

Title : Write a program to implement link state/distance vector routing protocol to find suitable path for transmission

Subject :- Computer Networks and Security Laboratory

Date :-

*

Code:-

```
#include <iostream>
#include <stdio.h>
using namespace std;

struct node {
    int dist[20];
    int from[20];
} route[10];

int main()
{
    int dm[20][20], no;
    cout << "Enter no of nodes." << endl;
    cin >> no;
    cout << "Enter the distance matrix:" << endl;
    for (int i = 0; i < no; i++) {
        for (int j = 0; j < no; j++) {
            cin >> dm[i][j];
            /* Set distance from i to i as 0 */
            dm[i][i] = 0;
            route[i].dist[j] = dm[i][j];
            route[i].from[j] = j;
        }
    }

    int flag;
    do {
        flag = 0;
        for (int i = 0; i < no; i++) {
            for (int j = 0; j < no; j++) {
                for (int k = 0; k < no; k++) {
                    if ((route[i].dist[j]) > (route[i].dist[k] + route[k].dist[j])) {
                        route[i].dist[j] = route[i].dist[k] + route[k].dist[j];
                        route[i].from[j] = k;
                        flag = 1;
                    }
                }
            }
        }
    } while (flag == 1);
}
```

```

    }
} while (flag);
for (int i = 0; i < no; i++) {
    cout << "Router info for router: " << i + 1 << endl;
    cout << "Dest\tNext Hop\tDist" << endl;
    for (int j = 0; j < no; j++) {
        printf("%d\t%d\t%d\n", j+1, route[i].from[j]+1, route[i].dist[j]);
    }
}
return 0;
}

```

Output:-

```

CNS Practical
File Edit Selection View Go Run Terminal Help ← →
EXPLORER ... J Calc_Client.java 4 J DNS.class J DNS.java J Subnet.java 9+ E dvr.cpp
Calc.Client.class
Calc.Client.java
Calc.Server.class
Calc.Server.java
Calculator.png
ClientUDP.class
ClientUDP.java
DNS.class
DNS.java
DNS.png
dvr.cpp
ServerUDP.class
ServerUDP.java
Subnet newv.png
Subnet.class
Subnet.java
SubnettingDemo.class
SubnettingDemo.java
SubnettingDemo.png
UDP Output.png
PROBLEMS 22 OUTPUT DEBUG CONSOLE PORTS TERMINAL
PS C:\Users\harsh\Downloads\CNS Practical\output> cd 'c:\Users\harsh\Downloads\CNS Practical\output'
PS C:\Users\harsh\Downloads\CNS Practical\output> & .\dvr.exe
● Enter no of nodes.
4
● Enter the distance matrix:
0 2 7 999
2 0 999 1
7 999 0 3
999 1 3 0
Router info for router: 1
Dest Next Hop Dist
1 1 0
2 2 2
3 2 6
4 2 3
Router info for router: 2
Dest Next Hop Dist
1 1 2
2 2 0
3 4 4
4 4 1
Router info for router: 3
Dest Next Hop Dist
1 2 6
2 4 4
3 3 0
4 4 3
Router info for router: 4
Dest Next Hop Dist
1 2 3
2 2 1
3 3 3
4 4 0
PS C:\Users\harsh\Downloads\CNS Practical\output>

```

The screenshot shows a terminal window titled 'CNS Practical' running on a Windows operating system. The window displays the output of a C++ program named 'dvr.cpp'. The program first asks for the number of nodes, which is entered as 4. It then prompts for the distance matrix and prints the resulting router information for each of the four routers. The router information includes the next hop and distance to all other routers in the network.