CS 542

PROJECT REPORT

LINK STATE ROUTING SIMULATOR

BY

BARANI KUMAR

A20434813

In this project , I have attempted to replicate the working of link state routing protocol with the help of python 2.7

Link state routing protocol :

Link state algorithm basically creates a map of link between each and every router in the network in the form of a graph .This graph clearly shows the exact connectivity link between the routers . Then using the dijkstra's algorithm , it create a shortest path from one router to another which finally results in the routing table

Dijkstra's algorithm

The Dijkstra algorithm is used to generate the shortest path from source to destination router.

The algorithm:

Sets the source router as the root and appends it to the path. And finds out the shortest distance from source router to all other adjacent routers. Finally sets the distance of the source to zero

And it repeats below two steps in a loop until all the routers are added to the path set

1. searches for the neighboring nodes with minimum distance from the source , and appends it to the path

2. sets the shortest distance of the router ,which is recently appended to the path , to the other routers which are in network.

Program Execution :

Note : Since I have coded the program with python 2.7 , please do install python 2.7 before compilation

- Open Command prompt in the project directory
- To run , the program type "python project.py"

Project package :

- Source code -> project.py
- Sample input file -> topology.txt
- A detailed project report -> report.pdf
- Project presentation with work flow -> project_ppt.pptx

Program Implementation:

This is the menu of the program , by selecting the required choice appropriate action will be taken by the program

```
--------------------Link State Routing Simulator-----------------------

|                          1->Create Network                            |
|                    2->Build a Connection Table                        |
|               3->Shortest route from source to destination             |
|                          4->Delete a router                           |
|                            5->Add a router                            |
|                            6->Best Router                             |
|                                7->Exit                                |

------------------------------------------------------------------------

Selection:1

Enter the Input topology file name with (.txt) : topology.txt
```

Step 1 : -

The topology matrix file is received as an input from the user in .txt file format . This file is then converted and stored in a List .

```
Selection:1

Enter the Input topology file name with (.txt) : topology.txt

-----------Original topology matrix-------------

0 2 10 1 -1
2 0 8 7 9
10 8 0 -1 4
1 7 -1 0 4
-1 9 4 4 0
Total number of routers : 5
```

Step 2 :-

Once after storing the link cost between all the routers in the list , the program requests the user to select a source router . Once chosen, then routing table for the source router is generated.

```
Selection:2

Select a source router : 1

Routing Table

Destination     Next Hop
1               None
2               2
3               4
4               4
5               4
```

This routing table is generated with help of Dijkstra's algorithm. Once after the execution of the algorithms, shortest distance from source to all the other routers will be generated and stored in appropriate variables

Step 3 :-

Now, to find the shortest path from one router to another , the program requests the user to choose a destination router . Once chosen, the shortest path and total cost from source router to destination will be displayed.

```
Selection:3

Select a destination router : 3

The shortest path from router 1 to router 3 :  1   4   5   3
The total cost is :  9
```

The path can be found by appending all the routers to the path list from destination router to source router and finally reverse the list. Routers in between the source and destination can be discovered by following the parent node from the parent table.

Step 4 :-

After displaying the shortest path and total cost from source to destination , the user will be prompted will 3 more options -> Delete router , Add a router , Best router

```
4->Delete a router
  5->Add a router
  6->Best Router
```

Step 5 :-

If user chooses to delete a router , the program requests user for the router to be deleted . Once input taken , program access the router matrix and pops the link cost values from the router chosen to other routers.

Now the updated network topology matrix will be displayed

```
Selection:4
Enter the router to be deleted:4

 Modofied topology matrix:

0 2 10 -1
2 0 8 9
10 8 0 4
-1 9 4 0

Routing Table

Destination     Next Hop
1               None
2               2
3               3
4               2

The shortest path from router 1 to router 3 :  1   3
The total cost is :  10
```

And displays the updated shortest path from source to destination router with cost that is already chosen by the user in step 2 and step 3

if the destination router/source router is equal to router which user wants to delete , then the program will display the updated matrix and routing table , and but doesn't show the shortest path and cost

```
Selection:4
Enter the router to be deleted:3

 Modofied topology matrix:

0 2 1 -1
2 0 7 9
1 7 0 4
-1 9 4 0

Routing Table

Destination     Next Hop
1               None
2               2
3               3
4               3

Router  3  is down
```

Step 6 :

If user chooses to add a router , the program requests for the link cost from new router to all the existing router . And it appends the new cost values to the router topology matrix .

And the program displays updated topology matrix ,routing table and shortest path from source to destination with cost that is already chosen by the user in step 2 and step 3.

```
Selection:5
Total number of routers 4
Enter the cost from new router to existing routers with (,) : 1,2,3,4

...........Updated Topology matrix.............
0 2 1 -1 4
2 0 7 9 3
1 7 0 4 2
-1 9 4 0 1
1 2 3 4 0

Routing Table

Destination     Next Hop
1               None
2               2
3               3
4               3
5               3

The shortest path from router 1 to router 3 :  1   3
The total cost is :  1
```

Step 7 :

Once user selects the 6th option , the program displays the optimal router which has minimum distance from all the other routers in network.

```
Selection:6
Total sum of distances for all the routers:
{9: 5, 10: 3, 11: 1, 15: 2}
The Best Router is : 5
```

Typically stores total distance of each router in a dictionary . dictionary is sorted in ascending order and first value is displayed which will be obviously the least sum when compared to other routers

Step 8 :-

Program exits once user chooses 7th option ,

```
Selection:7

Exiting the application
```

Exceptions :

If the users jumps to selection 3 without choosing a source router, the program will calculate and show the shortest path from router 1 to the selected destination router as default .

```
Selection:1

Enter the Input topology file name with (.txt) : topology.txt

-----------Original topology matrix-------------

0 2 10 1 -1
2 0 8 7 9
10 8 0 -1 4
1 7 -1 0 4
-1 9 4 4 0
Total number of routers : 5

---------------------Link State Routing Simulator-----------------------

|                          1->Create Network                            |
|                   2->Build a Connection Table                         |
|               3->Shortest route from source to destination             |
|                          4->Delete a router                           |
|                            5->Add a router                            |
|                            6->Best Router                             |
|                                7->Exit                                |

--------------------------------------------------------------------------

Selection:3

Select a destination router : 5

The shortest route from router 1 to router 5 :  1   4   5
The total cost between router 1 to router 5 : 5
```

If the user's source router and destination router are the same , the program will prompt an error message , and requests user to try again will different router

```
Selection:3

Select a destination router : 1
source router == destination router ,Please do try again with different router
```

Conclusion :

Given valid input ,this program is capable of find out the shortest path from source to destination in a network with N routers