11/21/2015

CS 542 Link State Routing Simulator

Sahil Chalke

CWID-A20354354

schalke@hawk.iit.edu

Index:

1. Introduction
2. Link State Routing Protocol.
3. Design of Program
4. Working of the Simulator (User Manual)
5. Conclusion
6. References

**Introduction:**

* Link State routing protocol is used in packet switching networks for computer communication.
* This protocol is used by every switching node in the network. Every node is the router in this case
* The routers construct the topology in the form of graph showing which routers are connected to which other routers in the network.
* The routers then calculate the shortest path to other routers using this topology so that it can successfully deliver the packet from its receiver to the destination.
* Every router forms its own Connection Table which represents the next hop according to the destination router. It will be explained later using example.

**Link State Routing Protocol:**

* Link state routing protocols are routing protocols whose algorithms calculate the best paths to network differently than Distance Vector routing protocols.
* Whereas Distance Vector protocols know routes by measures of distance and vector(direction) as reported by neighbouring routers, Link-State routing protocols calculate their network routes by building a complete topology of the entire network area and then calculating the best path from this topology or map of all the interconnected networks.
* It uses various subsidiary steps shown below to calculate the map of the network to every node.

1. **Determine the neighbours at each node**

First, each node needs to determine what other ports it is connected to, over fully working links; it does this using a reachability protocol which it runs periodically and separately with each of its directly connected neighbours.

1. **Distributing the information for the map**

Next, each node periodically (and in case of connectivity changes) sends a short message, the link-state advertisement, which:

* Identifies the node which is producing it.
* Identifies all the other nodes (either routers or networks) to which it is directly connected.
* Includes a sequence number, which increases every time the source node makes up a new version of the message.

1. **Creating the map**

Finally, with the complete set of link-state advertisements in hand, each node produces the graph for the map of the network. The algorithm iterates over the collection of link-state advertisements; for each one, it makes links on the map of the network, from the node which sent that message, to all the nodes which that message indicates are neighbours of the sending node.

1. **Notes about this stage**

The link-state message giving information about the neighbours is recomputed, and then flooded throughout the network, whenever there is a change in the connectivity between the node and its neighbours.

**Calculating the Connection table:**

* Each router in the network uses Dijkstra’s Algorithm to find its routing table.
* Dijkstra’s algorithm is used to find the shortest path from source node to destination node.
* The connection table is used to find the next hop in the network and it also contains the cost to destination router

**Dijkstra Algorithm:**

* Dijkstra’s algorithm is used to find the shortest path from source router to destination router. Below mentioned are the steps used to find the shortest path from source router to destination router.

1. Select the router as the root. Assign the cost to the neighbouring nodes as the link cost.
2. We now need to search the nodes which are not in the path. Select the one with the minimum shortest distance and it to the path.
3. Update the shortest distance for the remaining nodes using the node which was replaced as the shortest node.
4. Repeat steps 2 and 3 until all the routers are connected.

**Design of Project (Dijkstra’s Implementation):**

* The project uses Dijkstra’s Algorithm to find the minimum path from source to destination using Java.
* The program asks for the topology file-input.txt
* The program stores this file in matrix format and displays the topology matrix.
* Later the user has to choose various options from the menu displayed in the simulator.
* If the option for connection table is selected then the program displays the router forwarding table at each node. The user has the option whether to display the connection table at each node or only at particular node.
* If the user selects the option to find the shortest path between two nodes then the program then the program travels from the source node to the destination node using shortest path link cost traversal until it reaches the destination.

**Additional Features:**

* The project implements three additional features as described below

1. The project works for topologies with more than 8 routers i.e. we can have a topology file with more than 8 routers. It also works for the routers with less than 8 routers.

The working of these two cases will be explained in the further scenarios.

1. If the user selects option 5 then the program will display the connection table of each router at one go. The user is not needed to input the router number to view its connection table as is the case with point 2 which is basic functionalities.
2. The user can modify the link cost between the two routers. The basic assumption is that the routers must be the neighbouring routers in the topology. However, if the routers are not neighbours then the program will create the new link between these two routers which will disturb the original topology.

Once the cost is updated the connection table at each router is printed automatically and the program will ask the user to find the new shortest path between two routers.

The user needs to input the source router and the destination router.

**User Manual**

1. To run the program successfully we need to place the “input.txt” file and the ProjectFinal.jar into one folder.

Below is the sample input.txt file.

0 -1 5 1 -1

-1 0 -1 7 9

5 -1 0 -1 4

1 7 -1 0 2

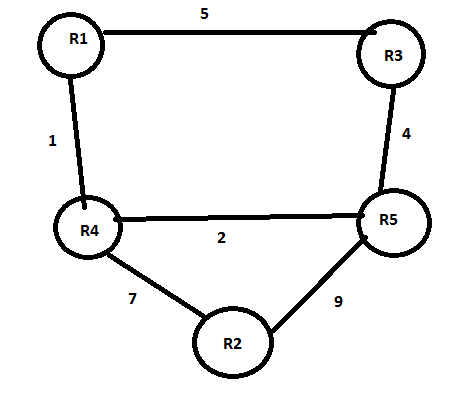
-1 9 4 2 0

1. Open command prompt and navigate to the folder where you have kept the executable jar file

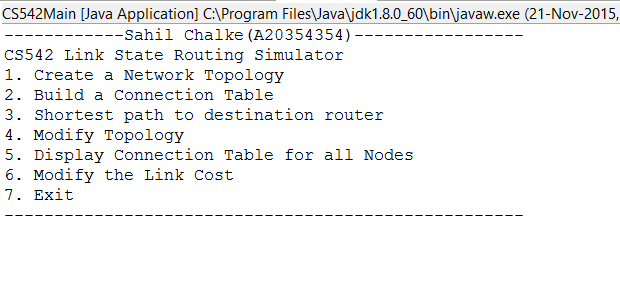
Type ‘java –jar ProjectFinal.jar’ and press Enter.

1. For step by step instruction please refer the Project Report (Part 4-Working of the Simulator)

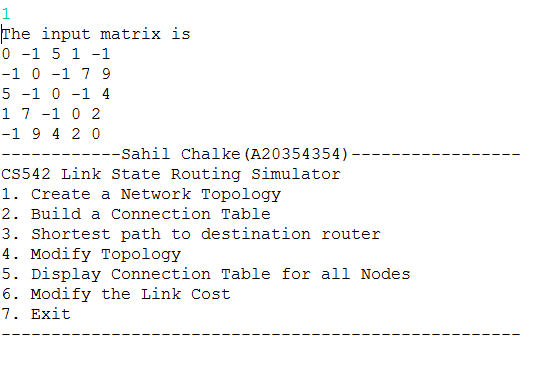
**The sample network topology shown in this user manual is given below:**



1. When you run the project he following options are displayed on the console.



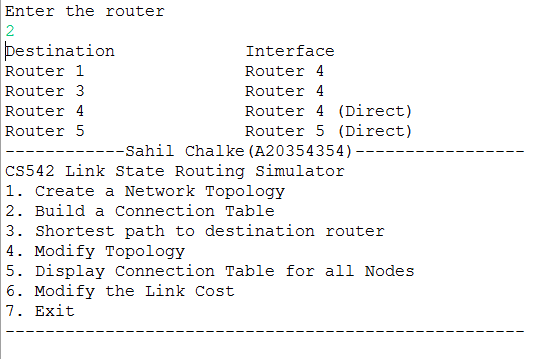
1. **Select option 1**: This option will display the input matrix topology as shown below :



The program will continue to run until the user selects option 7.

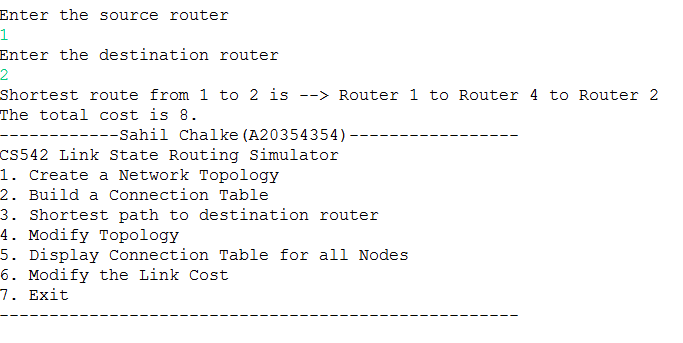
1. Now the user wishes to find the connection table at the particular Router then the user needs to input option 2:

The below screenshot will show this process:



1. To find the shortest path to the destination the user needs to select option 3:

Suppose the user wants to find the shortest path from Router 1 to Router 2 the following options should be selected. This is shown in the demo below:

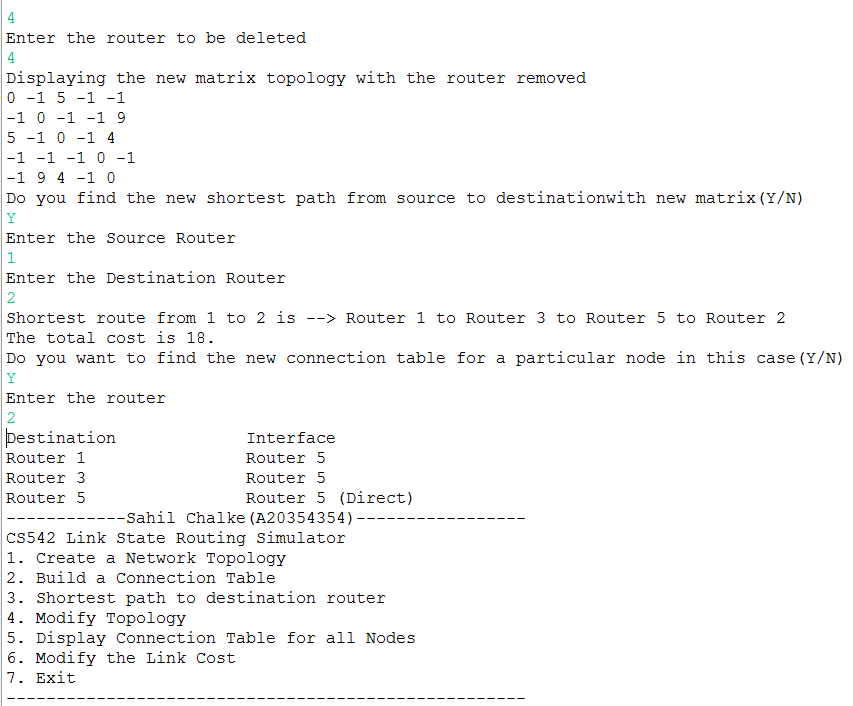


The shortest route will have the path which will be followed by the router and the total cost to reach the destination.

1. **Option 4:** The user needs to input the router number which needs to be deleted. If one of the router is down then the program will automatically modify the topology matrix and will ask the user whether he wants to find the new shortest path in this case. If the user inputs “Y” then he needs to input the source router and destination router. Once this step is completed then the program will ask whether the user needs to find the new connection table in this case. If the user inputs ”Y” then the user needs to input the router number which he wishes to find the connection table.

The process is shown in the below screenshot.

In the below example, if the Router 4 is deleted and we wish to find the shortest path from Router 1 to Router 2.Earlier the shortest path was from Router 1 to Router 4 to Router 2. However, when we delete the router number 4 the shortest path is from Router 1 to Router 5 to Router 2 and the total cost is 18. This is shown in the below figure



**Additional Functionalities:**

1. The simulator will work for routers more than 8 and also for routers which are less than 8.

**3 5**

**2**

**4 7**

**8**

**6**

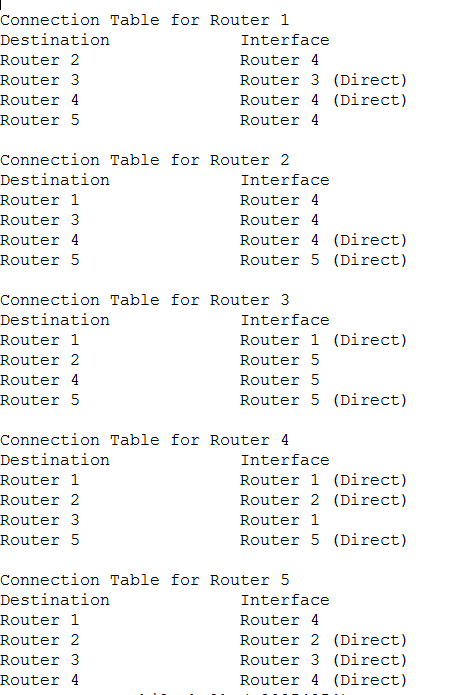
**3 2**

**1**

**2**

**9 5**

1. Select option 5:When the user inputs option 5 the connection table for all the routers will be displayed at one go. This process is displayed below



1. Select option 6: Now if the user wish to modify the link cost then he selects option 6.

Note: For modifying the link cost the two routers must be neighbours in the topology.

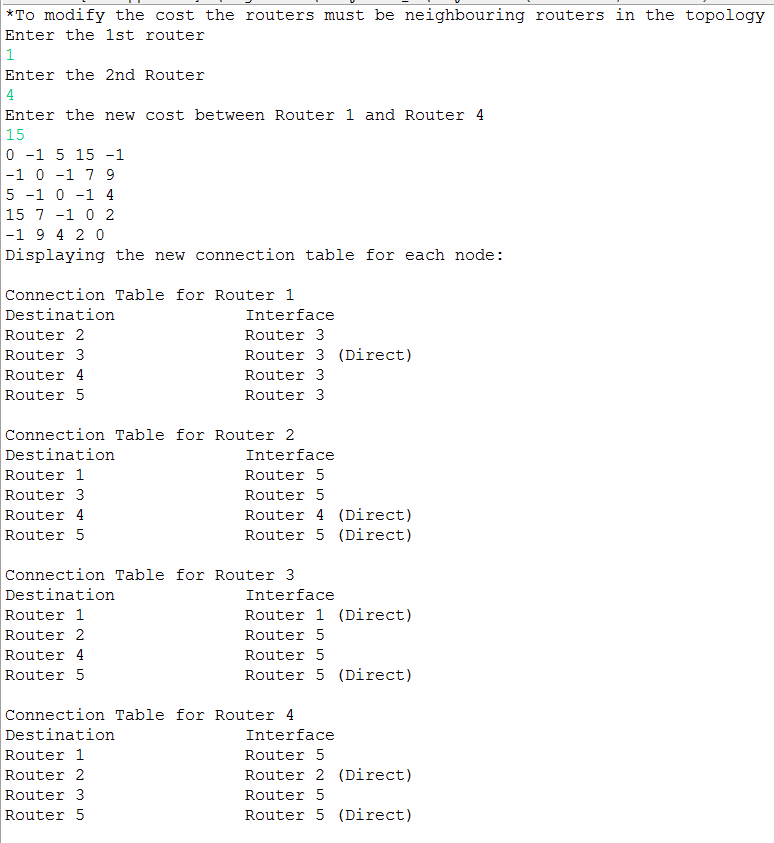
If the two routers are not the neighbours then the program will create the new link between the two routers which will disturb the entire topology.

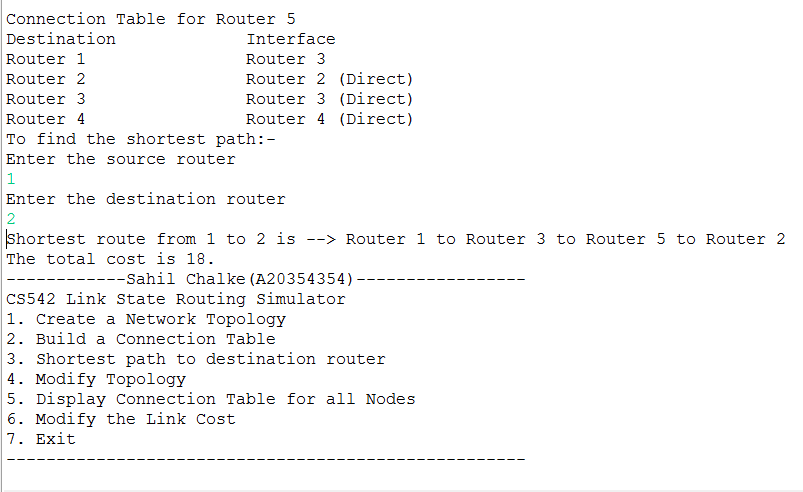
Once the lick cost is updated the program will automatically print the new connection table for all the routers. The program will then ask the user to input the new source router and destination router to get the new shortest path between the two routers.

In the example shown below, if the link cost between routers 1 and 4 is to be modified to 15 then the new connection table is printed for all the routers.

If the user needs to find the new shortest path between router 1 and 2 then it will be through Router 1 to Router 3 to Router 5 to Router 2 and the total cost is 18. This is because the cost from Router 1 to Router 4 is 15 and it cannot be from Router 1 to Router 4 to Router 2 as it will make the total cost 15 + 7 =22.

The entire process is shown below





1. To Exit the program press 7. The Program will terminate.

