EXPERIMENT - XII
LINK STATE ROUTING PROTOCOL

April 6, 2020

ADITHYA D RAJAGOPAL
ROLL NO : 9
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM

AIM

To implement and simulate algorithm for link state routing protocol.

THEORY

Link-State Routing Protocol

Link-State Routing Protocols are one of the two main classes of routing protocols used in packet switching networks for computer communications, the other being distance-vector routing protocols. Examples of link-state routing protocols include Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS).
The link-state protocol is performed by every switching node in the network (i.e., nodes that are prepared to forward packets; in the Internet, these are called routers). The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. Each collection of best paths will then form each node's routing table.

ALGORITHM

Algorithm 1 Link State Routing Algorithm

```
1: START
2: procedure LINKSTATEROUTING
3:     Input the number of routers and the cost matrix.
4:     Input the source router(u).
5:     for all router v do
6:         if v is a neighbour of u then
7:             dist[v] = cost[u][v]
8:         else
9:             dist[v] = ∞
10:        end if
11:        for i = 0 to n do
12:            Find a node w such that D(w) is minimum.
13:            dist[v] = min(dist[v] , dist[w] + cost[w][v])
14:            Print dist[v] for all nodes v.
15:        end for
16:    end for
17: end procedure
18: STOP
```

SOURCE CODE

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

void main()
{
 int n,i,j,min,v,w,cost[10][10],dist[10],source,flag[10],last[10];
 char s[10];
 printf("Enter the number of routers : ");
 scanf("%d",&n);
 printf("Enter * for non-neighbouring vertices...\n");
 printf("Enter the cost matrix:\n");
 for(i=0;i<n;i++)
 printf("\tN%d",i);
 printf("\n");
 for(i=0;i<n;i++)
 {
  printf("N%d\t",i);
  for(j=0;j<n;j++)
  {
   scanf("%s",s);
   if(strcmp(s,"*")==0)
   cost[i][j]=9999;
   else
   cost[i][j]=atoi(s);
  }
 }
 printf("Enter the source router : N");
 scanf("%d",&source);
 for(i=0;i<n;i++)
 {
  cost[i][i]=0;
  dist[i]=cost[source][i];
```

```
  flag[i]=0;
  last[i]=source;
 }
 flag[source]=1;
 for(i=0;i<n;i++)
 {
  min=9999;
  for(w=0;w<n;w++)
  if(flag[w]==0)
  if(dist[w]<min)
  {
   min=dist[w];
   v=w;
  }
  flag[v]=1;
  for(w=0;w<n;w++)
  if(!flag[w])
  if(min+cost[v][w]<dist[w])
  {
   dist[w]=min+cost[v][w];
   last[w]=v;
  }
 }
 for(i=0;i<n;i++)
 {
  printf("\n");
  printf("N%d ==> N%d\n",source,i);
 w=i;
  printf("Path Taken : N%d ",i);
  while(w!=source)
  {
   w=last[w];
   printf("<-- N%d ",w);
  }
  printf("\nShortest path cost : %d\n",dist[i]);
```

```
  }
}
```

OUTPUT

```
user@adithya:~/s6/np$ gcc p12.c
user@adithya:~/s6/np$ ./a.out
Enter the number of routers : 3
Enter * for non-neighbouring vertices...
Enter the cost matrix:
        N0      N1      N2
N0      0       1       5
N1      1       0       2
N2      5       2       0
Enter the source router : N2

N2 ==> N0
Path Taken : N0 <-- N1 <-- N2
Shortest path cost : 3

N2 ==> N1
Path Taken : N1 <-- N2
Shortest path cost : 2

N2 ==> N2
Path Taken : N2
Shortest path cost : 0
user@adithya:~/s6/np$
```

RESULT

Implemented and simulated algorithm for link state routing protocol using C.