Experiment 4: Distance Vector Routing

Aim: To generate routing tables for a network of routers using Distance Vector Routing

Objective: After carrying out this experiment, students will be able to:

- Generate routing tables for a given network using Distance Vector Routing
- Analyze the reasons why Distance Vector Routing is adaptive in nature

Problem statement: You are required to write a program that can generate routing tables for a network of routers. Take the number of nodes and the adjacency matrix as input from user. Your program should use this adjacency matrix and create routing tables for all the nodes in the network. The routing table should consist of one entry per destination. This entry should contain the total cost and the outgoing line to reach that destination.

Analysis: While analyzing your program, you are required to address the following points:

- Why is Distance Vector Routing classified as an adaptive routing algorithm?
- Limitations of Distance Vector Routing

MARKS DISTRIBUTION

Component	Maximum Marks	Marks Obtained
Preparation of Document	7	
Results	7	
Viva	6	
Total	**20**	

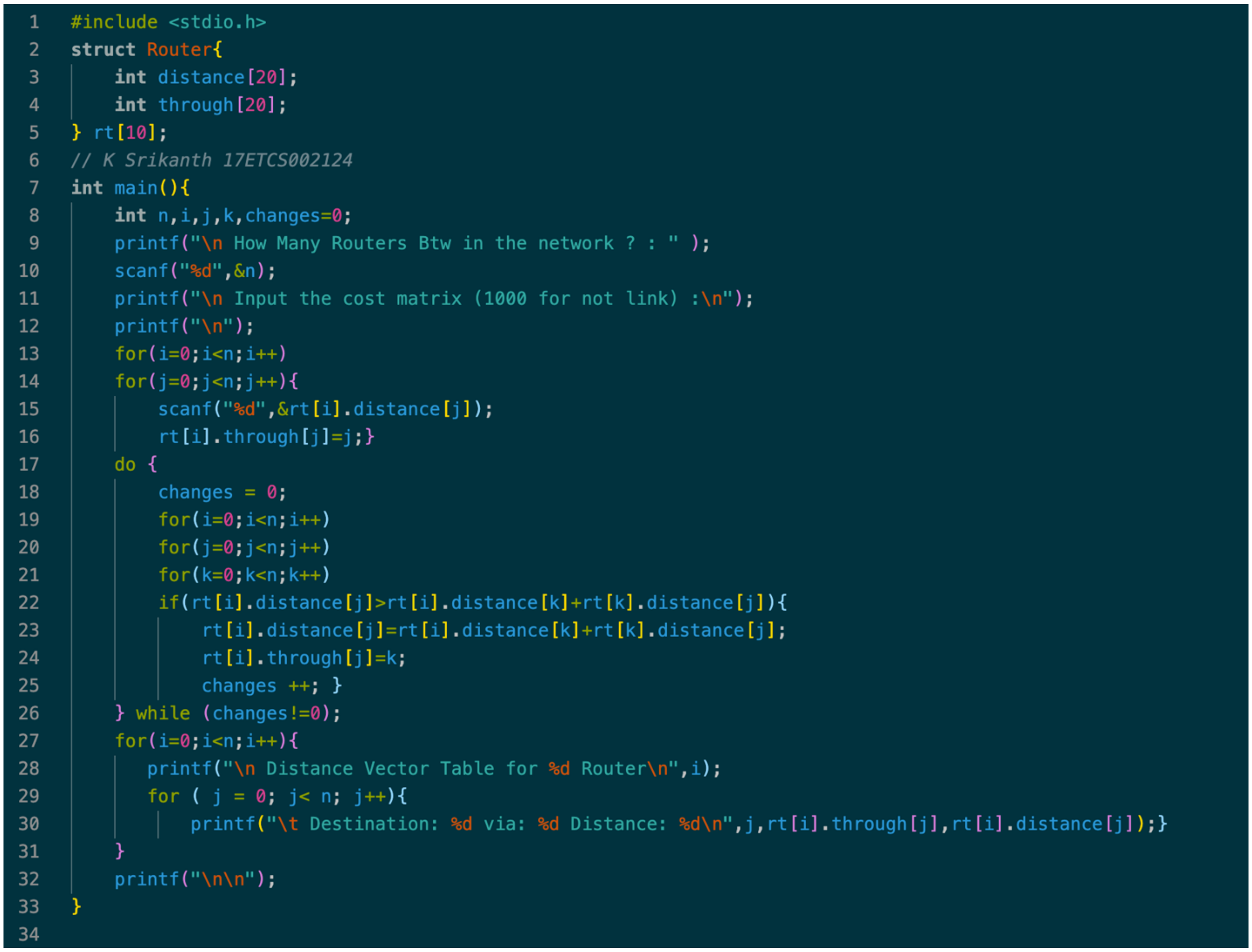
Submitted by: **17ETCS002124**

Register No: **K Srikanth**

1. Algorithm/Flowchart

```
1. Start

2.  Define a structure named ROUTER with two attributes

      a. Distance → Total distance to the destination router j
      b. Through → Next router in the path to reach destination

3.  Declare array of 10 routers

4. Input number of routers N

5. Input the cost Matrix:

6. Do

     For routers I :0 to N do
          For routers J :0 to N ( Destination ) do
               For routers K :0 to N ( Intermediate ) do
               If( distance( from I to J) >distance(from I to K)
          +distance(from K to J) )
               1. distance( from I to J) =distance(from I to K)
          +distance(from K to J)
               2. increment count of changes
               end for K
          end for J
     end for I

     while(changes>0): end do WHILE

7. For routers I: 0 to N do

     For routers J: 0 to N do

     1. Output distance (from I to J)
     2. Output next router in path (from I to J)
     End for J
     End for I

8. End of Program
```

2. Program



```c
#include <stdio.h>
struct Router{
    int distance[20];
    int through[20];
} rt[10];
// K Srikanth 17ETCS002124
int main(){
    int n,i,j,k,changes=0;
    printf("\n How Many Routers Btw in the network ? : " );
    scanf("%d",&n);
    printf("\n Input the cost matrix (1000 for not link) :\n");
    printf("\n");
    for(i=0;i<n;i++)
    for(j=0;j<n;j++){
        scanf("%d",&rt[i].distance[j]);
        rt[i].through[j]=j;}
    do {
        changes = 0;
        for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        for(k=0;k<n;k++)
        if(rt[i].distance[j]>rt[i].distance[k]+rt[k].distance[j]){
            rt[i].distance[j]=rt[i].distance[k]+rt[k].distance[j];
            rt[i].through[j]=k;
            changes ++; }
    } while (changes!=0);
    for(i=0;i<n;i++){
       printf("\n Distance Vector Table for %d Router\n",i);
       for ( j = 0; j< n; j++){
           printf("\t Destination: %d via: %d Distance: %d\n",j,rt[i].through[j],rt[i].distance[j]);}
    }
    printf("\n\n");
}
```

Figure 1 C Program Code for given problem statement

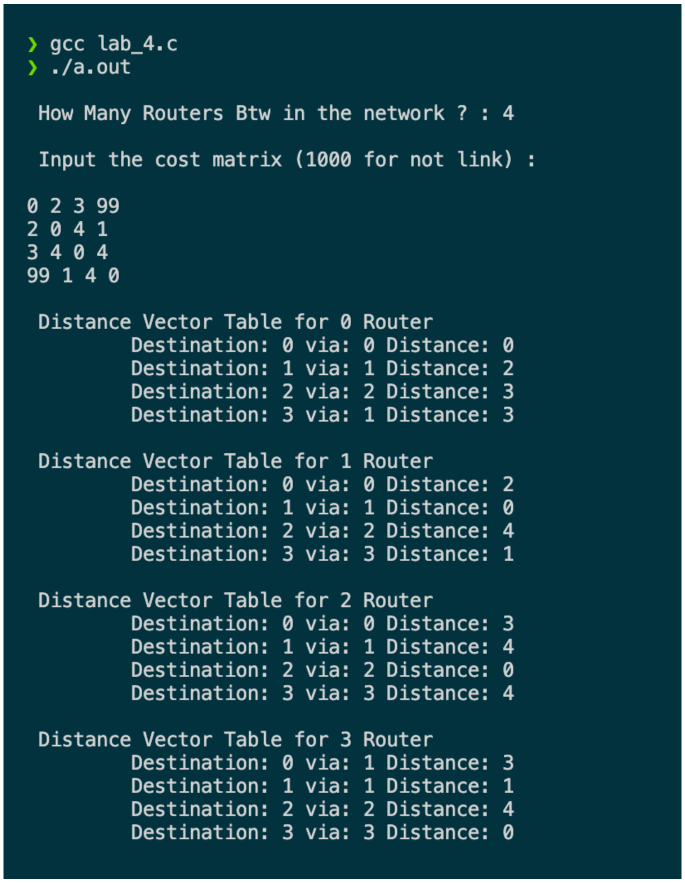
3. Results

To Run C Code

```
>> gcc filename.c
```

To Execute

```
>> ./a.out
```



```
❯ gcc lab_4.c
❯ ./a.out

 How Many Routers Btw in the network ? : 4

 Input the cost matrix (1000 for not link) :

0 2 3 99
2 0 4 1
3 4 0 4
99 1 4 0

 Distance Vector Table for 0 Router
         Destination: 0 via: 0 Distance: 0
         Destination: 1 via: 1 Distance: 2
         Destination: 2 via: 2 Distance: 3
         Destination: 3 via: 1 Distance: 3

 Distance Vector Table for 1 Router
         Destination: 0 via: 0 Distance: 2
         Destination: 1 via: 1 Distance: 0
         Destination: 2 via: 2 Distance: 4
         Destination: 3 via: 3 Distance: 1

 Distance Vector Table for 2 Router
         Destination: 0 via: 0 Distance: 3
         Destination: 1 via: 1 Distance: 4
         Destination: 2 via: 2 Distance: 0
         Destination: 3 via: 3 Distance: 4

 Distance Vector Table for 3 Router
         Destination: 0 via: 1 Distance: 3
         Destination: 1 via: 1 Distance: 1
         Destination: 2 via: 2 Distance: 4
         Destination: 3 via: 3 Distance: 0
```

Figure 2 C Program Output for the given problem statement (Image 1)

5. Analysis and Discussions

Why is Distance Vector Routing classified as an adaptive routing algorithm?

Using the distance vector table, a router can easily decide the next router to pass on the packets of data in the path to reach the destination router. It is called adaptive routing algorithm because it reads and respond to the changes in the network topology. i.e., if any node or link is added or removed from the network, the existing nodes in the network have to be made aware of this and their routing tables are kept updated.

Limitations of Distance Vector Routing

Since this routing algorithm assumes there is no traffic, it always sends the data packets in through shortest path, making those lines busy thereby increasing traffic. The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors. So, every time a new line is added, it has to be updated. Otherwise, the path would be ignored.

6. Conclusions

Each node constructs a one-dimensional array containing the "distances"(costs) to all other nodes and distributes that vector to its immediate neighbors. This helps node decide which neighboring node to pass on data packets to reach final node with minimum cost. Since it only considers the shortest paths, longest paths are left unused, and using only shortest paths increases traffic in those links. Although it has few listed drawbacks, it is widely used routing algorithm.

7. Comments

a. Limitations of the experiment

A link that is down or not present is assigned an infinite cost, but this program assumes it be 1000, so chances of missing original distances if given links weighing more than 1000.

b. Limitations of the results obtained

Minimum paths are not visually displayed in form of graphs.

c. Learning

Routing algorithms, Distance vector table, its application and significance**.**

d. Recommendations

Program to display results in tables.