9.5.3 Link State Routing

Distance vector routing was used in ARPANET upto 1979. After that it was replaced by the link state routing. Variants of this algorithm are now widely used. The link state routing is simple and each router has to perform the following five operations :

Router Operations

(*i*) Each router should discover its neighbours and obtain their network addresses.

(*ii*) Then it should measure the delay or cost to each of these neighbours.

(*iii*) It should construct a packet containing the network addresses and the delays of all the neighbours.

(*iv*) Send this packet to all other routers.

(*v*) Compute the shortest path to every other router.

The complete topology and all the delays are experimentally measured and this information is conveyed to each and every router.

Then a shortest path algorithm such as Dijkstra's algorithm can be used to find the shortest path to every other router.

Distance Vector Routing Algorithm-

> Distance Vector Routing is a dynamic routing algorithm.

It works in the following steps-

Step-01:

Each router prepares its routing table. By their local knowledge. each router knows about-

- All the routers present in the network
- Distance to its neighboring routers

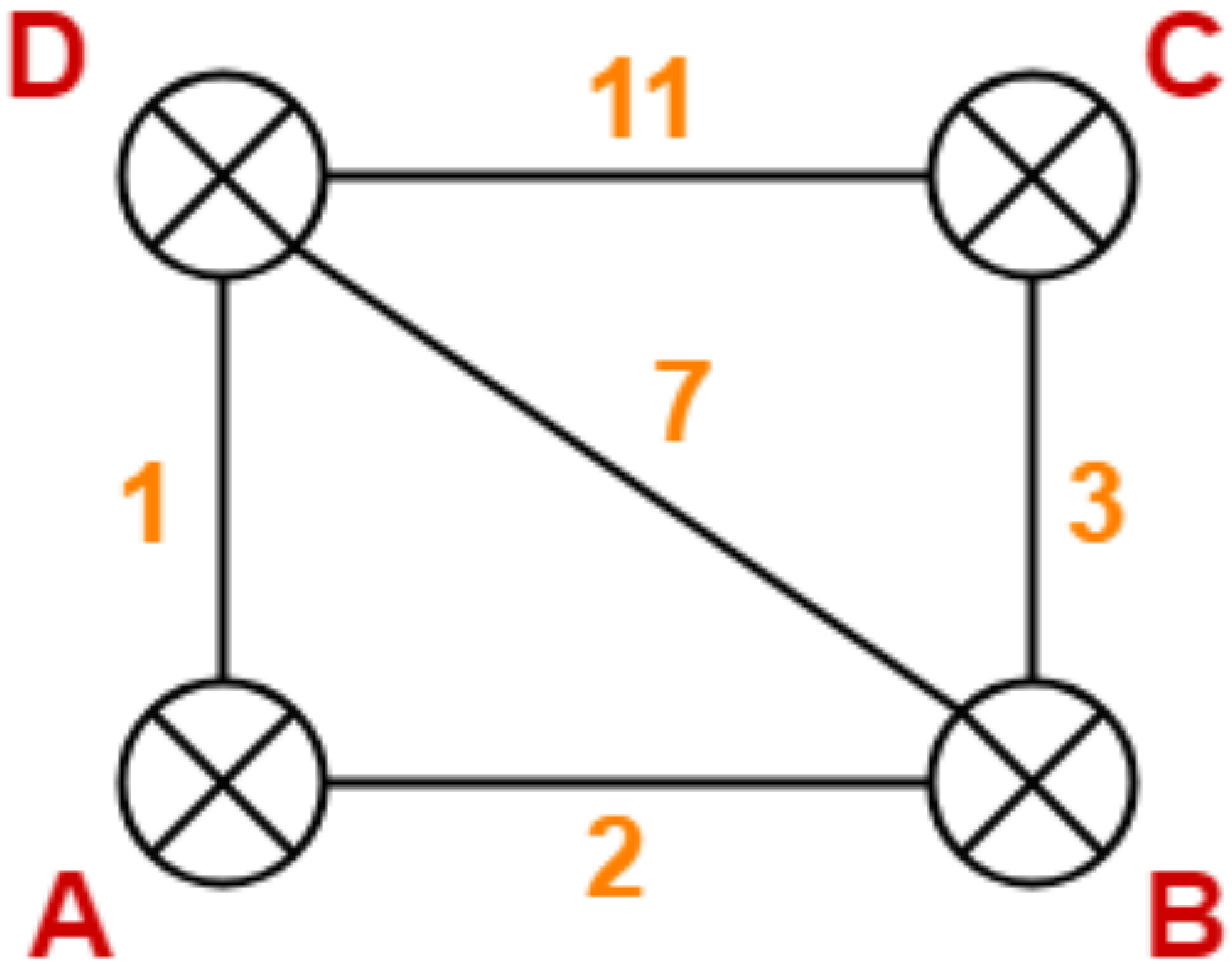
Step-02:

- Each router exchanges its distance vector with its neighboring routers.
- Each router prepares a new routing table using the distance vectors it has obtained from its neighbors.

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-

From B	From D
2	1
0	7
3	11
7	0
Cost(A→B) = 2	Cost(A→D) = 1

Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A

- 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-

From A

0
2
∞
1

Cost (B→A) = 2

From C

∞
3
0
11

Cost (B→C) = 3

From D

1
7
11
0

Cost (B→D) = 7

Destination	Distance	Next hop
A		
B	0	B
C		
D		

New Routing Table at Router B

- 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-

From B

2
0
3
7

Cost (C→B) = 3

From D

1
7
11
0

Cost (C→D) = 11

Destination	Distance	Next hop
A		
B		
C	0	C
D		

New Routing Table at Router C

- 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-

From A	From B	From C
0	2	∞
2	0	3
∞	3	0
1	7	11
Cost (D→A) = 1	Cost (D→B) = 7	Cost (D→C) = 11

Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D

- 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-

From B

2
0
3
3

Cost(A→B) = 2

From D

1
3
10
0

Cost(A→D) = 1

Destination	Distance	Next hop
A	0	A
B		
C		
D		

New Routing Table at Router A

- 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-

From A	From C	From D
0	5	1
2	3	3
5	0	10
1	10	0
Cost (B→A) = 2	Cost (B→C) = 3	Cost (B→D) = 3

Destination	Distance	Next hop
A		
B	0	B
C		
D		

New Routing Table at Router B

- 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-

From B

2
0
3
3

Cost (C→B) = 3

From D

1
3
10
0

Cost (C→D) = 10

Destination	Distance	Next hop
A		
B		
C	0	C
D		

New Routing Table at Router C

- 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-

From A	From B	From C
0	2	5
2	0	3
5	3	0
1	3	10
Cost (D→A) = 1	Cost (D→B) = 3	Cost (D→C) = 10

Destination	Distance	Next hop
A		
B		
C		
D	0	D

New Routing Table at Router D

- 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.

Identifying Unused Links-

After routing tables converge (becomes stable),

- Some of the links connecting the routers may never be used.
- In the above example, we can identify the unused links as-

We have-

- The value of next hop in the final routing table of router A suggests that only edges AB and AD are used.
- The value of next hop in the final routing table of router B suggests that only edges BA and BC are used.
- The value of next hop in the final routing table of router C suggests that only edge CB is used.

- The value of next hop in the final routing table of router D suggests that only edge DA is used.

Thus, edges BD and CD are never used.

Important Notes-

Note-01:

In Distance Vector Routing,

- Only distance vectors are exchanged.
- “Next hop”values are not exchanged.
- This is because it results in exchanging the large amount of data which consumes more bandwidth.

Note-02:

While preparing a new routing table-

- A router takes into consideration only the distance vectors it has obtained from its neighboring routers.
- It does not take into consideration its old routing table.

- The algorithm keeps on repeating periodically and never stops.
- This is to update the shortest path in case any link goes down or topology changes.

- Routing tables are prepared total (n-1) times if there are n routers in the given network.
- This is because shortest path between any 2 nodes contains at most n-1 edges if there are n nodes in the graph.

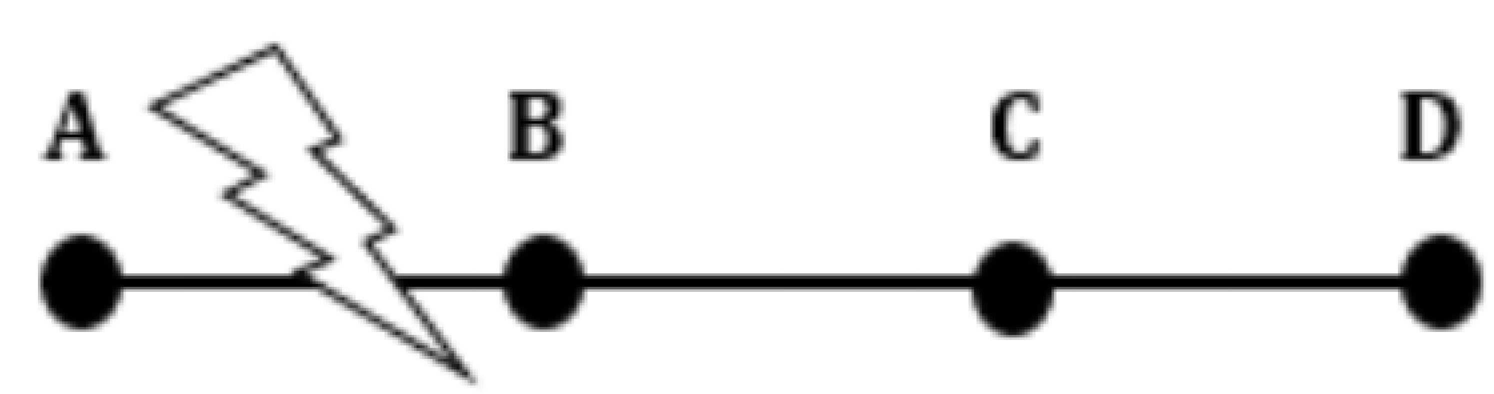
- Distance Vector Routing suffers from count to infinity problem.

Count to infinity problem:

1. One of the important issue in Distance Vector Routing is County of Infinity Problem.
2. Counting to infinity is just another name for a routing loop.
3. In distance vector routing, routing loops usually occur when an interface goes down.
4. It can also occur when two routers send updates to each other at the same time.

Example:

Link Between A & B is Broken



	A	B	C	D
A	0, -	1, A	2, B	3, C
B	1, B	0, -	2, C	3, D
C	2, B	1, C	0, -	1, C
D	3, B	2, C	1, D	0, -

- As you see in this graph, there is only one link between A and the other parts of the network.
- Now imagine that the link between A and B is cut.
- At this time, B corrects its table.
- After a specific amount of time, routers exchange their tables, and so B receives C's routing table.

- Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A -- it doesn't know B has no link to A).

- B receives this table and thinks there is a separate link between C and A, so it corrects its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said).

- Once again, routers exchange their tables.

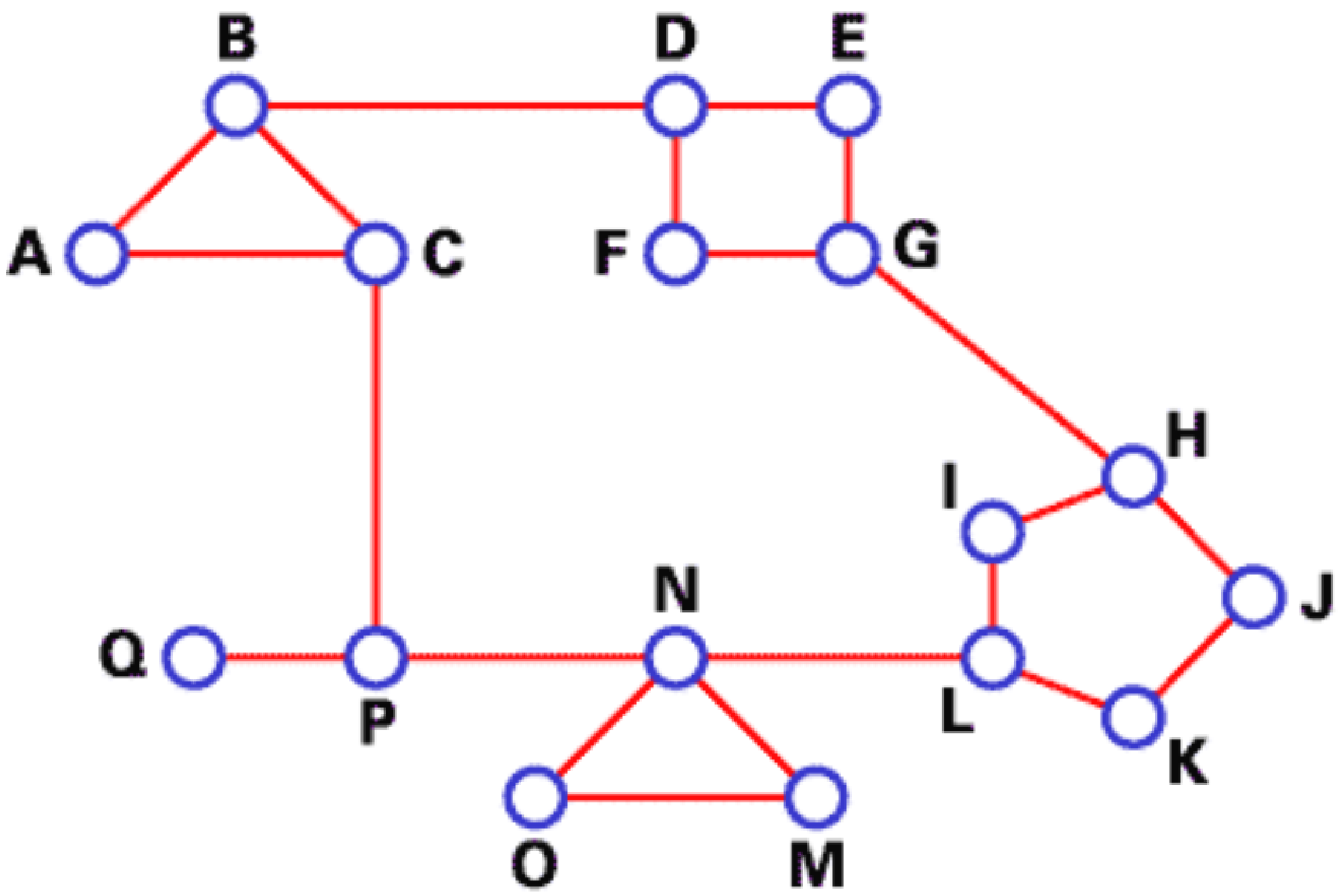
- When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).

- This process loops until all nodes find out that the weight of link to A is infinity.

- One way to solve this problem is for routers to send information only to the neighbors that are not exclusive links to the destination.

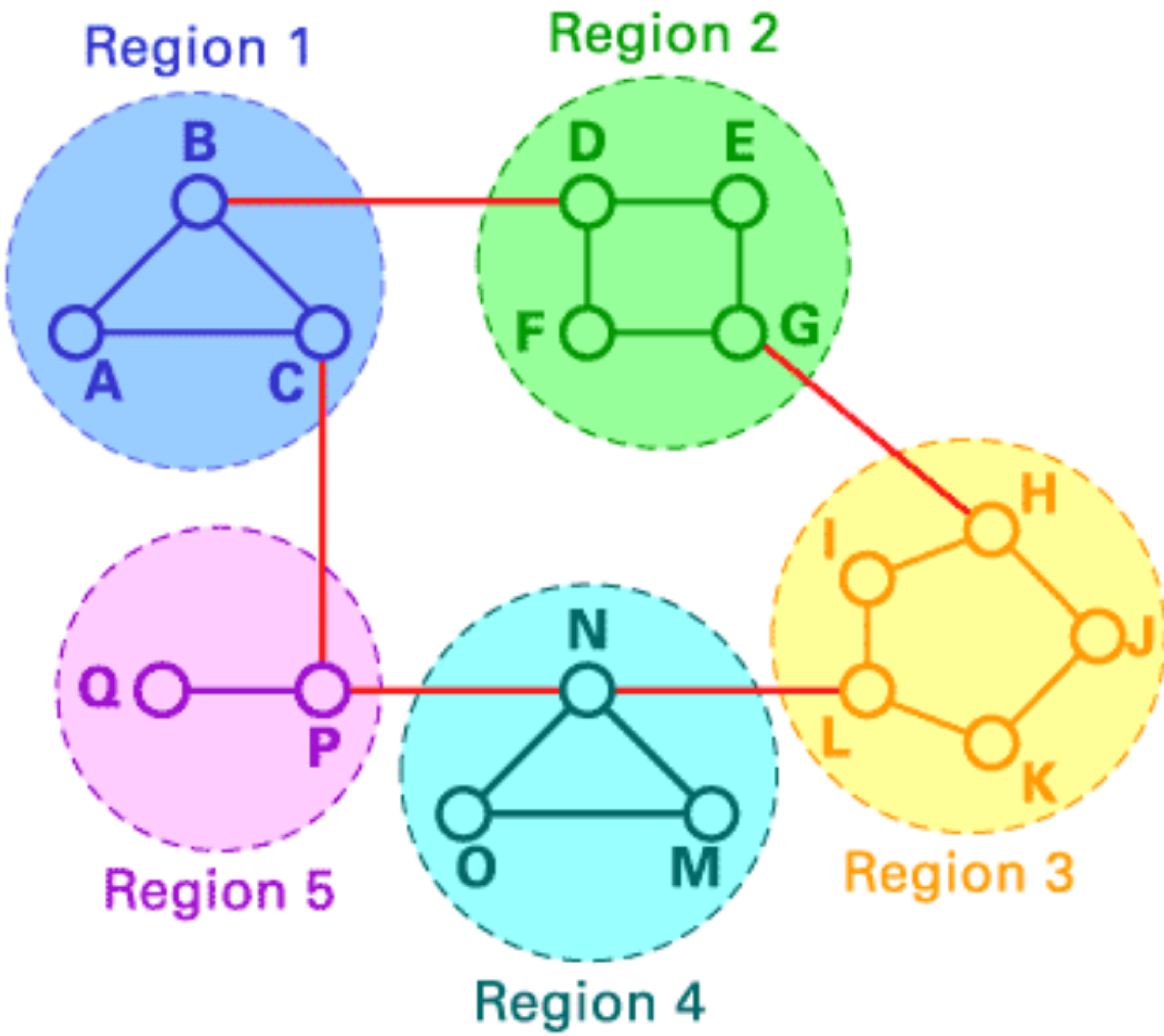
- For example, in this case, C shouldn't send any information to B about A, because B is the only way to A.

^ Hierarchical Routing



Destination	Line	Weight
A	---	---
B	B	1
C	C	1
D	B	2
E	B	3
F	B	3
G	B	4
H	B	5
I	C	5
J	C	6
K	C	5
L	C	4
M	C	4
N	C	3
O	C	4
P	C	2
Q	C	3

Network graph and A's routing table



Destination	Line	Weight
A	---	---
B	B	1
C	C	1
Region 2	B	2
Region 3	C	2
Region 4	C	3
Region 5	C	4

As you see, in both LS and DV algorithms, every router has to save some information about other routers. When the network size grows, the number of routers in the network increases. Consequently, the size of routing tables increases, as well, and routers can't handle network traffic as efficiently. We use **hierarchical routing** to overcome this problem. Let's examine this subject with an example:

We use DV algorithms to find best routes between nodes. In the situation depicted below, every node of the network has to save a routing table with 17 records. Here is a typical graph and routing table for A:

In hierarchical routing, routers are classified in groups known as **regions**. Each router has only the information about the routers in its own region and has no information about routers in other regions. So routers just save one record in their table for every other region. In this example, we have classified our network into five regions (see below).

If A wants to send packets to any router in region 2 (D, E, F or G), it sends them to B, and so on. As you can see, in this type of routing, the tables can be summarized, so network efficiency improves. The above example shows two-level hierarchical routing. We can also use three- or four-level hierarchical routing.