

Link State Routing Protocol

Aim

To implement and simulate algorithm for link state routing protocol.

Theory

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

Link State Routing Algorithm

1. START
2. START PROCEDURE LinkStateRouting
3. Input the no.of routers and the cost matrix
4. Input the source router
5. For all router v
6. If v is a neighbour of u
7. $\text{dist}[v] = \text{cost}[u][v]$
8. else $\text{dist}[v] = \infty$
9. Iterate N (no of routes in network)
10. Find a node w such that $D(w)$ is minimum
11. $\text{dist}[v] = \min (\text{dist}[v], \text{dist}[w] + \text{cost}[w][v])$
12. print $\text{dist}[v]$ for all node v
13. END PROCEDURE LinkStateRouting
14. STOP

Code

```
#include <stdio.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("\nEnter the number of routers: ");
scanf("%d",&count);
printf("\nEnter the cost matrix values: \n");
for(i=1;i<=count;i++){
for(j=1;j<=count;j++){
scanf("%d",&cost_matrix[i][j]);
if(cost_matrix[i][j]<0)
cost_matrix[i][j]=1000;
}
}
printf("\nEnter the source router: ");
scanf("%d",&src_router);
for(v=1;v<=count;v++){
flag[v]=0;
last[v]=src_router;
dist[v]=cost_matrix[src_router][v];
}
flag[src_router]=1;
for(i=1;i<=count;i++){
min=1000;
for(w=1;w<=count;w++){
if(!flag[w])
if(dist[w]<min){
v=w;
min=dist[w];
}
}
flag[v]=1;
for(w=1;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=1;i<=count;i++){
printf("\n%d==>%d:\n\tPath taken:%d",src_router,i,i);
w=i;
while(w!=src_router){
printf("<--%d",last[w]);
w=last[w];
}
printf("\n\tShortest path cost:%d ",dist[i]);
}
}

```

Output

```
[sachin@sachin ~]$ g++ exp12.cpp
[sachin@sachin ~]$ ./a.out

Enter the number of routers: 3

Enter the cost matrix values:
0 1 5
1 0 2
5 2 0

Enter the source router: 3

3==>1:
    Path taken:1<--2<--3
    Shortest path cost:3
3==>2:
    Path taken:2<--3
    Shortest path cost:2
3==>3:
    Path taken:3
```

Result

implement and simulate algorithm for link state routing protocol in cpp.