Simulation of Link-State Routing Protocol

Sagar Prakash Mane

Illinois Institute of Technology

CWID: A20379756

Smane1@hawk.iit.edu

Contents

Introduction

The goal of this project to develop a simulator to implement Link-State Routing Protocol which shows following functionalities:

- Creating a network topology.
- Simulate the process of generating connection table for each router in a given network.
- Computing optimal path with least cost between any two specific routers.
- Modify a network topology.
- Finding best router for broadcast.
- Adding router in a network topology.

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. Examples of link-state routing protocol include Open Shortest Path First(OSPF) and intermediate system to intermediate system (IS-IS). 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. Link state routing protocols are based on Shortest Path First(SPF) algorithm to find the best path between source and router.

Link State Routing Protocol Algorithm-Dijkstra's Algorithm

Dijkstra's algorithm is used to find shortest path between a source router and a destination router. We maintain. Following are the detailed steps used in Dijkstra's algorithm to find the shortest path from a source vertex to all other vertices in the given graph.

1. Create a set that keeps track of vertices included in shortest path. Initially, this set empty.
2. Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE for all vertices except source vertex.
3. While set doesn't include all vertices.
    - Pick a vertex u which is not there in set has minimum distance value.
    - Include u to set.
    - Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent Vertex v, if sum of distance value of u and weight of edge(u,v) is less than the distance value of v, then update the distance value of v.

Project Functionalities Description

Creating a Network Topology:

The network topology is given by a matrix, called the original topology(graph) matrix, which only indicates the cost of links between all directly connected routers. The program first asked the user for network topology file. It validates the data from file and store that data in matrix format.

Generating Connection Table:

The connection table display interface router from source router to reach all other routers. The program asked user for source router for generating connection table. The program display NONE if there is no path exist or self-router. All interface values are calculated after implementing Dijkstra's algorithm using source router.

Computing Optimal Path:

Optimal shortest path is calculated from source router to destination along with path cost. The program asked user for destination router for generating shortest path. Optimal shortest path is computed by implementing Dijkstra's algorithm.

Modify Network Topology:

Router can be deleted using this functionality. The program asked user for router number which user want to remove. New matrix is generated after removal of router from original network topology matrix and updated connection table display.

Best Router for Broadcast:

Router which has shortest paths to all the other routers in the network along with cost is display. The sum of the costs from this router to all other routers is minimum.

Adding Router:

New router is added using this functionality. The program asked user for router number and distance to other routers from that new router. New matrix is generated after adding new router to original network topology matrix and updated connection table display.

Dijkstra's Algorithm implementation

File Import matrix

Vertex	**1**	**2**	**3**	**4**	**5**
1	0	2	5	1	-1
2	2	0	8	7	9
3	5	8	0	-1	4
4	1	7	-1	0	2
5	-1	9	4	2	0

Let source is 1.

Array minDist[] initially empty and distances assigned to vertices are {0,INF,INF,INF,INF} where INT indicate infinite. Parent[1] is -1.

First iteration
Now pick the vertex with minimum distance value. The vertex 1 is picked, include it in visited[] array. So visited become {1}. After including 1 to visited, update distance values of its adjacent vertices. Adjacent vertices of 1 are 2,3 and 4. The distance values of 2,3 and 4 are updated as 2,5 and 1 (Values are obtained by matrix table). Update parent array by adjacent vertices of 1 i.e parent[2]=1, parent[3]=1 and parent[4]=1.
After first iteration minDist array is {0, 2, 5, 1, INF}

Second iteration
Pick the vertex with minimum distance value and not already included in visited array. The vertex 4 is picked and added to visited. So visited now become {1, 4}. Update the distance values of adjacent vertices of 4. Adjacent vertices of 4 is 1,5 and 2. The distance value of 5 is updated to 3. Update parent array i.e parent[5]=4
After second iteration minDist array is {0, 2, 5, 1, 3}

Third iteration
Pick the vertex with minimum distance value and not already included in visited array. The vertex 2 is picked and added to visited. So visited now become {1, 2, 4}. Update the distance values of adjacent vertices of 2. Adjacent vertices of 2 is 1, 3, 4 and 5. We are not updated minDist array because we select minimum distance to router.
After third iteration minDist array is {0, 2, 5, 1, 3}

Forth iteration
Pick the vertex with minimum distance value and not already included in visited array. The vertex 5 is picked and added to visited. So visited now become {1, 2, 4, 5}. Update the distance values of adjacent vertices of 5. Adjacent vertices of 5 is 2, 3 and 4.
After fourth iteration minDist array is {0, 2, 5, 1, 3}

At this point we get minimum distance to all router from source router. minDist array is {0, 2, 5, 1, 3} and parent array is {-1, 1, 1, 1, 4}

Using minDist and parent array we can find shortest path between source and destination router. Let destination router is 5.
Therefore cost from source router to destination router is minDist[5] which is 3.
For computing path we used parent array until we reach source router
Therefore,
parent[5] is 4
parent[4] is 1
Therefore path is 1→4→5

Test Data and Result

Test 1

Run the program and input case1.txt file

```
*****************************************

CS542 Simulate Link State Routing Protocol

*****************************************


Enter Master Command

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Add Router
(7) Exit

Master Command:
1
Input original network topology matrix data file:
case1.txt
Review original topology matrix:
 0  2  5  1 -1
 2  0  8  7  9
 5  8  0 -1  4
 1  7 -1  0  2
-1  9  4  2  0
```

Generating connection table using source router 1

```
Master Command:
2
Enter a source router:
1
==============================
Router 1 Connection Table
Destination      Interface
==============================
     1             None
     2               2
     3               3
     4               4
     5               4
```

Find the shortest path from source to destination 5

```
Master Command:
3
Enter a destination router:
5
Shortest path from 1 to 5 is 1-->4-->5
Cost is 3
```

Removing router 4

```
Master Command:
4
Enter a router to removed:
4
Review original topology matrix:
==============================
Router 1 Connection Table
Destination      Interface
==============================
     1             None
     2               2
     3               3
     5               3
```

Find best router for broadcast.

```
Master Command:
5
Best Router is 1
Total cost is 16
```

Adding new router

```
Master Command:
6
Enter new Router Number
6
Enter weight between Router 1 and 6
-1
Enter weight between Router 2 and 6
-1
Enter weight between Router 3 and 6
2
Enter weight between Router 5 and 6
1
Enter weight between Router 6 and 1
1
Enter weight between Router 6 and 2
-1
Enter weight between Router 6 and 3
3
Enter weight between Router 6 and 5
4
Review original topology matrix:
==============================
Router 1 Connection Table
Destination      Interface
==============================
     1             None
     2               2
     3               3
     5               3
     6               3
```

Test 2

Run the program and input case1.txt file

```
*****************************************

CS542 Simulate Link State Routing Protocol

*****************************************


Enter Master Command

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Add Router
(7) Exit

Master Command:
1
Input original network topology matrix data file:
case2.txt
Review original topology matrix:
 0 -1 -1 -1 -1
-1  0 -1 -1 -1
-1 -1  0 -1 -1
-1 -1 -1  0 -1
-1 -1 -1 -1  0
```

Generating connection table using source router 2

```
Master Command:
2
Enter a source router:
2
==============================
Router 2 Connection Table
Destination      Interface
==============================
     1            None
     2            None
     3            None
     4            None
     5            None
```

Find the shortest path from source to destination 5

```
Master Command:
3
Enter a destination router:
5
No Path Exits !!!
```

Removing router 3

```
Master Command:
4
Enter a router to removed:
3
Review original topology matrix:
==============================
Router 2 Connection Table
Destination      Interface
==============================
     1             None
     2             None
     4             None
     5             None
```

Find best router for broadcast.

```
Master Command:
5
There is no any best router
```

Test 3

Run the program and input case3.txt file

```
Master Command:
1
Input original network topology matrix data file:
case3.txt
Review original topology matrix:
 0 -1 19 24 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
16  0 -1 -1 25 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 9 -1  0 15 -1 22 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
24  5 15  0 14 17 12 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 25 -1 14  0 -1  8 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 22 17 -1  0 14 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 19  8 14  0  3 12 12 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1  6  0 -1  4 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 12 -1  0  3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 12 -1  3  0 16 -1 -1 12 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 16  0 -1 -1 -1  8 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  1 -1  4 -1 12 -1 -1 50
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  6  0 -1 -1  5 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 12 -1 -1 -1  0 -1 -1 -1  9 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1  8  4 -1 -1  0 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  5 -1 -1  0 -1 -1 39 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 12 -1 -1 -1 -1  0 -1  6 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  9 -1 -1 -1  0 -1 12
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 39  6 -1  0 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 16 -1 -1  0
```

Generating connection table using source router 3

```
Master Command:
2
Enter a source router:
3
==============================
Router 3 Connection Table
Destination      Interface
==============================
     1              1
     2              4
     3            None
     4              4
     5              4
     6              6
     7              4
     8              4
     9              4
     10             4
     11             4
     12             4
     13             4
     14             4
     15             4
     16             4
     17             4
     18             4
     19             4
     20             4
```

Find the shortest path from source to destination 18

```
Master Command:
3
Enter a destination router:
18
Shortest path from 3 to 18 is 3-->4-->7-->8-->10-->14-->18
Cost is 55
```

Find best router for broadcast.

```
Master Command:
5
Best Router is 7
Total cost is 512
```

Test 4

Run the program and input case4.txt file

```
*****************************************

CS542 Simulate Link State Routing Protocol

*****************************************


Enter Master Command

(1) Create a Network Topology
(2) Build a Connection Table
(3) Shortest Path to Destination Router
(4) Modify a Topology
(5) Best Router for Broadcast
(6) Add Router
(7) Exit

Master Command:
1
Input original network topology matrix data file:
case4.txt
Review original topology matrix:
 0 27 -1 20 -1
 9  0 34 19  9
 9 34  0 23 26
 3 29  3  0 29
23 25 30 34  0
```

Generating connection table using source router 1

```
Master Command:
2
Enter a source router:
1
==============================
Router 1 Connection Table
Destination      Interface
==============================
     1             None
     2               2
     3               4
     4               4
     5               2
```

Find the shortest path from source to destination 5

```
Master Command:
3
Enter a destination router:
5
Shortest path from 1 to 5 is 1-->2-->5
Cost is 36
```

Removing router 2

```
Master Command:
4
Enter a router to removed:
2
Review original topology matrix:
==============================
Router 1 Connection Table
Destination      Interface
==============================
     1             None
     3               4
     4               4
     5               4
```

Find the shortest path from source to destination 5

```
Master Command:
3
Enter a destination router:
5
Shortest path from 1 to 5 is 1-->4-->5
Cost is 49
```

Test Case Report

Test case ID	TEST1, TEST2, TEST3, TEST4.
Test case description	Verify Input of network topology file.
Steps	Enter master command 1 and enter txt file name.
Expected Output	Import file successfully and display file result in matrix format.
Actual Result	File import successfully and result display in matrix format.
Status	Pass.

Test case ID	TEST1, TEST2, TEST3, TEST4.
Test case description	Generating connection table.
Steps	Enter master command 1, enter txt file name, enter master command 2 and enter source router number.
Expected Output	Display connection table.
Actual Result	Connection table displayed successfully.
Status	Pass.

Test case ID	TEST1, TEST2, TEST3, TEST4.
Test case description	Find shortest path between source and destination router.
Steps	Enter master command 1, enter txt file name, enter master command 2, enter source router number, enter master command 3 and enter destination router number.
Expected Output	Display shortest path from source to destination router along with cost.
Actual Result	Shortest path displayed from source to destination along with cost.
Status	Pass.

Test case ID	TEST1, TEST2, TEST4.
Test case description	Remove router.
Steps	Enter master command 1, enter txt file name, enter master command 2, enter source router number, enter master command 4 and enter remove router number.
Expected Output	Display connection table without removed router.
Actual Result	Router removed successfully and connection table displayed.
Status	Pass.

Test case ID	TEST1, TEST2, TEST3.
Test case description	Best router to broadcast.
Steps	Enter master command 1, enter txt file name and enter master command 5.
Expected Output	Display the best router along with cost.
Actual Result	Displayed the best router along with cost.
Status	Pass.

Test case ID	TEST1.
Test case description	Adding new router.
Steps	Enter master command 1, enter txt file name, enter master command 6, enter new router number and enter distance between.
Expected Output	New router must be added and display new connection table including new router.
Actual Result	New router added successfully and connection table displayed.
Status	Pass.

Test case ID	TEST2.
Test case description	No path exits between source and destination router.
Steps	Enter master command 1, enter txt file name, enter master command 2 and enter source router number.
Expected Output	Connection table display NONE if no path exits between source and destination router
Actual Result	Connection table displayed NONE value.
Status	Pass.

Test case ID	TEST2.
Test case description	No path exits between source and destination router.
Steps	Enter master command 1, enter txt file name, enter master command 2, enter source router number, enter master command 3 and enter destination router.
Expected Output	Error message display if no path exits between source and destination router.
Actual Result	“No path Exits !!!” message displayed.
Status	Pass.

Test case ID	TEST2.
Test case description	Best router for broadcast.
Steps	Enter master command 1, enter txt file name and enter master command 5.
Expected Output	Error message display if there is no any best router i.e No path exits from source router to other routers.
Actual Result	“There is no any best Router” message displayed.
Status	Pass.

Conclusion

The implemented program of Simulation of Link-State Routing Protocol works for any network topology regardless of size of network. By giving valid data of network topology program will provide you the shortest path between source router and destination router.