Project on

CS542 LINK STATE ROUTING SIMULATOR

Link State Routing Simulation using "Dijkstra's Algorithm"

Submission By

Name: Gupta, Shiksha

CWID: A20380536

Course: CS542: Computer Networks-1

Index

Introduction

Link state is a routing protocol used in packet switching networks for computer communication. The basic concept of link-state routing is that every node creates a map of the connectivity to the network in the form of a graph, showing which nodes are connected to which other nodes. After that each node can independently calculate the best path from it to other nodes in network.

Link-state routing protocols have the reputation of being very complex, even intimidating. Link-state routing protocols are also known as shortest path first protocols and are built around Dijkstra's shortest path first (SPF) algorithm.

In link state routing, four different tasks are required to ensure that each node has the routing table showing the least-cost node to every other node.

1. To create a connection table for the source router.
2. Path-checking of a packet travelling from Source Node to Destination Node with the cost, by taking input the source and destination router from the user.
3. Best router for broadcasting the messages in the topology.
4. Calculation of a routing table based on the shortest path tree.

Dijkstra's algorithm will be used to calculate the direction as well as the shortest path between two routers. Python language is used to implement the Dijkstra's algorithm. The source code has two - dimensional arrays which are used to store original routing table, the distance between routers, values of distance during shortest path calculation, final table. Some integer variables are used to keep a count on the routers. The program interface displays a menu to user who can create a Network Topology, by choosing to input matrix of routing table from file.

Dijkstra's Algorithm

Below are steps used to implement the algorithm:

1. In the starting of the program, first we need to provide the network topology file. It will validate the data and store the data in the matrix format.

2. After storing the data in the matrix format, it is required to create the connection table. To perform the Dijkstra's algorithm, it is required to provide the source router as input.

3. The program follows following steps to implement the Dijkstra's algorithm:

- It considers the source node as the root of the tree and add it to the path. Then it sets the shortest distances for all the neighbors of the root to the cost between the root and those neighbors. Finally, it sets the shortest distance of the root to zero.
- Then it repeats the following two steps in loop until all nodes are added to the path:
    - It looks for the nodes which are not there in the path and selects the one with minimum shortest distance and then add it to the path.

- It then updates the shortest distance for all remaining nodes using the shortest distance of the node which are just moved to the path in the above step.

4. It keeps track of two types of nodes:

- The interface used to go to next router.
- The parent node of last added node.

5. Once both connection table and parent table are ready, the shortest path is found from given source to destination by following ways:

- Starting from the destination node, it follows the parent node from the parent table to reach to the source, and provide the reverse path.
- The total cost is found by adding the cost of all nodes in previous step.

6. If there is no path in given source and destination, the program returns with the message.

System Features

The document provides the basic system features in detail of how it works:

Input Topology File: The system provides its users an option to choose their own topology as an input to the system. It has values of n*n routers along with their distances from each other in form of a matrix.

Default Connection Table: The system has been developed in such a way that it facilitates the user to the Default Connection Tables possessed by source router, which is involved in the topology. This option proves to be very useful if any user needs having a detailed view of the connection table of the source router.

Show Connection Table: This feature is to check the Connection Table for the router number, for which the user wants to view individual connection table by taking as an input.

Modify the original topology: The system asks us about the router where it wants to delete a router in the topology. This helps in modification of the default topology and displays the modified topology connection table thereby showing the results of shortest distance between source and destination.

Shortest Path: The system is designed to find the shortest path between the mentioned Source router to Destination router. This allows user to quickly view the shortest path present between the given input routers.

Broadcasting: Checking which the best router to broadcast it. This is determined by calculating shortest path total from one router to all other routers and then taking the minimum of the values of all the router.

System Design

```
*********************************************************************************
*#################################################################################*
*#################################################################################*
*############   Welcome to CS542 Link State Routing Simulator  ############*
*#################################################################################*
*#################################################################################*
*                                                                                 *
*#################################################################################*
*###### Select the operation to be performed from the below option ######*
*#################################################################################*
*                                                                                 *
*#################################################################################*
*###### (1) Create a Network Topology                                      ######*
*###### (2) Build a forward Table                                         ######*
*###### (3) Shortest Path to Destination Router                           ######*
*###### (4) Change the status of the Router                               ######*
*###### (5) Best Router for Broadcast                                     ######*
*###### (6) Exit                                                         ######*
*#################################################################################*
*********************************************************************************


*********************************************************************************
Enter Choice:
```

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

Input original network topology matrix data file in (.txt) format:

input.txt
**********************************************************************


**********************************************************************
*######################################################################*
*##########             Original Topology Matrix             ##########*
*######################################################################*
-1    28    2    -1    1    -1    -1    -1
-1    -1    -1    9    -1    -1    -1    -1
-1    -1    -1    -1    -1    24    -1    27
-1    -1    -1    -1    -1    -1    8    7
-1    8    2    -1    -1    26    -1    -1
-1    -1    -1    -1    -1    -1    8    -1
-1    -1    -1    -1    -1    -1    -1    7
-1    -1    -1    -1    -1    -1    -1    -1
**********************************************************************


**********************************************************************
Total number of nodes present in the topology matrix:  8
**********************************************************************


**********************************************************************
Final topology matrix dictionary -->  {1: {2: 28, 3: 2, 5: 1}, 2: {4: 9}, 3: {6: 24, 8: 27}, 4: {7: 8, 8: 7}, 5: {2: 8, 3: 2, 6: 26}, 6: {7: 8}, 7: {8: 7}, 8: {}}
**********************************************************************


**********************************************************************
Enter Choice:
```

```
**********************************************************************************
Enter Choice:

2
**********************************************************************************


**********************************************************************************
Enter the source router:

1
**********************************************************************************


**********************************************************************************
Router 1 Connection Table:
Destination Router      Interface
         1               []
         2               [5]
         3               [3]
         4               [5]
         5               [5]
         6               [3]
         7               [5]
         8               [5]
**********************************************************************************


**********************************************************************************
Enter Choice:

|

**********************************************************************************
Enter Choice:

3
**********************************************************************************


**********************************************************************************
Enter the destination router:

2
**********************************************************************************


**********************************************************************************
Minimum Cost from 1 to 2 is 9

Shortest Path from 1 to 2 is [1, 5, 2]
**********************************************************************************


**********************************************************************************
Enter Choice:

|
```

```
**********************************************************************************
Enter Choice:

4
**********************************************************************************


**********************************************************************************
Enter the router to be deleted:

5
**********************************************************************************


**********************************************************************************
Router 1 Connection Table:
Destination Router      Interface
         1               []
         2               [2]
         3               [3]
         4               [2]
         6               [3]
         7               [3]
         8               [3]
**********************************************************************************


**********************************************************************************
Updated shortest path: [1, 2]

Updated shortest distance: 28
**********************************************************************************


**********************************************************************************
Enter Choice:
```

```
********************************************************************************
Enter Choice:

5
********************************************************************************


********************************************************************************
        Router          Total_Cost
         1               156
         2               inf
         3               inf
         4               inf
         6               inf
         7               inf
         8               inf
********************************************************************************


********************************************************************************
Best Router is 1 with lowest cost 156
********************************************************************************


********************************************************************************
Enter Choice:

```



```
********************************************************************************
Enter Choice:

6
********************************************************************************


********************************************************************************
Exit CS542 project. Good Bye!
********************************************************************************
```

Test Cases

Verification of input file before inserting topology

- If entered invalid command, it will display the error message, and prompt for it again.
- If entered command other then 1 to 6, it will display the error message, and exit the command prompt.

Verification of input file before inserting topology

- If invalid file format is entered, it will display the error message, and prompt for new file again.
- If invalid topology is entered, it will display the error message, and prompt for new file again.
- If empty topology file is entered, it will display the error message, and prompt for new file again.

Verification of source router before calculating Connection Table

- If entered invalid source router, it will display the error message, and prompt for it again.

Verification of Destination router before calculating Connection Table

- If entered invalid source router, it will display the error message, and prompt for it again.
- If both source router and destination router are same, it will display the error message, and prompt for it again.

Verification of Modify router before calculating Connection Table

- If entered invalid source router, it will display the error message, and prompt for it again.
- If entered router same as source or destination router already being used, then it will prompt for new start or destination router.

User Manual

Steps to execute the simulator:

- As we cannot run python 3.6 file as executable file.exe format we have to open it in python shell or spyder notebook.
- Open the source code in python shell or spyder notebook.
- Add the input matrix file in the same directory as the source code.
- In the input file, all the elements should be separated by single space.
- Either right click and run as python shell or from the toolbar select Run the code.
- Enter the desired choice to get the required output.

Conclusions

- The implemented program for Link State Routing Simulator works for any network topology regardless of the size of network.
- With every node having partial information about the network topology, it can create shortest path tree for the network.
- Given valid topology data, it will provide you with the shortest path between source router and destination router.
- You can also delete any node excluding source and destination router.
- It will give you the best router which has the lowest cost with every other router in the given topology.

References

- https://en.wikipedia.org/wiki/Routing
- https://en.wikipedia.org/wiki/Software_testing#Beta_testing
- https://en.wikipedia.org/wiki/Link-state_routing_protocol
- https://en.wikipedia.org/wiki/Dijkstra's_algorithm
- https://www.youtube.com/watch?v=gdmfOwyQlcI