Implementation of Link State Routing Protocol

Veereshwaran Rangasamy Chettiar Ramamoorthi

Dijkstra's Algorithm

- Link state protocols implement Shortest Path First(SPF) i.e., Dijkstra's Algorithm
- Dijkstra's algorithm finds the shortest path from one vertex to all other vertices in a weighted graph.
- It starts from the minimum weight edge from the source vertex and discovers new edges originating from the source vertex of the edge having minimum weight at each step.

Algorithm

- Read the given input topology and put it into a dictionary, mapping the source-destination and its distance value like a routing table for each router given.
- Get a source router(S_r)
- Initialize visited list to empty
- Copy all the routers into unvisited list
- For each router in unvisited:
  - Get the minimum distance router put it into visited list
  - Find whether the distance from the source router (S_r) to the available adjacent routers is smaller than the distance available in the routing table of S_r; if it is smaller, then replace the distance in the routing table of S_r
  - Update the current router as source for all the routers that has updated the distance values
- Compute the path by doing a recursive call on source router of each router until it reaches the source router (S_r)
- To find the path and distance from source S_r to a destination D_r fetch the computed path as well as distance from the route table

Implementation

- Dijkstra’s Algorithm is implemented in python 2.7
- Python file can be run using a python 2.7 interpreter
- python <filename.py>

Output

- The code produces the similar output as follows:

```
================:Menu:================
1.Create a network topology
2.Build a Connection Table
3.Shortest Path to destination router
4.Modify a topology
5.Exit
=====================================
```

Output

- If the user enters 1. Create a network topology, prompt the user to give the topology file. Once user enters filename, topology is displayed

```
Enter a command1
Input original network topology matix data file:new.txt
Review original topology matrix:
0       1       -1      6       -1      -1      3       -1      -1      -1
1       0       3       -1      2       -1      1       -1      -1      -1
-1      3       0       -1      2       5       -1      -1      1       -1
6       -1      -1      0       -1      -1      7       -1      -1      -1
-1      2       2       -1      0       1       -1      1       -1      -1
-1      -1      5       -1      1       0       -1      -1      9       -1
3       1       -1      7       -1      -1      0       9       -1      -1
-1      -1      -1      -1      1       -1      9       0       2       1
-1      -1      1       -1      -1      9       -1      2       0       2
-1      -1      -1      -1      -1      -1      -1      1       2       0
```

Output

- If the user selects 2. Build Connection Table, prompt the user for source router and a connection table is created and displayed.

```
Enter a command2
Enter the source router:2
Router2 Connection Table
Destination     Interface
==============================
0               2->1->0
1               2->1
3               2->1->0->3
4               2->4
5               2->4->5
6               2->1->6
7               2->8->7
8               2->8
9               2->8->9
```

Output

- If the user enters 3. Shortest Path to destination router, ask the user for destination router and print the path as well as distance to the destination router.

```
================:Menu:================
1.Create a network topology
2.Build a Connection Table
3.Shortest Path to destination router
4.Modify a topology
5.Exit
=======================================

Enter a command3
Enter a destination router: 5
Distance is 3 and Path is 2->4->5
```

Output

- If the user enters modify topology, ask the user for down router and rebuild the connection table and display to the user.

```
================:Menu:================
1.Create a network topology
2.Build a Connection Table
3.Shortest Path to destination router
4.Modify a topology
5.Exit
=======================================

Enter a command4
Enter the router to remove8
Router2 Connection Table
Destination     Interface
==============================
0               2->1->0
1               2->1
3               2->1->0->3
4               2->4
5               2->4->5
6               2->1->6
7               2->4->7
9               2->4->7->9
```

Output

- If the user enters 5. Exit, Exit the application

```
================:Menu:================
1.Create a network topology
2.Build a Connection Table
3.Shortest Path to destination router
4.Modify a topology
5.Exit
=======================================

Enter a command5
Exiting the menu
```

Output

- If the user enters any invalid inputs such as giving wrong router number or entering any invalid menu options, appropriate error messages are shown.

Testing

- The code has been tested with various input files.
- The menu options are checked under various scenarios and they gave correct outputs.