Experiment 3: Neighbour 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.
- Analyse 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:

Register No:

1. Algorithm/Flowchart

   Step1: Start

   Step2: generate random co-ordinates of 10 different nodes using rand().

   Step3: calculate the distance between each other using the distance formula:

   Distance=√((x_1-x_2)^2+(y_1-y_2)^2)

   Step4: display the distance in tabular manner using for loop.

   Step5: input the range from user

   Step6: display data according to given range using same formula as in step3.

   Step7: stop

2. Program:

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
   int i,j,x[12],y[12];
   int dist,a[99][99],r;
   for(i=1;i<=10;i++)
   {
       x[i] = rand() % 10+1;
       y[i] = rand() % 50+1;
       printf("\n x and y co-ordinates of node %d are: %d and %d \t",i,x[i],y[i]);
   }
   printf("\n\n\t\t<--------- THE DISTANCE TABLE IS:--------->\n\n");
   for (size_t i = 1; i < 11; i++)
   {
       printf("\tNode %d:", i);
   }
   printf("\n");
   for(i=1;i<=10;i++)
   {
    //    printf("%d \t",i);
       printf("Node %d: ", i);
       for(j=1;j<=10;j++)
       {
           dist = sqrt((pow((x[j]-x[i]),2))+ pow((y[j]-y[i]),2));
           printf("  %d \t", dist);
       }
       printf("\n");
   }
```


```
   printf("\n ENTER THE RANGE TO BE CONSIDERED AS NEIGHBOUR:");
   scanf("%d",&r);

   for(i=1;i<=10;i++)
   {
    //    printf("\n neighbours of %d node are : ", i);
       printf("\n NEIGHBOURS OF %d NODE ARE: ", i);
       for(j=1;j<=10;j++)
       {
           if(i!=j){
           dist = sqrt((pow((x[j]-x[i]),2))+pow((y[j]-y[i]),2));
           if(dist < r)

           {
             printf("%4d", j);
           }
        }
       }
   }
}
```

3. Results

```
x and y co-ordinates of node 1 are: 4 and 44
 x and y co-ordinates of node 2 are: 3 and 30
 x and y co-ordinates of node 3 are: 1 and 9
 x and y co-ordinates of node 4 are: 3 and 7
 x and y co-ordinates of node 5 are: 7 and 20
 x and y co-ordinates of node 6 are: 2 and 2
 x and y co-ordinates of node 7 are: 4 and 6
 x and y co-ordinates of node 8 are: 9 and 44
 x and y co-ordinates of node 9 are: 1 and 17
 x and y co-ordinates of node 10 are: 10 and 33
```


```
<--------- THE DISTANCE TABLE IS:--------->

        Node 1: Node 2: Node 3: Node 4: Node 5: Node 6: Node 7: Node 8: Node 9: Node 10:
Node 1:   0       14      35      37      24      42      38      5       27      12
Node 2:   14      0       21      23      10      28      24      15      13      7
Node 3:   35      21      0       2       12      7       4       35      8       25
Node 4:   37      23      2       0       13      5       1       37      10      26
Node 5:   24      10      12      13      0       18      14      24      6       13
Node 6:   42      28      7       5       18      0       4       42      15      32
Node 7:   38      24      4       1       14      4       0       38      11      27
Node 8:   5       15      35      37      24      42      38      0       28      11
Node 9:   27      13      8       10      6       15      11      28      0       18
Node 10:   12     7       25      26      13      32      27      11      18      0



 ENTER THE RANGE TO BE CONSIDERED AS NEIGHBOUR:10

 NEIGHBOURS OF 1 NODE ARE:    8
 NEIGHBOURS OF 2 NODE ARE:   10
 NEIGHBOURS OF 3 NODE ARE:    4   6   7   9
 NEIGHBOURS OF 4 NODE ARE:    3   6   7
 NEIGHBOURS OF 5 NODE ARE:    9
 NEIGHBOURS OF 6 NODE ARE:    3   4   7
 NEIGHBOURS OF 7 NODE ARE:    3   4   6
 NEIGHBOURS OF 8 NODE ARE:    1
 NEIGHBOURS OF 9 NODE ARE:    3   5
 NEIGHBOURS OF 10 NODE ARE:    2
PS F:\RUAS\5 sem\Computer networks\CN Lab\lab_3>
```

4. Analysis and Discussions

   Routing table using naïve algorithm approach.

   In computer networking a routing table, or routing information base (RIB), is a data table stored in a router or a network host that lists the routes to particular network destinations, and in some cases, metrics (distances) associated with those routes. The routing table contains information about the topology of the network immediately around it.

The construction of routing tables is the primary goal of routing protocols. Static routes are entries made in a routing table by non-automatic means and which are fixed rather than being the result of routing protocols and associated network topology discovery procedures.

The nodes calculate the distance between each other using the distance formula: Distance=√((x_1-x_2)^2+(y_1-y_2)^2).

5. Conclusions

In this experiment we got to know that routing table is used by other nodes to determine whether they can communicate with each other, their position and the distances between different nodes.

6. Comments

a. Limitations of the experiment:

Limitations of distance/routing table:

- They are not easy to implement in a large network.
- Managing the static configurations can become time consuming.
- If a link fails, a static route cannot reroute traffic.