Routing

* A Router is a process of selecting path along which the data can be transferred from source to the destination. Routing is performed by a special device known as a router.
* A Router works at the network layer in the OSI model and internet layer in TCP/IP model
* A router is a networking device that forwards the packet based on the information available in the packet header and forwarding table.
* The routing algorithms are used for routing the packets. The routing algorithm is nothing but a software responsible for deciding the optimal path through which packet can be transmitted.
* The routing protocols use the metric to determine the best path for the packet delivery.

The metric is the standard of measurement such as hop count, bandwidth, delay, current load on the path, etc. used by the routing algorithm to determine the optimal path to the destination.

* The routing algorithm initializes and maintains the routing table for the process of path

determination.

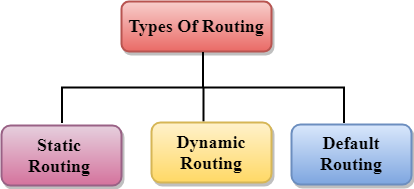
## The most common metric values are given below:

* **Hop count:** Hop count is defined as a metric that specifies the number of passes through internetworking devices such as a router, a packet must travel in a route to move from source to the destination. If the routing protocol considers the hop as a primary metric value, then the path with the least hop count will be considered as the best path to move from source to the destination.
* **Delay:** It is a time taken by the router to process, queue and transmit a datagram to an interface. The protocols use this metric to determine the delay values for all the links along the path end-to-end. The path having the lowest delay value will be considered as the best path.
* **Bandwidth:** The capacity of the link is known as a bandwidth of the link. The bandwidth is measured in terms of bits per second. The link that has a higher transfer rate like gigabit is preferred over the link that has the lower capacity like 56 kb. The protocol will determine the bandwidth capacity for all the links along the path, and the overall higher bandwidth will be considered as the best route.
* **Load:** Load refers to the degree to which the network resource such as a router or network link is busy. A Load can be calculated in a variety of ways such as CPU utilization, packets processed per second. If the traffic increases, then the load value will also be increased. The load value changes with respect to the change in the traffic.
* **Reliability:** Reliability is a metric factor may be composed of a fixed value. It depends on the network links, and its value is measured dynamically. Some networks go down more often than others. After network failure, some network links repaired more easily than other network links. Any reliability factor can be considered for the assignment of reliability ratings, which are generally numeric values assigned by the system administrator.

Types of Routing

Routing can be classified into three categories:

* Static Routing
* Default Routing
* Dynamic Routing



Static Routing

* Static Routing is also known as Nonadaptive Routing.
* It is a technique in which the administrator manually adds the routes in a routing table.
* A Router can send the packets for the destination along the route defined by the administrator.
* In this technique, routing decisions are not made based on the condition or topology of the networks

Advantages Of Static Routing

Following are the advantages of Static Routing:

* **No Overhead:** It has ho overhead on the CPU usage of the router. Therefore, the cheaper router can be used to obtain static routing.
* **Bandwidth:** It has not bandwidth usage between the routers.
* **Security:** It provides security as the system administrator is allowed only to have control over the routing to a particular network.

Disadvantages of Static Routing:

Following are the disadvantages of Static Routing:

* For a large network, it becomes a very difficult task to add each route manually to the routing table.
* The system administrator should have a good knowledge of a topology as he has to add each route manually.

Default Routing

* Default Routing is a technique in which a router is configured to send all the packets to the same hop device, and it doesn't matter whether it belongs to a particular network or not. A Packet is transmitted to the device for which it is configured in default routing.
* Default Routing is used when networks deal with the single exit point.
* It is also useful when the bulk of transmission networks have to transmit the data to the same hp device.
* When a specific route is mentioned in the routing table, the router will choose the specific route rather than the default route. The default route is chosen only when a specific route is not mentioned in the routing table.

Dynamic Routing

* It is also known as Adaptive Routing.
* It is a technique in which a router adds a new route in the routing table for each packet in response to the changes in the condition or topology of the network.
* Dynamic protocols are used to discover the new routes to reach the destination.
* In Dynamic Routing, RIP and OSPF are the protocols used to discover the new routes.
* If any route goes down, then the automatic adjustment will be made to reach the destination.

## The Dynamic protocol should have the following features:

* All the routers must have the same dynamic routing protocol in order to exchange the routes.
* If the router discovers any change in the condition or topology, then router broadcast this information to all other routers.

Advantages of Dynamic Routing:

* It is easier to configure.
* It is more effective in selecting the best route in response to the changes in the condition or topology.

Disadvantages of Dynamic Routing:

* It is more expensive in terms of CPU and bandwidth usage.
* It is less secure as compared to default and static routing.

## Difference between Static and Dynamic Routing:

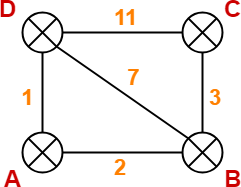
|  |  |  |
| --- | --- | --- |
| S.NO | Static Routing | Dynamic Routing |
| 1. | In static routing routes are user-defined. | In dynamic routing, routes are updated according to the  topology. |
| 2. | Static routing does not use  complex routing algorithms. | Dynamic routing uses  complex routing algorithms. |
| 3. | Static routing provides high  or more security. | Dynamic routing provides  less security. |
| 4. | Static routing is manual. | Dynamic routing is  automated. |
| 5. | Static routing is  implemented in small networks. | Dynamic routing is  implemented in large networks. |
| 6. | In static routing, additional resources are not required. | In dynamic routing,  additional resources are required. |
| 7. | In static routing, failure of  the link disrupts the rerouting. | In dynamic routing, failure  of the link does not interrupt the rerouting. |
| 8. | Less Bandwidth is required  in Static Routing. | More Bandwidth is required  in Dynamic Routing. |

|  |  |  |
| --- | --- | --- |
| 9. | Static Routing is difficult to  configure. | Dynamic Routing is easy to  configure. |
| 10. | Another name for static routing is non-adaptive  routing. | Another name for dynamic routing is adaptive routing. |

**Distance Vector Routing Example-**

Consider-

* There is a network consisting of 4 routers.
* The weights are mentioned on the edges.
* Weights could be distances or costs or delays.



**Step-01:**

Each router prepares its routing table using its local knowledge. Routing table prepared by each router is shown below-

**At Router A-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 0 | A |
| B | 2 | B |
| C | ∞ | – |

|  |  |  |
| --- | --- | --- |
| D | 1 | D |

**At Router B-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 2 | A |
| B | 0 | B |
| C | 3 | C |
| D | 7 | D |

**At Router C-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | ∞ | – |
| B | 3 | B |
| C | 0 | C |
| D | 11 | D |

**At Router D-**

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 1 | A |
| B | 7 | B |
| C | 11 | C |
| D | 0 | D |

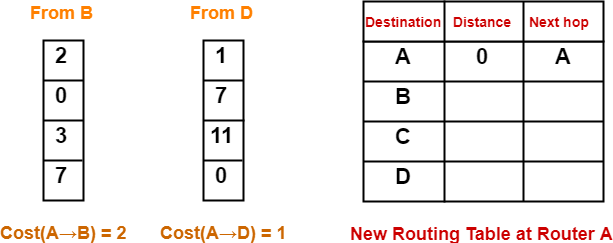
**Step-02:**

* Each router exchanges its distance vector obtained in Step-01 with its neighbors.
* After exchanging the distance vectors, each router prepares a new routing table.

This is shown below-

**At Router A-**

* Router A receives distance vectors from its neighbors B and D.
* Router A prepares a new routing table as-



* Cost of reaching destination B from router A = min { 2+0 , 1+7 } = 2 via B.
* Cost of reaching destination C from router A = min { 2+3 , 1+11 } = 5 via B.
* Cost of reaching destination D from router A = min { 2+7 , 1+0 } = 1 via D.

**Explanation For Destination B**

* Router A can reach the destination router B via its neighbor B or neighbor D.
* It chooses the path which gives the minimum cost.
* Cost of reaching router B from router A via neighbor B = Cost (A→B) + Cost (B→B)= **2 + 0** = 2
* Cost of reaching router B from router A via neighbor D = Cost (A→D) + Cost (D→B) = **1 + 7** = 8
* Since the cost is minimum via neighbor B, so router A chooses the path via B.
* It creates an entry (2, B) for destination B in its new routing table.
* Similarly, we calculate the shortest path distance to each destination router at every router.

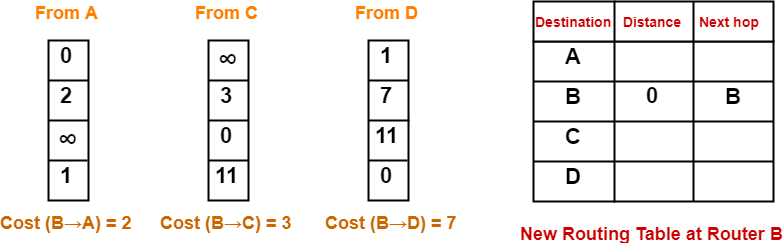
Thus, the new routing table at router A is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 0 | A |
| B | 2 | B |

|  |  |  |
| --- | --- | --- |
| C | 5 | B |
| D | 1 | D |

**At Router B-**

* Router B receives distance vectors from its neighbors A, C and D.
* Router B prepares a new routing table as-



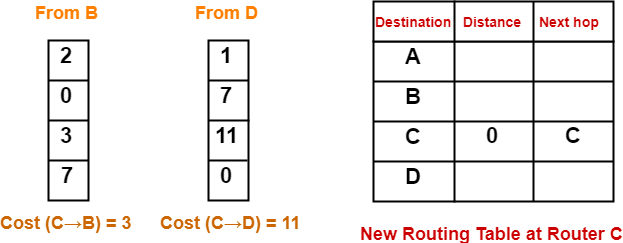
* Cost of reaching destination A from router B = min { 2+0 , 3+∞ , 7+1 } = 2 via A.
* Cost of reaching destination C from router B = min { 2+∞ , 3+0 , 7+11 } = 3 via C.
* Cost of reaching destination D from router B = min { 2+1 , 3+11 , 7+0 } = 3 via A.

Thus, the new routing table at router B is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 2 | A |
| B | 0 | B |
| C | 3 | C |
| D | 3 | A |

**At Router C-**

* Router C receives distance vectors from its neighbors B and D.
* Router C prepares a new routing table as-



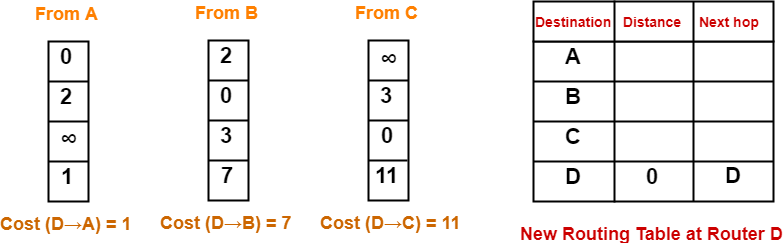
* Cost of reaching destination A from router C = min { 3+2 , 11+1 } = 5 via B.
* Cost of reaching destination B from router C = min { 3+0 , 11+7 } = 3 via B.
* Cost of reaching destination D from router C = min { 3+7 , 11+0 } = 10 via B.

Thus, the new routing table at router C is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 5 | B |
| B | 3 | B |
| C | 0 | C |
| D | 10 | B |

**At Router D-**

* Router D receives distance vectors from its neighbors A, B and C.
* Router D prepares a new routing table as-



* Cost of reaching destination A from router D = min { 1+0 , 7+2 , 11+∞ } = 1 via A.
* Cost of reaching destination B from router D = min { 1+2 , 7+0 , 11+3 } = 3 via A.
* Cost of reaching destination C from router D = min { 1+∞ , 7+3 , 11+0 } = 10 via B.

Thus, the new routing table at router D is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 1 | A |
| B | 3 | A |
| C | 10 | B |
| D | 0 | D |

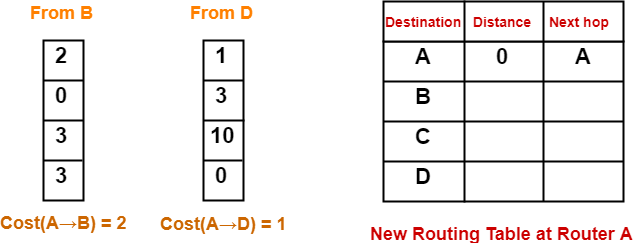
**Step-03:**

* Each router exchanges its distance vector obtained in Step-02 with its neighboring routers.
* After exchanging the distance vectors, each router prepares a new routing table.

This is shown below-

**At Router A-**

* Router A receives distance vectors from its neighbors B and D.
* Router A prepares a new routing table as-



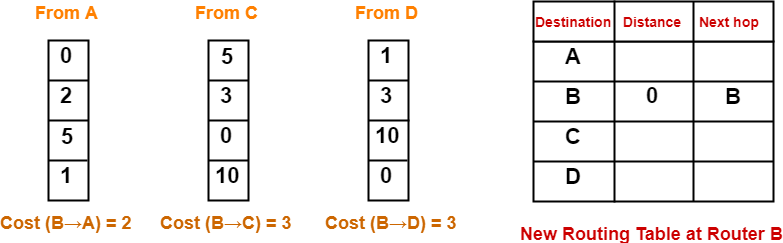
* Cost of reaching destination B from router A = min { 2+0 , 1+3 } = 2 via B.
* Cost of reaching destination C from router A = min { 2+3 , 1+10 } = 5 via B.
* Cost of reaching destination D from router A = min { 2+3 , 1+0 } = 1 via D.

Thus, the new routing table at router A is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 0 | A |
| B | 2 | B |
| C | 5 | B |
| D | 1 | D |

**At Router B-**

* Router B receives distance vectors from its neighbors A, C and D.
* Router B prepares a new routing table as-



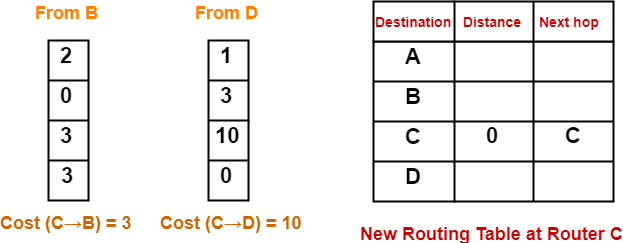
* Cost of reaching destination A from router B = min { 2+0 , 3+5 , 3+1 } = 2 via A.
* Cost of reaching destination C from router B = min { 2+5 , 3+0 , 3+10 } = 3 via C.
* Cost of reaching destination D from router B = min { 2+1 , 3+10 , 3+0 } = 3 via A.

Thus, the new routing table at router B is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 2 | A |
| B | 0 | B |
| C | 3 | C |
| D | 3 | A |

**At Router C-**

* Router C receives distance vectors from its neighbors B and D.
* Router C prepares a new routing table as-



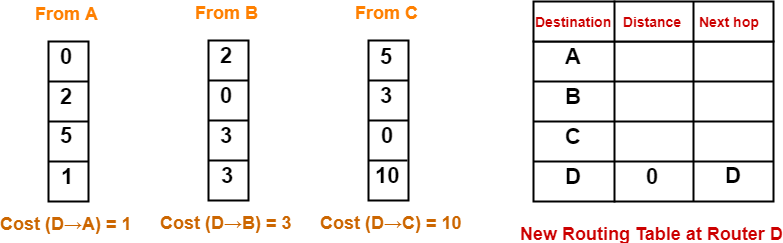
* Cost of reaching destination A from router C = min { 3+2 , 10+1 } = 5 via B.
* Cost of reaching destination B from router C = min { 3+0 , 10+3 } = 3 via B.
* Cost of reaching destination D from router C = min { 3+3 , 10+0 } = 6 via B.

Thus, the new routing table at router C is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 5 | B |
| B | 3 | B |
| C | 0 | C |
| D | 6 | B |

**At Router D-**

* Router D receives distance vectors from its neighbors A, B and C.
* Router D prepares a new routing table as-



* Cost of reaching destination A from router D = min { 1+0 , 3+2 , 10+5 } = 1 via A.
* Cost of reaching destination B from router D = min { 1+2 , 3+0 , 10+3 } = 3 via A.
* Cost of reaching destination C from router D = min { 1+5 , 3+3 , 10+0 } = 6 via A.

Thus, the new routing table at router D is-

|  |  |  |
| --- | --- | --- |
| **Destination** | **Distance** | **Next Hop** |
| A | 1 | A |
| B | 3 | A |
| C | 6 | A |
| D | 0 | D |

These will be the final routing tables at each router.

Link State Routing

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

## The three keys to understand the Link State Routing algorithm:

* **Knowledge about the neighborhood:** Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
* **Flooding:** Each router sends the information to every other router on the internetwork

except its neighbors. This process is known as Flooding. Every router that receives the

packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.

* **Information sharing:** A router sends the information to every other router only when

the change occurs in the information.

Link State Routing has two phases:

Reliable Flooding

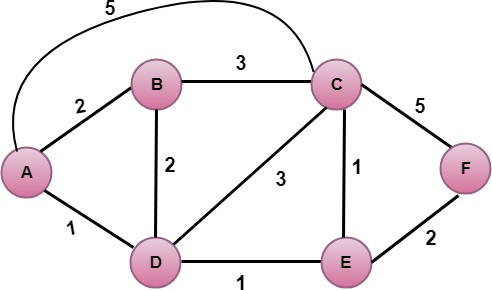
* **Initial state:** Each node knows the cost of its neighbors.
* **Final state:** Each node knows the entire graph.

Route Calculation

Each node uses Dijkstra's algorithm on the graph to calculate the optimal routes to all nodes.

* The Link state routing algorithm is also known as Dijkstra's algorithm which is used to find the shortest path from one node to every other node in the network.
* The Dijkstra's algorithm is an iterative, and it has the property that after kth iteration of the algorithm, the least cost paths are well known for k destination nodes.

Example



## In the above figure, source vertex is A.

Step 1:

The first step is an initialization step. The currently known least cost path from A to its directly attached neighbors, B, C, D are 2,5,1 respectively. The cost from A to B is set to 2, from A to D is set to 1 and from A to C is set to 5. The cost from A to E and F are set to infinity as they are not directly linked to A.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N** | **D(B),P(B)** | **D(C),P(C)** | **D(D),P(D)** | **D(E),P(E)** | **D(F),P(F)** |
| 1 | A | 2,A | 5,A | 1,A | ∞ | ∞ |

Step 2:

In the above table, we observe that vertex D contains the least cost path in step 1. Therefore, it is added in N. Now, we need to determine a least-cost path through D vertex.

## Calculating shortest path from A to B

* 1. v = B, w = D

2. D(B) = min( D(B) , D(D) + c(D,B) )

3. = min( 2, 1+2)>

4. = min( 2, 3)

1. The minimum value is 2. Therefore, the currently shortest path from A to B is 2.

## Calculating shortest path from A to C

1. v = C, w = D

7. D(B) = min( D(C) , D(D) + c(D,C) )

8. = min( 5, 1+3)

9. = min( 5, 4)

1. The minimum value is 4. Therefore, the currently shortest path from A to C is 4.</p>

## Calculating shortest path from A to E

1. v = E, w = D

12. D(B) = min( D(E) , D(D) + c(D,E) )

13. = min( ∞, 1+1)

14. = min(∞, 2)

1. The minimum value is 2. Therefore, the currently shortest path from A to E is 2. Note: The vertex D has no direct link to vertex E. Therefore, the value of D(F) is infinity.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N** | **D(B),P(B)** | **D(C),P(C)** | **D(D),P(D)** | **D(E),P(E)** | **D(F),P(F)** |
| 1 | A | 2,A | 5,A | 1,A | ∞ | ∞ |
| 2 | AD | 2,A | 4,D |  | 2,D | ∞ |

Step 3:

In the above table, we observe that both E and B have the least cost path in step 2. Let's consider the E vertex. Now, we determine the least cost path of remaining vertices through E.

## Calculating the shortest path from A to B.

1. v = B, w = E

17. D(B) = min( D(B) , D(E) + c(E,B) )

18. = min( 2 , 2+ ∞ )

19. = min( 2, ∞)

1. The minimum value is 2. Therefore, the currently shortest path from A to B is 2.

## Calculating the shortest path from A to C.

1. v = C, w = E

22. D(B) = min( D(C) , D(E) + c(E,C) )

23. = min( 4 , 2+1 )

24. = min( 4,3)

1. The minimum value is 3. Therefore, the currently shortest path from A to C is 3.

## Calculating the shortest path from A to F.

1. v = F, w = E

27. D(B) = min( D(F) , D(E) + c(E,F) )

28. = min( ∞ , 2+2 )

29. = min(∞ ,4)

1. The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N** | **D(B),P(B)** | **D(C),P(C)** | **D(D),P(D)** | **D(E),P(E)** | **D(F),P(F)** |
| 1 | A | 2,A | 5,A | 1,A | ∞ | ∞ |
| 2 | AD | 2,A | 4,D |  | 2,D | ∞ |
| 3 | ADE | 2,A | 3,E |  |  | 4,E |

Step 4:

In the above table, we observe that B vertex has the least cost path in step 3. Therefore, it is added in N. Now, we determine the least cost path of remaining vertices through B.

## Calculating the shortest path from A to C.

1. v = C, w = B

32. D(B) = min( D(C) , D(B) + c(B,C) )

33. = min( 3 , 2+3 )

34. = min( 3,5)

1. The minimum value is 3. Therefore, the currently shortest path from A to C is 3.

## Calculating the shortest path from A to F.

1. v = F, w = B

37. D(B) = min( D(F) , D(B) + c(B,F) )

38. = min( 4, ∞)

39. = min(4, ∞)

1. The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N** | **D(B),P(B)** | **D(C),P(C)** | **D(D),P(D)** | **D(E),P(E)** | **D(F),P(F)** |
| 1 | A | 2,A | 5,A | 1,A | ∞ | ∞ |
| 2 | AD | 2,A | 4,D |  | 2,D | ∞ |
| 3 | ADE | 2,A | 3,E |  |  | 4,E |
| 4 | ADEB |  | 3,E |  |  | 4,E |

Step 5:

In the above table, we observe that C vertex has the least cost path in step 4. Therefore, it is added in N. Now, we determine the least cost path of remaining vertices through C.

## a) Calculating the shortest path from A to F.

1. v = F, w = C

42. D(B) = min( D(F) , D(C) + c(C,F) )

43. = min( 4, 3+5)

44. = min(4,8)

1. The minimum value is 4. Therefore, the currently shortest path from A to F is 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N** | **D(B),P(B)** | **D(C),P(C)** | **D(D),P(D)** | **D(E),P(E)** | **D(F),P(F)** |
| 1 | A | 2,A | 5,A | 1,A | ∞ | ∞ |
| 2 | AD | 2,A | 4,D |  | 2,D | ∞ |
| 3 | ADE | 2,A | 3,E |  |  | 4,E |
| 4 | ADEB |  | 3,E |  |  | 4,E |
| 5 | ADEBC |  |  |  |  | 4,E |

Final table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **N** | **D(B),P(B)** | **D(C),P(C)** | **D(D),P(D)** | **D(E),P(E)** | **D(F),P(F)** |
| 1 | A | 2,A | 5,A | 1,A | ∞ | ∞ |
| 2 | AD | 2,A | 4,D |  | 2,D | ∞ |
| 3 | ADE | 2,A | 3,E |  |  | 4,E |
| 4 | ADEB |  | 3,E |  |  | 4,E |
| 5 | ADEBC |  |  |  |  | 4,E |
| 6 | ADEBCF |  |  |  |  |  |

Disadvantage:

Heavy traffic is created in Line state routing due to Flooding. Flooding can cause an infinite looping, this problem can be solved by using Time-to-leave field