

COMPUTER NETWORKS AND SECURITY LABORATORY

Assignment No. 7 B

NAME :- OJUS P. JAISWAL

ROLL NO. :- TACO19108

YEAR AND DIV :- TE A

Ques :- Write a program to implement Distance vector routing protocol to find suitable path for transmission.

Solution :-

Program :

```
#include<stdlib.h>
#include<stdio.h>
#define NUL 1000
#define NODES 10
struct node
{
    int t[NODES][3];
};
struct node n[NODES];
typedef struct node NOD;
int main()
{
    void init(int,int);
    void inp(int,int);
    void caller(int,int);
    void op1(int,int,int);
    void find(int,int);
    int i,j,x,y,no;
```

```

do{
    printf("\n Enter the no of nodes required:");
    scanf("%d",&no);
}while(no>10||no<0);
for(i=0;i<no;i++)
{
    init(no,i);
    inp(no,i);
}
printf("\nThe configuration of the nodes after initalization is as follows:");
for(i=0;i<no;i++)
op1(no,i,0);
for(j=0;j<no;j++)
{
    for(i=0;i<no;i++)
        caller(no,i);
}
printf("\nThe config of the nodes after the comp of the paths is as
follows:");
for(i=0;i<no;i++)
op1(no,i,1);
while(1)
{
    printf("\n Enter 0 to exit or any other key to find the shortest path:");
    scanf("%d",&j);
}

```

```

        if(!j)
            break;
        do{
            printf("\n Enter the nodes btn which path is to be found:");
            scanf("%d%d",&x,&y);
            }while((x<0 | x>no) && (y<0 | y>no));
        printf("\nThe most suitable route from node %d to %d is as follows\n",x,y);
        find(x,y);
        printf("%d",y);
        printf("\nThe length of the shortest path between node %d & %d is
%d",x,y,n[x- 1].t[y-1][2]);
    }
}

void init(int no,int x)
{
    int i; for(i=0;i<no;i++)
    {
        n[x].t[i][1]=i;
        n[x].t[i][2]=999;
        n[x].t[i][3]=NUL;
    }
    n[x].t[x][2]=0;
    n[x].t[x][3]=x;
}

void inp(int no,int x)

```

```

{
    int i;
    printf("\nEnter the dists from the nodes %d to other node...",x+1);
    printf("\nPls enter 999 if there is no direct \n");
    for(i=0;i<no;i++)
    {
        if(i!=x)
        {
            do
            {
                printf("\n Enter dist to node %d=",i+1);
                scanf("%d",&n[x].t[i][2]);
            }while(n[x].t[i][2]<0 | n[x].t[i][2]>999);
            if(n[x].t[i][2]!=999)
                n[x].t[i][3]=i;
        }
    }
}

```

```

void caller(int no,int x)
{
    void compar(int,int,int);
    int i;
    for(i=0;i<no;i++)

```

```

    {
        if(n[x].t[i][2]!=999 && n[x].t[i][2]!=0)
        {
            compar(x,i,no);
        }
    }
}

```

```

void compar(int x,int y,int no)
{
    int i,z;
    for(i=0;i<no;i++)
    {
        z=n[x].t[y][2]+n[y].t[i][2];
        if(n[x].t[i][2]>z)
        {
            n[x].t[i][2]=z;
            n[x].t[i][3]=y;
        }
    }
}

```

```

void op1(int no,int x,int z)
{

```

```

int i,j;

printf("\n The routing table for node no %d is as follows",x+1);

printf("\n\n\t\t\tDESTINATION\tDISTANCE\tNEXT_HOP");

for(i=0;i<no;i++)
{
if((!z && n[x].t[i][2]>=999) || (n[x].t[i][2]>=(999*no)))

    printf("\n\t\t\t %d \tNO LINK \t NO HOP",n[x].t[i][1]+1);

else

if(n[x].t[i][3]==NUL)

    printf("\n\t\t\t %d \t\t %d \t\t NO HOP",n[x].t[i][1]+1,n[x].t[i][2]);

else

    printf("\n\t\t\t %d \t\t %d
\t\t %d",n[x].t[i][1]+1,n[x].t[i][2],n[x].t[i][3]+1);

}
}

```

```

void find(int x,int y)
{

    int i,j;

    i=x-1;

    j=y-1;

    printf("%d-->",x);

    if(n[i].t[j][3]!=j)

    {

        find(n[i].t[j][3]+1,y);
    }
}

```

return;

}

}

Output :

```
input

Enter the no of nodes required:3

Enter the dists from the nodes 1 to other node...
Pls enter 999 if there is no direct

Enter dist to node 2=10

Enter dist to node 3=999

Enter the dists from the nodes 2 to other node...
Pls enter 999 if there is no direct

Enter dist to node 1=999

Enter dist to node 3=15

Enter the dists from the nodes 3 to other node...
Pls enter 999 if there is no direct

Enter dist to node 1=20

Enter dist to node 2=25

The configuration of the nodes after initialization is as follows:
The routing table for node no 1 is as follows

      DESTINATION    DISTANCE    NEXT_HOP
      1              0            1
      2              10           2
      3      NO LINK      NO HOP

The routing table for node no 2 is as follows

      DESTINATION    DISTANCE    NEXT_HOP
      1      NO LINK      NO HOP
      2              0            2
      3              15           3

The routing table for node no 3 is as follows

      DESTINATION    DISTANCE    NEXT_HOP
      1              20           1
      2              25           2
      3              0            3

The config of the nodes after the comp of the paths is as follows:
The routing table for node no 1 is as follows

      DESTINATION    DISTANCE    NEXT_HOP
      1              0            1
      2              10           2
      3              25           2

The routing table for node no 2 is as follows

      DESTINATION    DISTANCE    NEXT_HOP
      1              35           3
      2              0            2
      3              15           3

The routing table for node no 3 is as follows

      DESTINATION    DISTANCE    NEXT_HOP
      1              20           1
      2              25           2
      3              0            3

Enter 0 to exit or any other key to find the shortest path:1

Enter the nodes btn which path is to be found:1 3

The most suitable route from node 1 to 3 is as follows
1-->2-->3
The length of the shortest path between node 1 & 3 is 25
Enter 0 to exit or any other key to find the shortest path:

...Program finished with exit code 9
Press ENTER to exit console
```