Experiment 3: Neighbor Table Determination

Aim: To create neighbor table for a given network topology

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

- Generate neighbor table for all the nodes in a given topology.
- Analyze how this is useful in the process of routing data

Problem statement: You are required to write a program that calculates neighbor table for all the nodes in a given network. Consider a network with 10 nodes that is deployed in an area of 500 m^2. Your program should initially determine the distance between each node and all other nodes. Then the range of the nodes is given as input to the user. Using this range information, determine the neighbors of all the nodes.

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

- How this is useful in the process of routing data?
- For a 3D topology, how would your program need to be changed?

MARKS DISTRIBUTION

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

Submitted by: K Srikanth

Register No: 17ETCS002124

1. **Algorithm/Flowchart**

1. Start

2. Define a structure named ROUTER with following features: x and y coordinates

3. Declare an array of node ROUTER:R[node]numbered from 0to9

4. For i=0 to node-1 do

a. Input x and y coordinates of ROUTER-i
b. End for

5. Declare node × node 2d array with name d of type float
6. For i=0 to node-1 do
a. For j=0 to node-1 do
(Distance between two ROUTERs I and j)

$$d[i][j] = \sqrt{(r(i).x - r(j).x)^2 + (r(i).y - r(j).y)^2}$$

b. End for

c. End for

7. Input the router number:i
8. Input the range within which, Found ROUTERs have to be displayed
9. For j=0 to node-1 do
a. If d[i][j] ≤ range and d[i][j] > 0
i. Print i: ROUTER-number
ii.End if
iii.End for

10. Stop

2. Program

Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
# define nodes 10
struct router{
    int x,y;};
// K Srikanth 17ETCS002124
struct router r[nodes];
int main(){
    int i,j,range;
    float d[nodes][nodes];
    printf("Input the Following 10 Routers : in X & Y Positions\n");
    for ( i = 0; i < nodes; i++){
      scanf("%d %d",&r[i].x,&r[i].y);
    }
    i =0,j=0;
    printf("Distances Between Each Router and Other Router from 0 to 9\n");
    for (i = 0; i < nodes; i++){
        printf("Router_%d\t |",i);
        for ( j = 0; j < nodes; j++) {
            d[i][j]=sqrt(pow((abs((r[i].x)-(r[j].x))),2)+pow((abs((r[i].y)-(r[j].y))),2));
            printf("%.2f |",d[i][j]); }
        printf("\n");
    }
    printf("\nInput the Router Number : ");
    scanf("%d",&i);
    printf("Enter the Range : ");
    scanf("%d",&range);
    printf("The Router found with in %d units from %d are \t",range,i);
    for (j = 0; j<=nodes-1; j++){
        if(d[i][j]<=range && d[i][j]>0){
            printf("%d\t",j);}
    }

}
```

Figure 1 C Program for the given problem statement

3. Results

To Run C Code

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

To Execute

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

```
❯ ./a.out
Input the Following 10 Routers : in X & Y Positions
1 1
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
Distances Between Each Router and Other Router from 0 to 9
Router_0        |0.00 |1.00 |2.00 |3.00 |4.00 |5.00 |6.00 |7.00 |8.00 |9.00 |
Router_1        |1.00 |0.00 |1.00 |2.00 |3.00 |4.00 |5.00 |6.00 |7.00 |8.00 |
Router_2        |2.00 |1.00 |0.00 |1.00 |2.00 |3.00 |4.00 |5.00 |6.00 |7.00 |
Router_3        |3.00 |2.00 |1.00 |0.00 |1.00 |2.00 |3.00 |4.00 |5.00 |6.00 |
Router_4        |4.00 |3.00 |2.00 |1.00 |0.00 |1.00 |2.00 |3.00 |4.00 |5.00 |
Router_5        |5.00 |4.00 |3.00 |2.00 |1.00 |0.00 |1.00 |2.00 |3.00 |4.00 |
Router_6        |6.00 |5.00 |4.00 |3.00 |2.00 |1.00 |0.00 |1.00 |2.00 |3.00 |
Router_7        |7.00 |6.00 |5.00 |4.00 |3.00 |2.00 |1.00 |0.00 |1.00 |2.00 |
Router_8        |8.00 |7.00 |6.00 |5.00 |4.00 |3.00 |2.00 |1.00 |0.00 |1.00 |
Router_9        |9.00 |8.00 |7.00 |6.00 |5.00 |4.00 |3.00 |2.00 |1.00 |0.00 |

Input the Router Number : 3
Enter the Range : 8
The Router found with in 8 units from 3 are     0       1       2       4       5       6       7       8       9
```

Figure 2 Output for C code (Figure 1)

4. Analysis and Discussions

- **How this is useful in the process of routing data**

The above process helps the ROUTER to find out nearest neighboring ROUTER in the subnet and transmit packets of data.

- **For a 3D topology, how would your program need to be changed?**

In 3d topology, the ROUTER position is defined by 3 coordinates: x, y and z We can use the similar formulae to obtain the distance between any two ROUTERs i and j with x, y and z coordinates.

(Distance between Routers i and j)

$$\boldsymbol{d[i][j] = \sqrt{(r(i).x - r(j).x)^2 + (r(i).y - r(j).y)^2 + (r(i).y - r(j).y)^2}}$$

5. Conclusions

The main purpose of a router is to connect multiple networks and forward packets destined either for its own networks or other networks. When a router receives a packet, it determines the best path which involves the evaluation of multiple paths to the same destination network through the neighboring routers and selecting the optimum or shortest path to reach that network. The above program helps the router to find out nearest neighboring router in the destined path.

6. Comments

a. Limitations of the experiment

The algorithm is simple greedy approach to find out neighbors within certain range, the algorithm needs to select the router that can reach the destination network with minimum overall costs.

b. Limitations of the results obtained

The result obtained is just the closest routers from the given router, from a neighboring table.

c. Learning

Routing techniques, network topologies and neighboring table determination.