‘OPEN SHORTEST PATH FIRST’

TEAM MEMBERS:

- Soumyak Paul |Class Roll: 3
- Enrollment No.: 12017009001495
- Debjit Chattopadhyay |Class Roll: 4
- Enrollment No.: 12017009001087

SECTION: CSE 3D

SUBJECT: DESIGN AND ANALYSIS OF ALGORITHM

INDEX

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of our faculties. I would like to extend my sincere thanks to all of them.

I am highly indebted to all the faculties for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents for their kind co-operation and encouragement which help me in completion of this project.

PROBLEM STATEMENT:

To implement a routing protocol for Internet Protocol (IP) networks which finds the minimum cost based on shortest path between two routers. The shortest path between the source router and its adjacent routers has to be found and the path has to be displayed.

ALGORITHM:

```
function Dijkstra(Graph, source):
      dist[source]  := 0                    // Distance from source to source is set to 0
 for each vertex v in Graph:           // Initializations
           if v ≠ source
                    dist[v]  := infinity           // Unknown distance function from source to each
          node set to infinity
         add v to Q                         // All nodes initially in Q


     while Q is not empty:                  // The main loop
         v := vertex in Q with min dist[v]  // In the first run-through, this vertex is the source
node
         remove v from Q


         for each neighbor u of v:           // where neighbor u has not yet been removed from Q.
            alt := dist[v] + length(v, u)
            if alt < dist[u]:              // A shorter path to u has been found
               dist[u]  := alt           // Update distance of u


     return dist[]
end function
```

SOURCE CODE:

```
#include<stdio.h>
#include<string.h>
int main()
{
	int count,src_router,i,j,k,w,v,min; int
	cost_matrix[100][100],dist[100],last[100]; int
	flag[100];
	printf("Enter the no of routers\n"); scanf("%d",&count);
	printf("\n Enter the cost matrix values:");
	for(i=0;i<count;i++)
	{
	 for(j=0;j<count;j++)
		{
		printf("\n%d->%d:",i,j);
			scanf("%d",&cost_matrix[i][j]);
			if(cost_matrix[i][j]<0)
			{ cost_matrix[i][j]=1000;
			}
		}
		}
	printf("\n Enter the source router:");
	scanf("%d",&src_router);
	for(v=0;v<count;v++)
	{
		flag[v]=0;
		last[v]=src_router;
		dist[v]=cost_matrix[src_router][v];
```

```
        }
        flag[src_router]=1;
        for(i=0;i<count;i++)
        {
                min=1000;
                for(w=0;w<count;w++)
                {
                if(!flag[w])
                        if(dist[w]<min)
                                {
                                 v=w;
                                min=dist[w];
                                }
                }
         flag[v]=1;
        for(w=0;w<count;w++
        {
                if(!flag[w])
                if(min+cost_matrix[v][w]<dist[w])
                {
                dist[w]=min+cost_matrix[v][w];
                last[w]=v;
                }
        }
}
 for(i=0;i<count;i++)
        {
         printf("\n%d==>%d:Path taken:%d",src_router,i,i);
        w=i;
        while(w!=src_router)
```

```
                    {
                    printf("<--%d",last[w]);
                    w=last[w];
                    }
                    printf("\n Shortest path cost:%d",dist[i]); }
}
```

- ## SAMPLE INPUT AND OUTPUT:

```
 Enter the cost matrix values:
0->0:0

0->1:11

0->2:22

1->0:12

1->1:0

1->2:23

2->0:32

2->1:21

2->2:0

 Enter the source router:1

1==>0:Path taken:0<--1
 Shortest path cost:12
1==>1:Path taken:1
 Shortest path cost:0
1==>2:Path taken:2<--1
 Shortest path cost:23
--------------------------------
Process exited after 45.82 seconds with return value 0
Press any key to continue . . .
```

1.

2.

```
Enter the no of routers
4

 Enter the cost matrix values:
0->0:12

0->1:63

0->2:45

0->3:22

1->0:34

1->1:88

1->2:11

1->3:22

2->0:42

2->1:33

2->2:23

2->3:84
```

3.

```
3->0:23

3->1:71

3->2:22

3->3:8

 Enter the source router:2

2==>0:Path taken:0<--2
 Shortest path cost:42
2==>1:Path taken:1<--2
 Shortest path cost:33
2==>2:Path taken:2
 Shortest path cost:23
2==>3:Path taken:3<--1<--2
 Shortest path cost:55
```

COMPLEXITY ANALYSIS:

Time Complexity: The time complexity of the above code/algorithm looks O(V^2) as there are two nested while loops. If we take a closer look, we can observe that the statements in inner loop are executed O(V+E) times (similar to BFS).

So overall time complexity is O(E+V)*O(LogV) which is O((E+V)*LogV) = **O(ELogV).**

Space Complexity: O(V).

APPLICATION AND LIMITATIONS:

File Server: We want to designate a file server in a local area network. Now, we consider that most of time transmitting files from one computer to another computer is the connect time. So we want to minimize the number of "hops" from the file server to every other computer on the network.

Popularity and convergence: OSPF is the first widely deployed routing protocol. It can converge with a network in a few seconds and it is one of the protocols that can provide loopfree paths. Aside from these features, OSPF allows the imposition of policies for the propagation of routes in the network.

Where has it been implemented?

1)Microsoft's Windows NT 4.0 Server, Windows 2000 Server and Windows Server 2003 all have OSPF v2 in the Routing and Remote Access Services. Microsoft Removed the support with the Windows Server 2008 and later version of Server Operating Systems by the company.

2)Multi-Protocol Routing module in Netware has support for OSPF.

3)OpenBSD Operating system has an implementation of OpenBGPD protocol which has OpenOSPFD implementation.

Limitations of OSPF:

- OSPF is a processor intensive protocol to use.
- Because it maintains more than one copy of routing information, it consumes more memory.
- OSPF is a more complex protocol to understand and learn compared to other Internet Protocols.

CONCLUSION

Open Shortest Path First as a routing protocol has an important place in internet infrastructure. Being able to find the shortest path easily and quickly helps in reducing unnecessary network load and the ability to find another path in case of error at the optimal one helps in increasing the stability of the network. Completing this project helped us in understanding the implementation of Dijkstra’s algorithm in a real-life situation.