23/11/2020 CN LAB KARTHIK VENUGOPAL

1BM18CS043

Distance Vector Routing

Program

```
class Network.
    def __init__(self,n).
        self.matrix = []
        self.n=n
    def addlink (self, u, v, w).
        Self.matrix append ((u, v, w))

    def printtable (self, dist, src)
        print ("Vector Table of {}".format (chr (ord ('A')+src)))
        print ("{0}\t{1}".format ("Dest", "cost"))
        for i in range (self.n)
            print ("{0}\t{1}".format (chr (ord('A')+i),
                dist[i]))

    def algor (self, src):
        dist= [99] * self.n
        dist [src]= 0
        for _ in range (self.n-1):
            for u, v, w in self matrix:
                ~~if~~ if dis[u] != 99 and dist[u]+w < dis[v]:
                    dist[v] = dist [u]+w
        self.printtable(dist,src)
```

23/11/2020 CN LAB KARTHIK VENUGOPAL

1BM18CS043

```
def main().
    matrix=[]
    print(" Enter No of Nodes ")
    n= int(input())
    print(" Enter the Adjacency matrix ")
    for i in range(n):
        m=list(map(int, input().split(" ")))
        matrix.append(m)
     g =Network(n)
     for i in range(n).
        for j in range(n):
            if matrix[i][j]==1.
                g.addlink(i,j,1)
     for _ in range(n):
        g.algo()

main()
```

In DVR protocol each router informs its neighbours of topology changes periodically ~~[illegible]~~ i.e each router maintains a distance vector table containing the distance between itself and all the destinations nodes. Distances are calculated ~~from~~ using neighbouring distance vectors