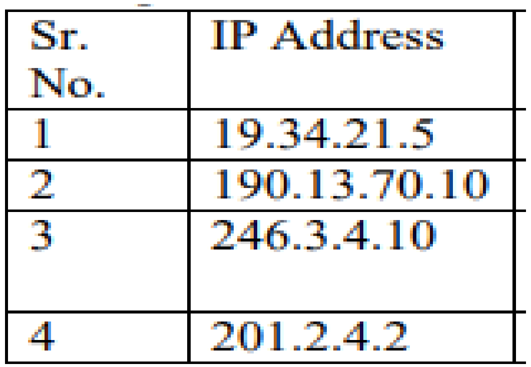
Find the class of the following IP address.

- Solution:



Sr. No.	IP Address	Class
1	1.22.200.10	**Class A**
2	241.240.200.2	**Class E**
3	227.3.6.8	**Class D**
4	180.170.0.2	**Class B**

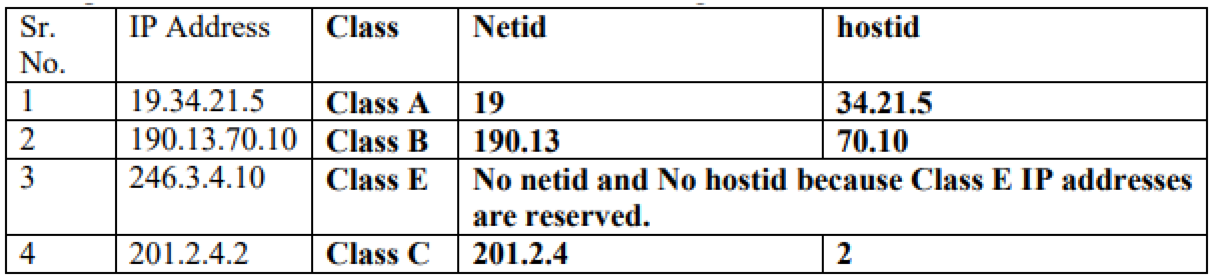
Find the netid and hostid for the following.



Sr. No.	IP Address
1	19.34.21.5
2	190.13.70.10
3	246.3.4.10
4	201.2.4.2

Find the netid and hostid for the following.

Solution:



Sr. No.	IP Address	Class	Netid	hostid
1	19.34.21.5	**Class A**	**19**	**34.21.5**
2	190.13.70.10	**Class B**	**190.13**	**70.10**
3	246.3.4.10	**Class E**	**No netid and No hostid because Class E IP addresses are reserved.**	
4	201.2.4.2	**Class C**	**201.2.4**	**2**

Extracting Information in a Block of Classful IP Address

- A block is a range of IP address.
- Given any address in the block, we normally like to know the number of addresses, the first address, and the last address about the block.
- Before extracting this information, we need to know the class of the IP address.
- Once the class of the block is found, we know the value of ‘n’, the length of netid in bits.
- The number of addresses N in the block can be found using $N = 2^{32-n}$.
- To find the first address, we keep the ‘n’ leftmost bits and set the (32 – n) rightmost bits all to 0s.
- To find the last address, we keep the ‘n’ leftmost bits and set the (32 – n) rightmost bits all to 1s.

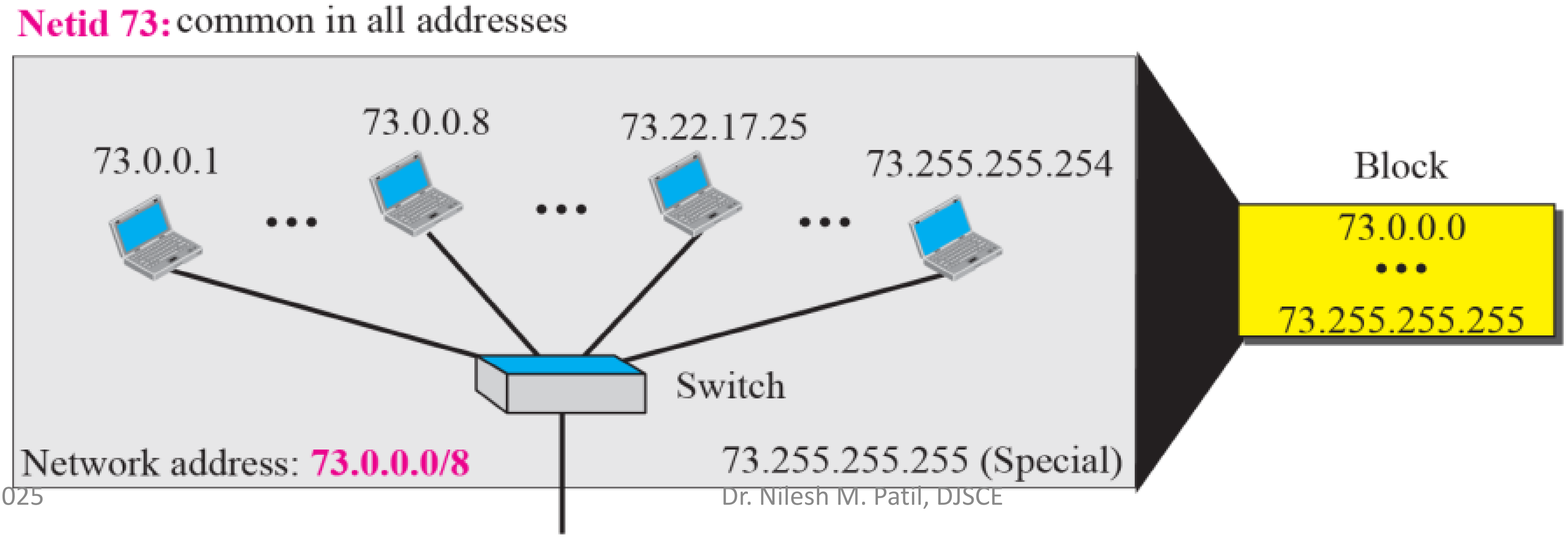
Q. An address in the block is given as 73.25.16.27. Find the number of addresses in the block, the first address, and the last address.

Solution:

- Since 73 is between 0 and 127, the class of IP address is A. The value of n for class A is 8.
- The number of addresses N in the block can be found using

$$N = 2^{32-n} = 2^{32-8} = 2^{24} = 16{,}777{,}216$$

- To find the first address, we keep the n = 8 leftmost bits and set the (32 – 8) = 24 rightmost bits all to 0s. Hence, the first address is **73.0.0.0** which is also called the network address and is not assigned to any host. It is used to define the network.
- To find the last address, we keep the n = 8 leftmost bits and set the (32 – n) = 24 rightmost bits all to 1s. Hence, the last address is **73.255.255.255** which is normally used for special purpose.
- Figure shows a possible configuration of the network that uses this block.



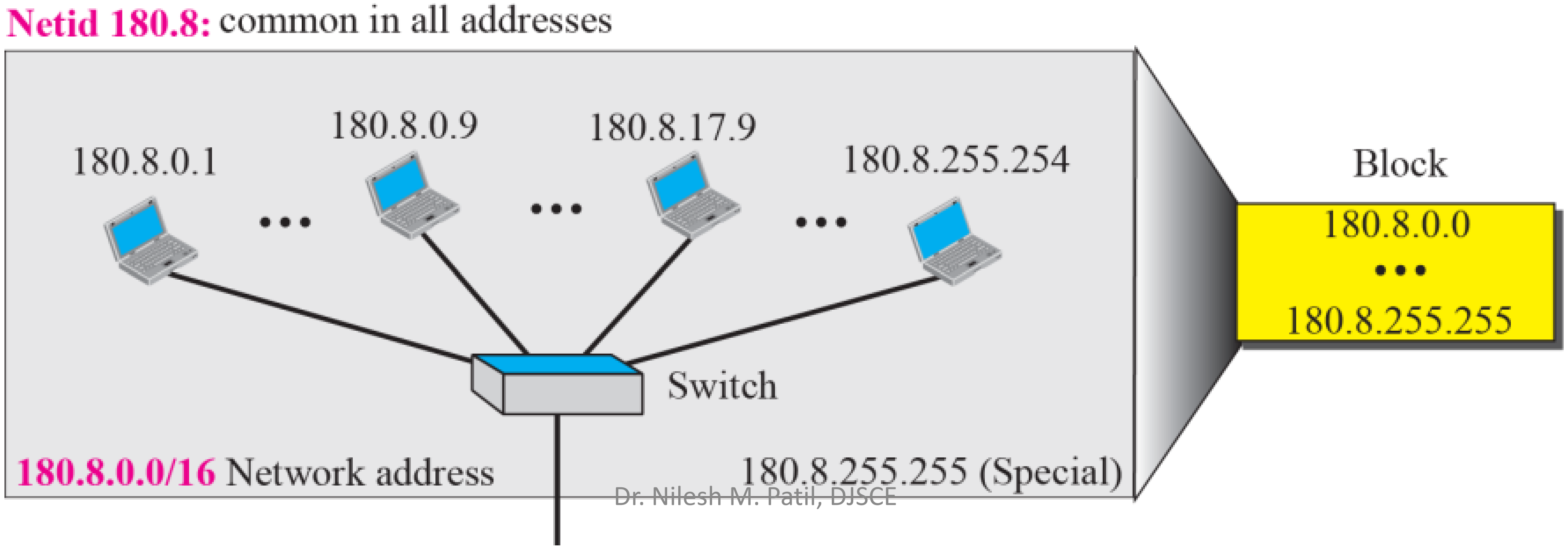
Q. An address in a block is given as 180.8.17.9. Find the number of addresses in the block, the first address, and the last address.

Solution:

- Since 180 is between 128 and 191, the class of IP address is B. The value of n for class B is 16.
- The number of addresses N in the block can be found using

$$N = 2^{32-n} = 2^{32-16} = 2^{16} = 65,536$$

- To find the first address, we keep the n = 16 leftmost bits and set the (32 – 16) = 16 rightmost bits all to 0s. Hence, the first address is **180.8.0.0** which is also called the network address and is not assigned to any host. It is used to define the network.
- To find the last address, we keep the n = 16 leftmost bits and set the (32 – n) = 16 rightmost bits all to 1s. Hence, the last address is **180.8.255.255** which is normally used for special purpose.
- Figure shows a possible configuration of the network that uses this block.



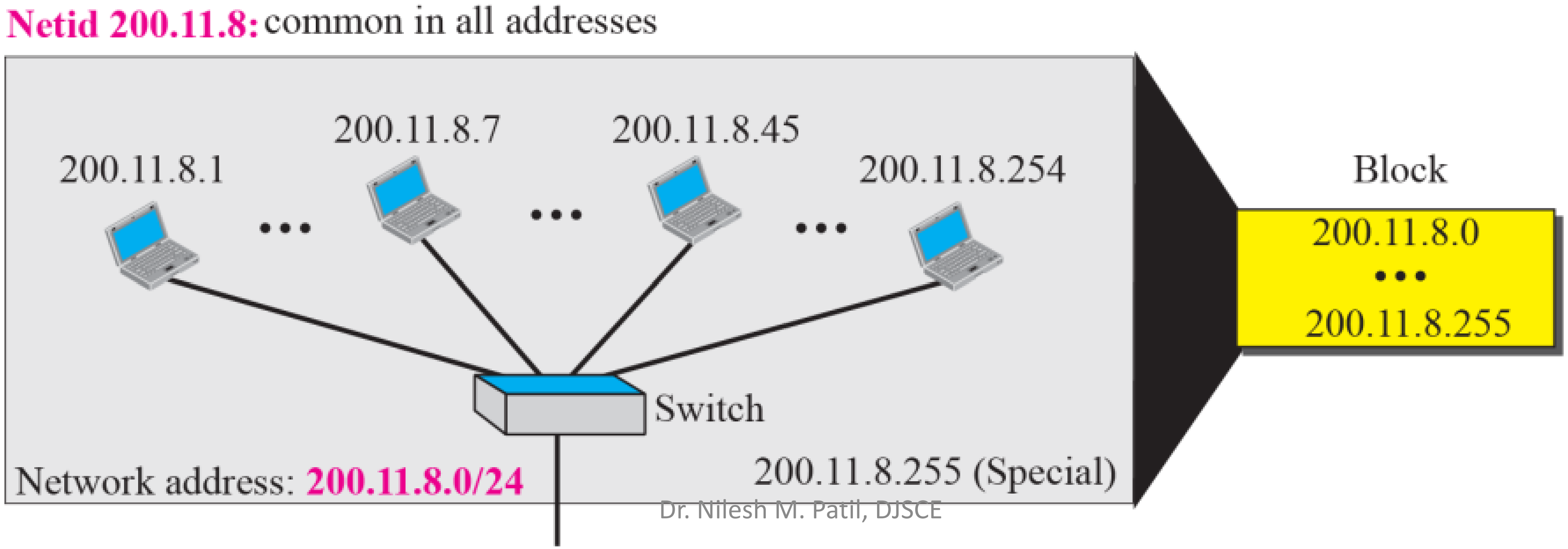
Q. An address in a block is given as 200.11.8.45. Find the number of addresses in the block, the first address, and the last address.

Solution:

- Since 200 is between 192 and 223, the class of IP address is C. The value of n for class C is 24.
- The number of addresses N in the block can be found using

$$N = 2^{32-n} = 2^{32-24} = 2^8 = 256$$

- To find the first address, we keep the n = 24 leftmost bits and set the (32 – 24) = 8 rightmost bits all to 0s. Hence, the first address is **200.11.8.0** which is also called the network address and is not assigned to any host. It is used to define the network.
- To find the last address, we keep the n = 24 leftmost bits and set the (32 – 24) = 8 rightmost bits all to 1s. Hence, the last address is **200.11.8.255** which is normally used for special purpose.
- Figure shows a possible configuration of the network that uses this block.



Classless Addressing

- To reduce the wastage of IP addresses in a block, we use sub-netting.
- What we do is that we use host id bits as net id bits of a Classfull IP address.
- We give the IP address and define the number of bits for mask along with it (usually followed by a ‘/’ symbol), like, 192.168.1.1/28.
- Here, subnet mask is found by putting the given number of bits out of 32 as 1, like, in the given address, we need to put 28 out of 32 bits as 1 and the rest as 0, and so, the subnet mask would be 255.255.255.240.

Subnetting

- Dividing a large block of addresses into several contiguous sub-blocks and assigning these sub-blocks to different smaller networks is called subnetting.
- It is also called as subnet routing or subnet addressing.
- It is a practice that is widely used when classless addressing is done.

Benefits of Subnetting

Reduced network traffic

Optimized network performance

Simplified network management

Masking

- A process that extracts the address of the physical network from an IP address is called masking.
- If we do the subnetting, then masking extracts the subnetwork address from an IP address.
- To find the subnetwork address, two methods are used. They are boundary level masking and non-boundary level masking.
- In boundary level masking, two masking numbers are considered (i.e., 0 or 255).
- In non-boundary level masking, other value apart from 0 and 255 are considered

Rules for boundary level masking

- In this mask number is either 0 or 255.
- If the mask number is 255 in the mask IP address, then the IP address is repeated in the subnetwork address.
- If the mask number is 0 in the mask IP address, then the 0 is repeated in the subnetwork address.

Rules for non-boundary level masking

- In this mask number is greater than 0 and less than 255.
- If the mask number is 255 in the mask IP address, then the IP address is repeated in the subnetwork address.
- If the mask number is 0 in the mask IP address, then the 0 is repeated in the subnetwork address.
- For any other mask numbers, bitwise AND operator is used. Bitwise ANDing is done in between mask number (byte) and IP address (byte).
- The default mask in different classes are:

Class A – 255.0.0.0

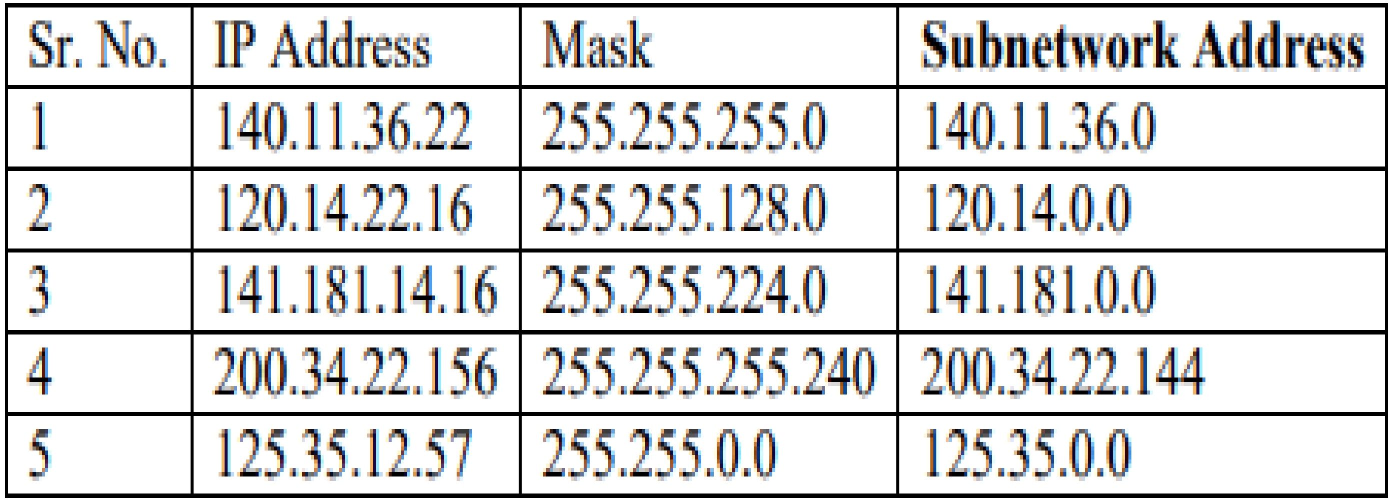
Class B – 255.255.0.0

Class C – 255.255.255.0

How to do Logical AND on Calculator?

1. Turn ON the calculator.
2. Change the mode to Base-N.
3. Select the Base as per the given question on Calculator (i.e., DEC or HEX or BIN).
4. Enter the corresponding number from IP address and press = sign.
5. Then, press SHIFT + 3, and select the AND operation and then enter the corresponding number from the mask.
6. Press = sign and get the result.

Find the subnetwork address for the following.



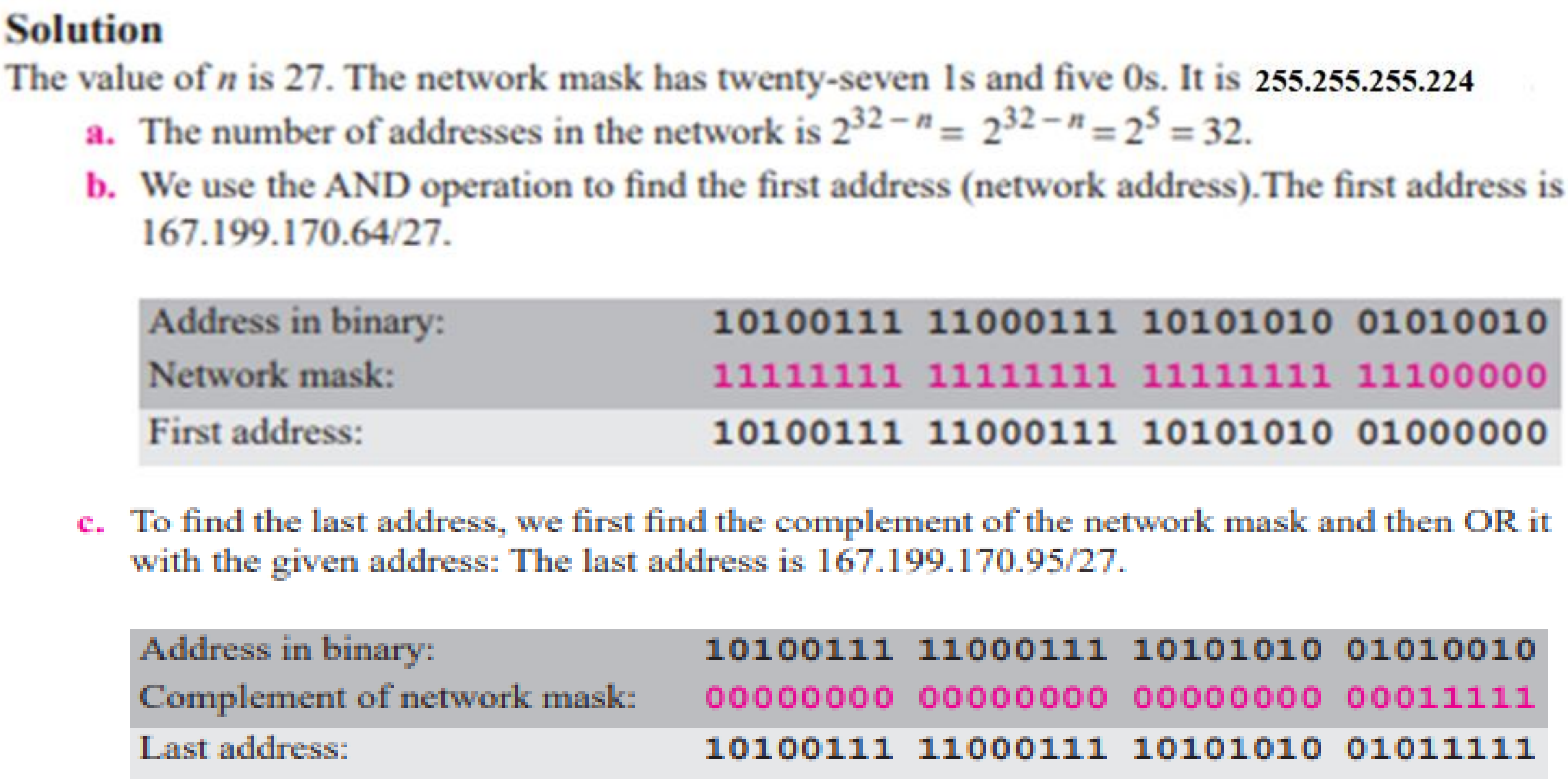
Sr. No.	IP Address	Mask	**Subnetwork Address**
1	140.11.36.22	255.255.255.0	140.11.36.0
2	120.14.22.16	255.255.128.0	120.14.0.0
3	141.181.14.16	255.255.224.0	141.181.0.0
4	200.34.22.156	255.255.255.240	200.34.22.144
5	125.35.12.57	255.255.0.0	125.35.0.0

Extracting Block Information in Classless Addressing

- An address in slash notation (CIDR) contains all information we need about the block: the first address (network address), the number of addresses, and the last address.
- These three pieces of information can be found as follows:

1. Number of addresses in the block $N = 2^{32} - n$, in which n is the prefix length and N is the number of addresses in the block.
2. First address = (any address) AND (network mask)
3. Last address = (any address) OR [NOT (network mask)]

Q. One of the addresses in a block is 167.199.170.82/27. Find the number of addresses in the network, the first address, and the last address.



Solution

The value of n is 27. The network mask has twenty-seven 1s and five 0s. It is 255.255.255.224

a. The number of addresses in the network is $2^{32-n} = 2^{32-n} = 2^5 = 32$.

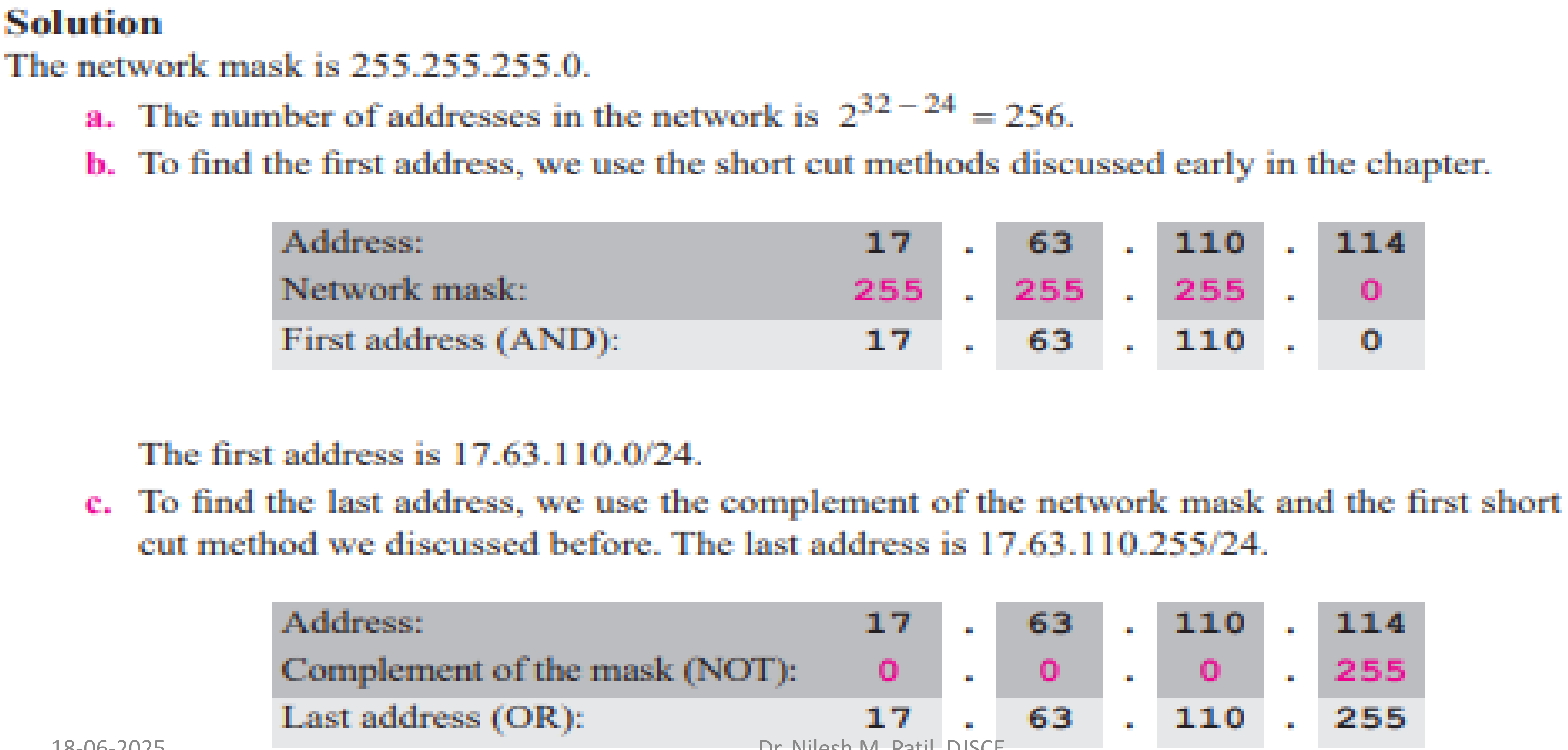
b. We use the AND operation to find the first address (network address). The first address is 167.199.170.64/27.

Address in binary:	10100111	11000111	10101010	01010010
Network mask:	11111111	11111111	11111111	11100000
First address:	10100111	11000111	10101010	01000000

c. To find the last address, we first find the complement of the network mask and then OR it with the given address: The last address is 167.199.170.95/27.

Address in binary:	10100111	11000111	10101010	01010010
Complement of network mask:	00000000	00000000	00000000	00011111
Last address:	10100111	11000111	10101010	01011111

Q. One of the addresses in a block is 17.63.110.114/24. Find the number of addresses, the first address, and the last address in the block.



Solution

The network mask is 255.255.255.0.

a. The number of addresses in the network is $2^{32-24} = 256$.

b. To find the first address, we use the short cut methods discussed early in the chapter.

Address:	17	.	63	.	110	.	114
Network mask:	255	.	255	.	255	.	0
First address (AND):	17	.	63	.	110	.	0

The first address is 17.63.110.0/24.

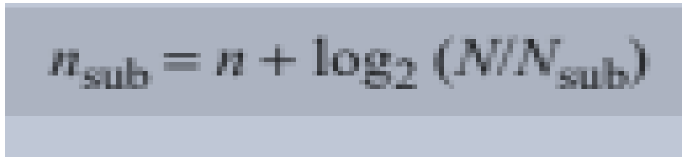
c. To find the last address, we use the complement of the network mask and the first short cut method we discussed before. The last address is 17.63.110.255/24.

Address:	17	.	63	.	110	.	114
Complement of the mask (NOT):	0	.	0	.	0	.	255
Last address (OR):	17	.	63	.	110	.	255

Designing Subnets

- The subnetworks in a network should be carefully designed to enable the routing of packets.
- We assume the total number of addresses granted to the organization is N, the prefix length is n, the assigned number of addresses to each subnetwork is N_{sub}, the prefix length for each subnetwork is n_{sub}, and the total number of subnetworks is s.
- Then, the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

1. The number of addresses in each subnetwork should be a power of 2.
2. The prefix length for each subnetwork should be found using the following formula:



$$n_{\text{sub}} = n + \log_2 (N/N_{\text{sub}})$$

3. The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger networks.

Q. An organization is granted the block 130.34.12.64/26. The organization needs four subnetworks, each with an equal number of hosts. Design the subnetworks and find the information about each network.

Solution:

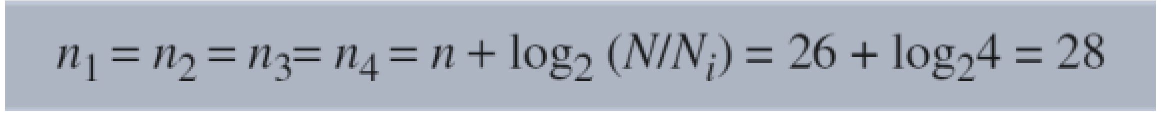
- The number of addresses N in the block can be found using

$$N = 2^{32-n} = 2^{32-26} = 2^{6} = 64$$

- Network Mask : 255.255.255.192
- First address = (any address) AND (network mask) = 130.34.12.64/26 AND 255.255.255.192 = 130.34.12.64/26
- Last address = (any address) OR [NOT (network mask)] =130.34.12.64/26 OR [NOT 255.255.255.192]
  = 130.34.12.64/26 OR 0.0.0.63 = 130.34.12.127/26

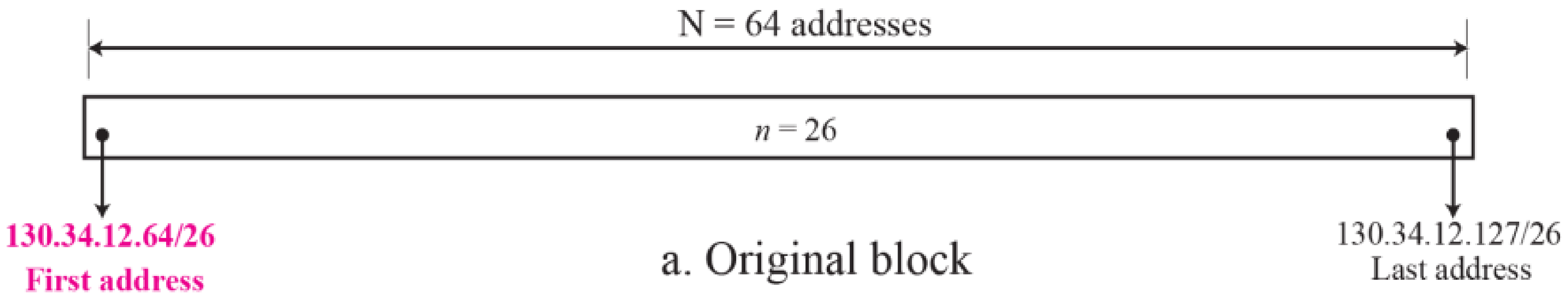
We now design the subnetworks:

1. We grant 16 addresses for each subnetwork to meet the first requirement (64/16 is a power of 2).
2. The subnetwork mask for each subnetwork is:

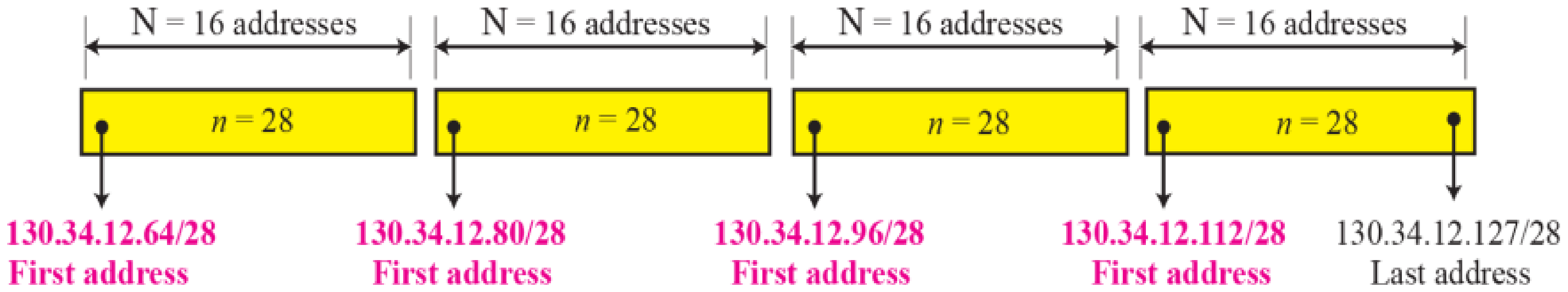


$$n_1 = n_2 = n_3 = n_4 = n + \log_2 (N/N_i) = 26 + \log_2 4 = 28$$

3. We grant 16 addresses to each subnet starting from the first available address.



a. Original block



b. Subblocks

Q. An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets as shown below:

❑ One subblock of 120 addresses.
❑ One subblock of 60 addresses.
❑ One subblock of 10 addresses.

Solution:

There are $2^{32-24} = 256$ addresses in this block.

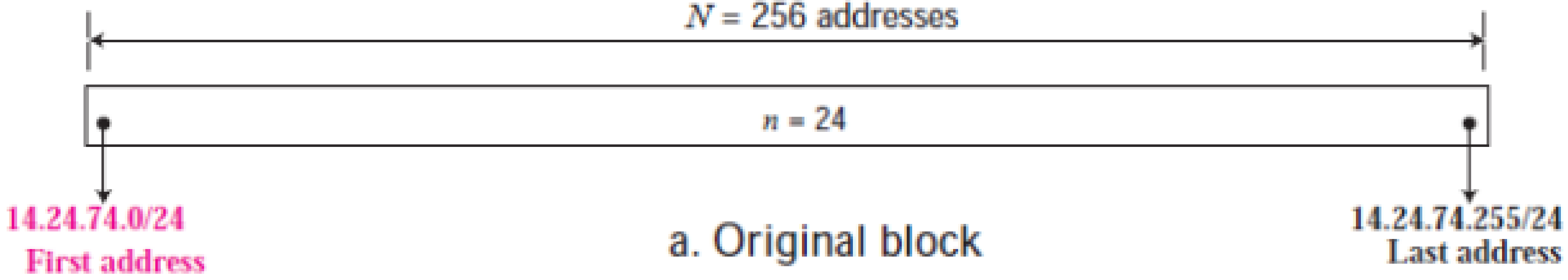
- Network Mask : 255.255.255.0
- First address = (any address) AND (network mask) = 14.24.74.0/24 AND 255.255.255.0 = 14.24.74.0/24
- Last address = (any address) OR [NOT (network mask)] =14.24.74.0/24 OR [NOT 255.255.255.0] = 14.24.74.0/24 OR 0.0.0.255 = 14.24.74.255/24

a. Given, One subblock of 120 addresses. The number of addresses in the first subblock is not a power of 2. We allocate 128 addresses. The first can be used as network address and the last as the special address. There are still 126 addresses available. The subnet mask for this subnet can be found as $n_1 = 24 + \log_2 (256/128) = 25$. The first address in this block is 14.24.74.0/25; the last address is 14.24.74.127/25.

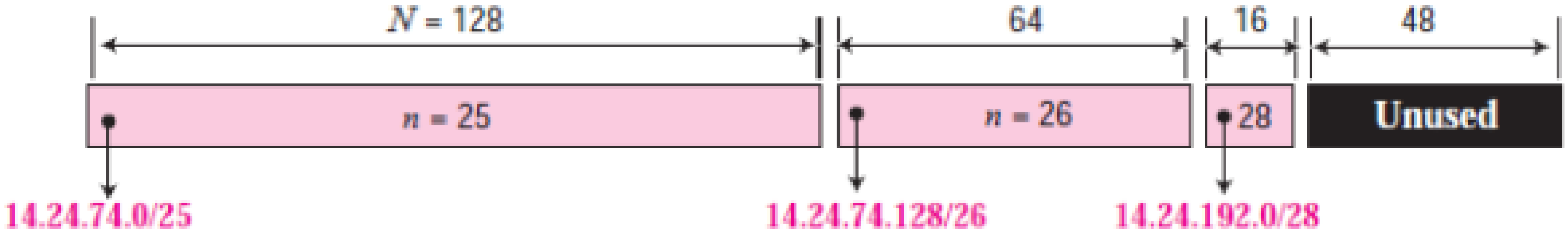
b. Given, One subblock of 60 addresses. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The first can be used as network address and the last as the special address. There are still 62 addresses available. The subnet mask for this subnet can be found as $n_1 = 24 + \log_2 (256/64) = 26$. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

c. Given, One subblock of 10 addresses. The number of addresses in the third subblock is not a power of 2 either. We allocate 16 addresses. The first can be used as network address and the last as the special address. There are still 14 addresses available. The subnet mask for this subnet can be found as $n_1 = 24 + \log_2 (256/16) = 28$. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.

d. If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.209. The last address is 14.24.74.255. We don't know about the prefix length yet.



a. Original block



b. Subblocks

IPv4 Datagram Header

- Datagram is a part of IPV4 Header. It represents the Header and payload size.
- Datagram of IPV4 is a 16 bit field which is equal to 65535 bytes.
- IPV4 Datagram is the **combination** of **header size** and **payload** (data). IPV4 Datagram will always be in the range of 20 bytes to 65535 bytes (2^{16} bytes).

1. Header Size

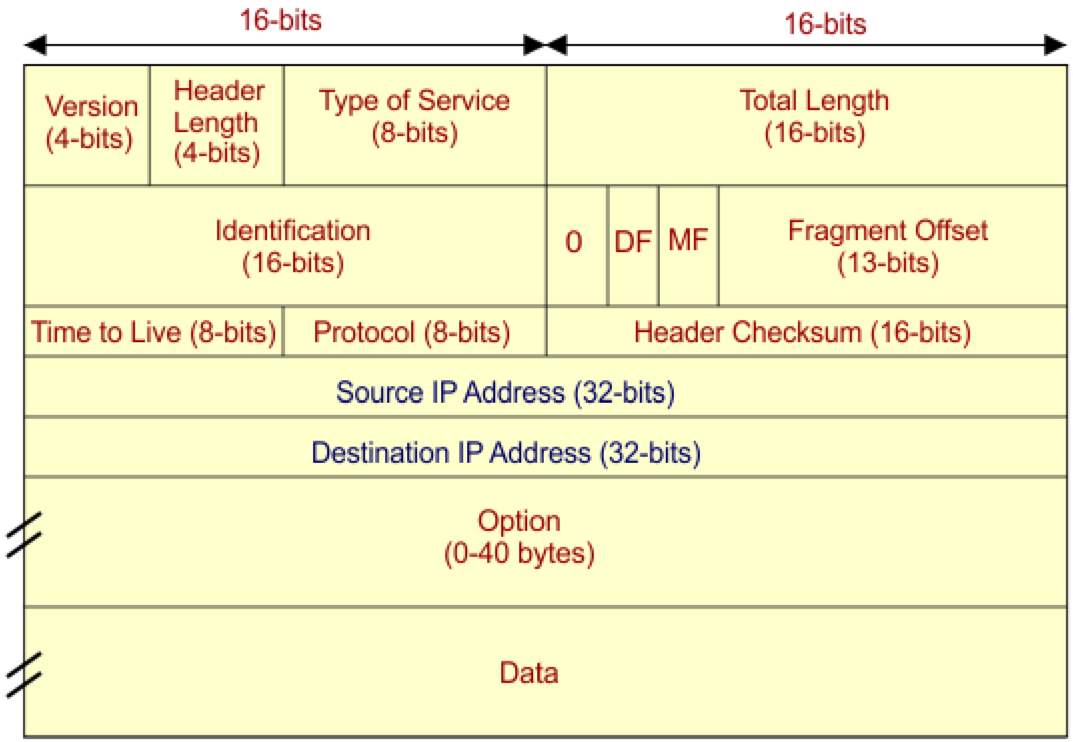
- The minimum size of header is 20 bytes or 160 bits and maximum of 60 bytes or 480 bits. First five rows of IPV4 header are mandatory, having total size of 20 bytes.

2. Payload

- The minimum size of Payload (Data) is 0 bytes and maximum of 65515 bytes.

IPv4 Datagram Header Size Calculations

- If header size is 20 bytes (minimum) and payload is 0 bytes (minimum) then datagram size = 20 bytes
- If header size is 20 bytes (minimum) and payload is 65515 (maximum) then datagram size = 65535 bytes
- If header size is 60bytes then the payload maximum value will be 65475. Because 65475 + 60 bytes are equal to maximum IPV4 datagram size which is equal to 65535.



IPV4 Datagram Header

IPV4 Datagram Header Attributes

1. ***VERSION:*** *Version of the IP protocol is of 4 bits. These 4 bits are always fixed as 0100 to represent 4 in decimal for IPv4.*
2. ***HLEN:*** *IPv4 header length is of 4 bits. The minimum value for this field is 5 and the maximum is 15 bytes.* Header length = Header length field value x 4 bytes E.g. If header length field contains decimal value 11 (represented in binary 1011) then Header length = 11 x 4 = 44 bytes.
3. **Type Of Service:** It is an 8-bit field that is used for Quality of Service. The division of 8-bits are explained under:

Precedence (3 bits)	Delay (1 bit)	Throughput (1 bit)	Reliability (1 bit)	Cost (1 bit)	Reserved (1 bit)

- **Precedence (3 bits):** First 3 bits define the precedence. Precedence means priority i.e., immediate, routine etc. If a router is congested and needs to discard a packets, it will discard packets having lowest priority first. Bits values will be 0 or 1.
- **Delay (1 bit):** if we want a minimum delay in data packets then this field will be 1 otherwise 0. It is mostly used in video calling where it needs no delay.
- **Throughput (1 bit):** If need is of high output, then its field bit will be 1 otherwise 0.
- **Reliability (1 bit):** If need is of high reliability, then its field bit will be 1 otherwise 0. It is used where no data loss is tolerated.
- **Cost (1 bit):** If need is of low cost, then this field bit will be 1 otherwise 0. It is required when we need to select the shortest path to reach destination.
- **Last bit** is reserved for future purpose which mostly controls the congestion Notification. Congestion Notification means it inform sender to minimize the speed of sending data.

4. **Total Length:** It is Total length of the datagram. It is a 16-bit field which can represent 2^{16} = 65536 value . It has minimum size of 20 bytes and max value of 65535 bytes.
5. **Identification:** It is a 16-bit field. It is helpful for the identification of the fragments of an original datagram. When an IP datagram is fragmented, each fragmented datagram is assigned the same identification header number. This header number is useful during the re-assembly of fragmented datagrams.
6. **Flag Bits:** It use 3 flag bits. First flag bit is Reserved. Second Flag bit (DF Bit). DF bit stands for Do Not Fragment bit. DF value may be 0 or 1. When DF value is 0 then It gives the permission to the intermediate devices (i.e., routers) to fragment the datagram if required. When DF value is 1 then It indicates the intermediate devices (i.e., router) not to fragment the datagram at any cost. Third flag bit is MF. MF bit stands for More Fragments bit. MF value may be 0 or 1. When MF bit value is 0 then it tells to the receiver that the current datagram-fragment is the last fragment and no more segment will appear of same datagram. When MF bit value is 1 then it tells more fragments are still to come after this fragment. MF bit is set to 1 for all the fragments except the last one.
7. **Fragment Offset:** Fragment Offset is a 13 bit field. It tells the position of a fragmented datagram in the original un-fragmented IP datagram.
8. **Time to live:** Time to live (TTL) is 8-bit field. It indicates the maximum number of hops a datagram can take to reach the destination. The main purpose of TTL is to prevent the IP datagrams from looping around forever in a routing loop. If the value of TTL becomes zero before reaching the destination, then datagram is discarded.
9. ***Protocol:*** it is **an 8-bit number that defines what protocol is used inside the IP packet**. TCP, UDP, ICMP or IGMP protocols can be filtered on, although they are most common. Protocol number of ICMP is 1 (in binary 00000001), IGMP is 2 (in binary 00000010), TCP is 6 and UDP is 17.
10. **Header Checksum**: It is a 16 bits field. It is used for checking errors in the datagram header.
11. **Source IP address:** 32 bits IP address of the sender.
12. **Destination IP address:** 32 bits IP address of the receiver.
13. **Option**: Due to options field, datagram-header-size can be of variable length (20 bytes to 60 bytes).

An IPv4 datagram has arrived with the following information in the header (in hexadecimal): Ox45 00 00 54 00 03 58 50 20 06 00 00 7C 4E 03 02 B4 OE OF 02

a. Is the packet corrupted? b. Are there any options? c. Is the packet fragmented? d. What is the size of the data?
e. How many more routers can the packet travel to? f. What is the identification number of the packet?
g. What is the type of service?

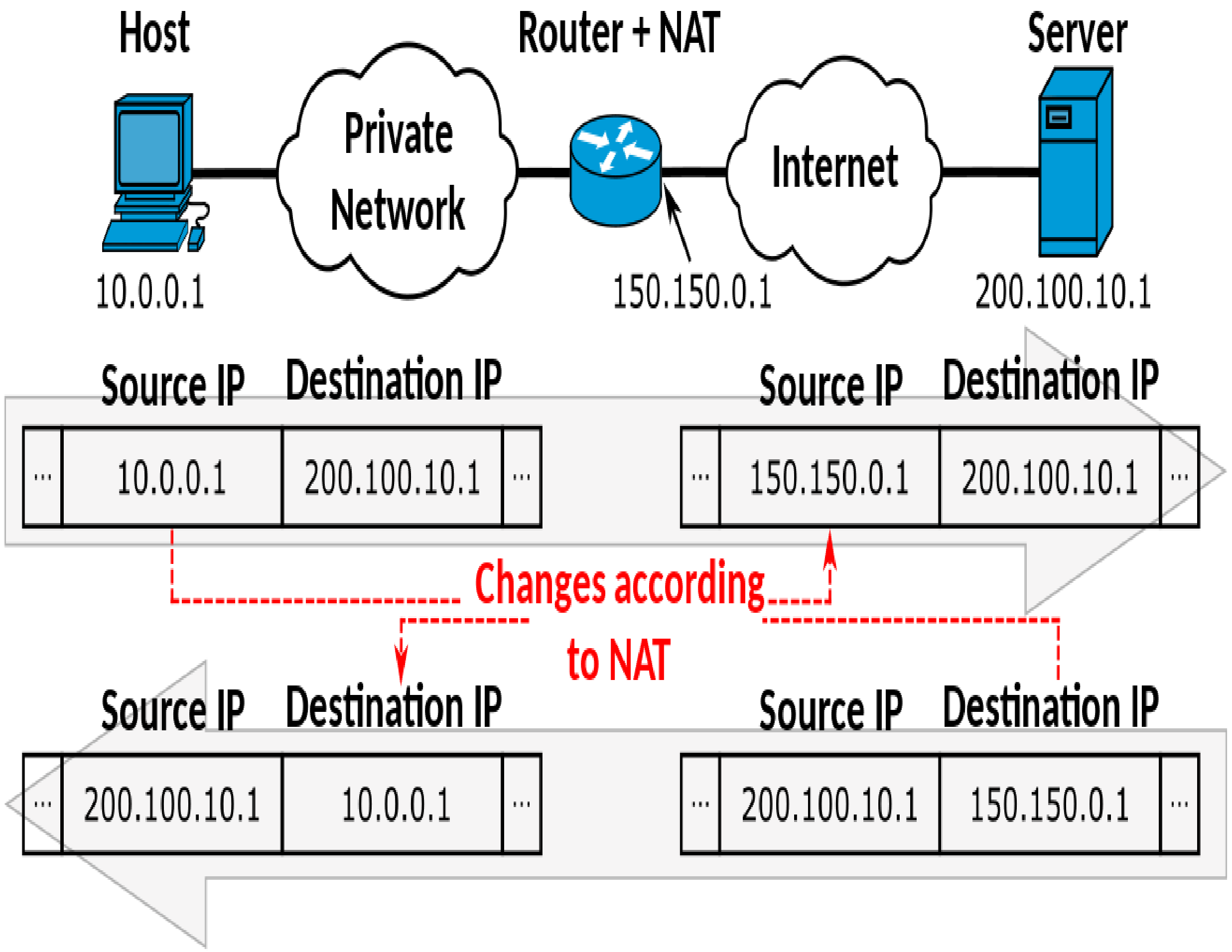
a. Is the packet corrupted? To check if the packet is corrupted, we need to calculate the header checksum. However, the checksum field (bytes 10-11) is given as 00 00. This could mean either the checksum was not computed (which is allowed) or it was computed and happened to be zero. Without additional information, we can't definitively say if the packet is corrupted.
b. Are there any options? No, there are no options. The IHL (Internet Header Length) is 5 (from the first nibble of the first byte, 0x4'5'), which means the header is 20 bytes long (5 * 4 bytes). If there were options, the IHL would be greater than 5.
c. Is the packet fragmented? No, the packet is not fragmented. The flags are in the 7th byte (58), which in binary is 0101 1000. The "More Fragments" flag (second bit from right) is 0, and the fragment offset (last 13 bits of bytes 6-7) is also 0.
d. What is the size of the data? The total length of the packet is given in bytes 2-3 (00 54), which is 84 bytes. Subtracting the header length (20 bytes) gives us the data size: 84 - 20 = 64 bytes.
e. How many more routers can the packet travel to? The TTL (Time to Live) is given in the 8th byte (20), which is 32 in decimal. This means the packet can travel through up to 32 more routers.
f. What is the identification number of the packet? The identification number is in bytes 4-5 (00 03), which is 3 in decimal.
g. What is the type of service? The Type of Service is given in the second byte (00). This indicates normal service with default values for delay, throughput, and reliability.

Network Address Translation (NAT)

- To access the Internet, one public IP address is needed, but we can use a private IP address in our private network.
- The idea of NAT is to allow multiple devices to access the Internet through a single public address.
- To achieve this, the translation of a private IP address to a public IP address is required.
- **Network Address Translation (NAT)** is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts.
- Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination.
- It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on a router or firewall.

Simplified overview of how NAT works

- A device on the private network sends a request to access the internet.
- The router on the private network receives the request and replaces the private IP address of the device with its own public IP address.
- The router then sends the request to the internet using its public IP address.
- When the response is received from the internet, the router uses its translation table to map the public IP address back to the private IP address of the original device and forwards the response to that device.



Example of NAT

- Let’s say that there is a laptop connected to a home network using NAT.
- That network eventually connects to a router that addresses the internet.
- Suppose that someone uses that laptop to search for directions to their favorite restaurant.
- The laptop is using NAT. So, it sends this request in an IP packet to the router, which passes that request along to the internet and the search service you’re using.
- But before your request leaves your home network, the router first changes the internal IP address from a private local IP address to a public IP address.
- Your router effectively translates the private address you’re using to one that can be used on the internet, and then back again.
- Now you know that your humble little cable modem or DSL router has a little, automated translator working inside of it.
- If the packet keeps a private address, the receiving server won’t know where to send the information back to. This is because a private IP address cannot be routed onto the internet.
- If your router were to try doing this, all internet routers are programmed to automatically drop private IP addresses.
- The nice thing is, though, that all routers sold today for home offices and small offices can readily translate back and forth between private IP addresses and publicly-routed IP addresses.

Types of NAT

- Static NAT: Here, a specific private IP address is always mapped to a specific public IP address.
- Dynamic NAT: Here, a pool of public IP addresses are used to map to private IP addresses on an as-needed basis.
- Port Address Translation (PAT): Here, a single public IP address is used to map multiple private IP addresses by using different ports to identify each device.

Advantages of NAT

- NAT conserves legally registered IP addresses.
- It provides privacy as the device’s IP address, sending and receiving the traffic, will be hidden.
- Eliminates address renumbering when a network evolves.

Disadvantage of NAT

- Translation results in switching path delays.
- Certain applications like online gaming, and VOIP (Skype) will not function while NAT is enabled.
- Complicates tunneling protocols such as IPsec.
- Also, the router being a network layer device, should not tamper with port numbers(transport layer) but it has to do so because of NAT.

Routing Algorithms

- One of the functions of the network layer is to route the packets from the source machine to the destination machine.
- The major area of the network layer design includes the algorithms which choose the routes and the data structures which are used.
- **Routing algorithm** is a part of network layer software which is responsible for deciding the route/path over which a packet is to be sent.

- **Types of Routing Algorithms:**
- **Static Routing or Non-Adaptive Routing**

(i) It follows user defined routing and routing table is not changed until network administrator changes it.

(ii) Static Routing uses simple routing algorithms and provides more security than dynamic routing.

- **Dynamic Routing or Adaptive Routing**

(i) As the name suggests, dynamic routing changes the routing table once any changes to network occurs or network topology changes.

(ii) During network change, dynamic routing sends a signal to router, recalculates the routes and send the updated routing information.

Static Routing Vs Dynamic Routing

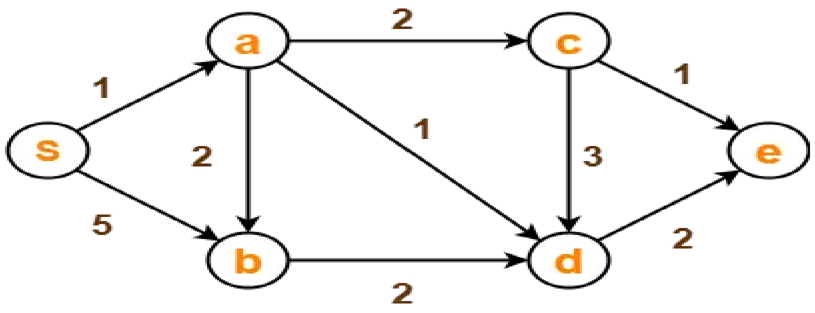
Sr. No.	Key	Static Routing	Dynamic Routing
1.	**Routing pattern**	In static routing, user defined routes are used in routing table.	In dynamic routing, routes are updated as per the changes in network.
2.	**Routing Algorithm**	No complex algorithm used to figure out shortest path.	Dynamic routing employs complex algorithms to find the shortest routes.
3.	**Security**	Static routing provides higher security.	Dynamic routing is less secure.
4.	**Automation**	Static routing is a manual process.	Dynamic routing is an automatic process.
5.	**Applicability**	Static routing is used in smaller networks.	Dynamic routing is implemented in large networks.
6.	**Protocols**	Static routing may not follow any specific protocol.	Dynamic routing follows protocols like BGP, RIP and EIGRP.
7.	**Additional Resources**	Static routing does not require any additional resources.	Dynamic routing requires additional resources like memory, bandwidth etc.

Routing Algorithms
Dijkstra’s Shortest Path Algorithm

- Dijkstra's shortest path algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge weights, producing the shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.
- The algorithm works by maintaining a set of vertices whose shortest distance from the source vertex is known, and gradually expanding this set until all vertices have been included.
- At each step, the algorithm chooses the vertex with the smallest known distance from the source, adds it to the set of known vertices, and updates the distances to its neighbors. This process continues until the destination vertex is reached or all vertices have been visited.
- To implement Dijkstra's algorithm, one can use a priority queue to keep track of the vertices with the smallest known distance, with the distance to each vertex being the priority.
- The algorithm begins by initializing the distance to the source vertex to 0 and the distance to all other vertices to infinity. It then inserts the source vertex into the priority queue.
- The algorithm then repeatedly extracts the vertex with the smallest distance from the priority queue and updates the distances to its neighbors.
- If the new distance is smaller than the previous distance, the neighbor's distance is updated and it is inserted into the priority queue. This process continues until the destination vertex is extracted from the priority queue or the priority queue is empty.
- Once the algorithm has finished, the shortest path from the source vertex to any other vertex can be found by following the edges of the shortest path tree from the source to the destination vertex.

Routing Algorithms
Dijkstra’s Shortest Path Algorithm

Using Dijkstra’s Algorithm, find the shortest distance from source vertex ‘S’ to remaining vertices in the following graph



Step-01:

The following two sets are created-

- Unvisited set : {S , a , b , c , d , e}
- Visited set : { }

Step-02:

The two variables Π and d are created for each vertex and initialized as-

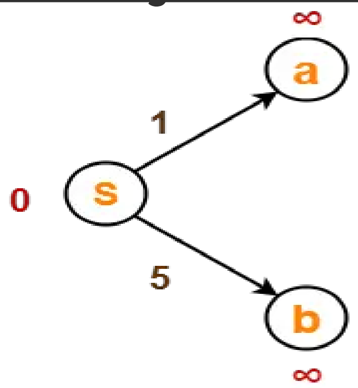
- Π[S] = Π[a] = Π[b] = Π[c] = Π[d] = Π[e] = NIL
- d[S] = 0
- d[a] = d[b] = d[c] = d[d] = d[e] = ∞

Step-03:

Vertex ‘S’ is chosen.

- This is because shortest path estimate for vertex ‘S’ is least.
- The outgoing edges of vertex ‘S’ are relaxed.

Before Edge Relaxation-



Now,

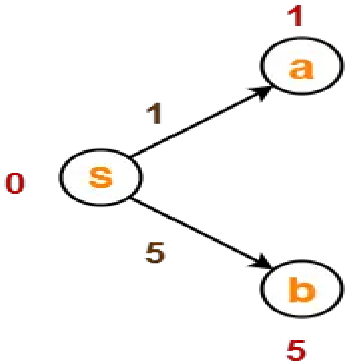
- d[S] + 1 = 0 + 1 = 1 < ∞

∴ d[a] = 1 and Π[a] = S

- d[S] + 5 = 0 + 5 = 5 < ∞

∴ d[b] = 5 and Π[b] = S

After edge relaxation, our shortest path tree is-



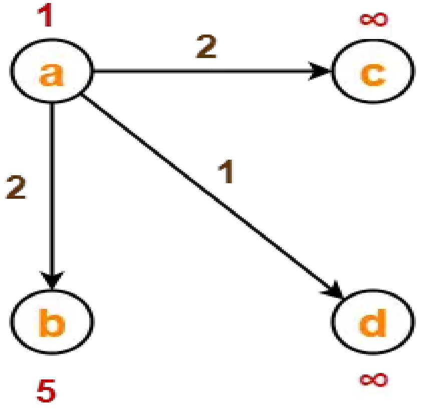
Now, the sets are updated as-

- Unvisited set : {a , b , c , d , e}
- Visited set : {S}

Step-04:

- Vertex ‘a’ is chosen.
- This is because shortest path estimate for vertex ‘a’ is least.
- The outgoing edges of vertex ‘a’ are relaxed.

Before Edge Relaxation-



Now,

- $d[a] + 2 = 1 + 2 = 3 < \infty$

$\therefore$ $d[c] = 3$ and $\Pi[c] = a$

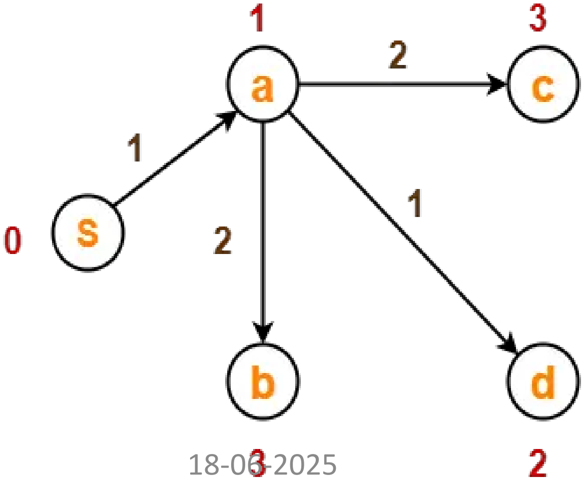
- $d[a] + 1 = 1 + 1 = 2 < \infty$

$\therefore$ $d[d] = 2$ and $\Pi[d] = a$

- $d[b] + 2 = 1 + 2 = 3 < 5$

$\therefore$ $d[b] = 3$ and $\Pi[b] = a$

After edge relaxation, our shortest path tree is-



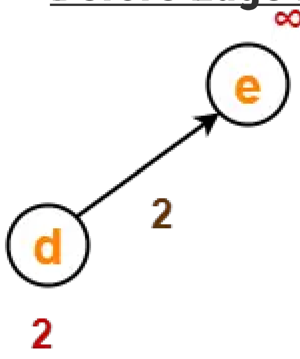
Now, the sets are updated as-

- Unvisited set : {b , c , d , e}
- Visited set : {S , a}

Step-05:

- Vertex ‘d’ is chosen.
- This is because shortest path estimate for vertex ‘d’ is least.
- The outgoing edges of vertex ‘d’ are relaxed.

Before Edge Relaxation-

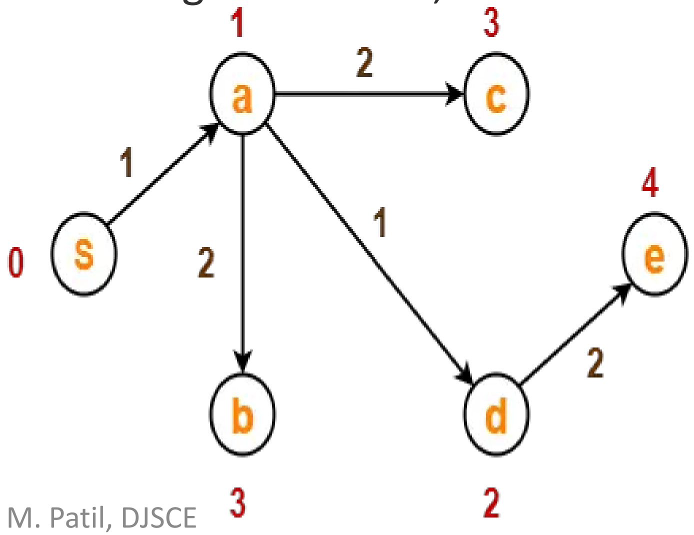


Now,

- $d[d] + 2 = 2 + 2 = 4 < \infty$

$\therefore$ $d[e] = 4$ and $\Pi[e] = d$

After edge relaxation, our shortest path tree is-



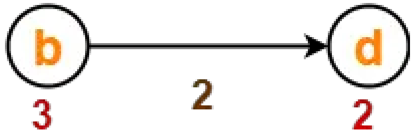
Now, the sets are updated as-

- Unvisited set : {b , c , e}
- Visited set : {S , a , d}

Step-06:

- Vertex ‘b’ is chosen.
- This is because shortest path estimate for vertex ‘b’ is least.
- Vertex ‘c’ may also be chosen since for both the vertices, shortest path estimate is least.
- The outgoing edges of vertex ‘b’ are relaxed.

Before Edge Relaxation-



Now,

- d[b] + 2 = 3 + 2 = 5 > 2

∴ No change

After edge relaxation, our shortest path tree remains the same as in Step-05.

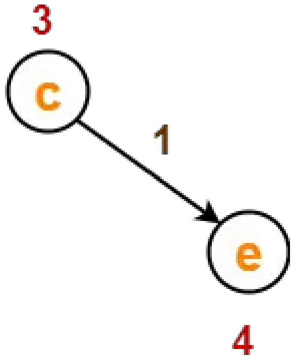
Now, the sets are updated as-

- Unvisited set : {c , e}
- Visited set   : {S , a , d , b}

Step-07:

- Vertex ‘c’ is chosen.
- This is because shortest path estimate for vertex ‘c’ is least.
- The outgoing edges of vertex ‘c’ are relaxed.

Before Edge Relaxation-



Now,

- d[c] + 1 = 3 + 1 = 4 = 4

∴ No change

After edge relaxation, our shortest path tree remains the same as in Step-05.

Now, the sets are updated as-

- Unvisited set : {e}
- Visited set : {S , a , d , b , c}

Step-08:

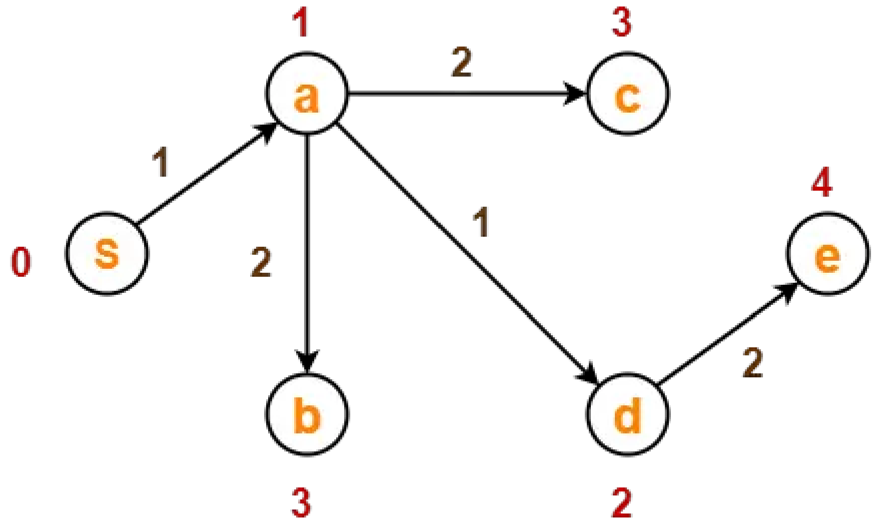
- Vertex ‘e’ is chosen.
- This is because shortest path estimate for vertex ‘e’ is least.
- The outgoing edges of vertex ‘e’ are relaxed.
- There are no outgoing edges for vertex ‘e’.
- So, our shortest path tree remains the same as in Step-05.

Now, the sets are updated as-

- Unvisited set : { }
- Visited set : {S , a , d , b , c , e}

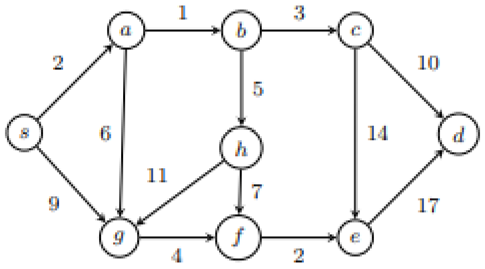
Now,

- All vertices of the graph are processed.
- Our final shortest path tree is as shown below.
- It represents the shortest path from source vertex ‘S’ to all other remaining vertices.

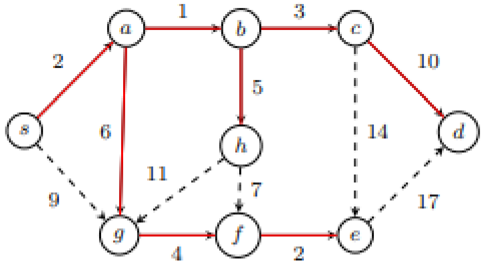


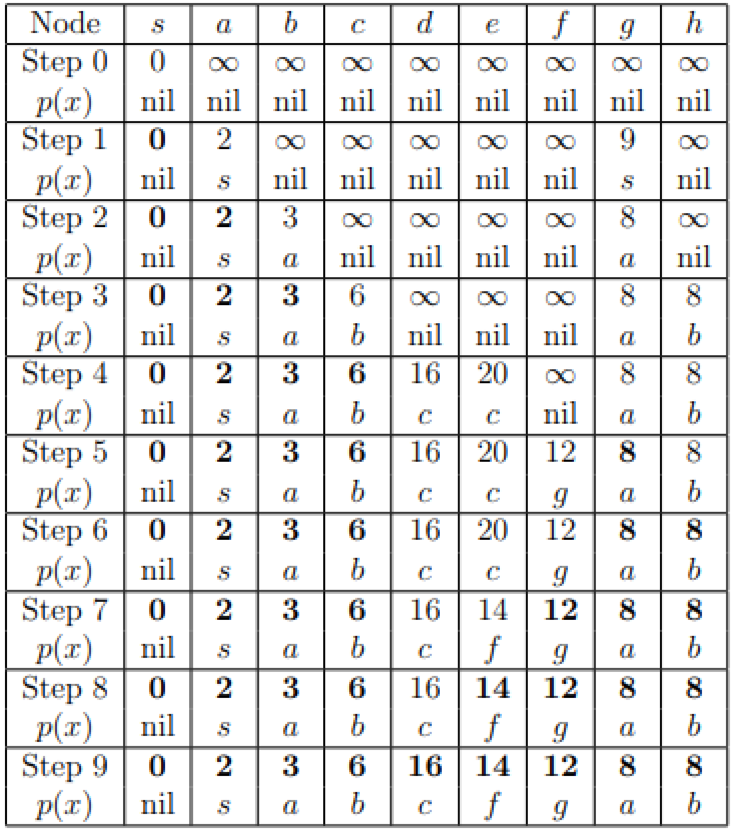
Shortest Path Tree

The order in which all the vertices are processed is :
S , a , d , b , c , e.

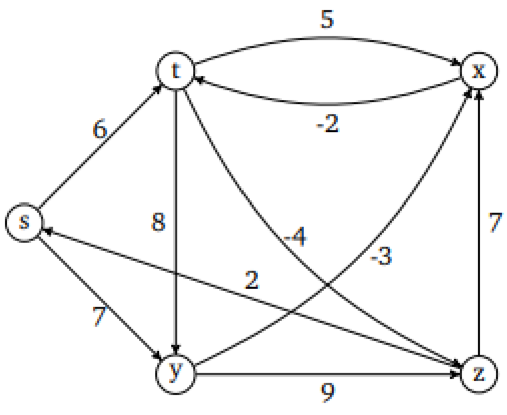


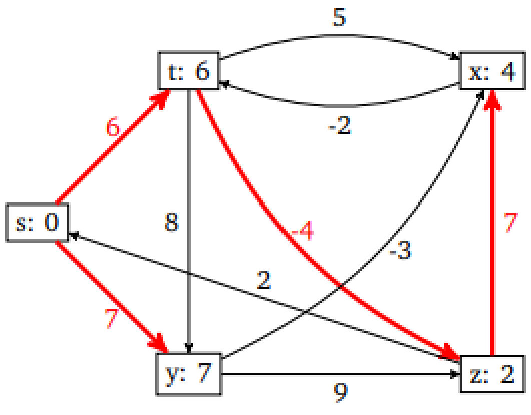
The shortest paths are resulting in a tree, highlighted in red.

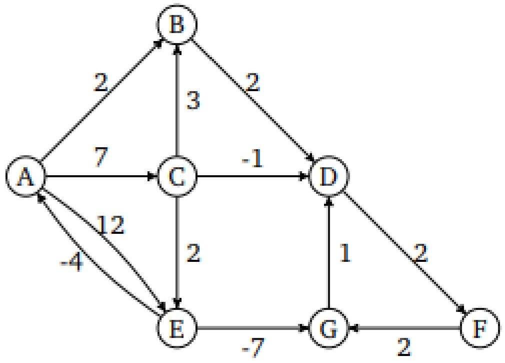


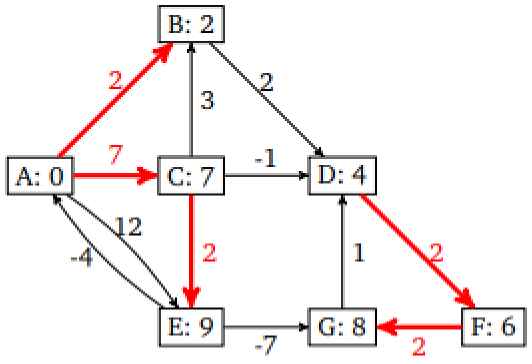


Node	s	a	b	c	d	e	f	g	h
Step 0	0	∞	∞	∞	∞	∞	∞	∞	∞
$p(x)$	nil	nil	nil	nil	nil	nil	nil	nil	nil
Step 1	**0**	2	∞	∞	∞	∞	∞	9	∞
$p(x)$	nil	s	nil	nil	nil	nil	nil	s	nil
Step 2	**0**	**2**	3	∞	∞	∞	∞	8	∞
$p(x)$	nil	s	a	nil	nil	nil	nil	a	nil
Step 3	**0**	**2**	**3**	6	∞	∞	∞	8	8
$p(x)$	nil	s	a	b	nil	nil	nil	a	b
Step 4	**0**	**2**	**3**	**6**	16	20	∞	8	8
$p(x)$	nil	s	a	b	c	c	nil	a	b
Step 5	**0**	**2**	**3**	**6**	16	20	12	**8**	8
$p(x)$	nil	s	a	b	c	c	g	a	b
Step 6	**0**	**2**	**3**	**6**	16	20	12	**8**	**8**
$p(x)$	nil	s	a	b	c	c	g	a	b
Step 7	**0**	**2**	**3**	**6**	16	14	**12**	**8**	**8**
$p(x)$	nil	s	a	b	c	f	g	a	b
Step 8	**0**	**2**	**3**	**6**	16	**14**	**12**	**8**	**8**
$p(x)$	nil	s	a	b	c	f	g	a	b
Step 9	**0**	**2**	**3**	**6**	**16**	**14**	**12**	**8**	**8**
$p(x)$	nil	s	a	b	c	f	g	a	b





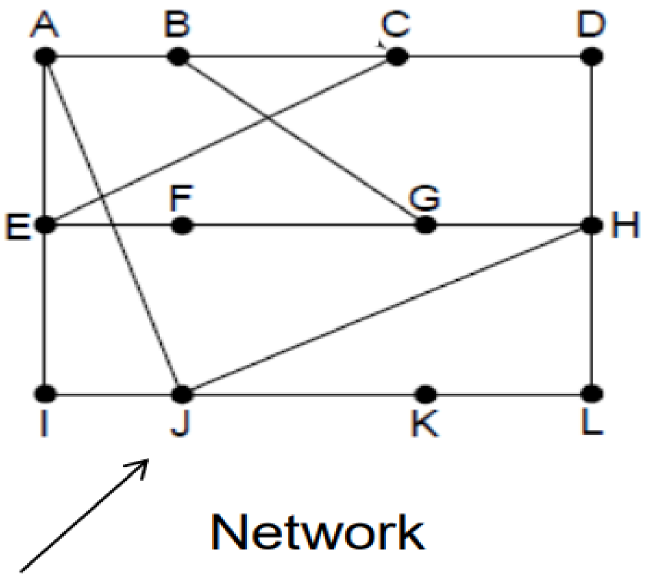




Distance Vector Routing

- Distance vector routing is a routing algorithm used in computer networks to determine the best path for routing data packets between nodes.
- It works by each node maintaining a table that lists the cost of sending a packet to all other nodes in the network and then using this information to determine the best path for each packet.
- The basic idea behind distance vector routing is that each node sends its routing table to its neighbors at regular intervals, and each node updates its own routing table based on the information it receives from its neighbors. This process continues until all nodes have a consistent view of the network topology and the best path for each destination.
- The cost of sending a packet from one node to another can be measured in terms of various metrics, such as the number of hops, the available bandwidth, or the delay.
- The most commonly used metric is the number of hops, which represents the number of intermediate nodes that the packet must traverse to reach its destination.
- One of the main advantages of distance vector routing is its simplicity and low overhead. However, it can also suffer from several drawbacks, such as slow convergence, routing loops, and count-to-infinity problems, which can lead to suboptimal routing and network congestion.
- A broken link between the routers should be updated to every other router in the network immediately. The distance vector routing takes a considerable time for the updating. This problem is also known as **count-to-infinity.**
- To address these issues, more advanced routing protocols, such as Link State Routing (LSR) and Border Gateway Protocol (BGP), have been developed.

Distance Vector Routing

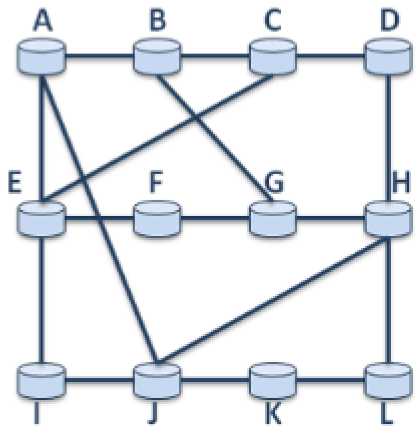


Network

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6		

New vector for J

Vectors received at J from Neighbors A, I, H and K



Vectors received from Routers A, I, H, and K to Router-J

	A	I	H	K	J	
A	0	22	18	19	A	6
B	10	34	29	26	A	18
C	23	16	16	34	I	26
D	38	25	6	22	H	18
E	12	5	28	20	I	15
F	21	18	17	38	I	28
G	16	29	4	29	H	16
H	15	18	0	17	H	10
I	19	0	12	20	I	8
J	7	9	5	8	--	0
K	22	20	20	0	K	4
L	27	31	7	7	K	13
	JA	JI	JH	JK	New routing table for J	
Delay:	6	8	10	4		

Newly estimated delay from Router-J

Link State Routing

- Link-state routing protocol is a type of routing protocol used in computer networks to determine the shortest path for data transmission between two nodes.
- In this protocol, every node in the network maintains a complete map of the network topology, including all the links and their status.
- The nodes then use this information to calculate the shortest path to any other node in the network.
- Link-state routing protocols have the following characteristics:

1. Each node maintains a complete map of the network, which is updated periodically or when a change in the network topology occurs.
2. The shortest path is calculated using Dijkstra's algorithm or a similar algorithm, which takes into account the link cost and the status of each link.
3. Each node communicates with its neighboring nodes to exchange information about the network topology and to update its own map.
4. When a change in the network topology occurs, the affected nodes send out update messages to inform the rest of the network.
5. Link-state routing protocols are more efficient than distance-vector protocols in large networks because they converge faster and have less routing overhead.

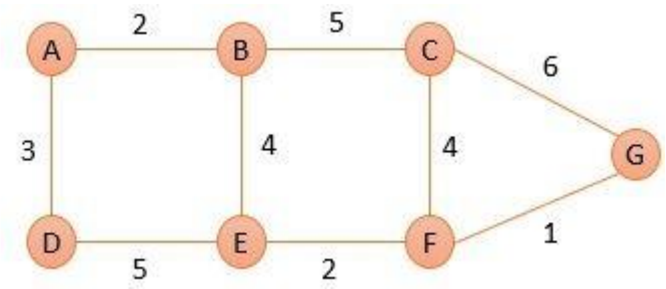
- Examples of link-state routing protocols include OSPF (Open Shortest Path First) and IS-IS (Intermediate System-to-Intermediate System). These protocols are commonly used in enterprise networks and the internet backbone.

Phases in Link State Routing

- The LSR process can be divided into several phases:

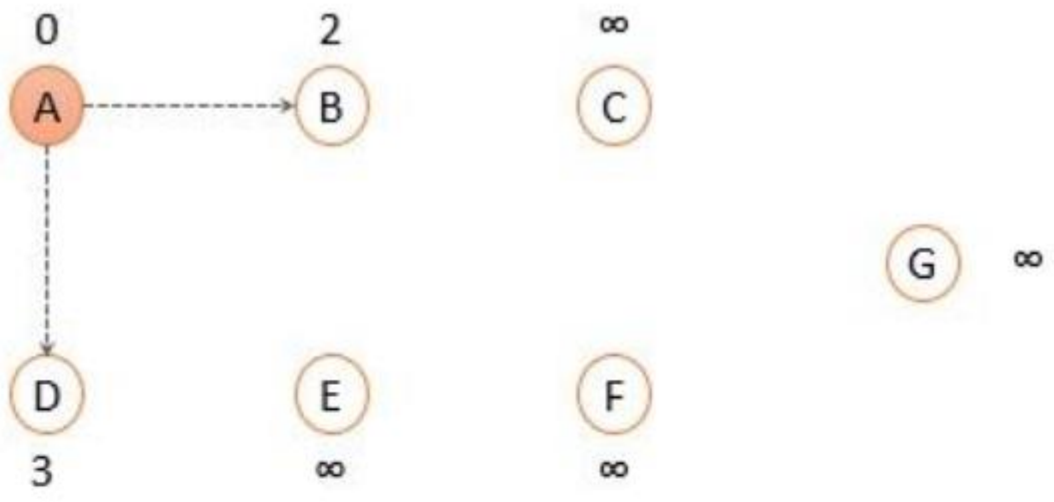
1. **Initialization phase:** The first phase is the initialization phase, where each router in the network learns about its own directly connected links. This information is then stored in the router's link state database.
2. **Flooding phase:** The second phase is the flooding phase, where each router floods its link state information to all other routers in the network. This allows each router to learn about the entire network topology.
3. **Path calculation phase:** The third phase is the shortest path calculation phase, where each router uses the link state information to calculate the shortest path to every other router in the network. This is typically done using Dijkstra's algorithm.
4. **Route installation phase:** The fourth and final phase is the route installation phase, where each router installs the calculated shortest paths in its routing table. This allows the router to forward packets along the optimal path to their destination.

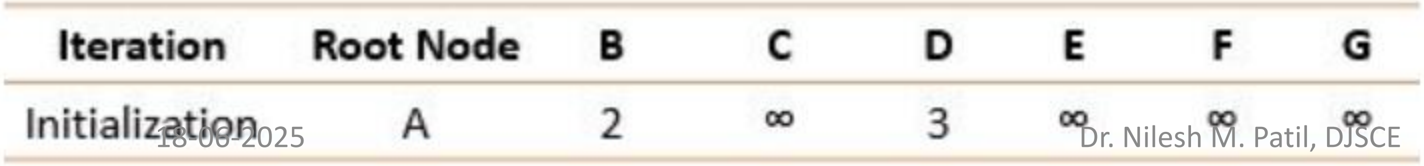
Using Link State Routing, find the shortest path tree for node A.



Solution:

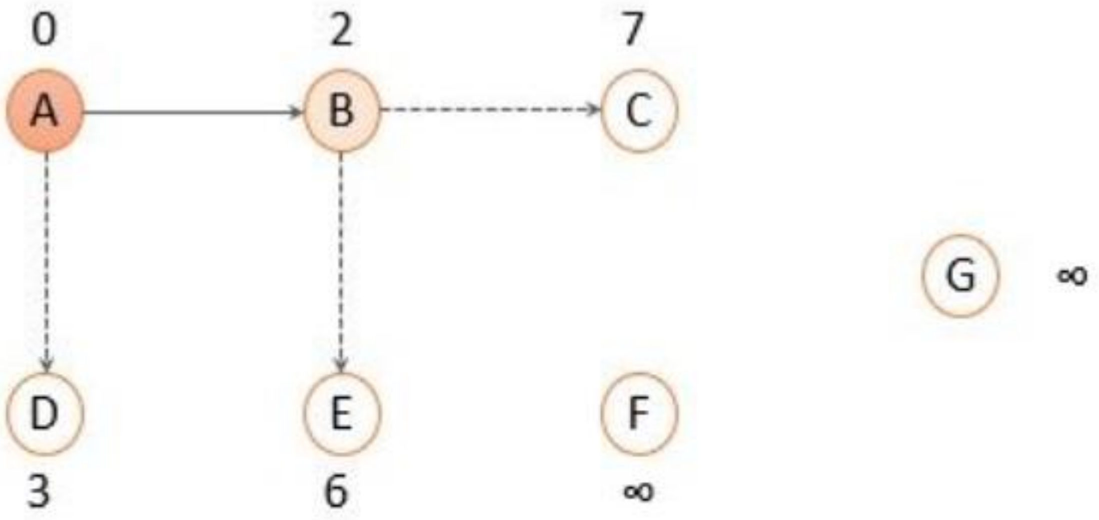
First, we will initialize node A and set the shortest distance to its immediate neighbors.

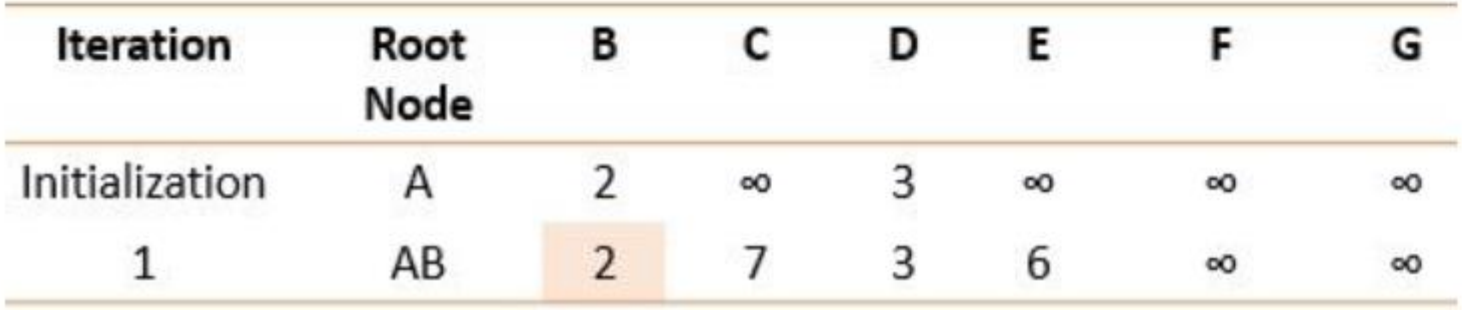




Iteration	Root Node	B	C	D	E	F	G
Initialization	A	2	∞	3	∞	∞	∞

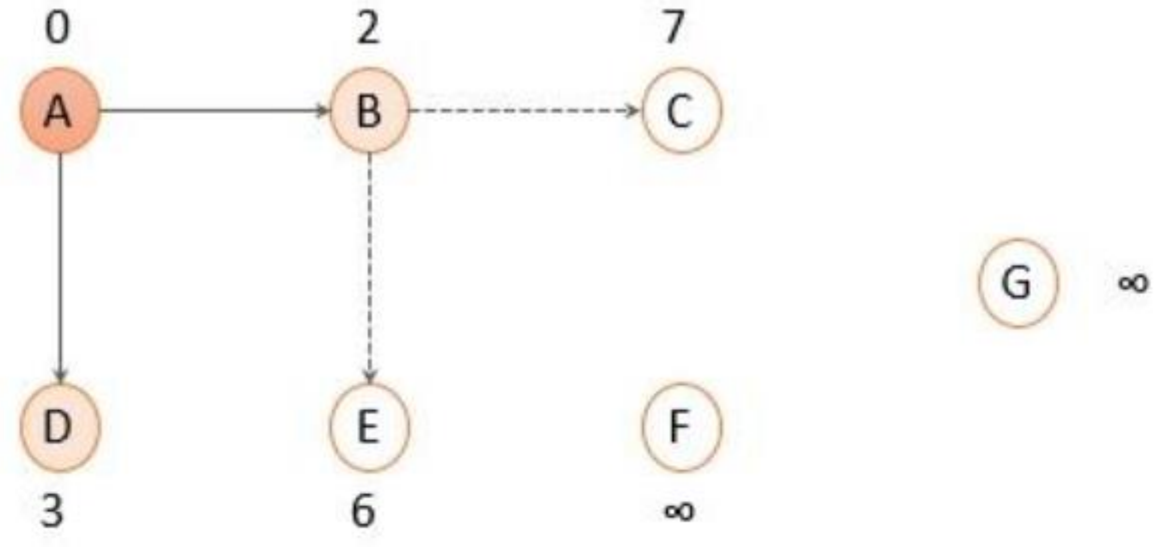
Among the two immediate neighbors of node A, we will add a node to the path that has the shortest distance i.e. node B with the shortest distance 2.

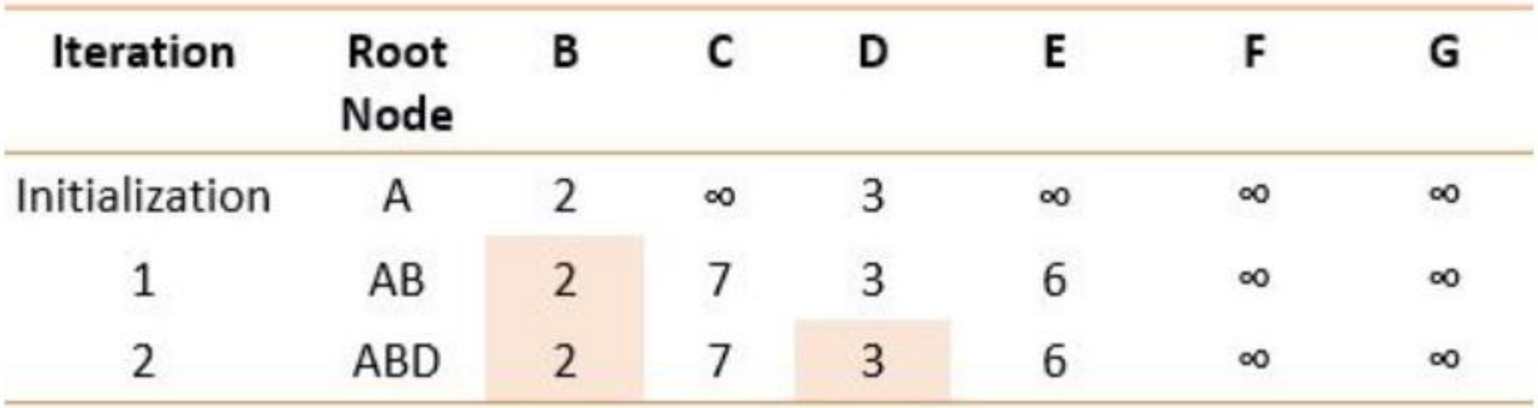




Iteration	Root Node	B	C	D	E	F	G
Initialization	A	2	∞	3	∞	∞	∞
1	AB	2	7	3	6	∞	∞

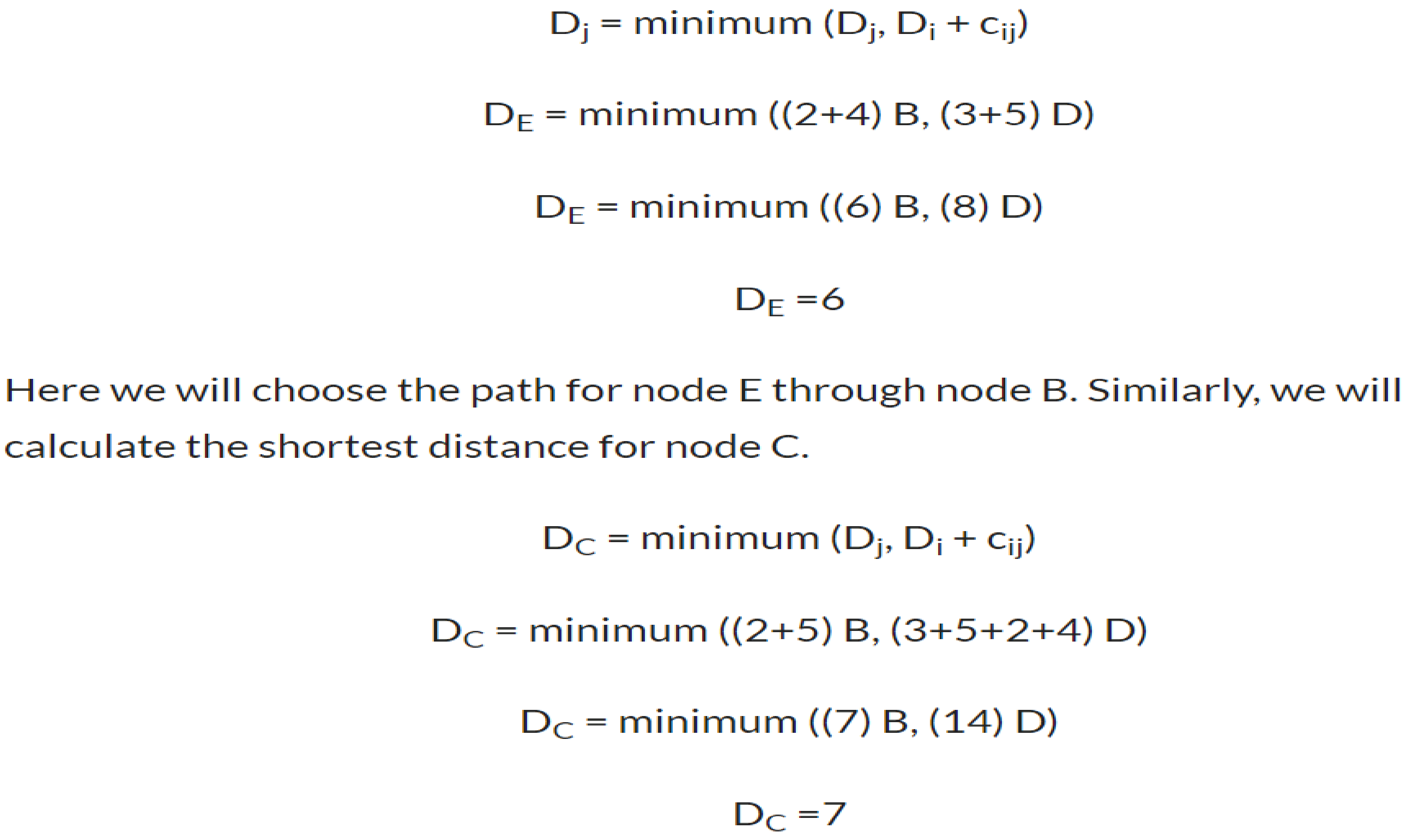
Now search for the nodes that are not added to the path yet (C, D, E) and select the one with the shortest distance i.e. node D.





Iteration	Root Node	B	C	D	E	F	G
Initialization	A	2	∞	3	∞	∞	∞
1	AB	2	7	3	6	∞	∞
2	ABD	2	7	3	6	∞	∞

Now, nodes B and D that we added to the path were directly connected to root node A and were at the shortest distance as compared to any other path. Further, we have node E and node C which can be reached by node A through node B or through node D. Let us check out which path is shortest.



$$D_j = \text{minimum } (D_j, D_i + c_{ij})$$

$$D_E = \text{minimum } ((2+4)\ B, (3+5)\ D)$$

$$D_E = \text{minimum } ((6)\ B, (8)\ D)$$

$$D_E = 6$$

Here we will choose the path for node E through node B. Similarly, we will calculate the shortest distance for node C.

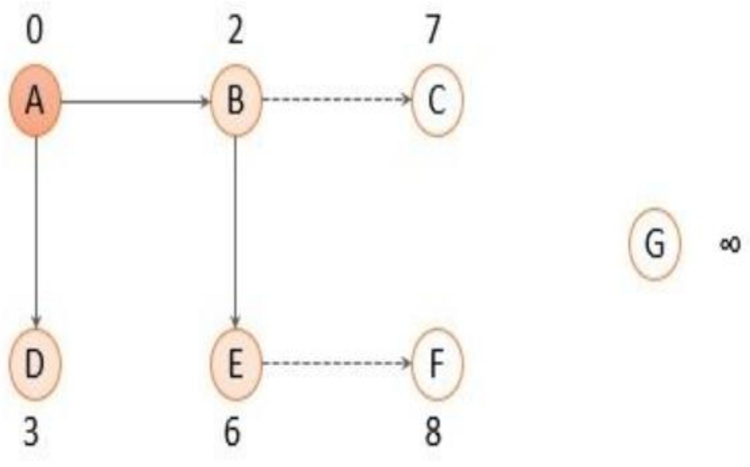
$$D_C = \text{minimum } (D_j, D_i + c_{ij})$$

$$D_C = \text{minimum } ((2+5)\ B, (3+5+2+4)\ D)$$

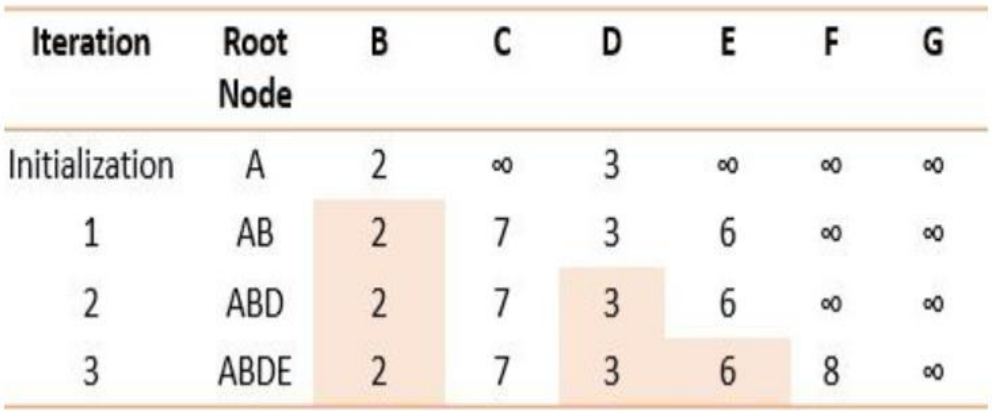
$$D_C = \text{minimum } ((7)\ B, (14)\ D)$$

$$D_C = 7$$

Now, node E has the shortest distance so we will add node E to the path.

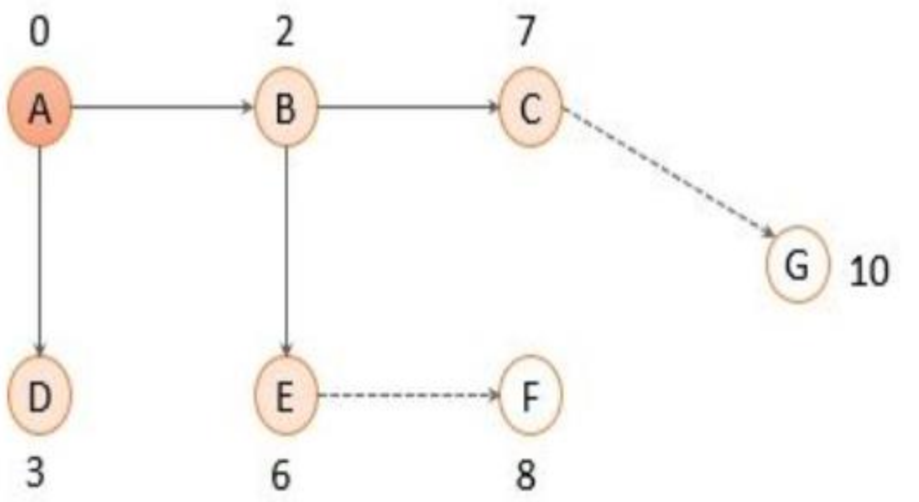


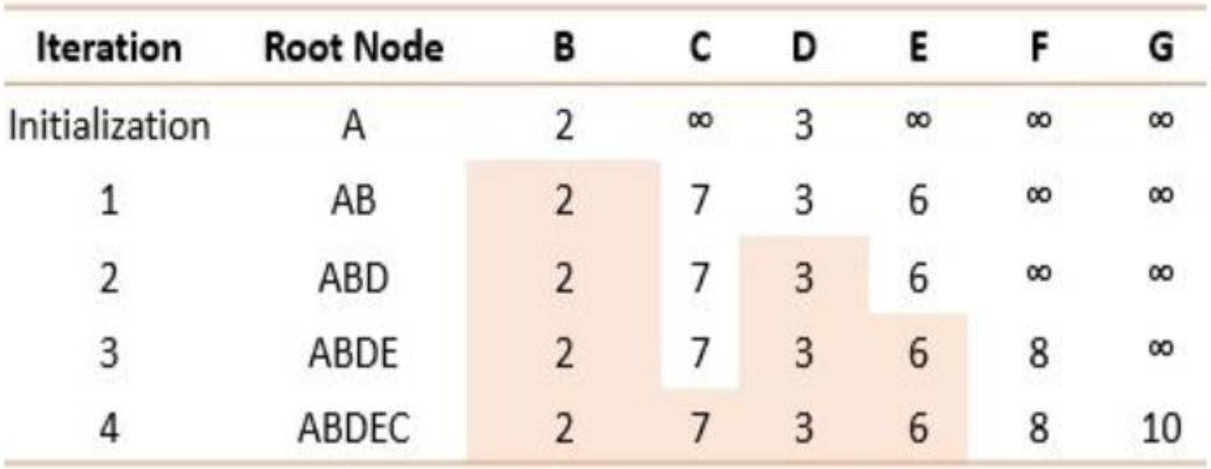
Iteration 3



Iteration	Root Node	B	C	D	E	F	G
Initialization	A	2	∞	3	∞	∞	∞
1	AB	2	7	3	6	∞	∞
2	ABD	2	7	3	6	∞	∞
3	ABDE	2	7	3	6	8	∞

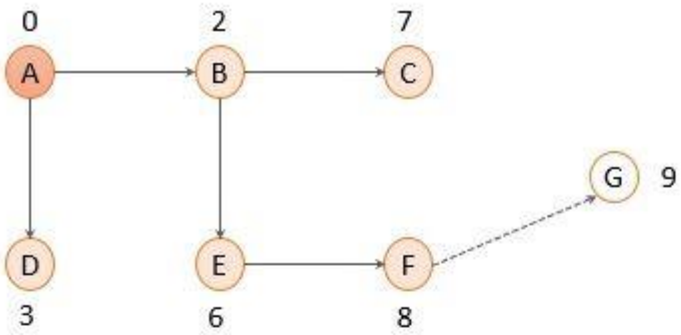
After adding node E to the path again explore the nodes that are not added to the path and there enlist their cost as the distance. Again, select the node with the shortest distance so we will add node C to the path.

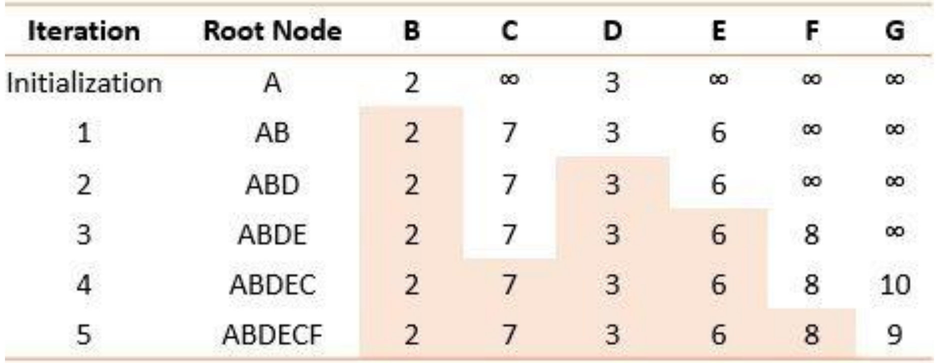




Iteration	Root Node	B	C	D	E	F	G
Initialization	A	2	∞	3	∞	∞	∞
1	AB	2	7	3	6	∞	∞
2	ABD	2	7	3	6	∞	∞
3	ABDE	2	7	3	6	8	∞
4	ABDEC	2	7	3	6	8	10

Now we are left with node F and node G we will calculate the shortest distance for node F and G as we calculate for node E and C. Now, among F and G the node with the shortest distance is node F so, we will update and add node F to the path.

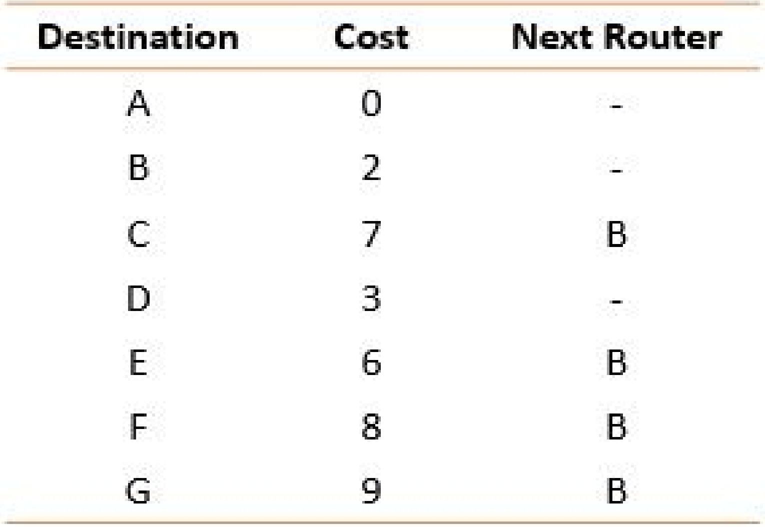




Iteration	Root Node	B	C	D	E	F	G
Initialization	A	2	∞	3	∞	∞	∞
1	AB	2	7	3	6	∞	∞
2	ABD	2	7	3	6	∞	∞
3	ABDE	2	7	3	6	8	∞
4	ABDEC	2	7	3	6	8	10
5	ABDECF	2	7	3	6	8	9

Also, the root node A can reach node G through node F with the shortest distance i.e.9. So finally we add node G.

Using the shortest path tree, each router creates a routing table that has three columns destination, cost, and next router. In the figure below, I have given a routing table for router A. Similarly, the routing table is created for all the routers in the domain.



Destination	Cost	Next Router
A	0	-
B	2	-
C	7	B
D	3	-
E	6	B
F	8	B
G	9	B

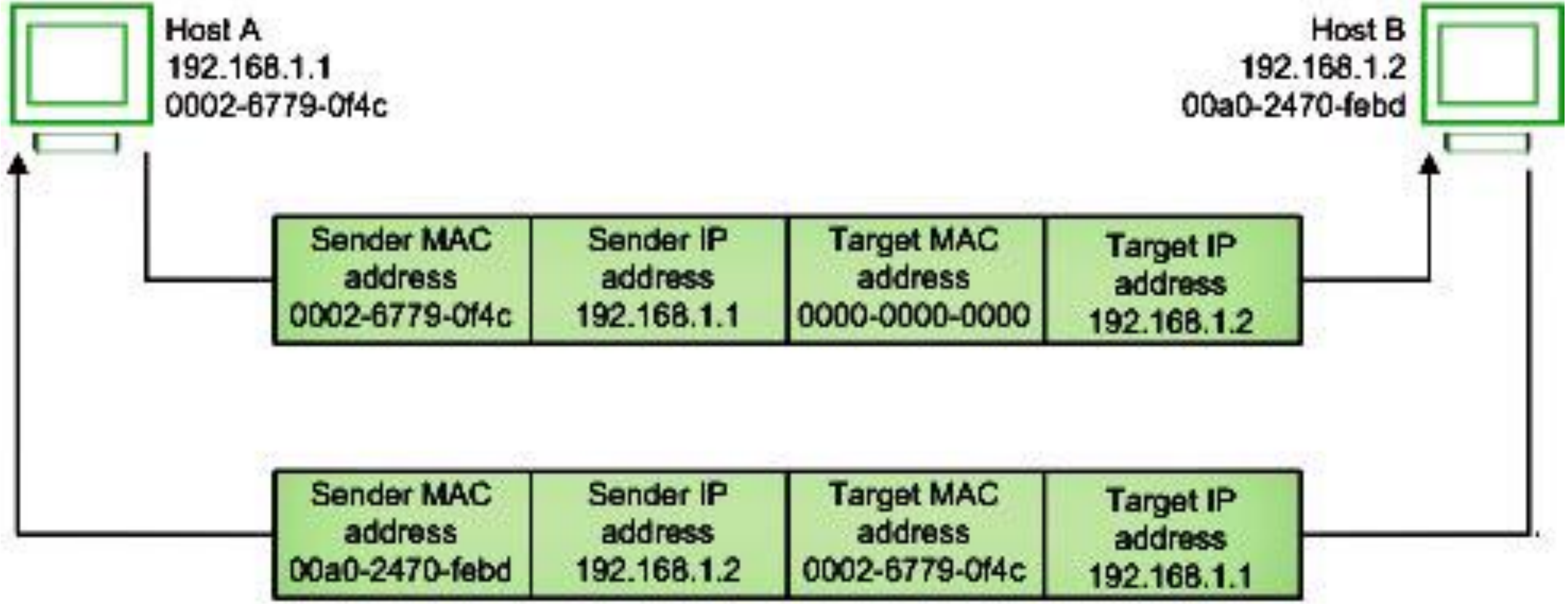
Routing Table for Node A

Distance Vector Routing Vs Link State Routing

Distance Vector Routing	Link State Routing
Bellman ford algorithm is used in distance vector routing	Dijkstra’s algorithm is used in link-state routing
It is simple to use	It needs trained network administrators
Chances of traffic are less	There is more chance of traffic
Convergence time is moderate i.e good news forward fastly as compared to bad news	Convergence time is fast
There is less utilisation of CPU and memory	There is more utilisation of CPU and memory
There is a problem of persistent looping	Only transient looping occurs
Best path is determined by the least number of hops	Best path is determined by the least cost
There is no hierarchical structure	There is a hierarchical structure
Require smaller bandwidth as there is no flooding, small packet and local sharing	Require larger bandwidth for flooding problems and for transmitting large link state packets
It updates tables with information about its neighbours. So, it works based on local information	It has the information of the whole network. So, it works based on global information
It updates on a broadcast basis	It updates on a multicast basis

ARP Vs RARP Routing Protocol

Parameters	ARP	RARP
Full Form	The term ARP is an abbreviation for Address resolution protocol.	The term RARP is an abbreviation for Reverse Address Resolution Protocol.
Basics	The ARP retrieves the receiver's physical address in a network.	The RARP retrieves a computer's logical address from its available server.
Broadcast Address	The nodes use ARP broadcasts in the LAN with the help of the MAC address.	The RARP utilises IP addresses for broadcasting.
Table Maintained By	The ARP table is maintained by the Local Host.	The RARP table is maintained by the RARP Server.
Usage	The router or the host uses ARP to find another router/host's address (physical address) in LAN.	RARP is used by thin clients that have limited facilities.
Reply Information	The primary use of the ARP reply is to update the ARP table.	The primary use of the RARP reply is to configure the local host's IP address.
Mapping	The ARP maps the node's IP address (32-bit logical address) to the MAC address/physical address (48-bit address).	The RARP maps the 48-bit address (MAC address/physical address) to the logical IP address (32-bit).



Host A
192.168.1.1
0002-6779-0f4c

Host B
192.168.1.2
00a0-2470-febd

Sender MAC address	Sender IP address	Target MAC address	Target IP address
0002-6779-0f4c	192.168.1.1	0000-0000-0000	192.168.1.2

Sender MAC address	Sender IP address	Target MAC address	Target IP address
00a0-2470-febd	192.168.1.2	0002-6779-0f4c	192.168.1.1

ICMP Protocol

- The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues.
- ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner.
- Commonly, the ICMP protocol is used on network devices, such as routers.
- ICMP is crucial for error reporting and testing, but it can also be used in distributed denial-of-service (DDoS) attacks.

Types of ICMP messages

- **Information Messages** – In this message, the sender sends a query to the host or router and expects an answer. For example, A host wants to know if a router is alive or not.
- **Error-reporting message** – This message report problems that a router or a host (destination) may encounter when it processes an IP packet.
- **Query Message** – It helps a router or a network manager to get specific information from a router or another host.

Category	Type	Message
Error-Reporting Messages	3	Destination unreachable
	4	Source quench
	11	Time Exceeded
	12	Parameter Problem
	5	Redirection
Query Message	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router Solicitation or advertisement

•**Source Quench** − It requests to decrease the traffic rate of message sending from source to destination.

•**Time Exceeded** − When fragments are lost in a network the fragments hold by the router will be dropped and then ICMP will take the source IP from the discarded packet and inform the source, that datagram is discarded due to the time to live field reaches zero, by sending time exceeded message.

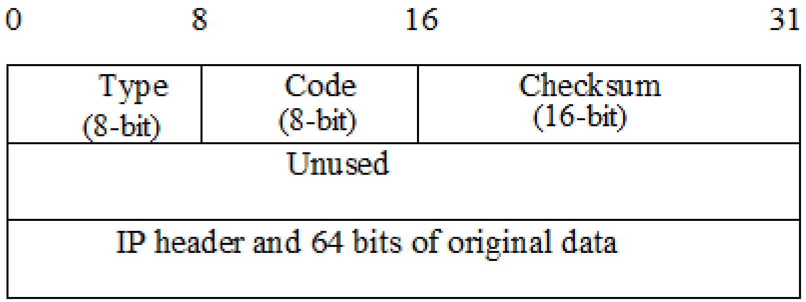
•**Fragmentation Required** − When a router is unable to forward a datagram because it exceeds the MTU of the next-hop network and the DF (Don't Fragment) bit is set, the router is required to return an ICMP Destination Unreachable message to the source of the datagram, with the Code indicating fragmentation is needed and DF (Don’t Fragment) set.

•**Destination Unreachable** − This error message indicates that the destination host, network, or port number that is specified in the IP packet is unreachable. This may happen due to the destination host device is down, an intermediate router is unable to find a path to forward the packet, and a firewall is configured to block connections from the source of the packet.

•**Redirect Message** − A redirect error message is used when a router needs to tell a sender that it should use a different path for a specific destination. It occurs when the router knows a shorter path to the destination.

ICMP Basic Error Message Format

- A basic ICMP error message would have the following format :



0	8	16 31
Type (8-bit)	Code (8-bit)	Checksum (16-bit)
Unused		
IP header and 64 bits of original data		

- **Type** – The type field identifies the type of the message.
- **Code** – The code field in ICMP describes the purpose of the message.
- **Checksum** – The checksum field is used to validate ICMP messages.

IGMP

- The Internet Group Management Protocol (IGMP) is a protocol that allows several devices to share one IP address so they can all receive the same data.
- IGMP is a network layer protocol used to set up multicasting on networks that use the Internet Protocol version 4 (IPv4).
- Specifically, IGMP allows devices to join a multicasting group.
- Multicasting is when a group of devices all receive the same messages or packets.
- Multicasting works by sharing an IP address between multiple devices.

How does IGMP work?

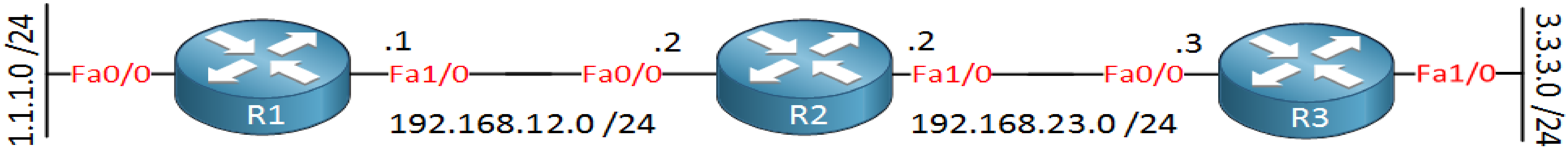
- Computers and other devices connected to a network use IGMP when they want to join a multicast group.
- A router that supports IGMP listens to IGMP transmissions from devices in order to figure out which devices belong to which multicast groups.
- IGMP uses IP addresses that are set aside for multicasting.
- Multicast IP addresses are in the range between 224.0.0.0 and 239.255.255.255. (In contrast, anycast networks can use any regular IP address.)
- Each multicast group shares one of these IP addresses.
- When a router receives a series of packets directed at the shared IP address, it will duplicate those packets, sending copies to all members of the multicast group.
- IGMP multicast groups can change at any time. A device can send an IGMP "join group" or "leave group" message at any point.
- IGMP works directly on top of the Internet Protocol (IP). Each IGMP packet has both an IGMP header and an IP header.
- Just like ICMP, IGMP does not use a transport layers protocol such as TCP or UDP.

Types of IGMP messages

- The IGMP protocol allows for several kinds of IGMP messages:
- Membership reports: Devices send these to a multicast router in order to become a member of a multicast group.
- "Leave group" messages: These messages go from a device to a router and allow devices to leave a multicast group.
- General membership queries: A multicast-capable router sends out these messages to the entire connected network of devices to update multicast group membership for all groups on the network.
- Group-specific membership queries: Routers send these messages to a specific multicast group, instead of the entire network.

RIP Protocol

- RIP stands for Routing Information Protocol.
- RIP is an intra-domain routing protocol used within an autonomous system.
- Here, intra-domain means routing the packets in a defined domain, for example, web browsing within an institutional area.
- RIP is based on the distance vector-based strategy, so we consider the entire structure as a graph where nodes are the routers, and the links are the networks.
- In a routing table, the first column is the destination, or we can say that it is a network address.
- The cost metric is the number of hops to reach the destination. The number of hops available in a network would be the cost. The hop count is the number of networks required to reach the destination.
- In RIP, infinity is defined as 16, which means that the RIP is useful for smaller networks or small autonomous systems. The maximum number of hops that RIP can contain is 15 hops, i.e., it should not have more than 15 hops as 16 is infinity.
- The next column contains the address of the router to which the packet is to be sent to reach the destination.



Routing Table

1.1.1.0 /24	Fa0/0	0
192.168.12.0 /24	Fa1/0	0
192.168.23.0 /24	Fa1/0	1
3.3.3.0 /24	Fa1/0	2

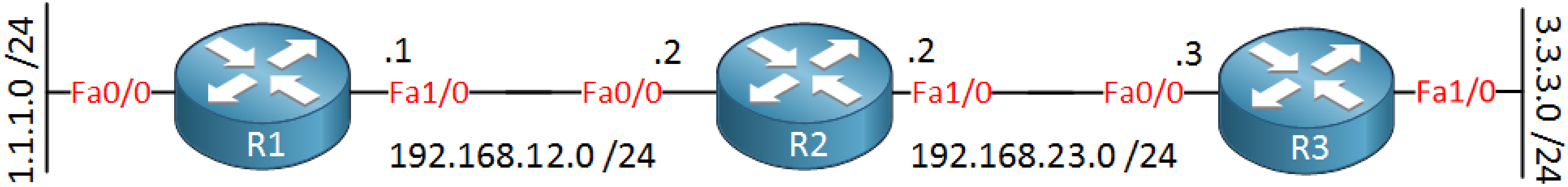
Routing Table

192.168.12.0 /24	Fa0/0	0
192.168.23.0 /24	Fa1/0	0
1.1.1.0 /24	Fa0/0	1
3.3.3.0 /24	Fa1/0	1

Routing Table

192.168.23.0 /24	Fa0/0	0
3.3.3.0 /24	Fa1/0	0
192.168.12.0 /24	Fa0/0	1
1.1.1.0 /24	Fa0/0	2

In red, you can see which interface, and in green, you can see the metric.



Routing Table

1.1.1.0 /24	Fa0/0	0
192.168.12.0 /24	Fa1/0	0
192.168.23.0 /24	Fa1/0	1
3.3.3.0 /24	Fa1/0	2

Routing Table

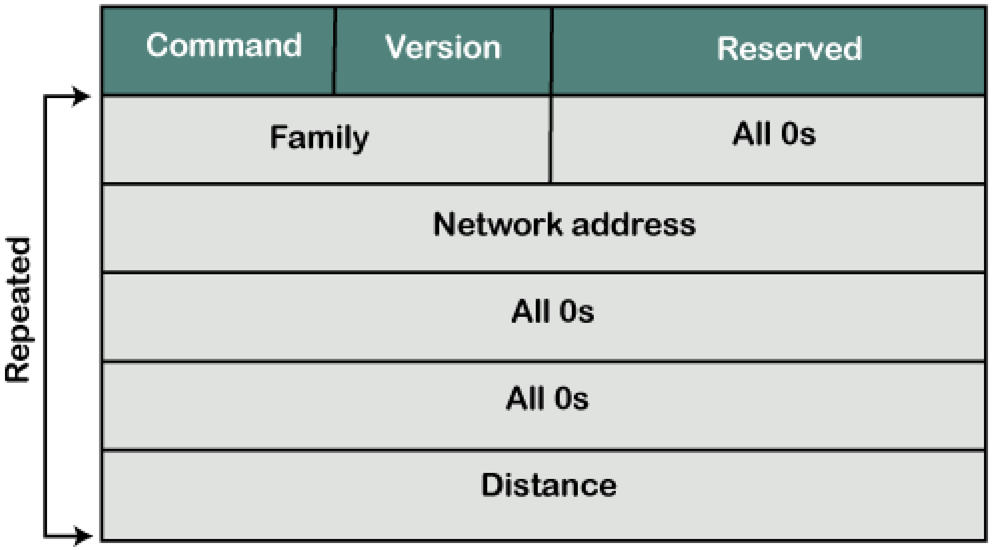
192.168.12.0 /24	Fa0/0	0
192.168.23.0 /24	Fa1/0	0
1.1.1.0 /24	Fa0/0	1
3.3.3.0 /24	Fa1/0	1

Routing Table

192.168.23.0 /24	Fa0/0	0
3.3.3.0 /24	Fa1/0	0
192.168.12.0 /24	Fa0/0	1
1.1.1.0 /24	Fa0/0	2

Take a look at R1, and you will see that it has learned about the 192.168.23.0 /24 and 3.3.3.0 /24 networks from R2. You see that it has added these two items: Interface (Fa1/0) and Metric (hop count). 192.168.23.0 /24 is one hop away, and 3.3.3.0 /24 is two hops away.

RIP Message Format



Command | Version | Reserved

Family | All 0s

Network address

All 0s

All 0s

Distance

Repeated

- Command: It is an 8-bit field that is used for request or reply. The value of the request is 1, and the value of the reply is 2.
- Version: Here, version means that which version of the protocol we are using. Suppose we are using the protocol of version1, then we put the 1 in this field.
- Reserved: This is a reserved field, so it is filled with zeroes.
- Family: It is a 16-bit field. As we are using the TCP/IP family, so we put 2 value in this field.
- Network Address: It is defined as 14 bytes field. If we use the IPv4 version, then we use 4 bytes, and the other 10 bytes are all zeroes.
- Distance: The distance field specifies the hop count, i.e., the number of hops used to reach the destination.

Advantages of RIP

- It is easy to configure
- It has less complexity
- The CPU utilization is less.

OSPF

- The OSPF stands for **Open Shortest Path First**. It is a widely used and supported routing protocol.
- It uses the **Shortest Path First** algorithm which is also called **Dijkstra's algorithm**.
- There are three versions of OSPF:
  - **OSPFv1:** This is the first version of OSPF created in the year 1989. It is no longer in use.
  - OSPFv2: This is the second version of OSPF created in 1998. It is used in IPv4. This version is important for **CCNA 200-301**.
  - **OSPFv3:** This is the latest version of OSPF created in the year 2008. This version is used for **IPv6** and as well as for IPv4.
- The routers store information about networks in **Link State Advertisements** (LSAs) which are organized in a structure called the **Link State Database** (LSDB).
- Routers **flood** the LSAs until all the routers in the OSPF area develop the same map of the network (LSDB).
- This flooding is similar to **switches** when they receive a broadcast frame or an unknown unicast frame.
- In OSPF, flooding means that they send the LSAs to all of their OSPF neighbors.

How OSPF Works?

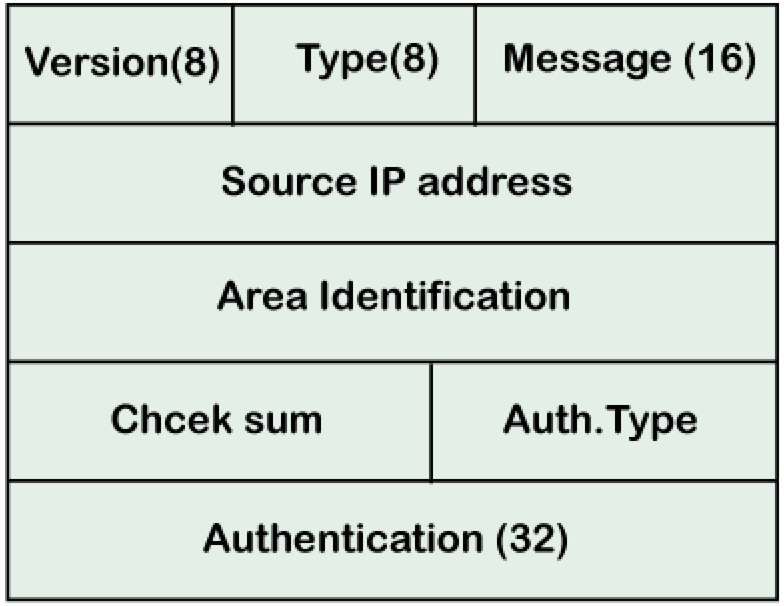
- The first step is to become neighbors with other routers connected to the same segment.
- Then the routers exchange the Link State Advertisements (LSAs) with neighbor routers.
- After that, each router independently calculates the best routes to each destination. Then it inserts all of these into the routing table.

Different States of OSPF

- The network devices that use the OSPF protocol undergo certain states. The various states of OSPF are as follows:
- **Down:** No “Hello” packets are received on the interface in the down state. The downstate means that the OSPF adjacency process has not begun yet.
- **INIT:** The “Hello” packets are received from other routers in the INIT state.
- **2WAY:** In this state, a bidirectional connection is formed. Both routers receive “Hello” packets from other routers.
- **Exstart:** In this state, the exchange of NULL DBD (Database Descriptor) takes place. The election of the **master** and **slave** router occurs. The router with the **higher router ID** becomes the master while the router with the **lower router ID** becomes the slave. This decides which router sends the DBD first.
- **Exchange:** The actual DBDs are exchanged in this state.
- **Loading:** The Link State Advertisements (LSAs) along with Link State Updates (LSUs) and Link State Requests (LSRs) are exchanged in this state.
- **Full:** All the information is synced in this state. It is only after the Full state that the OSPF routing begins!

Therefore, there are a total of **seven states** of OSPF that the routers undergo before the actual OSPF routing begins.

OSPF Message Format



<table>
<tr><td>Version(8)</td><td colspan="2">Type(8)</td><td>Message (16)</td></tr>
<tr><td colspan="4">Source IP address</td></tr>
<tr><td colspan="4">Area Identification</td></tr>
<tr><td colspan="2">Chcek sum</td><td colspan="2">Auth.Type</td></tr>
<tr><td colspan="4">Authentication (32)</td></tr>
</table>

- **Version:** It is an 8-bit field that specifies the OSPF protocol version.
- **Type:** It is an 8-bit field. It specifies the type of the OSPF packet.
- **Message:** It is a 16-bit field that defines the total length of the message, including the header. Therefore, the total length is equal to the sum of the length of the message and header.
- **Source IP address:** It defines the address from which the packets are sent. It is a sending routing IP address.
- **Area identification:** It defines the area within which the routing takes place.
- **Checksum:** It is used for error correction and error detection.
- **Authentication type:** There are two types of authentication, i.e., 0 and 1. Here, 0 means for none that specifies no authentication is available and 1 means for pwd that specifies the password-based authentication.
- **Authentication:** It is a 32-bit field that contains the actual value of the authentication data.

Advantages of OSPF

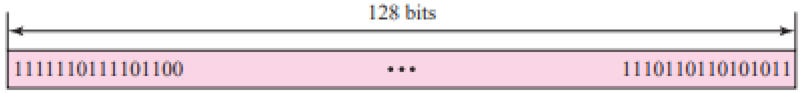
- OSPF can be configured on both IPv4 and IPv6 versions of IPs.
- It can carry out load balancing.
- It uses the SPF algorithm to present a loop-free technology.
- It is not Cisco proprietary. It can run on many routers.
- It is a classless protocol.
- It has unlimited hop counts.
- It works very fast.

Disadvantages of OSPF

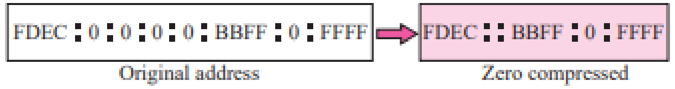
- It needs extra storage. Therefore, it means that it needs an extra CPU process to run the SPF algorithm.
- It needs more RAM to save adjacency topology.
- It is very complex. Therefore, it’s very difficult to troubleshoot.

IPV6

- An IPv6 address is 128 bits or 16 bytes (octet) long.
- The address length in IPv6 is four times of the length address in IPv4.



- **Notations**
- Dotted-Decimal Notation-221.14.65.11.105.45.170.34.12.234.18.0.14.0.115.255
- Colon Hexadecimal Notation-FDEC BA98 7654 3210 ADBF BBFF 2922 FFFF

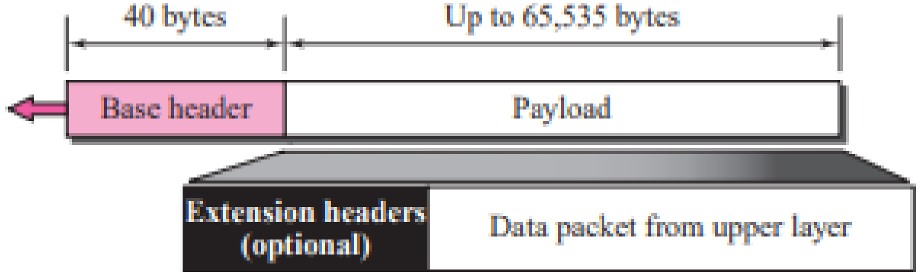


- Mixed Representation-FDEC:14AB:2311:BBFE:AAAA:BBBB:130.24.24.18

Three Address Types

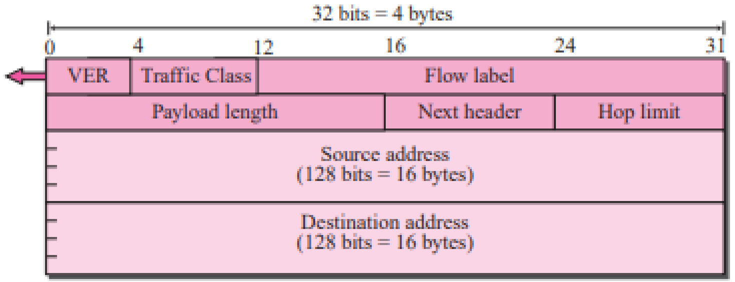
- **Unicast Address**: A unicast address defines a single interface (computer or router). The packet sent to a unicast address will be routed to the intended recipient. As we see shortly, IPv6 has designated a large block from which unicast addresses can be assigned to interfaces.
- **Anycast Address**: An anycast address defines a group of computers that all share a single address. A packet with an anycast address is delivered to only one member of the group, the most reachable one. An anycast communication is used, for example, when there are several servers that can respond to an inquiry. The request is sent to the one that is most reachable. The hardware and software generate only one copy of the request; the copy reaches only one of the servers. IPv6 does not designate a block for anycasting; the addresses are assigned from the unicast block.
- **Multicast Address**: A multicast address also defines a group of computers. However, there is a difference between anycasting and multicasting. In multicasting, each member of the group receives a copy. As we will see shortly, IPv6 has designated a block for multicasting from which the same address is assigned to the members of the group.
- **Broadcasting and Multicasting**: It is interesting that IPv6 does not define broadcasting, even in a limited version, as IPv4 does. IPv6 considers broadcasting as a special case of multicasting.

IPv6 datagram

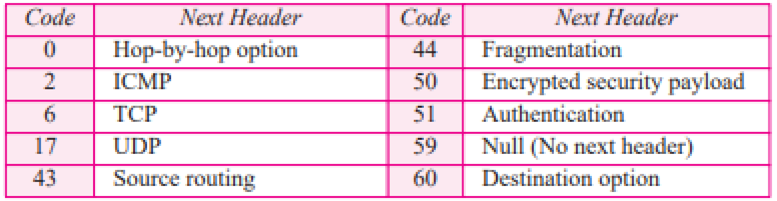


- Each packet is composed of a mandatory base header followed by the payload.
- The payload consists of two parts: optional extension headers and data from an upper layer.
- The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.

IPv6 Header Format



Next Header Codes



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

- **Version**. This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
- **Traffic Class**. This 8-bit field is used to distinguish different payloads with different delivery requirements. It replaces the service class field in IPv4.
- **Flow label**. The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data. We will discuss this field later.
- **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.
- **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. However, if source routing is used, this field contains the address of the next router.

Difference Between IPv4 and IPv6

IPv4	IPv6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end, connection integrity is Unachievable	In IPv6 end-to-end, connection integrity is Achievable
It can generate 4.29×10^{9} address space	The address space of IPv6 is quite large it can produce 3.4×10^{38} address space
The Security feature is dependent on the application	IPSEC is an inbuilt security feature in the IPv6 protocol
Address representation of IPv4 is in decimal	Address representation of IPv6 is in hexadecimal
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation is performed only by the sender
In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
In IPv4 checksum field is available	In IPv6 checksum field is not available
It has a broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4 has a header of 20-60 bytes.	IPv6 has a header of 40 bytes fixed
IPv4 can be converted to IPv6	Not all IPv6 can be converted to IPv4
IPv4 consists of 4 fields which are separated by addresses dot (.)	IPv6 consists of 8 fields, which are separated by a colon (:)
IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C, Class D , Class E.	IPv6 does not have any classes of the IP address.
IPv4 supports VLSM(Variable Length subnet mask).	IPv6 does not support VLSM.
Example of IPv4: 66.94.29.13	Example of IPv6: 2001:0000:3238:DFE1:0063:0000:0000:FEFB

Congestion:

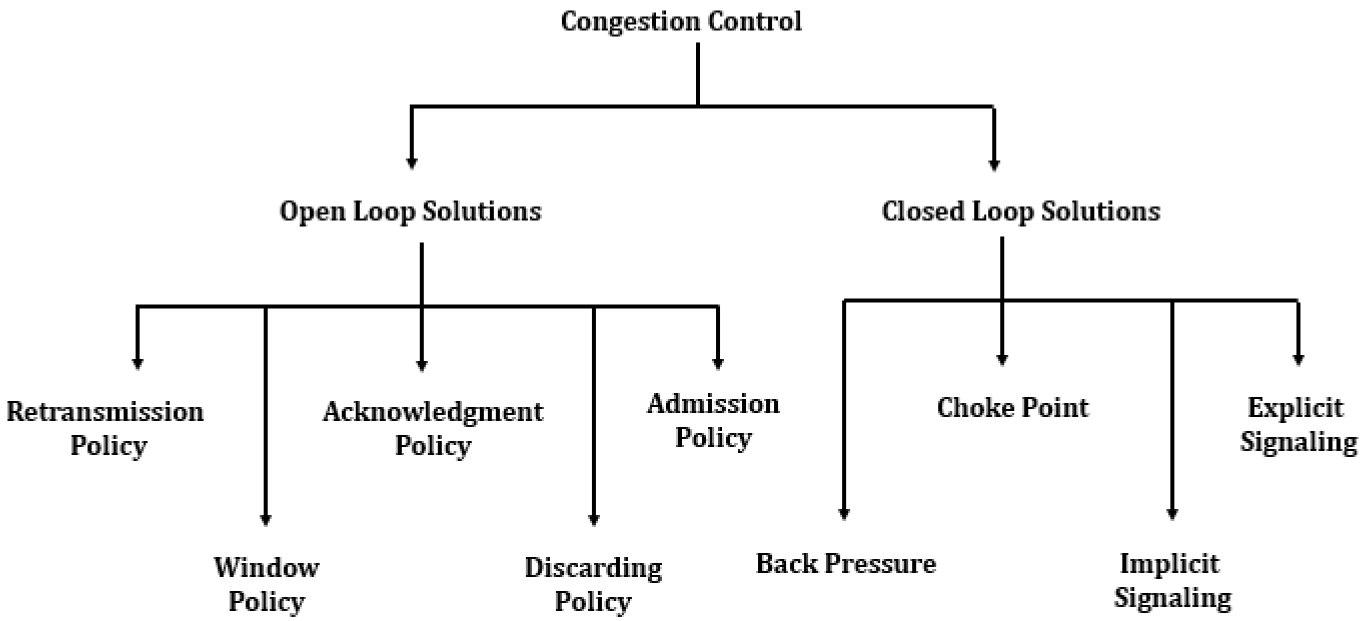
1. Congestion is an important issue that can arise in packet switched network.
2. Congestion is a situation in Communication Networks in which too many packets are present in a part of the subnet.
3. Congestion in a network may occur when the load on the network is greater than the capacity of the network.
4. Due to Congestion the performance degrades.

Factors that Causes the Congestion:

- Packet arrival rate exceeds the outgoing link capacity.
- Insufficient memory to store arriving packets.
- Bursty traffic.
- Slow processor.

Congestion Control:

1. Congestion Control is the techniques and mechanisms which can either prevent congestion from happening or remove congestion after it has taken place.
2. Congestion control mechanisms are divided into two categories, one category prevents the congestion from happening and the other category removes congestion after it has taken place.



I) Open Loop Congestion Control:

- In Open Loop Congestion Control, policies are used to prevent the congestion before it happens.
- Congestion control is handled either by the source or by the destination.

II) Closed Loop Congestion Control:

- Closed loop congestion control mechanisms try to remove the congestion after it happens.
- It uses some kind of feedback.
- **NEED of Congestion Control:**

1. It is not possible to completely avoid the congestion but it is necessary to control it.
2. Congestions lead to a large Queue Length.
3. It results in Buffer Overflow & Loss of Packets.
4. So, congestion control is necessary to ensure that the user gets the negotiated Quality of Services.

Congestion prevention policies: (1/3)

I) Retransmission Policy:

- The sender retransmits a packet, if it feels that the packet it has sent is lost or corrupted.
- However, retransmission increases the congestion in the network.
- But we need to implement good retransmission policy to prevent congestion.
- The retransmission policy and the retransmission timers need to be designed to optimize efficiency and at the same time prevent the congestion.

II) Window Policy:

- To implement window policy, selective reject window method is used for congestion control.
- Selective Reject method is preferred over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver.
- Thus, this duplication may make congestion worse.
- Selective reject method sends only the specific lost or damaged packets.

Congestion prevention policies:(2/3)

III) Acknowledgement Policy:

- The acknowledgement policy imposed by the receiver may also affect congestion.
- If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.
- Acknowledgements also add to the traffic load on the network.
- Thus, by sending fewer acknowledgements we can reduce the load on the network.
- To implement it, several approaches can be used:
- A receiver may send an acknowledgement only if it has a packet to be sent.
- A receiver may send an acknowledgement when a timer expires.
- A receiver may also decide to acknowledge only N packets at a time.

Congestion prevention policies:(3/3)

IV) Discarding Policy:

 A router may discard less sensitive packets when congestion is likely to happen.

 Such a discarding policy may prevent congestion and at the same time may not harm the integrity of the transmission.

V) Admission Policy:

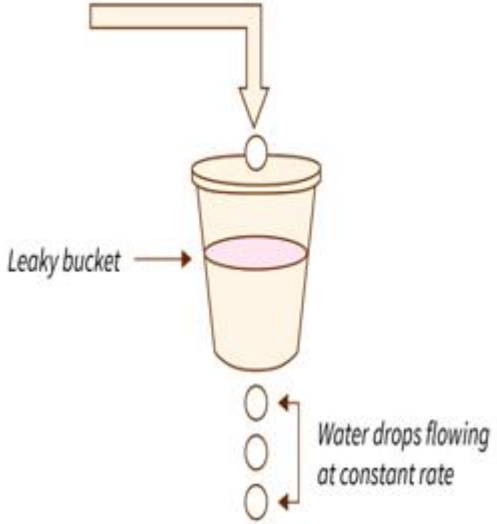
 An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.

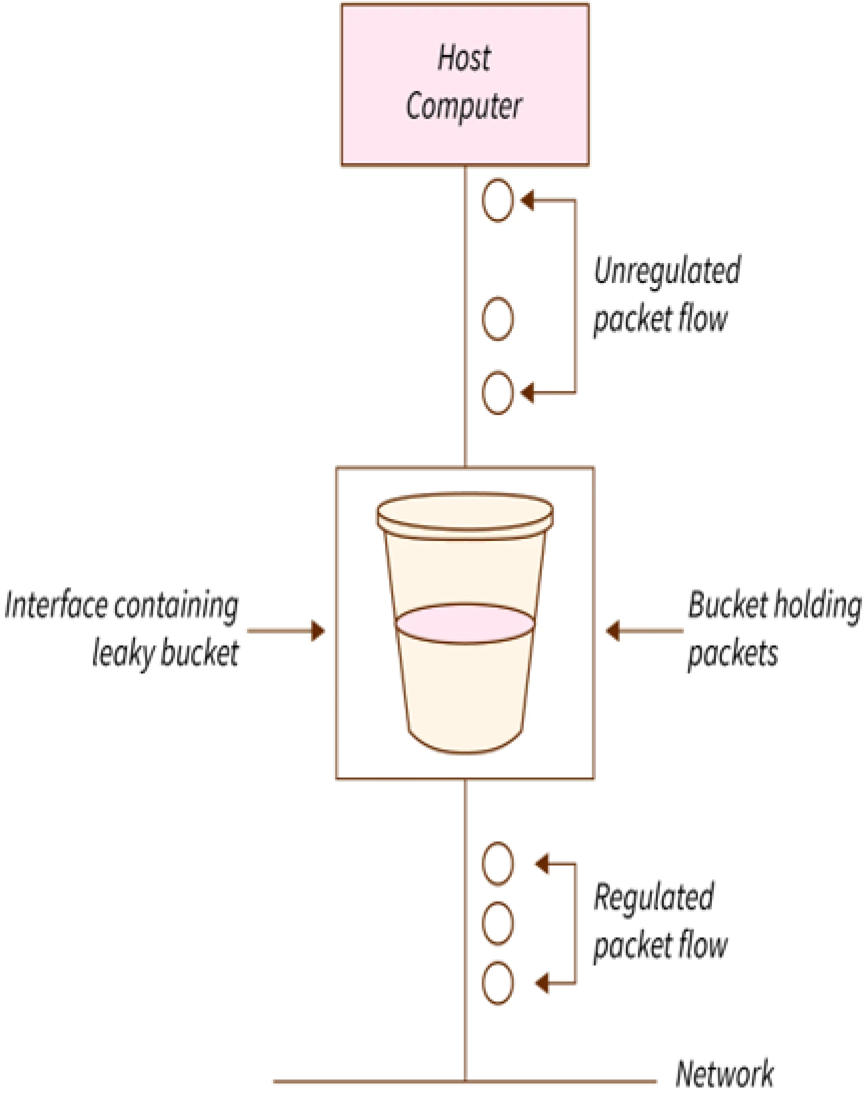
 Switches in a flow, first check the resource requirement of a flow before admitting it to the network.

 A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion.

Leaky Bucket Algorithm

- The leaky bucket algorithm is a method of congestion control where multiple packets are stored temporarily. These packets are sent to the network at a constant rate that is decided between the sender and the network. This algorithm is used to implement congestion control through traffic shaping in data networks.
- To understand the algorithm, let us first discuss the analogy of a leaky bucket.
- Consider a bucket with a small hole at the bottom. Now imagine that water is poured into the bucket at random intervals. At each interval, the amount of water poured into the bucket is not fixed. Now it does not matter how much water is inside the bucket, the water comes out at a constant rate from the hole. Consider the image below for more clarity.
- The rate at which water leaks (called the leak rate) is independent of the amount of water inside the bucket.
- If the bucket becomes full, the water poured will be lost.
- The same idea of the leaky bucket can be applied to the data packets.

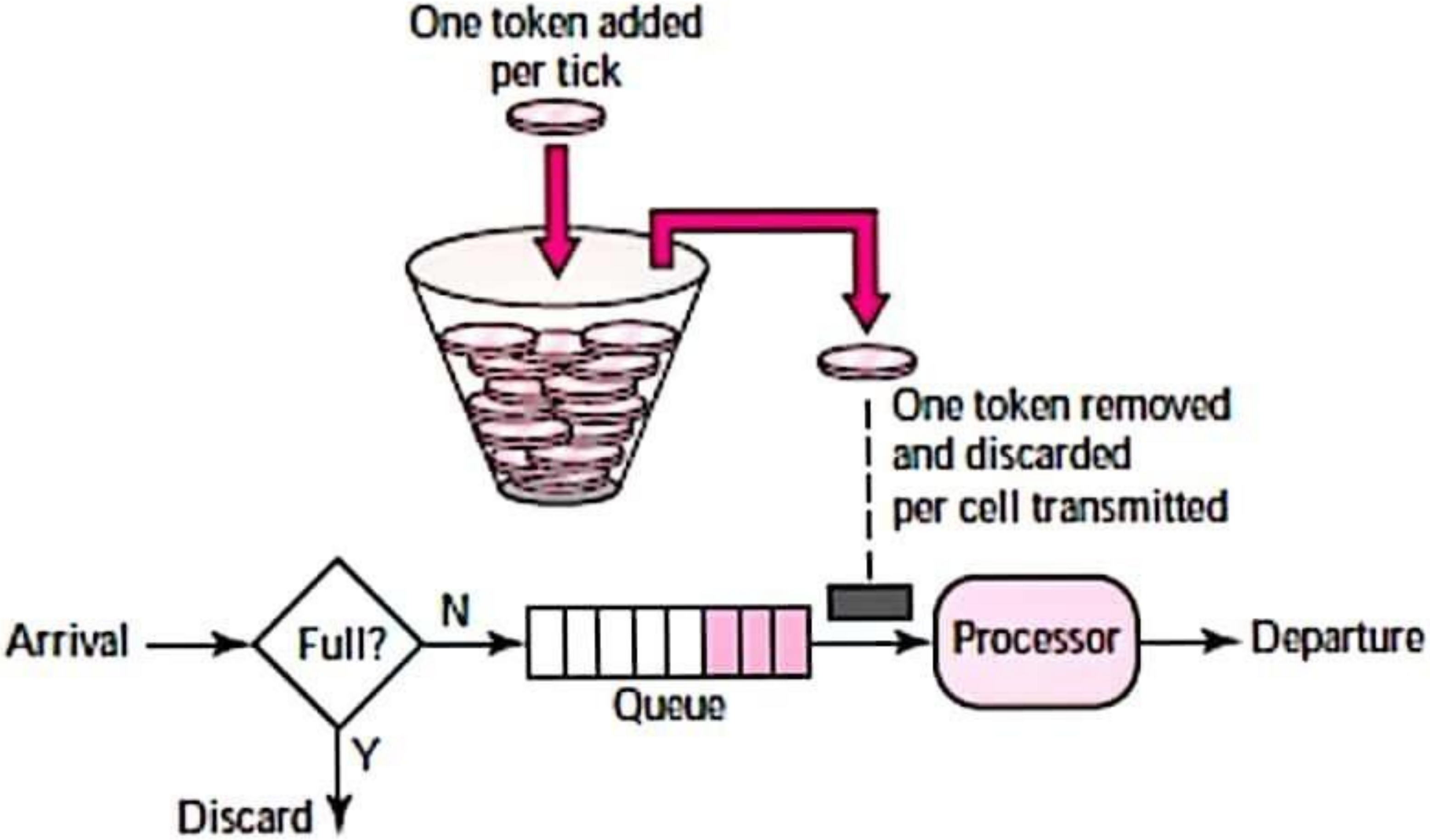


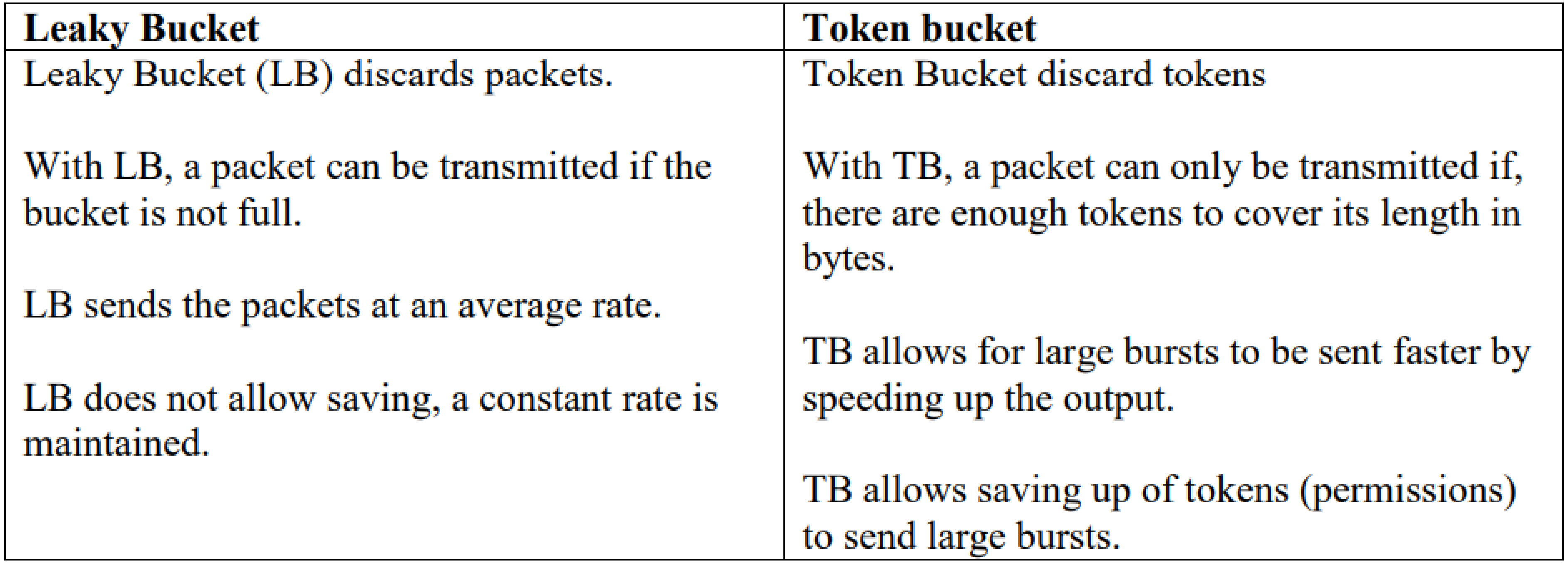


- Consider that, each network interface has a leaky bucket.
- Now, when the sender wants to transmit packets, the packets are thrown into the bucket. These packets get accumulated in the bucket present at the network interface.
- If the bucket is full, the packets are discarded by the buckets and are lost.
- This bucket will leak at a constant rate. This means that the packets will be transmitted to the network at a constant rate. This constant rate is known as the Leak Rate or the Average Rate.
- In this way, bursty traffic is converted into smooth, fixed traffic by the leaky bucket.
- Queuing and releasing the packets at different intervals help in reducing network congestion and increasing overall performance.

Token Bucket Algorithm

- The leaky bucket algorithm allows only an average (constant) rate of data flow. Its major problem is that it cannot deal with bursty data.
- A leaky bucket algorithm does not consider the idle time of the host. For example, if the host was idle for 10 seconds and now it is willing to send data at a very high speed for another 10 seconds, the total data transmission will be divided into 20 seconds and average data rate will be maintained. The host is having no advantage of sitting idle for 10 seconds.
- To overcome this problem, a token bucket algorithm is used. A token bucket algorithm allows bursty data transfers.
- A token bucket algorithm is a modification of leaky bucket in which leaky bucket contains tokens.
- In this algorithm, a token(s) are generated at every clock tick. For a packet to be transmitted, system must remove token(s) from the bucket.
- Thus, a token bucket algorithm allows idle hosts to accumulate credit for the future in form of tokens.
- For example, if a system generates 100 tokens in one clock tick and the host is idle for 100 ticks. The bucket will contain 10,000 tokens. Now, if the host wants to send bursty data, it can consume all 10,000 tokens at once for sending 10,000 cells or bytes. Thus a host can send bursty data as long as bucket is not empty.


One token added per tick
One token removed and discarded per cell transmitted
Arrival
Full?
N
Y
Discard
Queue
Processor
Departure



Leaky Bucket	**Token bucket**
Leaky Bucket (LB) discards packets. With LB, a packet can be transmitted if the bucket is not full. LB sends the packets at an average rate. LB does not allow saving, a constant rate is maintained.	Token Bucket discard tokens With TB, a packet can only be transmitted if, there are enough tokens to cover its length in bytes. TB allows for large bursts to be sent faster by speeding up the output. TB allows saving up of tokens (permissions) to send large bursts.

Thank you