CNL ASSIGNMENT 10

DISTANCE VECTOR ROUTING ALGORITHM

ANTRIKSH SHARMA
20070122021
CS-A1

Objective: To study and implement the Distance Vector Routing Algorithm

Theory:

A distance-vector routing protocol requires that a router informs its neighbours of topology changes periodically. Compared to link-state protocols, which require a router to inform all the nodes in a network of topology changes, distance-vector routing protocols have less computational complexity and message overhead. The term distance vector refers to the fact that the protocol manipulates vectors (arrays) of distances to other nodes in the network. The vector distance algorithm was the original ARPANET routing algorithm and was also used in the internet under the name of RIP.

Routers using distance-vector protocol do not have knowledge of the entire path to a destination.

Instead, they use two methods:

1. Direction in which router or exit interface a packet should be forwarded.
2. Distance from its destination

Distance-vector protocols are based on calculating the direction and distance to any link in a network. "Direction" usually means the next hop address and the exit interface. "Distance" is a measure of the cost to reach a certain node.

The least cost route between any two nodes is the route with minimum distance. Each node maintains a vector (table) of minimum distance to every node. The cost of reaching a destination is calculated using various route metrics. RIP uses the hop count of the destination whereas IGRP takes into account other information such as node delay and available bandwidth.

Updates are performed periodically in a distance-vector protocol where all or part of a router's routing table is sent to all its neighbours that are configured to use the same distance-vector routing protocol. RIP supports cross-platform distance vector routing whereas IGRP is a Cisco Systems proprietary distance vector routing protocol. Once a router has this information it is able to amend its own routing table to reflect the changes and then inform its neighbours of the changes.

LAB TASK 1: Implement the Go-Back-N Protocol

```
#include<stdio.h>
#include<iostream>
using namespace std;
struct node
{
    unsigned dist[6];
    unsigned from[6];
}DVR[10];
int main()
{
    cout<<"\n\n------------------- Distance Vector Routing Algorithm----------
";
    int costmat[6][6];
    int nodes, i, j, k;
    cout<<"\n\n Enter the number of nodes : ";
    cin>>nodes; //Enter the nodes
    cout<<"\n Enter the cost matrix : \n" ;
    for(i = 0; i < nodes; i++)
     {
        for(j = 0; j < nodes; j++)
        {
            cin>>costmat[i][j];
            costmat[i][i] = 0;
            DVR[i].dist[j] = costmat[i][j]; //initialise the distance equal to
cost matrix
            DVR[i].from[j] = j;
        }
    }
            for(i = 0; i < nodes; i++) //We choose arbitary vertex k and we
calculate the
            //direct distance from the node i to k using the cost matrix and add
the distance from k to node j
            for(j = i+1; j < nodes; j++)
            for(k = 0; k < nodes; k++)
                if(DVR[i].dist[j] > costmat[i][k] + DVR[k].dist[j])
                {   //We calculate the minimum distance
                    DVR[i].dist[j] = DVR[i].dist[k] + DVR[k].dist[j];
                    DVR[j].dist[i] = DVR[i].dist[j];
                    DVR[i].from[j] = k;
                    DVR[j].from[i] = k;
                }
        for(i = 0; i < nodes; i++)
        {
            cout<<"\n\n For router: "<<i+1;
            for(j = 0; j < nodes; j++)
                cout<<"\t\n node "<<j+1<<" via "<<DVR[i].from[j]+1<<" Distance
"<<DVR[i].dist[j];
        }
    cout<<" \n\n ";
    return 0;
}
```

PROBLEMS OUTPUT TERMINAL JUPYTER COMMENTS DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! ht

PS F:\SIT\5. SEM 5\CNL> cd "f:\SIT\5. SEM 5\CNL\Go Back to N\" ; if


-------------------- Distance Vector Routing Algorithm-----------

 Enter the number of nodes : 3

 Enter the cost matrix :
1 2 3
4 5 6
7 8 9


 For router: 1
 node 1 via 1 Distance 0
 node 2 via 2 Distance 2
 node 3 via 3 Distance 3

 For router: 2
 node 1 via 1 Distance 4
 node 2 via 2 Distance 0
 node 3 via 3 Distance 6

 For router: 3
 node 1 via 1 Distance 7
 node 2 via 2 Distance 8
 node 3 via 3 Distance 0


PS F:\SIT\5. SEM 5\CNL\Go Back to N>
```