Computer Networking Lab-05

Group 35

Shubham Kumar Jha	220123081
Srinjoy Som	220123074
Yash Singhal	220123072
Priya Sharma	220101081

Distance Vector Routing:

Distance Vector Routing (DVR) Protocol is a method used by routers to find the best path for data to travel across a network. Each router keeps a table that shows the shortest distance to every other router based on total cost (congestion or physical distance) to reach them. Routers share this information with their neighbours, allowing them to update their tables and find the most efficient routes.

Each router maintains a Distance Vector table containing the distance between itself and All possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbours' distance vectors.

A router transmits its distance vector to its neighbours in a routing packet.
Each router receives and saves the most recently received distance vector from each of its neighbours.
A router recalculates its distance vector when:

1. It receives a distance vector from a neighbor containing different information than before.
2. It discovers that a link to a neighbor has gone down.

From time-to-time, each node sends its own distance vector estimate to neighbors. When a node **x** receives a new DV estimate from any neighbor **v**, it saves **v's** distance vector, and it updates its own DV using the Bellman-Ford equation:

$$Dx(y) = \min \{ C(x,v) + Dv(y), Dx(y) \} \text{ for each node } y$$

So to simulate it we have made a router class which has an id , its neighbors and stores the distance vector to other routers in the network. At each iteration we store the vectors that each router sends to the other routers because in the vanilla algorithm it is same for all neighbors.And then it simply runs the Distance vector algorithm.Also it checks if any changes are made when no changes are made and the states become stable , the algorithm stops.

```
 Enters two nodes and the path cost
4 5 6
dest             cost     nexthop
1                 0        1
2                 3        2
3                 5        2
4                 6        2
5                 12        2
dest             cost     nexthop
1                 3        1
2                 0        2
3                 2        3
4                 3        3
5                 9        3
dest             cost     nexthop
1                 5        1
2                 2        2
3                 0        3
4                 1        4
5                 7        4
dest             cost     nexthop
1                 6        3
2                 3        3
3                 1        3
4                 0        4
5                 6        5
dest             cost     nexthop
1                 12        4
2                 9        4
3                 7        4
4                 6        4
5                 0        5
No of iterataions to reach stability 3
```

Fig: before link disconnection

```
enter which link failed in format (a , b ) if link betw
4 5
dest             cost     nexthop
1                 0        1
2                 3        2
3                 5        2
4                 6        2
5                 100        2
dest             cost     nexthop
1                 3        1
2                 0        2
3                 2        3
4                 3        3
5                 100        3
dest             cost     nexthop
1                 5        1
2                 2        2
3                 0        3
4                 1        4
5                 100        4
dest             cost     nexthop
1                 6        3
2                 3        3
3                 1        3
4                 0        4
5                 100        3
dest             cost     nexthop
1                 100        4
2                 100        4
3                 100        4
4                 100        -1
5                 0        5
No of iterataions to reach stability 94
```

Fig: After Link failure, it takes 94 iterations to get to a stable state and set path cost to inf(100 in simulation)

Disadvantages of Distance Vector Routing Algorithm:

1. It is slower to converge than the link state.
2. It is at risk from the count-to-infinity problem.
3. It creates more traffic than link-state since a hop count change must be propagated to all routers and processed on each router. Hop count updates take place on a periodic basis, even if there are no changes in the network topology, so bandwidth-wasting broadcasts still occur.
4. For larger networks, distance vector routing results in larger routing tables than link-state since each router must know about all other routers. This can also lead to congestion on WAN links.

Poisoned Reverse Mechanism

Poisoned Reverse is a technique that, instead of just preventing a router from sending routing information back to the source, the router sends back the information but with an "infinity" value for the distance. This further ensures that the source will never consider using that route.

Poison Reverse states that routes that are received via one interface must be broadcast back out from that interface with an unreachable metric.

Most of the code is same as of the last problem just one thing is changed that, as here instead of a router sending routing information back to the source, the router sends back the information but with an "infinity" value for the distance. So we change the value in the sent distance vector to infinity before passing to the neighbour.

```
dest            cost     nexthop
1                0        1
2                3        2
3                5        2
4                6        2
5                12        2
dest            cost     nexthop
1                3        1
2                0        2
3                2        3
4                3        3
5                9        3
dest            cost     nexthop
1                5        1
2                2        2
3                0        3
5                7        4
dest            cost     nexthop
1                6        3
2                3        3
3                1        3
4                0        4
5                6        5
dest            cost     nexthop
1                12        4
2                9        4
3                7        4
4                6        4
5                0        5
No of iterataions to reach stability 3
```

Fig: before link disconnection

```
dest            cost    nexthop
1                0       1
2                3       2
3                5       2
4                6       2
5                100       2
dest            cost    nexthop
1                3       1
2                0       2
3                2       3
4                3       3
5                100       3
dest            cost    nexthop
1                5       1
2                2       2
3                0       3
4                1       4
5                100       4
dest            cost    nexthop
1                6       3
2                3       3
3                1       3
4                0       4
5                100       2
dest            cost    nexthop
1                100       4
2                100       4
3                100       4
4                100       -1
5                0       5
No of iterataions to reach stability 41
```

Fig: After Link failure, it takes 94 iterations to get to a stable state and set path cost to inf(100 in simulation)

Split Horizon Mechanism

The Split Horizon technique is used to prevent a router from sending information about a route back in the direction from which it came, helping to avoid routing loops and the count-to-infinity problem.

Split Horizon states that a route cannot be promoted out of an interface if the next hop for the advertised route is located on that interface. By not broadcasting the route out of the incorrect interface, Split Horizon passively attempts to prevent the routing loop.

Again most of the code is same as of the last problem just one thing that instead of sending back the information with an "infinity" value for the distance, the router does not send the info at all to the router it actually came from to avoid routing loops so we simulate this by checking when the condition is true and in that case skip the iteration for that destination because that value would not be passed to the router at all.

```
dest             cost     nexthop
1                 0        1
2                 3        2
3                 5        2
4                 6        2
5                 12        2
dest             cost     nexthop
1                 3        1
2                 0        2
3                 2        3
4                 3        3
5                 9        3
dest             cost     nexthop
1                 5        1
2                 2        2
3                 0        3
5                 7        4
dest             cost     nexthop
1                 6        3
2                 3        3
3                 1        3
4                 0        4
5                 6        5
dest             cost     nexthop
1                 12        4
2                 9        4
3                 7        4
4                 6        4
5                 0        5
No of iterataions to reach stability 3
```

Fig: before link disconnection

```
dest            cost     nexthop
1                0        1
2                3        2
3                5        2
4                6        2
5                100        2
dest            cost     nexthop
1                3        1
2                0        2
3                2        3
4                3        3
5                100        3
dest            cost     nexthop
1                5        1
2                2        2
3                0        3
4                1        4
5                100        4
dest            cost     nexthop
1                6        3
2                3        3
3                1        3
4                0        4
5                100        2
dest            cost     nexthop
1                100        4
2                100        4
3                100        4
4                100        -1
5                0        5
No of iterataions to reach stability 41
```

Fig: After Link failure, it takes 94 iterations to get to a stable state and set path cost to inf(100 in simulation)