Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

## Fall Semester 2023-24 UG Seniors (B.Tech 2021 Batch)

## Continuous Assessment Test – II

## Key

(1)(a) Consider a scenario where Station A needs to transmit a message containing 10 frames to Station B using sliding window protocol. Set the sender's window size is 3. Assume, every 5th transmission is lost, while acknowledgments from the receiver for other transmissions are never lost. After every 5th transmission loss, the remaining frames within the window are sent to the receiver. Subsequently a timeout occurs. Draw the timing diagram and determine the total number of data transmission made by the sender (including retransmissions) if it follows:-

i. Go-back-N protocol **(3 marks)**

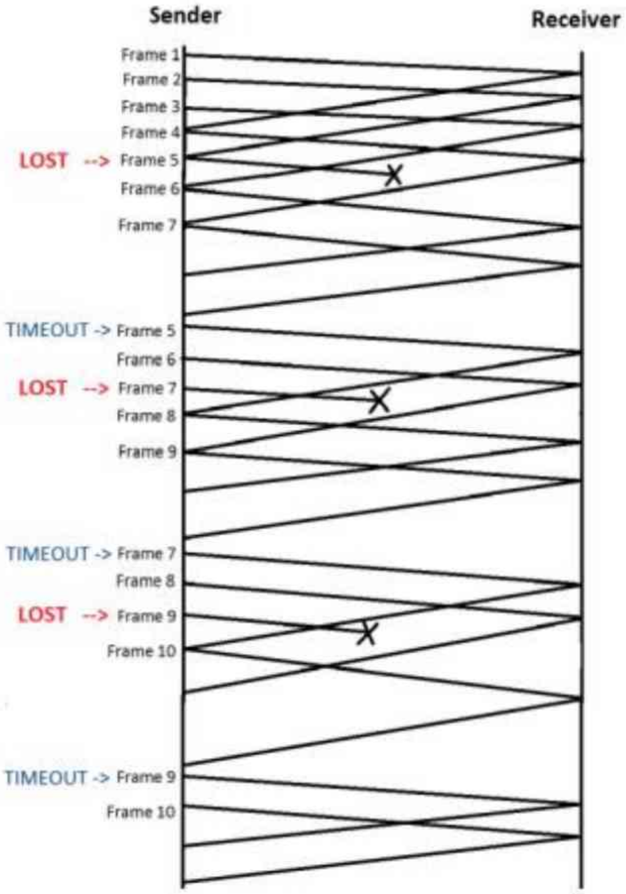
ii. Selective repeat protocol **(3 marks)**

Also, calculate the ratio of the effective number of frames sent to the total number of frames sent in each of the above cases.

Sol:

In this problem, for every 5th frame that is trying to be sent from the sender to receiver, an error in transmission occurs, and the frame is lost.

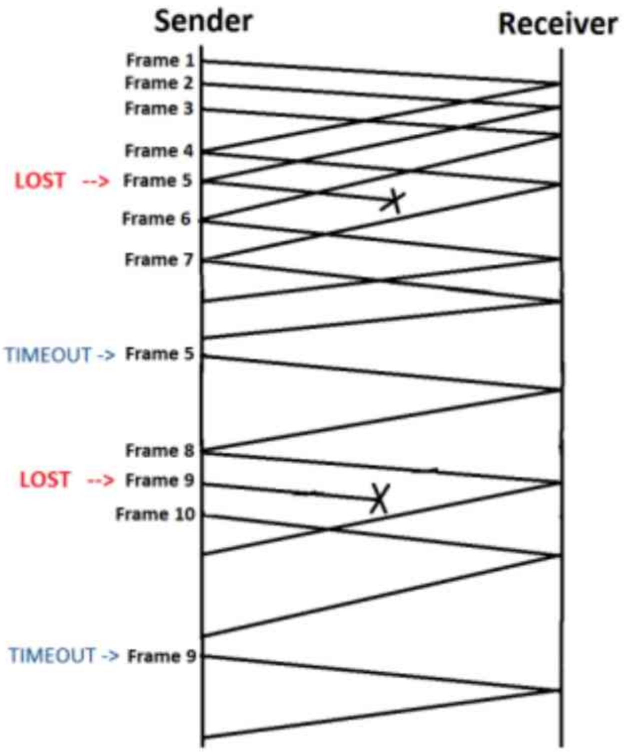
i) Given below is a diagram showing the sequence of transmissions from the sender to the receiver. The go-back-n protocol is followed, and every 5th frame transmitted is lost. (With necessary sliding in the sender side when ack received)



It can be seen that in order to transmit 10 frames of data from the sender to the receiver, 18 frames in total were sent. Hence the efficiency of the transmission is 10/18 = 55.55%.

(ii). Given below is a diagram showing the sequence of transmissions from the sender to the receiver. The selective repeat protocol is followed, and every 5th frame being transmitted is lost.

(With necessary sliding in the sender side when acknowledgment received)



It can be seen that in order to transmit 10 frames of data from the sender to the receiver, 12 frames in total were sent. Hence the efficiency of the transmission is 10/12 = 83.33%.

(b) A slotted ALOHA network transmits 800-bit frames using a shared channel with 400 kbps bandwidth. Find the throughput if the system (all stations together) produces ( 4 marks)

i. 2000 frames per second

ii. 1000 frames per second

iii. 500 frames per second

Also find the vulnerability.

Sol:

The frame transmission time is 800/400 kbps or 2 ms.

G= No. of frames * Tfr

Tfr = 2ms.

a. If the system creates 2000 frames per second, this is 2 frame per millisecond. The load is 4. In this case $S = G \times e^{-G}$ or S = 0.0732 (7.3 percent). This means that the throughput is 2000 × 0.0732= 146 and so only 146 frames out of 2000 will probably survive.
b. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 2. In this case $S = G \times e^{-G}$ or S = 0.2706 (27.06 percent). This means that the throughput is 1000 × 0.2706 = 271. Only 271 frames out of 1000 will probably survive.
c. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is 1. In this case $S = G \times e^{-G}$ or S = 0.3678 (36.78 percent). This means that the throughput is 500 × 0.3678 = 184. Only 184 frames out of 500 will probably survive.

Vulnerability = $T_{fr}$ = 2ms.

(2) An ISP is granted a block of addresses starting with 188.50.0.0/16 The ISP wants to distribute these blocks to 100 customers as follows.
**(10 marks)**

a. The first group has 30 medium-size businesses; each needs 128 addresses.

b. The second group has 50 small businesses; each needs 64 addresses.

c. The third group has 20 households; each needs 32 addresses

Design the subblocks and give the slash notation for each subblock. Find out how many addresses are still available after these allocations.

Sol:

ISP granted block of addresses starting with 188.50.0.0/16. So Total no of addresses available now is $2^{16}$=$65536$.

Group 1 has 30 medium size businesses each need 128 addresses.So we need $\log_2 128 = 7$ bit to define each host.The prefix length for group 1 is then $32 - 7 = 25$.

The addresses are :-

1st business :- $188.50.0.0/25 - - - - - - - - 188.50.0.127/25$

2nd business :-$188.50.0.128/25 - - - - - - - - 188.50.0.255/25$

3rd  business :-$188.50.1.0/25 - - - - - - - - 188.50.1.127/25$

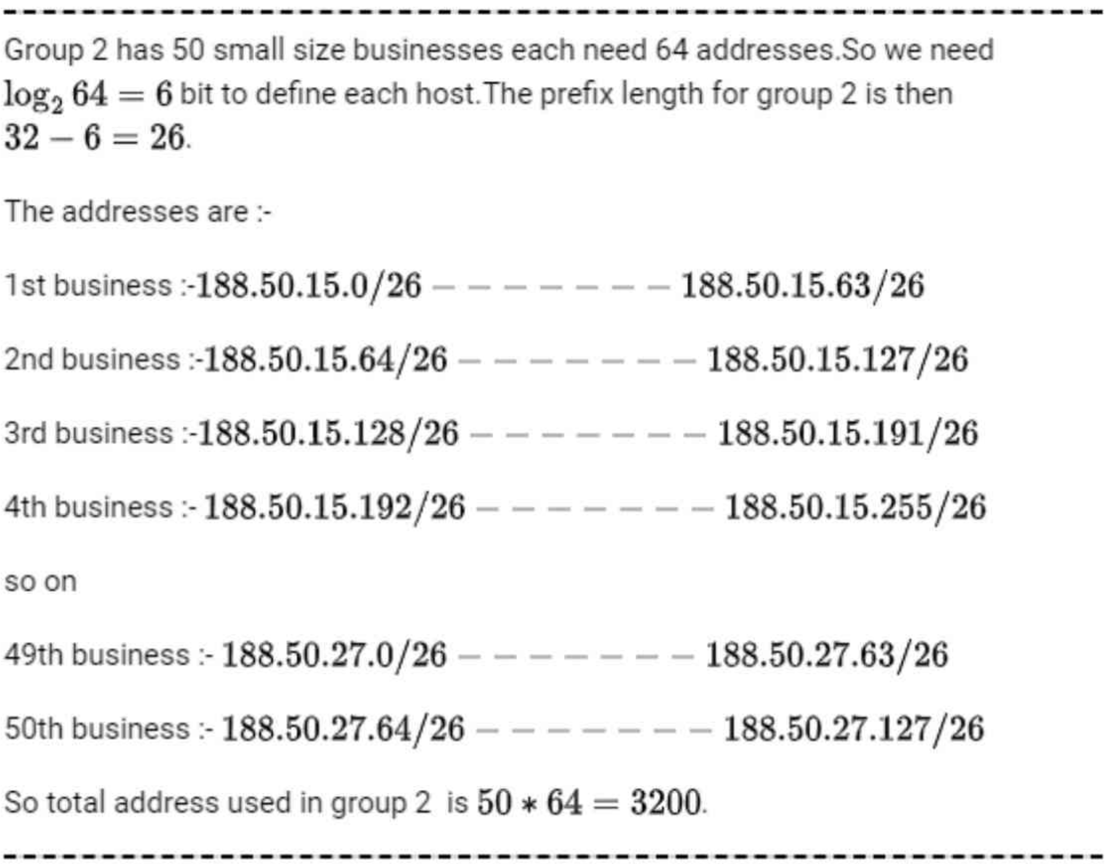
4th business :- $188.50.1.128/25 - - - - - - - - 188.50.1.255/25$

so on

29th business:-$188.50.14.0/25 - - - - - - - - 188.50.14.127/25$

30th business:-$188.50.14.128/25 - - - - - - - - 188.50.14.255/25$

So total address used in group 1  is $30 * 128 = 3840$.



---

Group 2 has 50 small size businesses each need 64 addresses.So we need $\log_2 64 = 6$ bit to define each host.The prefix length for group 2 is then $32 - 6 = 26$.

The addresses are :-

1st business :-$188.50.15.0/26 - - - - - - - - 188.50.15.63/26$

2nd business :-$188.50.15.64/26 - - - - - - - - 188.50.15.127/26$

3rd business :-$188.50.15.128/26 - - - - - - - - 188.50.15.191/26$

4th business :- $188.50.15.192/26 - - - - - - - - 188.50.15.255/26$

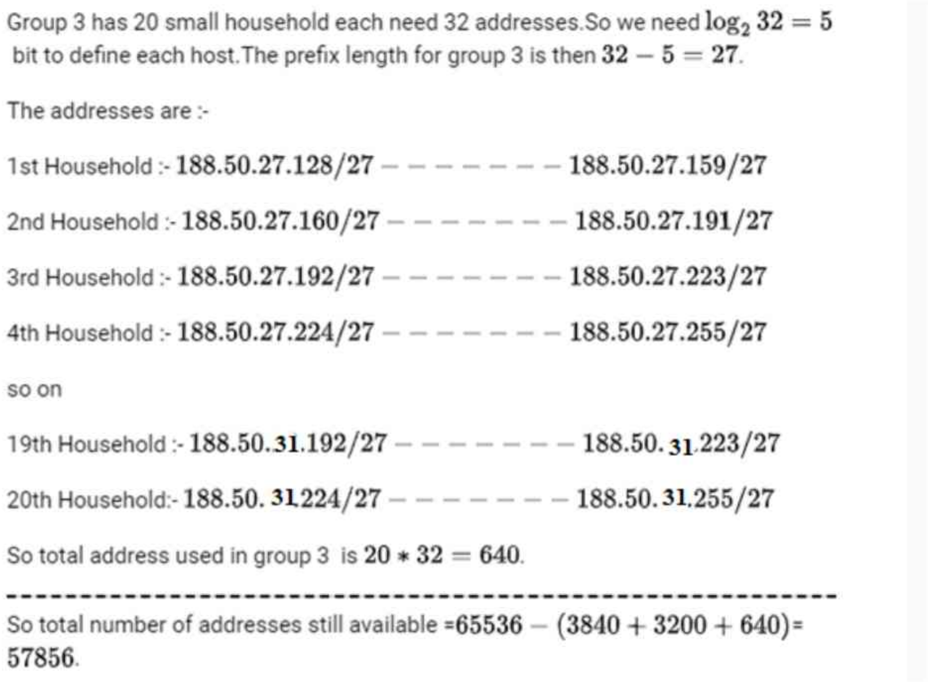
so on

49th business :- $188.50.27.0/26 - - - - - - - - 188.50.27.63/26$

50th business :- $188.50.27.64/26 - - - - - - - - 188.50.27.127/26$

So total address used in group 2 is $50 * 64 = 3200$.

---



Group 3 has 20 small household each need 32 addresses.So we need $\log_2 32 = 5$ bit to define each host.The prefix length for group 3 is then $32 - 5 = 27$.

The addresses are :-

1st Household :- $188.50.27.128/27 - - - - - - - 188.50.27.159/27$

2nd Household :- $188.50.27.160/27 - - - - - - - 188.50.27.191/27$

3rd Household :- $188.50.27.192/27 - - - - - - - 188.50.27.223/27$

4th Household :- $188.50.27.224/27 - - - - - - - 188.50.27.255/27$

so on

19th Household :- $188.50.31.192/27 - - - - - - - 188.50.31.223/27$

20th Household:- $188.50.31.224/27 - - - - - - - 188.50.31.255/27$

So total address used in group 3 is $20 * 32 = 640$.

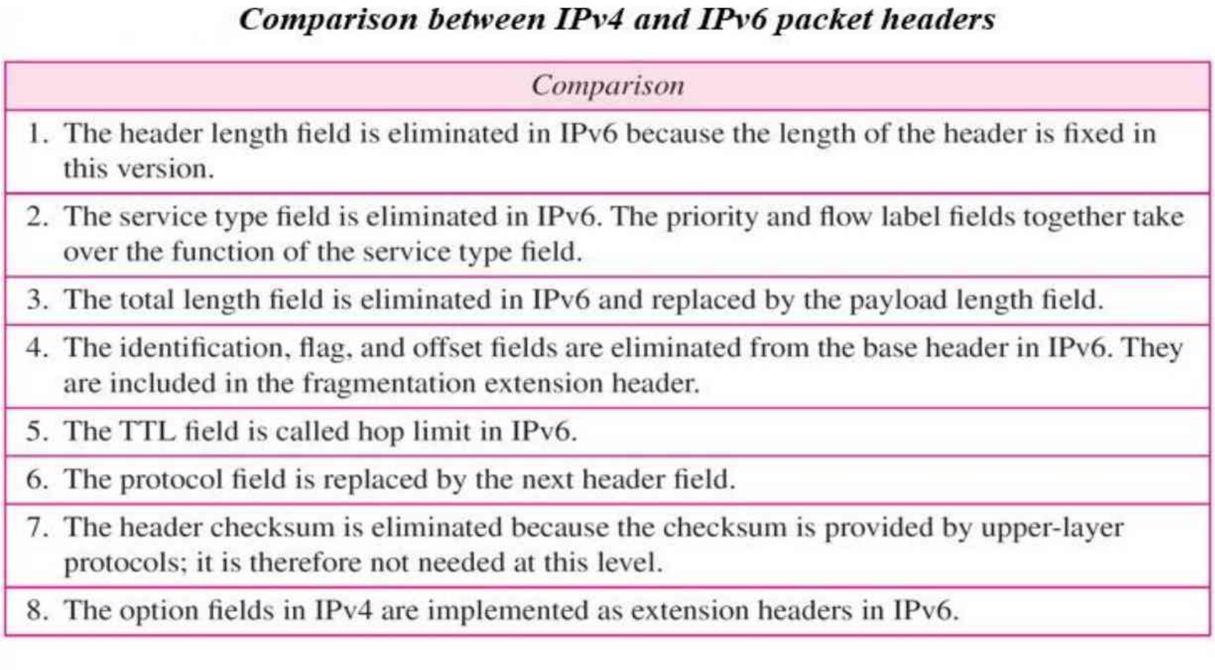
---

So total number of addresses still available =$65536 - (3840 + 3200 + 640)$= $57856$.

3.(a) Discuss the differences between an IPv4 and IPv6 packet header (List any six). (4 marks)

| | |
|---|---|
| IPv4 is a 32-bit address. | IPv6 is a 128-bit address. |
| IPv4 is a numeric address that consists of 4 fields which are separated by dot (.). | IPv6 is an alphanumeric address that consists of 8 fields, which are separated by colon. |
| IPv4 has 5 different classes of IP address that includes Class A, Class B, Class C, Class D, and Class E. | IPv6 does not contain classes of IP addresses. |
| IPv4 has a limited number of IP addresses. | IPv6 has a large number of IP addresses. |
| It supports VLSM (Virtual Length Subnet Mask). Here, VLSM means that Ipv4 converts IP addresses into a subnet of different sizes. | It does not support VLSM. |
| It supports manual and DHCP configuration. | It supports manual, DHCP, auto-configuration, and renumbering. |
| It generates 4 billion unique addresses | It generates 340 undecillion unique addresses. |
| In IPv4, end-to-end connection integrity is unachievable. | In the case of IPv6, end-to-end connection integrity is achievable. |
| In IPv4, security depends on the application. This IP address is not developed in keeping the security feature in mind. | In IPv6, IPSEC is developed for security purposes. |
| In IPv4, the IP address is represented in decimal. | In IPv6, the representation of the IP address in hexadecimal. |
| Fragmentation is done by the senders and the forwarding routers. | Fragmentation is done by the senders only. |
| It does not provide any mechanism for packet flow identification. | It uses flow label field in the header for the packet flow identification. |
| The checksum field is available in IPv4. | The checksum field is not available in IPv6. |
| IPv4 is broadcasting. | On the other hand, IPv6 is multicasting, which provides efficient network operations. |
| It does not provide encryption and authentication. | It provides encryption and authentication. |
| It consists of 4 octets. | It consists of 8 fields, and each field contains 2 octets. Therefore, the total number of |

| | |
|---|---|
| | octets in IPv6 is 16. |



**Comparison between IPv4 and IPv6 packet headers**

| Comparison |
|---|
| 1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version. |
| 2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field. |
| 3. The total length field is eliminated in IPv6 and replaced by the payload length field. |
| 4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header. |
| 5. The TTL field is called hop limit in IPv6. |
| 6. The protocol field is replaced by the next header field. |
| 7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level. |
| 8. The option fields in IPv4 are implemented as extension headers in IPv6. |

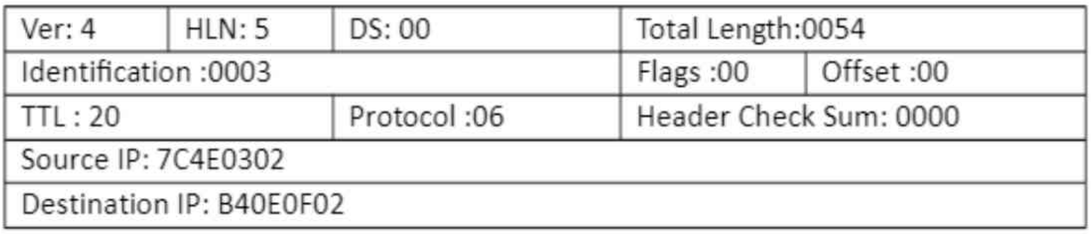
(b) An IPv4 datagram has arrived with the following information in the header(in hexadecimal): (6)

*45 00 00 54 00 03 00 00 20 06 00 00 7C 4E 03 02 B4 0E 0F 02*

i. Is the packet fragmented? Justify.
ii. What is the size of the data?
iii. How many routers thus the packet travel to? Justify.
iv. What is the identification number of the packet?
v. Give the offset value?
vi. Write the source IP address and Destination IP address

Sol:

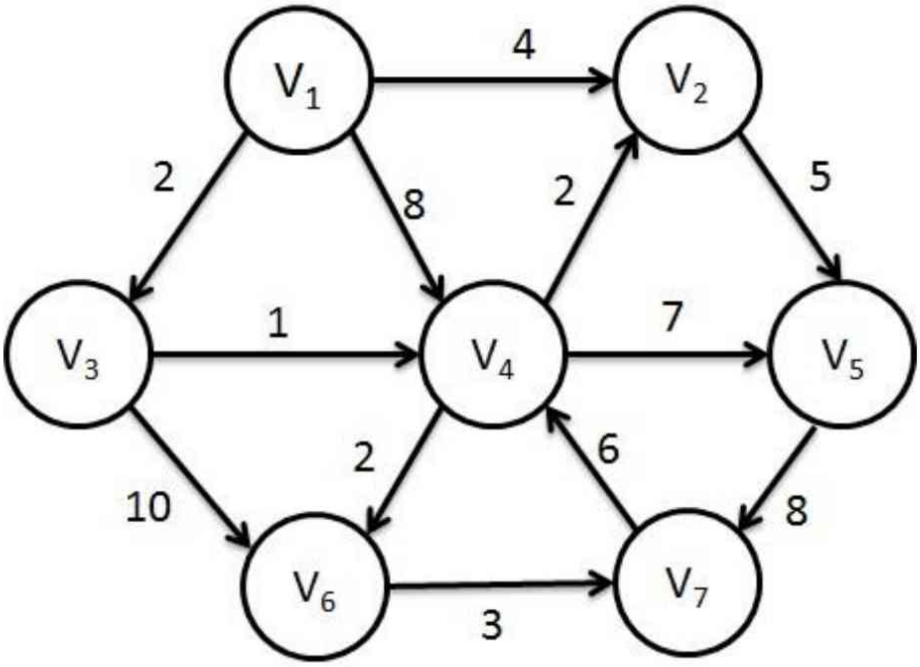
i) The packet is not fragmented because the offset value is 0 and flag value is 0
ii) The size of the data= Total length – HLEN = 84-20=64 bytes
iii) The packet can travel to 32 more routers
iv) The identification number of this packet is 0003
v) The offset value is 0
vi) Source IP address: 0x7C4E0302=124.78.3.2 and Destination address: 0xB40E0F02= 180.14.15.2



| | | | | |
|---|---|---|---|---|
| Ver: 4 | HLN: 5 | DS: 00 | Total Length:0054 | |
| Identification :0003 | | | Flags :00 | Offset :00 |
| TTL : 20 | | Protocol :06 | Header Check Sum: 0000 | |
| Source IP: 7C4E0302 | | | | |
| Destination IP: B40E0F02 | | | | |

4) Use the Dijkstra's algorithm to solve the shortest path problem for the following graph. Find shortest path from $\mathbf{V_1}$ to all other vertices. Draw a table showing the intermediate distance values of all the vertices at each iteration of the algorithm. (10 marks)

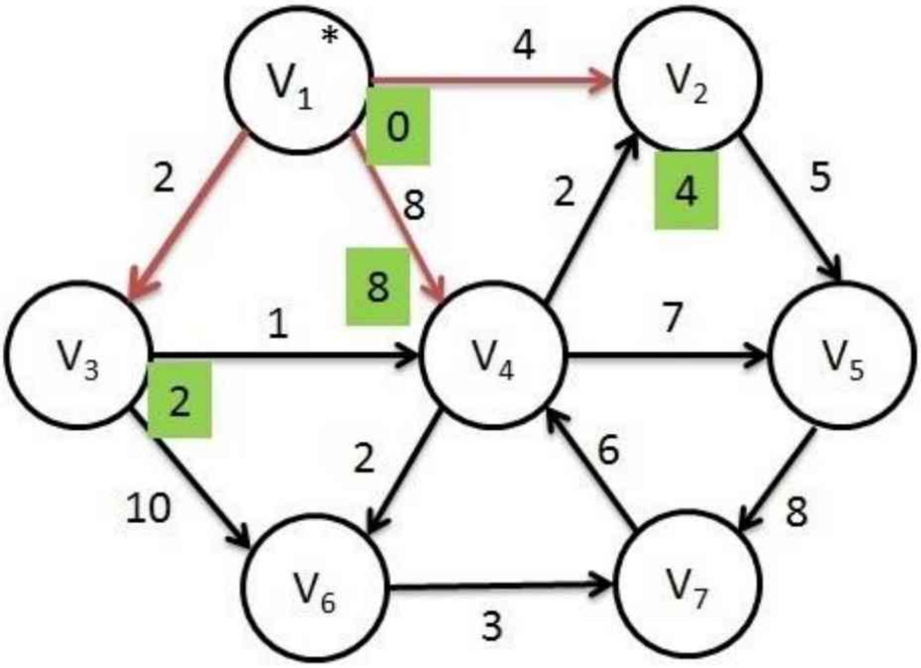
.

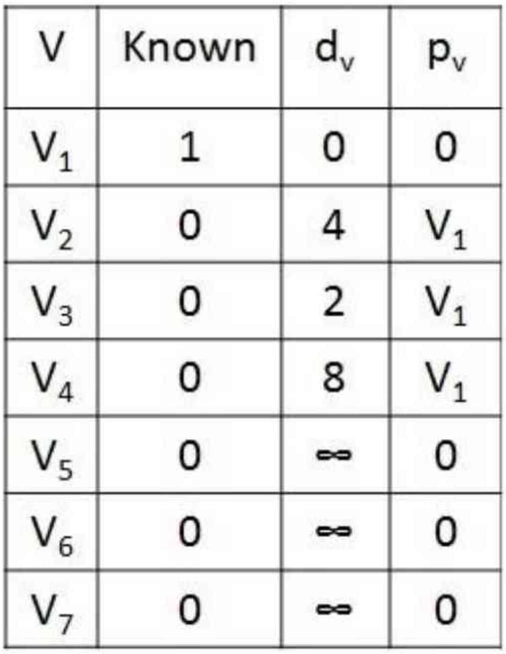


Sol:

Dijkstras Algorithm - Step 1

First, we select the source vertex as $V_1$, with path length 0 and we set known value to 1 and update the distance value of adjacent vertices such as $V_2$, $V_3$, and $V_4$.



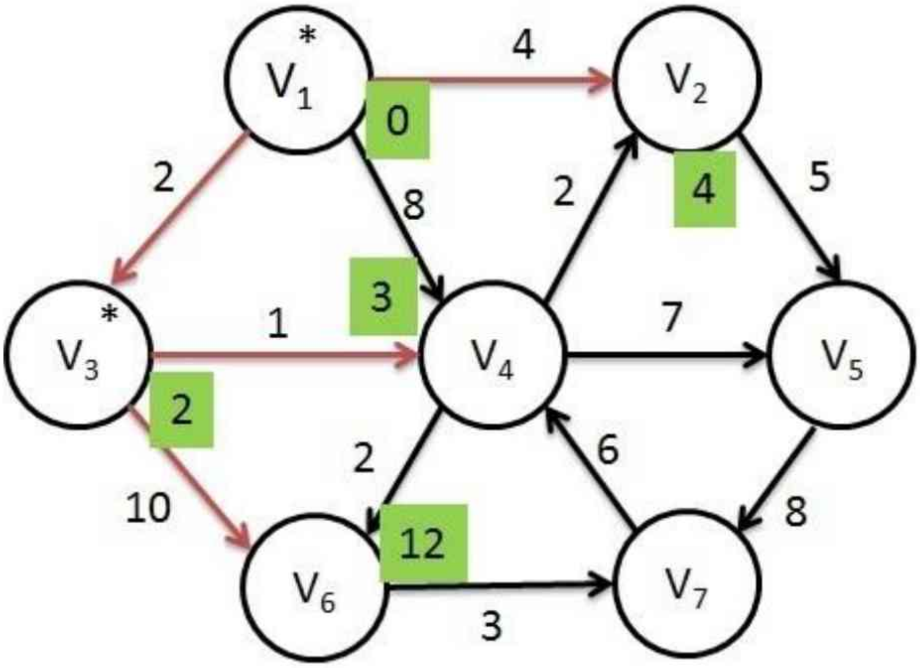


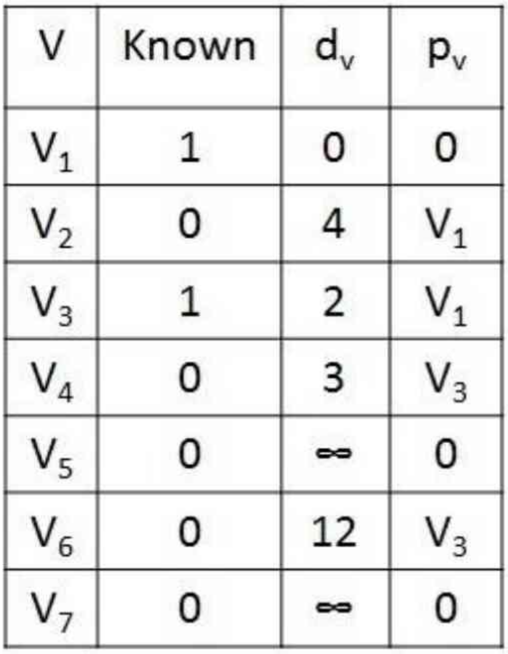
| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 0 | 4 | $V_1$ |
| $V_3$ | 0 | 2 | $V_1$ |
| $V_4$ | 0 | 8 | $V_1$ |
| $V_5$ | 0 | ∞ | 0 |
| $V_6$ | 0 | ∞ | 0 |
| $V_7$ | 0 | ∞ | 0 |

Fig 2. After $V_1$ is known

## Dijkstras Algorithm - Step 2

Next, $V_3$ is selected and set known value to 1 and update the adjacent vertices $V_4$ and $V_6$. Actually, $V_4$ is already reached with cost of 8, but the cost of going through $v_3$ will be 3 which is smaller than the previous cost. So we can replace that old one with new distance value.





| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 0 | 4 | $V_1$ |
| $V_3$ | 1 | 2 | $V_1$ |
| $V_4$ | 0 | 3 | $V_3$ |
| $V_5$ | 0 | ∞ | 0 |
| $V_6$ | 0 | 12 | $V_3$ |
| $V_7$ | 0 | ∞ | 0 |

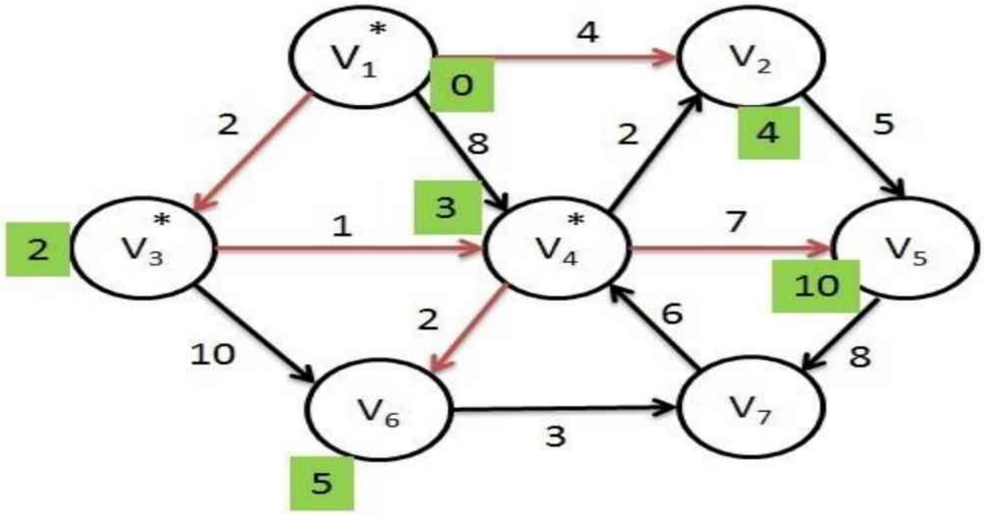
$$d_{v4}=d_{v3}+C_{v3,\,v4}$$
$$=2+1=3$$

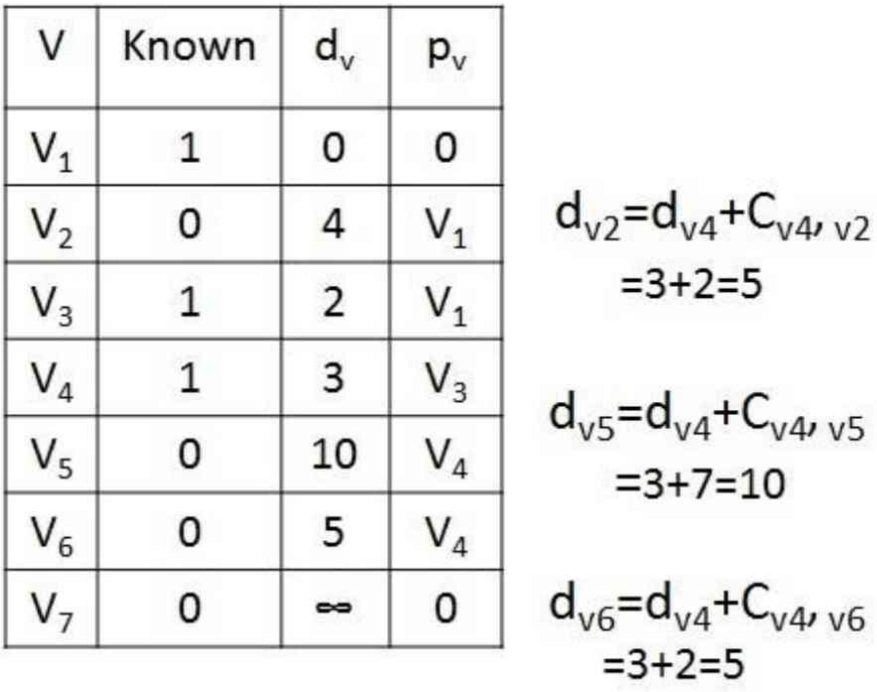
$$d_{v6}=d_{v3}+C_{v3,\,v6}$$
$$=2+10=12$$

Fig 3. After $V_3$ is known

## Dijkstras Algorithm - Step 3

Next, $V_4$ is selected and marked known. The vertices $V_2$, $V_5$ and $V_6$ are adjacent. The vertex $V_2$ is adjacent but not adjusted, because the cost of going via $V_4$ is 3+2=5 which is greater than the previous distance 4. So simply discards the new value. Similarly for $V_6$, the cost is changed to 5.





| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 0 | 4 | $V_1$ |
| $V_3$ | 1 | 2 | $V_1$ |
| $V_4$ | 1 | 3 | $V_3$ |
| $V_5$ | 0 | 10 | $V_4$ |
| $V_6$ | 0 | 5 | $V_4$ |
| $V_7$ | 0 | ∞ | 0 |

$d_{v2}=d_{v4}+C_{v4,\,v2}$
$=3+2=5$

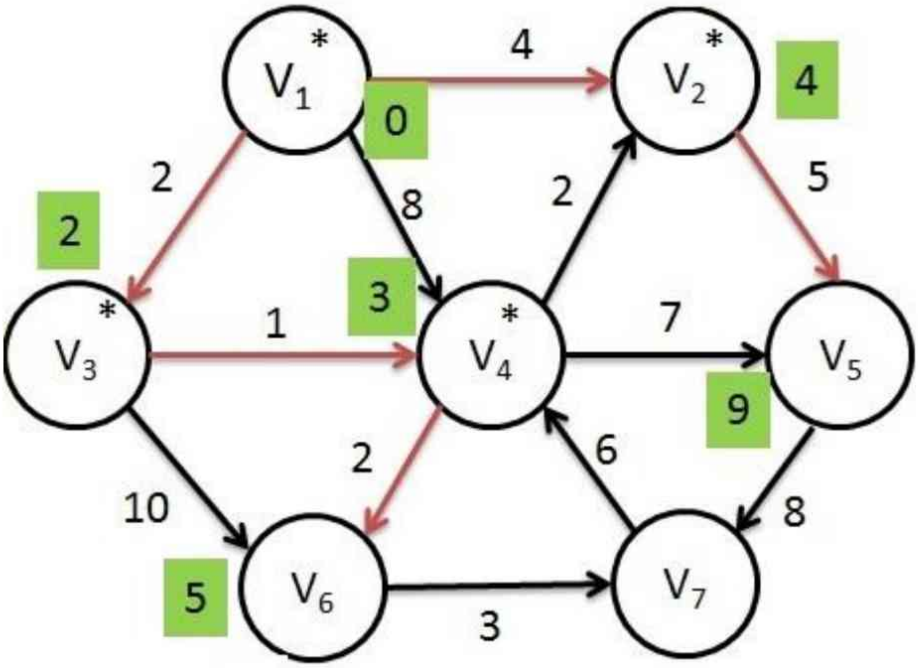
$d_{v5}=d_{v4}+C_{v4,\,v5}$
$=3+7=10$

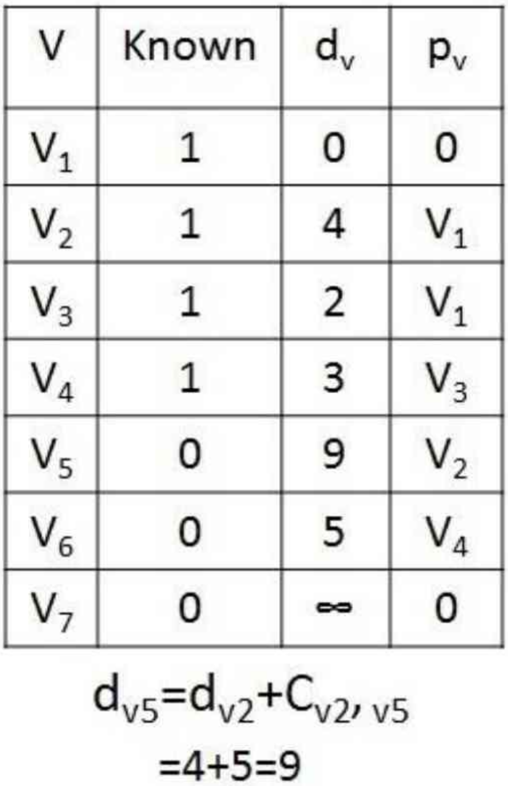
$d_{v6}=d_{v4}+C_{v4,\,v6}$
$=3+2=5$

Fig 4. After $V_4$ is known

## Dijkstras Algorithm - Step 4

Next, $V_2$ is selected and marked known. $V_5$ is the only adjacent vertex which is adjusted to 9(4+5) < 10.





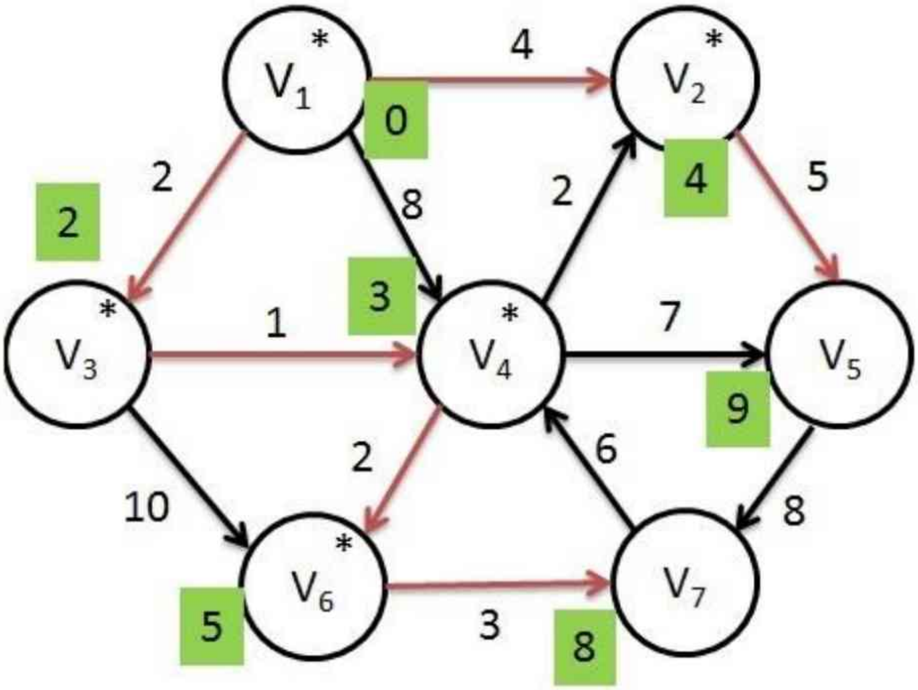
| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 1 | 4 | $V_1$ |
| $V_3$ | 1 | 2 | $V_1$ |
| $V_4$ | 1 | 3 | $V_3$ |
| $V_5$ | 0 | 9 | $V_2$ |
| $V_6$ | 0 | 5 | $V_4$ |
| $V_7$ | 0 | ∞ | 0 |

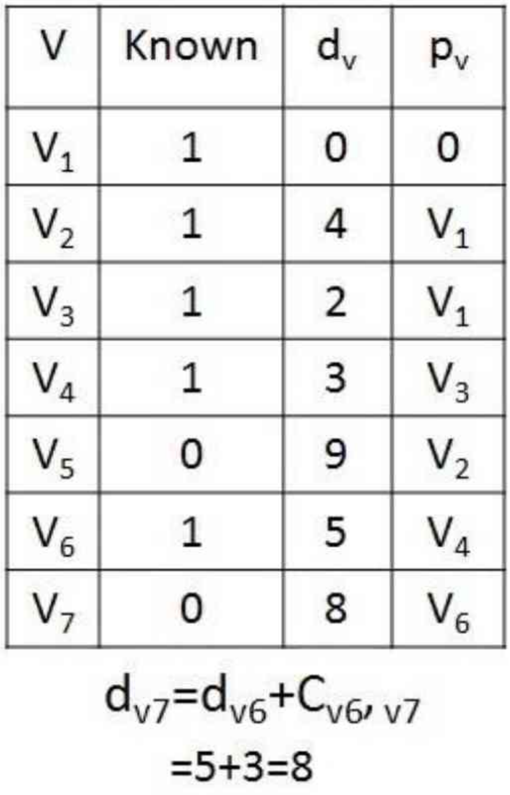
$$d_{v5}=d_{v2}+C_{v2,\,v5}$$
$$=4+5=9$$

Fig 5. After $V_2$ is known

## Dijkstras Algorithm - Step 5

Next, selected vertex is $V_6$ and marked known. $V_7$ is the only adjacent vertex which is adjusted to 8(5+3).





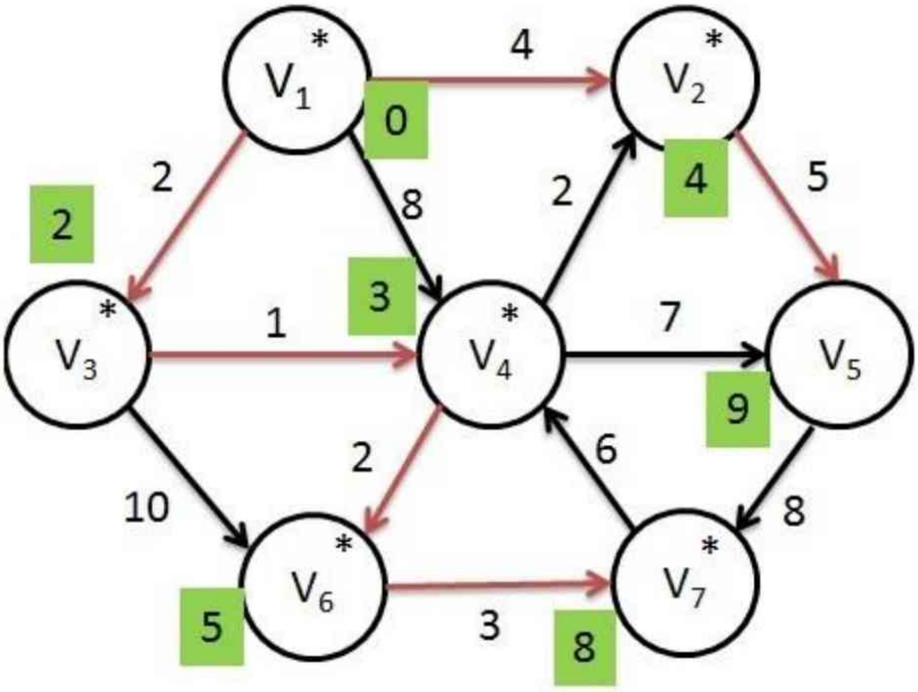
| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 1 | 4 | $V_1$ |
| $V_3$ | 1 | 2 | $V_1$ |
| $V_4$ | 1 | 3 | $V_3$ |
| $V_5$ | 0 | 9 | $V_2$ |
| $V_6$ | 1 | 5 | $V_4$ |
| $V_7$ | 0 | 8 | $V_6$ |

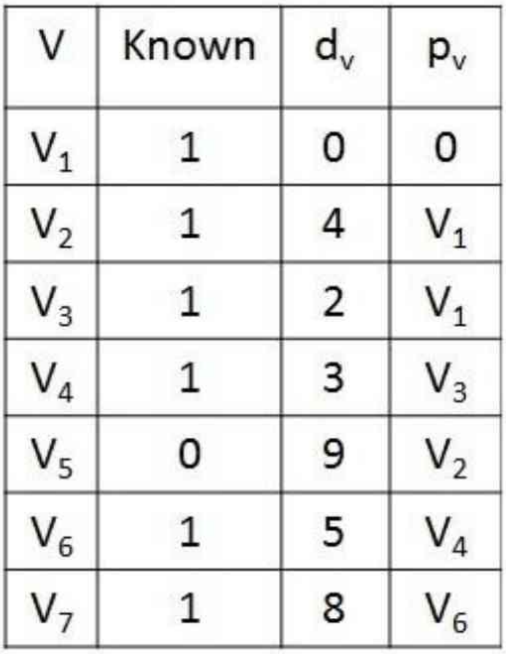
$$d_{v7}=d_{v6}+C_{v6,\,v7}$$
$$=5+3=8$$

Fig 6. After $V_6$ is known

## Dijkstras Algorithm - Step 6

Next, selected vertex is $V_7$ and set known to 1. $V_4$ is adjacent but already known, so no work is done on it.



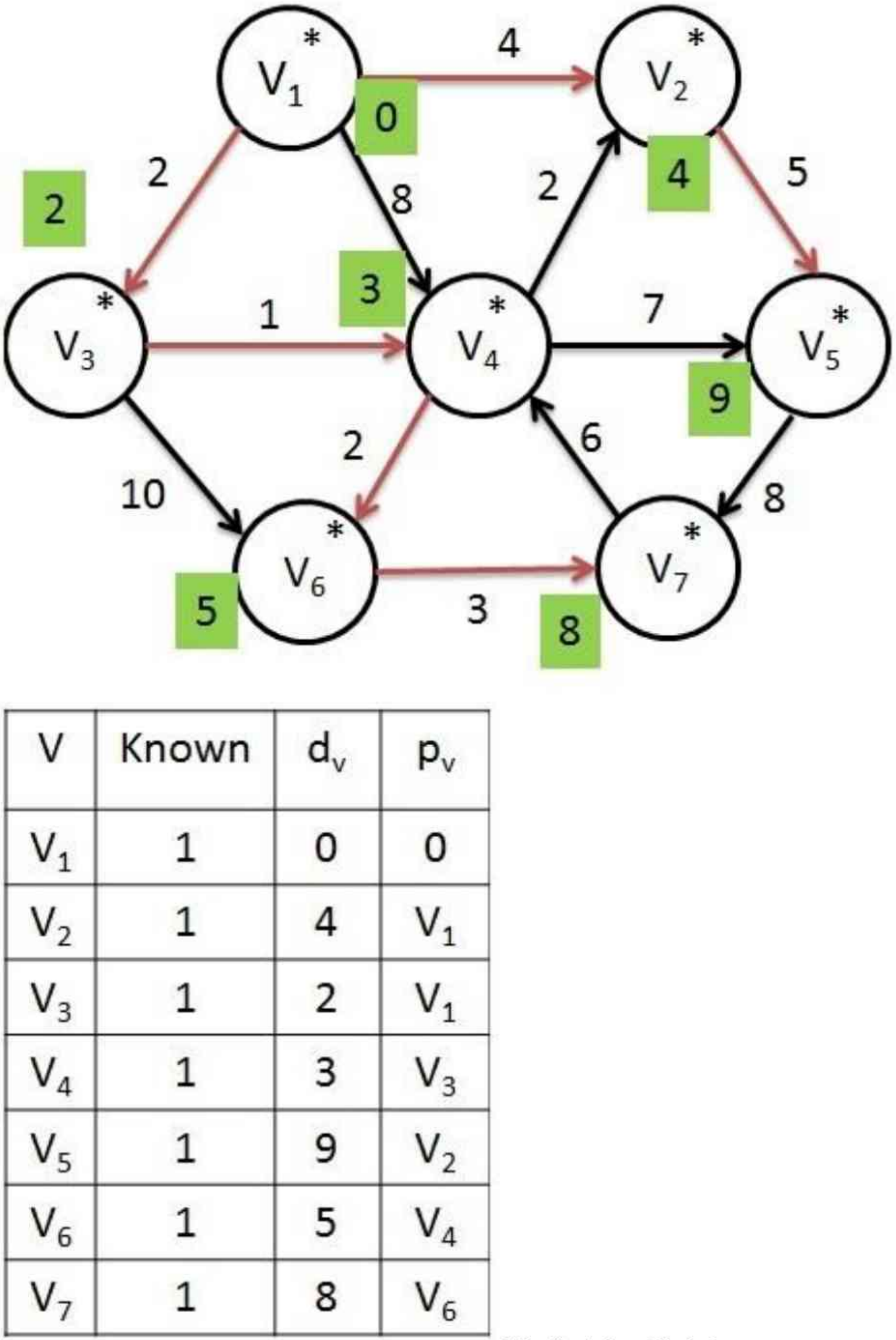


| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 1 | 4 | $V_1$ |
| $V_3$ | 1 | 2 | $V_1$ |
| $V_4$ | 1 | 3 | $V_3$ |
| $V_5$ | 0 | 9 | $V_2$ |
| $V_6$ | 1 | 5 | $V_4$ |
| $V_7$ | 1 | 8 | $V_6$ |

Fig 7. After $V_7$ is known

## Dijkstras Algorithm - Step 7

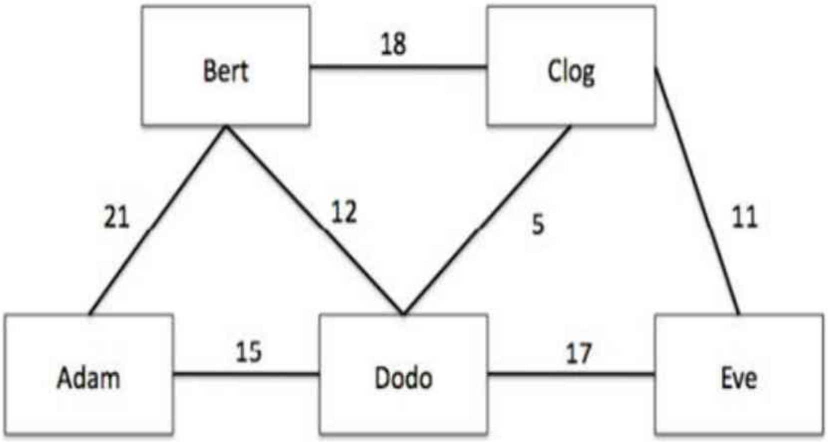
Finally, $V_5$ is selected and marked known. Now the Dijkstra's algorithm is terminated and print out the shortest path from a start vertex to all other vertices.



| V | Known | $d_v$ | $p_v$ |
|---|---|---|---|
| $V_1$ | 1 | 0 | 0 |
| $V_2$ | 1 | 4 | $V_1$ |
| $V_3$ | 1 | 2 | $V_1$ |
| $V_4$ | 1 | 3 | $V_3$ |
| $V_5$ | 1 | 9 | $V_2$ |
| $V_6$ | 1 | 5 | $V_4$ |
| $V_7$ | 1 | 8 | $V_6$ |

Fig 8. After $V_5$ is known

5)The CSE department at Princeton University bought new Sun Fire V210 servers. They decided to run a distance-vector protocol for routing between these servers (even though it is a rather small network). They are currently configured as the picture below, with respective edge costs.



The CSE staff asked for your help. Write down each step of building the distance vector routing table for 'Eve'. You can use abbreviations e.g., 'A' for Adam and 'E' for Eve.

i. Show the initial routing table of server Eve. **(2)**
ii. Show the routing table of the server Eve after 1st and 2nd iteration of the algorithm. **(4)**
iii. Explain how a node transmits routing updates to its neighbors. **(2)**
iv. How the node/link failures are identified? **(2)**

Solution:

i) Initial routing table of Eve (E): (2 marks)

| Destination | cost | Next hop |
|---|---|---|
| C | 11 | |
| D | 19 | |
| A | ∞ | - |
| B | ∞ | - |

Steps:

Initial routing table of Eve is constructed based on the immediate neighbours of Eve's cost and next hop. If immediate neighbours are not there, there cost is considered as ∞.

(ii) Routing table of Eve(E ) after 1st iteration and 2nd iteration (4 marks)

Routing table of Eve(E ) after 1st iteration:

| Destination | cost | Next hop |
|---|---|---|
| C | 11 | |
| D | 19 | |
| A | ∞ | - |
| B | ∞ | - |

E's initial routing table

| To | Cost |
|---|---|
| D | 0 |
| A | 15 |
| C | 15 |
| B | 12 |

Message from D
E to D: cost:17

| Dest | Cost | Next hop |
|---|---|---|
| A | 32 | D |
| B | 29 | D |
| C | 11 | C |
| D | 17 | D |

→ E's routing table after 1's iteration

(1) To A: 17+15=32. (32<∞)
(2) To B: 17+12=29(29<∞)
(3) To C: 17+5=22(11<22). Keep 11
(4) To D: 17+0=17(17=17). Keep 17.

Routing table of Eve(E ) after 2nd iteration:

| Dest | Cost | Next hop |
|---|---|---|
| A | 32 | D |
| B | 29 | D |
| C | 11 | C |
| D | 17 | D |

E's routing table after 1st Iteration

| To | Cost |
|---|---|
| C | 0 |
| B | 18 |
| D | 5 |
| E | 11 |

Message from C
E to C: Cost=11

| Dest | Cost | Next hop |
|---|---|---|
| A | 32 | D |
| B | 29 | D OR C |
| C | 11 | C |
| D | 16 | C |

→ E's routing table after 2nd Iteration

(1)To A: No data about A
(2) To B: 11=18=29. (Same cost)
(3) To C: 11+0=11(same cost)
(4) To D: 11+5=16(16<17.)

Routing updates are done using Bellman-Ford equation:
dx(y)=min(c(x,y)+dv(y))

(iii) 1. Triggered Update (2 marks)
2. Periodic Update

(iv) 1)A node didn't receive periodic update for long time (2 marks)
2) Through probe-ack