UNIT-3
Network Layer

B SAI BABA,M.Tech(Ph.D),VIT,Bhimavaram

Syllabus

- ★ **Network Layer:**
  - Network layer design issues
- ★ **Routing Algorithms:**
  - Optimality Principle
  - Shortest Path Algorithm
  - Distance Vector Routing Algorithm
  - Flooding
  - Link State Routing
  - Hierarchical Routing
  - Broadcast Routing
  - Multicast Routing
- ★ **Congestion Control Algorithms**

Network Layer Design Issues

★ **The Network Layer or Layer 3** of the OSI (Open Systems Interconnection) model is concerned delivery of data packets from the source to the destination **across multiple hops or links.**

★ The design issues can be elaborated as−

1. **Store − and − Forward Packet Switching**
2. **Services to Transport Layer**
3. **Providing Connection Oriented Service**
4. **Providing Connectionless Service**
5. **Routing**
6. **Congestion Control**

Store – and – Forward Packet Switching

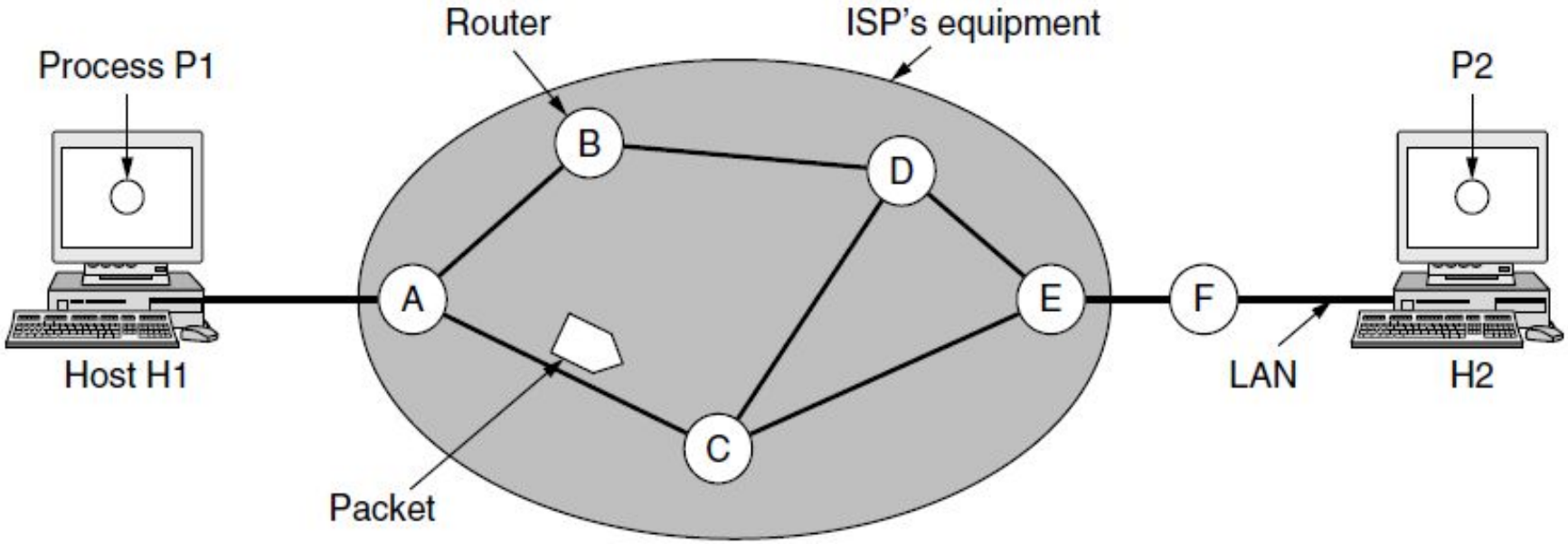


Figure 5-1. The environment of the network layer protocols.

Store − and − Forward Packet Switching

★ The network layer operates in an environment that uses store and forward packet switching.

★ The node which has a packet to send, **delivers it to the nearest router.**

★ The packet is stored in the router until it has fully arrived and its checksum is verified for error detection.

★ Once, this is done, **the packet is forwarded to the next router.**

★ Since, each router needs to store the entire packet before it can forward it to the next hop, the mechanism is called **store − and − forward switching**.

Services to Transport Layer

★ The Network Layer provides service to its immediate upper layer, namely **Transport Layer**, through the network − transport layer interface.

★ **The two types of services provided are −**

1. **Connection − Oriented Service** :

   In this service, **a path is setup between the source and the destination**, and all the data packets belonging to a message are routed along this path.

2. **Connectionless Service :**

   In this service, each packet of the message is considered as **an independent entity** and is **individually routed from the source to the destination**.

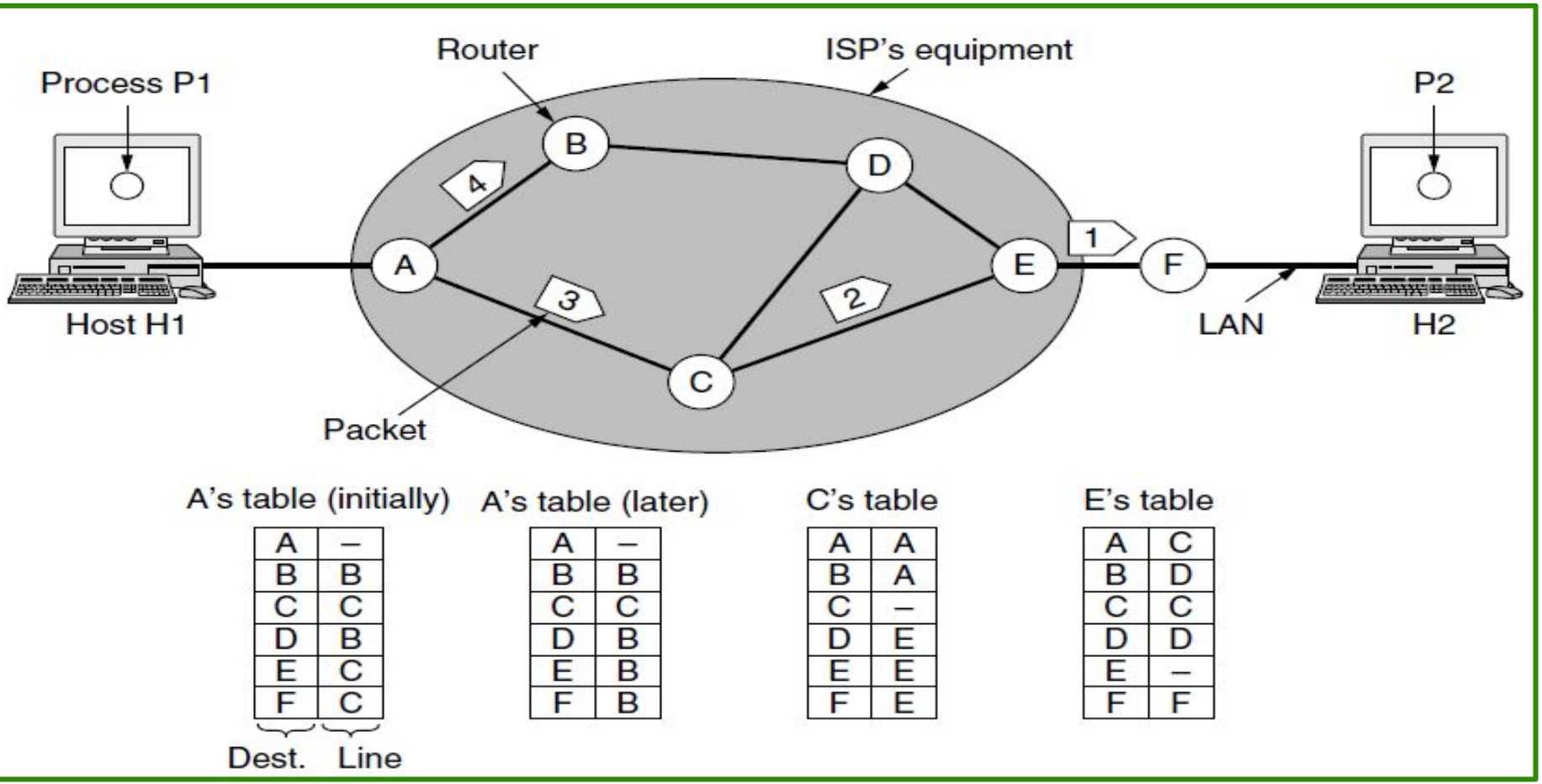
Providing Connectionless Service

★ **If connectionless service is offered**, packets are injected into the network individually and routed independently of each other.**No advance setup is needed.**

★ In this context, the packets are frequently called **datagrams** and the network is called **a datagram network**.

★ In this section, we will examine datagram networks :

Figure 5-2. Routing within a datagram network



A's table (initially)

Dest.	Line
A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

C's table

A	A
B	A
C	–
D	E
E	E
F	E

E's table

A	C
B	D
C	C
D	D
E	–
F	F

- ★ Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A using some point-to-point protocol.
- ★ At this point the ISP takes over. Every **router** has **an internal table** telling it where to send packets for each of the possible destinations.
- ★ **Each table entry is a pair** consisting of **a destination** and **the outgoing line to use for that destination**. Only directly connected lines can be used.
- ★ For example, in Fig. 5-2, **A has only two outgoing lines—to B and to C**—so every incoming packet must be sent to one of these routers, even if the ultimate destination is to some other router.
- ★ A's initial routing table is shown in the figure under the label ''initially.''

- ★ At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link and had their checksums verified. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame.
- ★ Packet 1 is then forwarded to E and then to F. When it gets to F, **it is sent within a frame** over the LAN to H2. **Packets 2 and 3** follow the same route.
- ★ However, something different happens to **packet 4**.
- ★ When it gets to A it is sent to router B, even though it is also destined for F.
- ★ For some reason, A decided to send packet 4 via a different route than that of the first three packets. Perhaps it has learned of **a traffic jam** somewhere along **the ACE path** and updated its routing table, as shown under the label "later."
- ★ The algorithm that manages the tables and makes the routing decisions is called **the routing algorithm.**

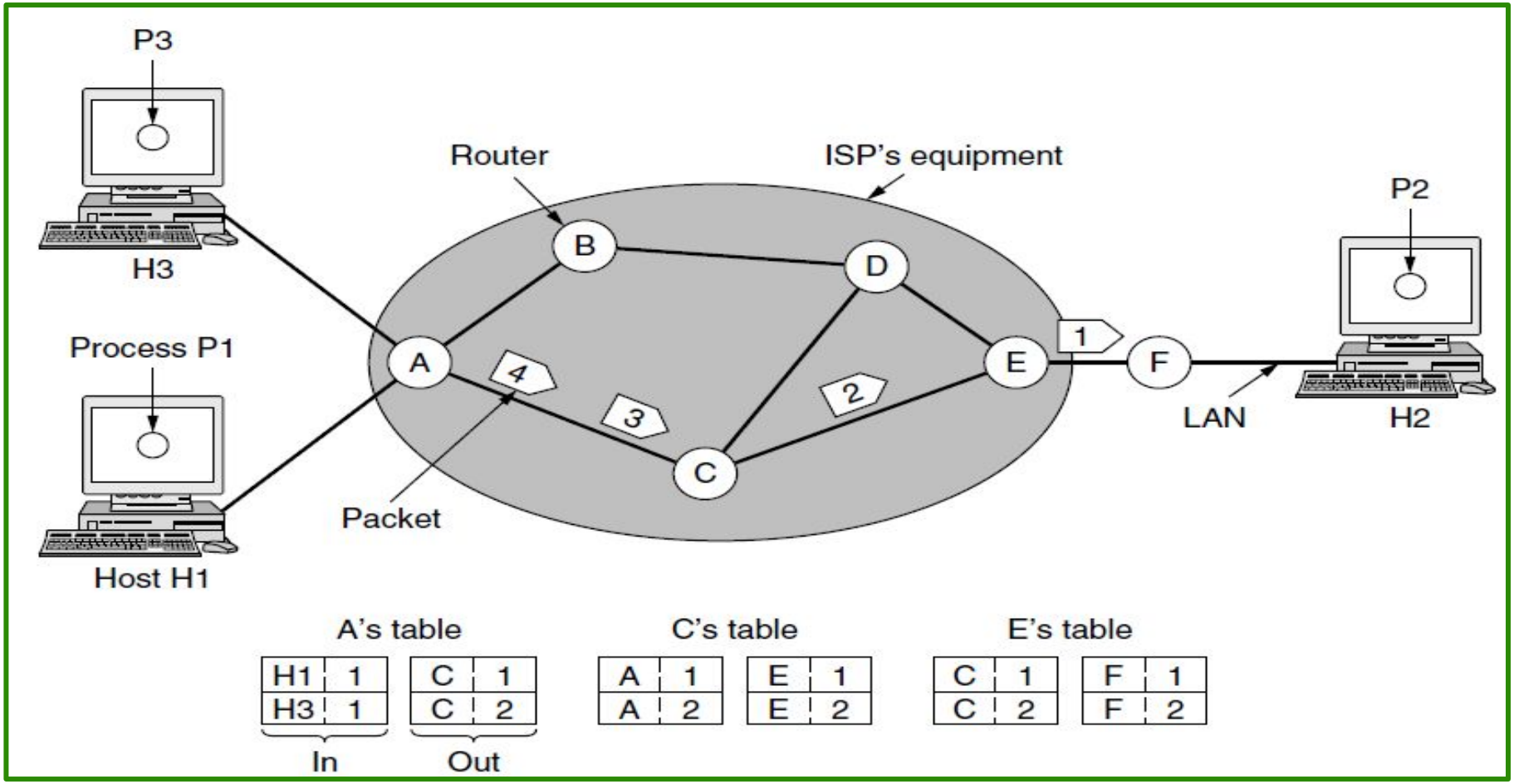
Providing Connection Oriented Service

★ If connection-oriented service is used, **a path** from the source router all the way to the destination router **must be established before any data packets can be sent**.

★ This connection is called a **VC (virtual circuit),**

★ The network is called **a virtual-circuit network.**

Figure 5-3. Routing within a virtual-circuit network



- ★ The idea behind virtual circuits is **to avoid having to choose a new route** for every packet sent, as in Fig. 5-2.
- ★ Instead, when a connection is established,a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers.
- ★ That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works.
- ★ When the connection is released, the virtual circuit is also terminated.
- ★ **With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.**
- ★ As an example, consider the situation shown in Fig. 5-3. Here, host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables.

- ★ The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1.
- ★ Similarly, the first entry at C routes the packet to E, also with connection identifier 1.
- ★ Now let us consider what happens if H3 also wants to establish a connection to H2.
- ★ It chooses connection identifier 1 (because it is initiating the connection and this is it's only connection) and tells the network to establish the virtual circuit.This leads to the second row in the tables.
- ★ Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection.
- ★ Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets. In some contexts, this process is called **label switching.**

Comparison of Virtual Circuit and Datagram Network

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithms

Routing Algorithm

★ In order to transfer the packets from source to the destination, the network layer must determine **the best route** through which packets can be transmitted.

★ Whether the network layer provides **datagram service or virtual circuit service**, the main job of the network layer is to provide **the best route**. The routing protocol provides this job.

★ The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the **"least-cost path"** from source to the destination.

★ Routing is the process of forwarding the packets from source to the destination but **the best route to send the packets is determined by the routing algorithm.**

Optimality Principle

Introduction :

★ A general statement is made about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle(Bellman,1975).

Statement of the optimality principle :

★ It states that if the router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route. Call the route from I to J, r1 and the rest of the route r2. it could be concatenated with r1 to improve the route from I to K, contradicting our statement that r1r2 is optimal only if a route better than r2 existed from J to K.

Introduction :

★ A general statement is made about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle(Bellman,1975).

Explanation:

★ The purpose of a routing algorithm at a router is to decide which output line an incoming packet should go.

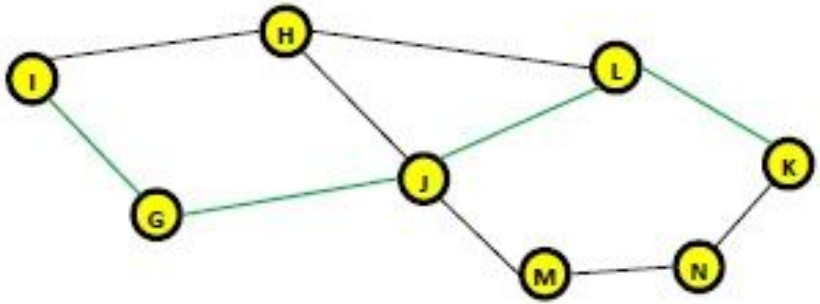
★ **The optimal path** from a particular router to another may be the **least cost path, the least distance path, the least time path, the least hops path** or a combination of any of the above.

The optimality principle can be logically proved as follows −

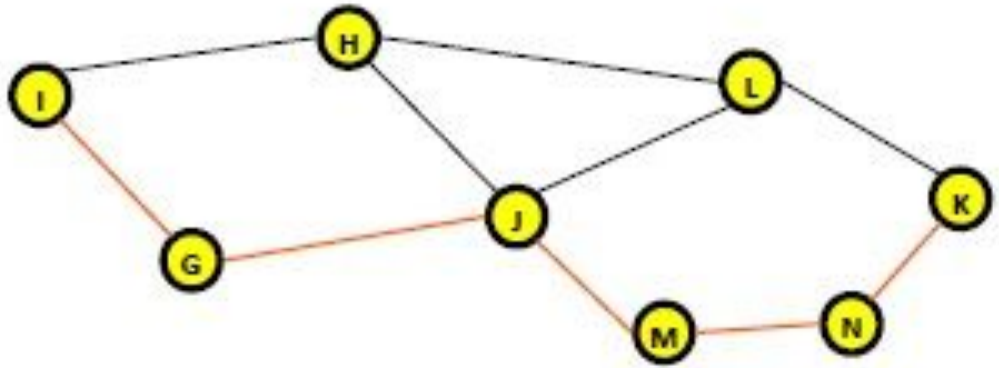
If the router J is on the optimal path from router I to router K, If a better route could be found between router J and router K, the path from router I to router K via J would be updated via this route. Thus, the optimal path from J to K will again lie on the optimal path from I to K.

Example

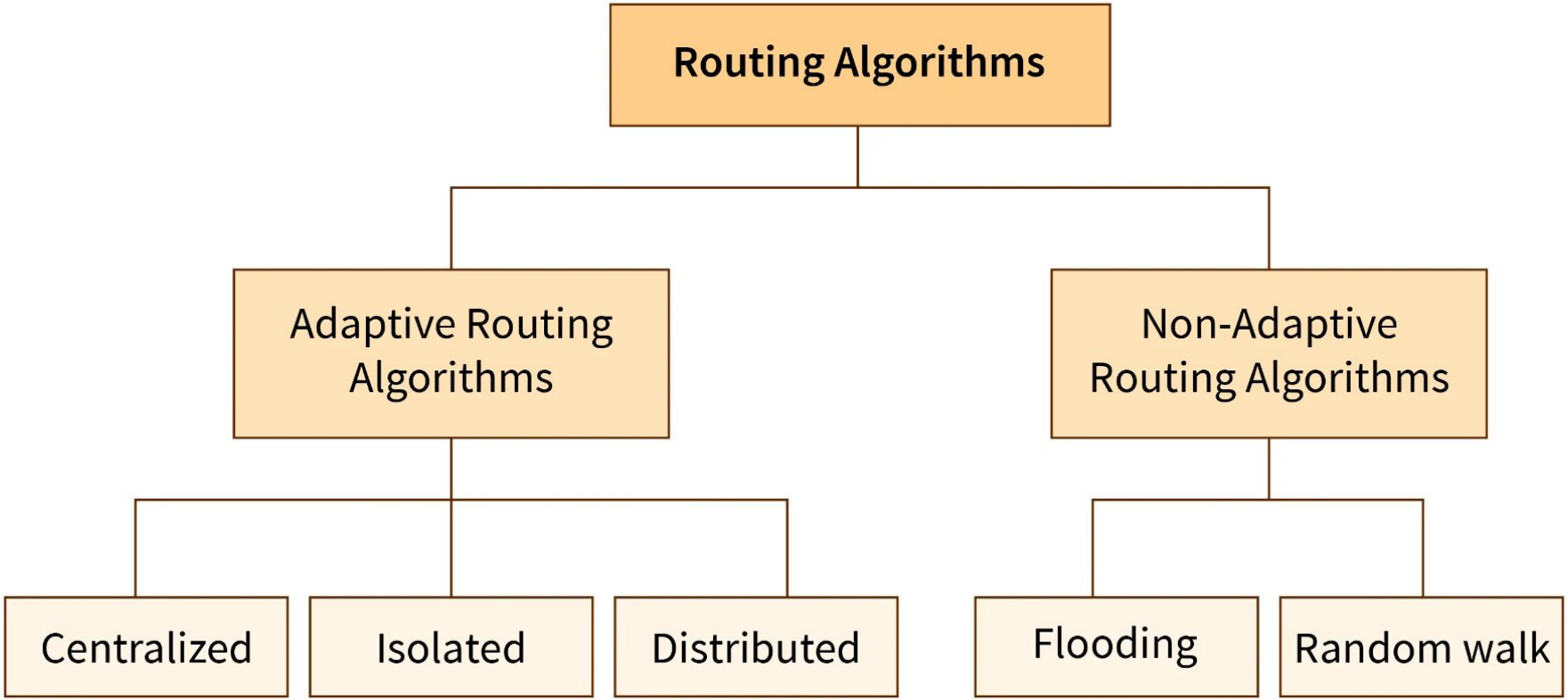
Consider a network of routers, {G, H, I, J, K, L, M, N} as shown in the figure. Let the optimal route from I to K be as shown via the green path, i.e. via the route I-G-J-L-K. According to the optimality principle, the optimal path from J to K with be along the same route, i.e. J-L-K.



Now, suppose we find a better route from J to K is found, say along J-M-N-K. Consequently, we will also need to update the optimal route from I to K as I-GJ- M-N-K, since the previous route ceases to be optimal in this situation. This new optimal path is shown line orange lines in the following figure −



Types of Routing Algorithm



Adaptive Routing Algorithm

★ Adaptive routing algorithm is also called **a dynamic routing algorithm**.

★ In this algorithm, **the routing decisions** are made based on **network traffic and topology.**

★ The parameters which are used in adaptive routing algorithms are **distance, hop, estimated transit time.**

★ The adaptive routing algorithm is of **three types** –

- **Centralized algorithm**
- **Isolation algorithm**
- **Distributed algorithm**

Centralized algorithm:

★ In centralized routing, **one centralized node** has the total network information and takes the routing decisions.

★ It finds the least-cost path between source and destination nodes by using **global knowledge about the network.** So, it is also known as global routing algorithm.

★ The advantage of this routing is that only the central node is required to store network information and so the resource requirement of the other nodes may be less.

★ Eg: **Link state routing algorithm**

Isolated algorithm:

★ This algorithm procures the routing information by using **local information** instead of gathering information from other nodes.

Distributed algorithm:

★ This is a **decentralized algorithm** where each node receives information from its neighbouring nodes and takes the decision based upon the received information.

★ The least-cost path between source and destination is computed **iteratively in a distributed manner**.

★ An advantage is that each node can **dynamically change routing decisions** based upon the changes in the network.

★ Example : **distance vector routing algorithm.**

Non-adaptive Routing Algorithm

- ★ Non-adaptive routing algorithm is also called **a static routing algorithm**.
- ★ In a non-adaptive routing algorithm, the routing decisions are **not made based on network traffic and topology.**
- ★ This algorithm is used by static routing.
- ★ Non-adaptive routing algorithms are **simple** as compared to Adaptive routing algorithms in terms of **complexity.**
- ★ The non-adaptive routing algorithm is of **two types** –
  - **Flooding**
  - **Random walks**

Shortest Path Algorithm

Introduction:

★ **Dijkstra's algorithm and the Bellman-Ford algorithm** are two different algorithms used to **find the shortest path** in a weighted graph.

Dijkstra's algorithm	Bellman-Ford algorithm
It works only for graphs with **non-negative edge weights**.	It works for graphs with both **non-negative and negative edge weights.**

Dijkstra's algorithm:

★ The shortest path algorithm was first introduced by **Edsger W. Dijkstra in 1956.**

★ This algorithm, commonly known as Dijkstra's algorithm, is used to find the shortest path between nodes in a graph, particularly in **weighted graphs** where each **edge has a non-negative weight.**

★ Dijkstra's algorithm is widely used in various applications, including **computer networking, transportation systems**, and more, to find the shortest path from a source node to all other nodes in the graph.

Procedure of Shortest Path Algorithm

★ Initially mark all nodes (except source) with infinite distance.
- working node = source node
- Sink node = destination node

★ While the working node is not equal to the sink

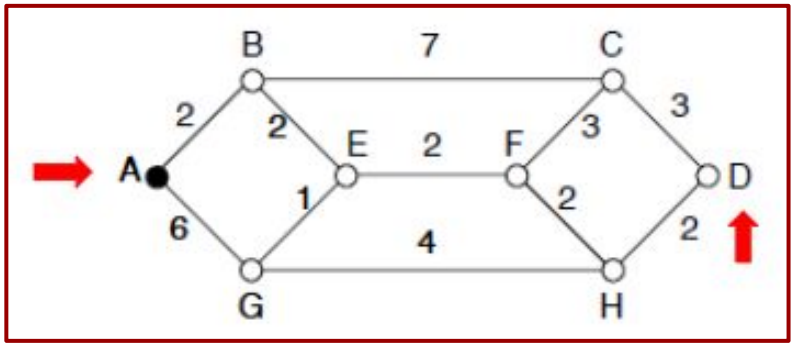
1. Mark the working node as permanent.

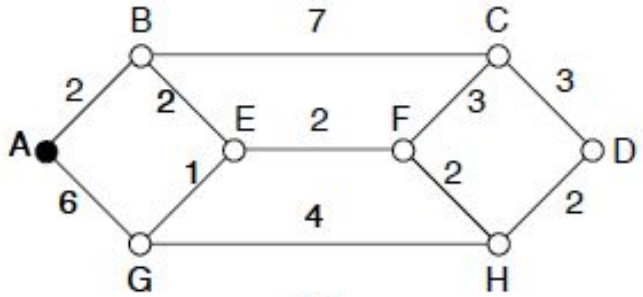
2. Examine all adjacent nodes in turn

- If the sum of label on working node plus distance from working node to adjacent node is less than current labeled distance on the adjacent node, this implies a shorter path. Re label the distance on the adjacent node and label it with the node from which the probe was made.
- 3. Examine all tentative nodes (not just adjacent nodes) and mark the node with the smallest labeled value as permanent. This node becomes the new working node.

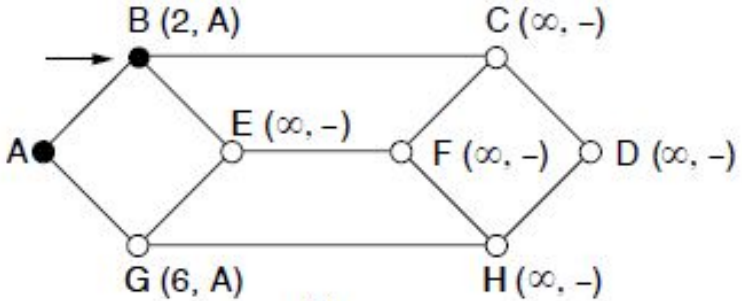
★ Reconstruct the path backwards from sink to source

- The idea is to build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link.
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- The concept of a shortest path deserves some explanation.
  - One way of measuring path length is the **number of hops**. Using this metric, the paths ABC and ABE in Fig. are equally long.
  - Another metric is **the geographic distance in kilometers**, in which case ABC is clearly much longer than ABE

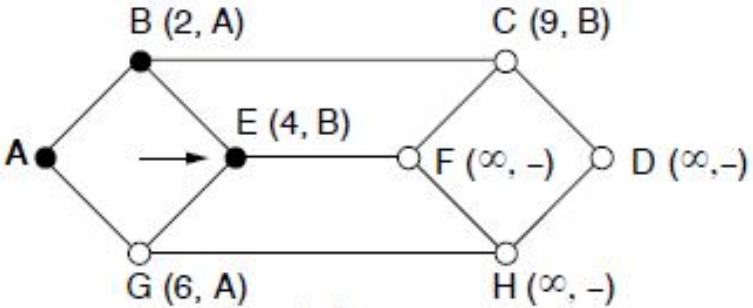



B
7
C
2
2
E
2
F
3
3
A
D
1
2
6
4
2
G
H

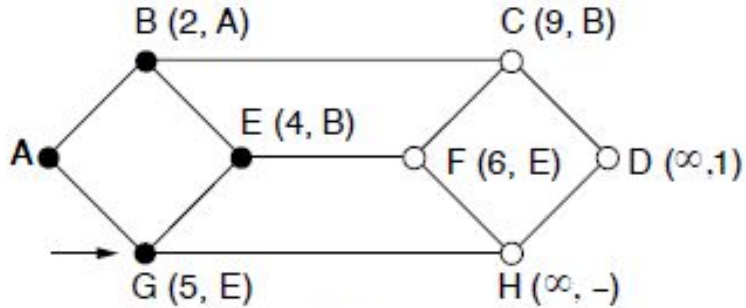
(a)


B (2, A)
C (∞, –)
E (∞, –)
A
F (∞, –)
D (∞, –)
G (6, A)
H (∞, –)

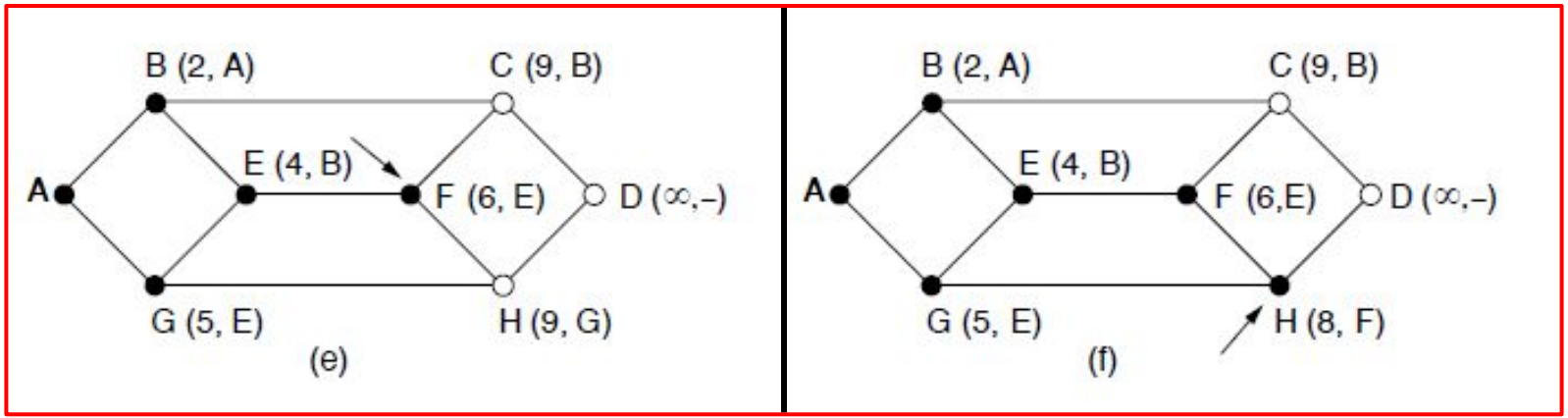
(b)


B (2, A)
C (9, B)
E (4, B)
A
F (∞, –)
D (∞,–)
G (6, A)
H (∞, –)

(c)


B (2, A)
C (9, B)
E (4, B)
A
F (6, E)
D (∞,1)
G (5, E)
H (∞, –)

(d)

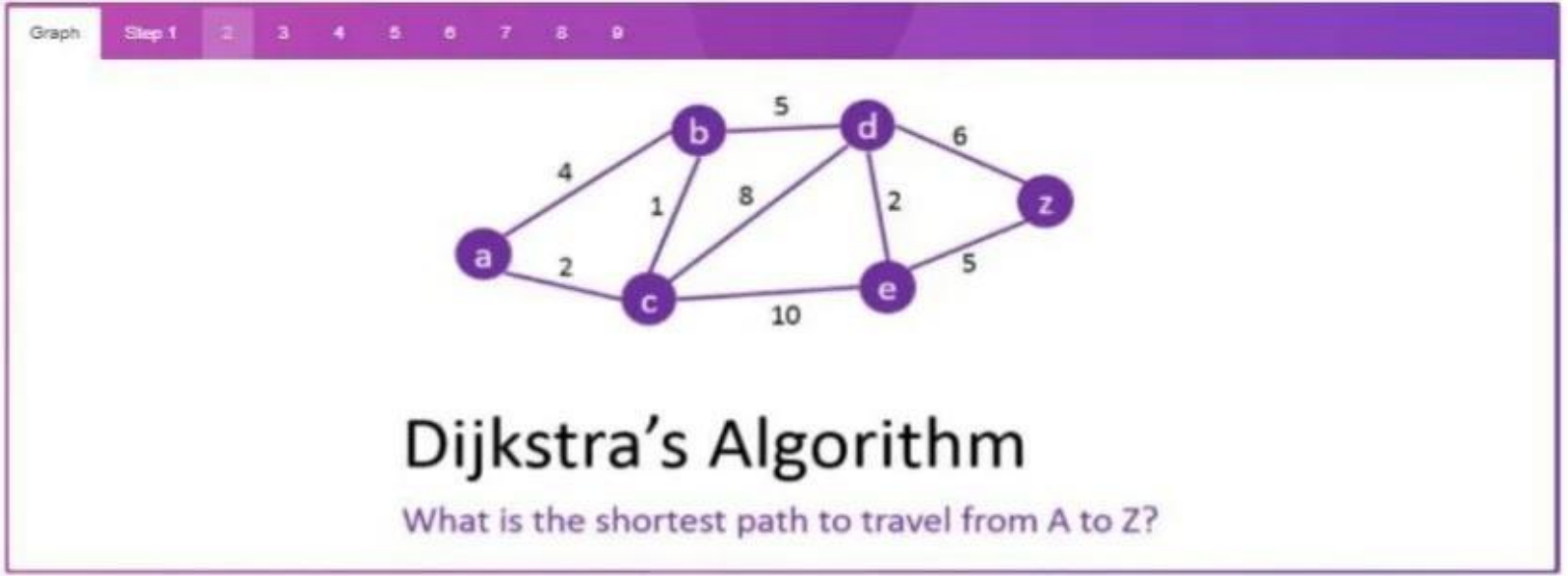


- **The first six steps used in computing the shortest path from A to D.**
- **The arrows indicate the working node.**

Example :2

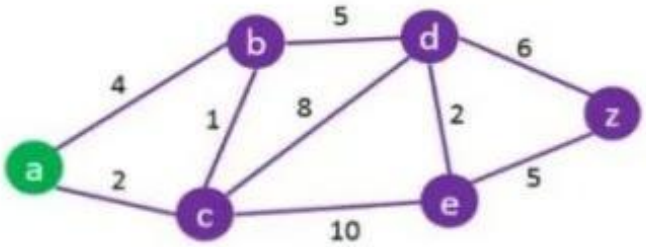
Dijkstra Algorithm (Shortest Path Algorithm)

- How Routers Decide Shortest Path Using dijkstra Algorithm ?



Step 1

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

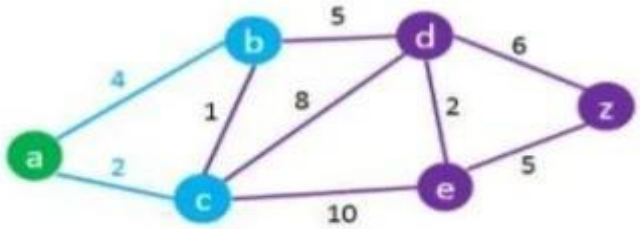


Node	Status	Shortest Distance From A	Previous Node
A	Current Node	0	
B		∞	
C		∞	
D		∞	
E		∞	
Z		∞	

Start by setting the starting node (A) as the current node.

Step 2

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

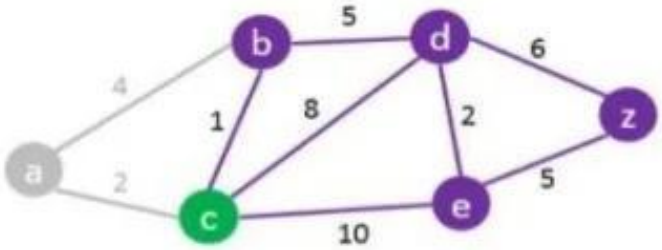


Node	Status	Shortest Distance From A	Previous Node
A	Current Node	0	
B		∞ 4	A
C		∞ 2	A
D		∞	
E		∞	
Z		∞	

Check all the nodes connected to A and update their **"Distance from A"** and set their **"previous node"** to "A".

Step 3

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

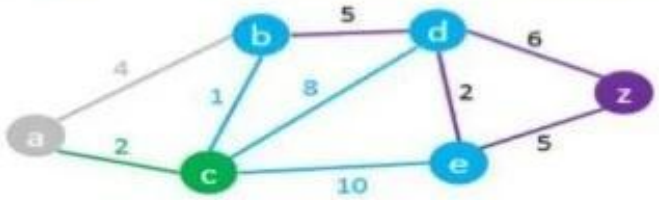


Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B		∞ 4	A
C	Current Node	∞ 2	A
D		∞	
E		∞	
Z		∞	

Set the current node (A) to **"visited"** and use the closest unvisited node to A as the **current node** (e.g. in this case: Node C).

Step 4

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9



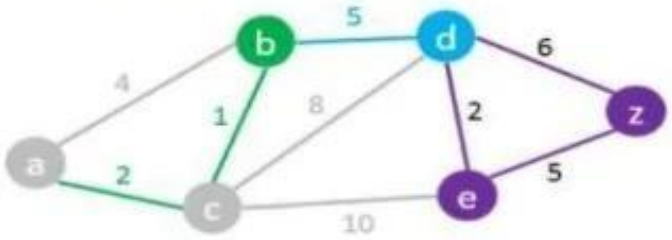
Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B		4 2+1=3	C
C	Current Node	2	A
D		∞ 2+8=10	C
E		∞ 2+10=12	C
Z		∞	

Check all unvisited nodes connected to the current node and add the distance from A to C to all distances from the connected nodes. Replace their values only if the new distance is lower than the previous one.

C -> B: 2 + 1 = 3 < 4 – Change Node B
C -> D: 2 + 8 = 10 < ∞ – Change Node D
C -> E: 2 + 10 = 12 < ∞ – Change Node E

Step 5

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

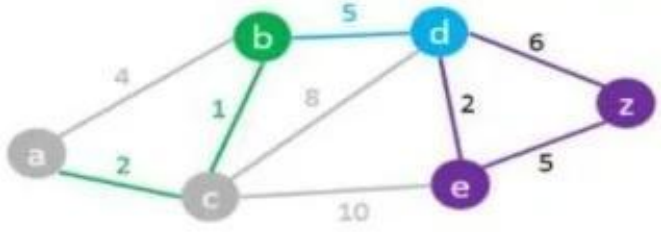


Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Current Node	3	C
C	Visited Node	2	A
D		10	C
E		12	C
Z		∞	

Set the current node C status to Visited.
We then repeat the same process always picking the closest unvisited node to A as the current node.
In this case node B becomes the current node.

Step 6

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9



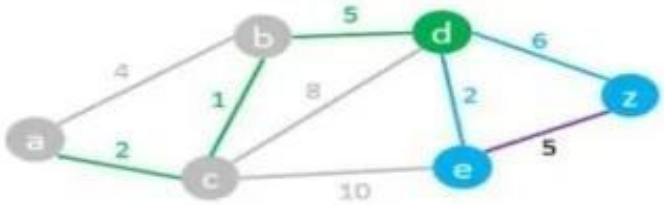
Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Current Node	3	C
C	Visited Node	2	A
D		~~10~~ 3+5=8	B
E		12	C
Z		∞	

B -> D 3+5 = 8 < 10 – Change Node D

Next "Current Node" will be D as it has the shortest distance from A amongst all unvisited nodes.

Step 7

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9



Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Current Node	8	B
E		~~12~~ 8 + 2 = 10	D
Z		∞ 8 + 6 = 14	D

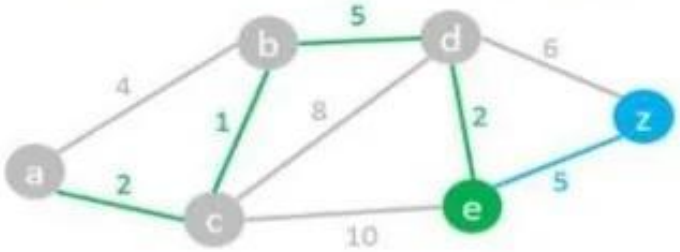
D -> E 8+2 = 10 < 12 – Change Node E
D -> Z 8+6 = 14 < ∞ – Change Node Z

We found a path from A to Z but it may not be the shortest one yet. So we need to carry on the process.

Next **"Current Node"**: E

Step 8

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

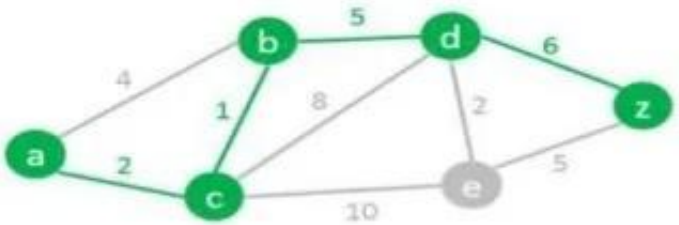


Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Visited Node	8	B
E	Current Node	10	D
Z		14 10 +5 ~~=15~~	D

E -> Z 10+5 = 15 > 14 – We do not change node Z.

Step 9

Graph | Step 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9



Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Visited Node	8	B
E	Visited Node	10	D
Z	Current Node	14	D

We found the shortest path from A to Z.
Read the path from Z to A using the previous node column:
Z > D > B > C > A
So the Shortest Path is:
A – C – B – D – Z with a length of 14