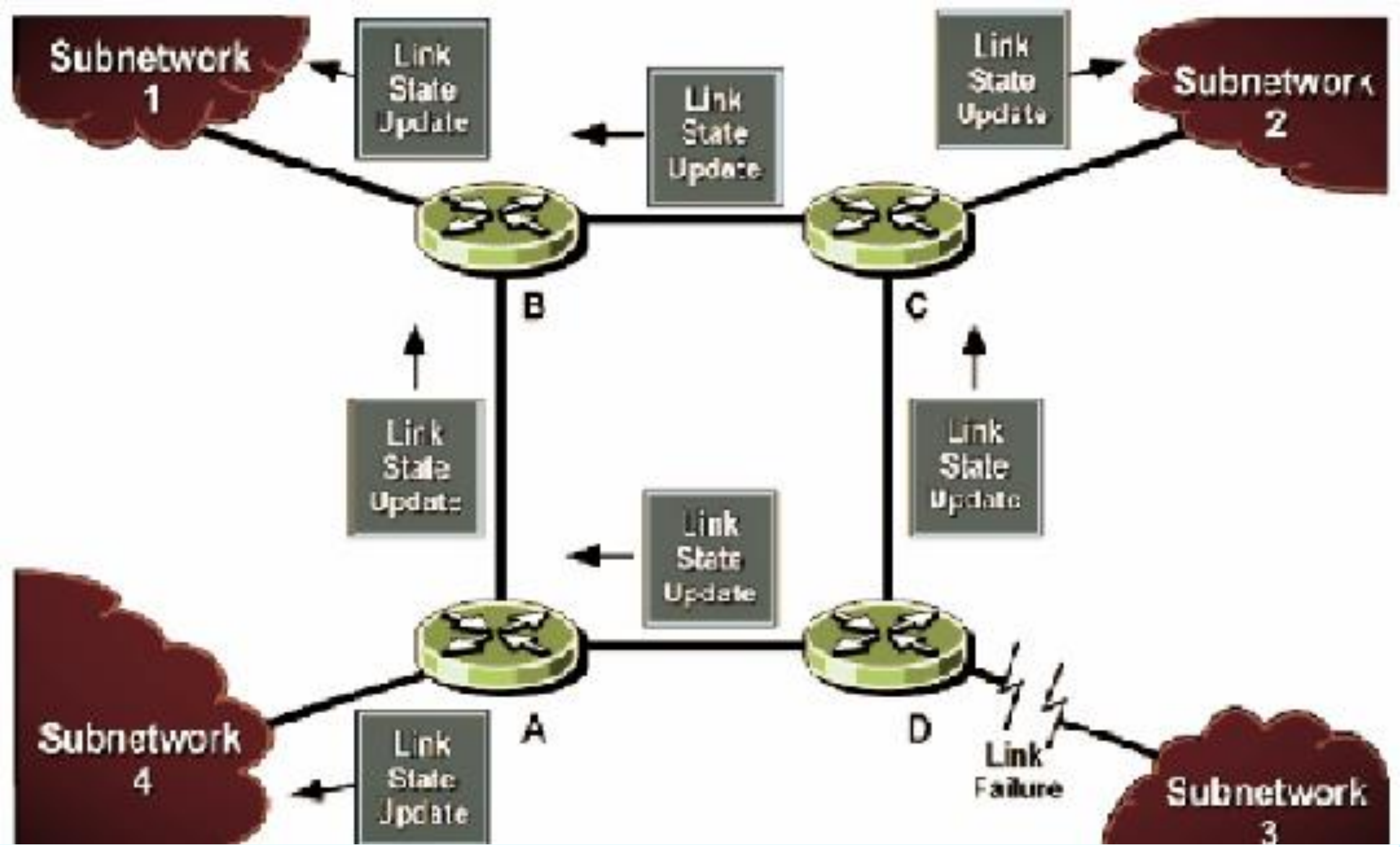
LINK STATE ROUTING

MANAS GUDURI (A20345155)
SAI RAVALI NANNURU (A20354346)
RAJA SEKHAR REDDY VENNA(A20345234)
GAUTAM MISHRA (A20345311)

CS 542 PROJECT

INTRODUCTION: LINK STATE ROUTING PROTOCOL

- Link State Protocols are one of the two main protocols used in packet switching networks.
- Built around E W Dijkstra's algorithm from graph theory which acts as working principle behind protocol.
- Every switching node performs the Link State routing Protocol.
- Every router has information about every other router in the neighbourhood and passes the routing information to others without changing.
- Each router calculates its own best path. For example OSPF (Open Shortest Path First) and ISIS

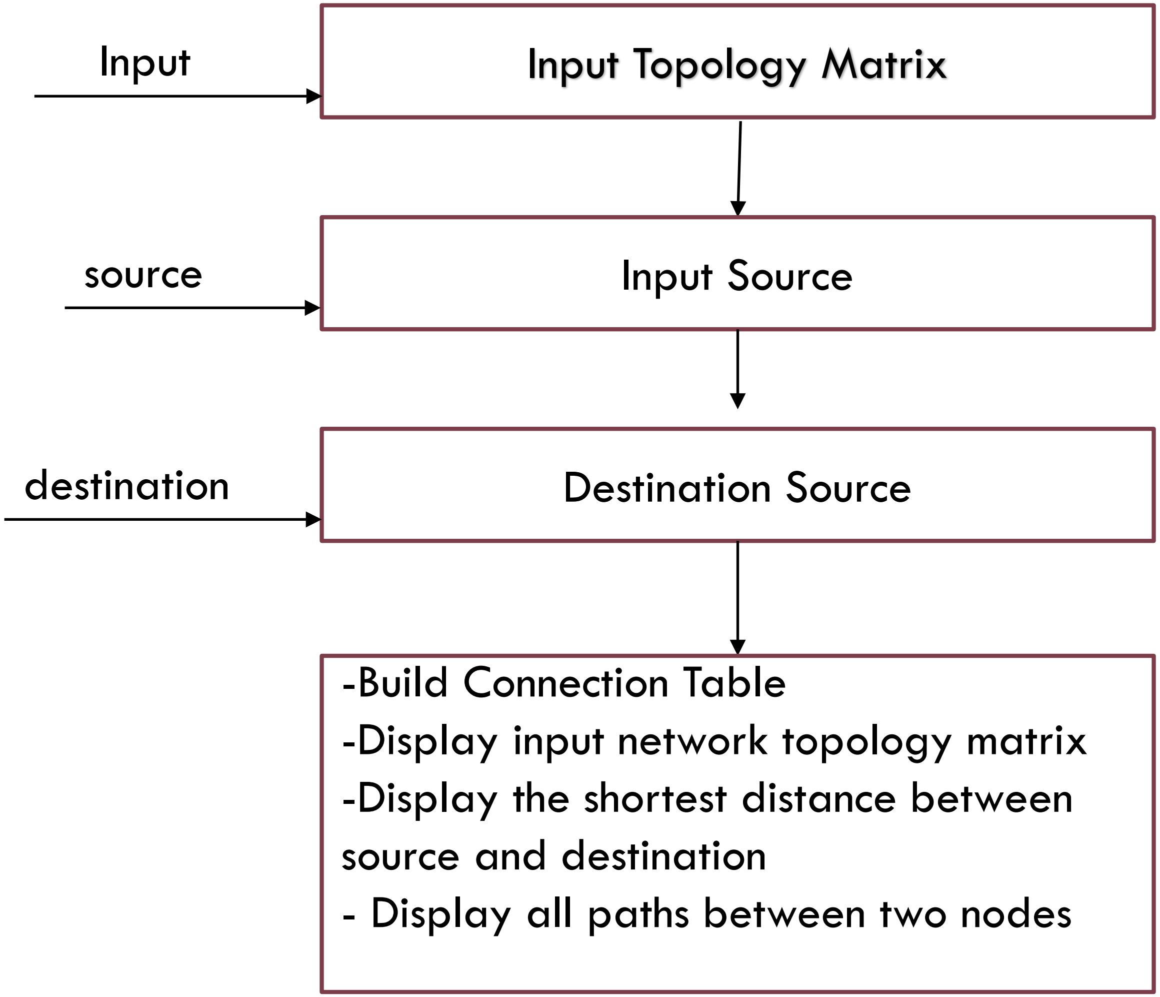


LINK STATE PROTOCOL STRUCTURE

It allows better control of routing updates , decrease computation time and improves the overall performance of the network.

DESIGN IMPLEMENTATION

DESIGN



Input

Input Topology Matrix

source

Input Source

destination

Destination Source

-Build Connection Table

-Display input network topology matrix

-Display the shortest distance between source and destination

- Display all paths between two nodes

Code Implementation

- The main purpose is to implement the Link-State routing Protocol using a program. Goals are:
  - Building a path from one router to another with the given topology matrix.
  - Generating routing tables.
  - Finding the optimal least cost path given source and destination.
  - Using Dijkstra's algorithm, obtain the shortest path as well as the direction.
- In the code, we use 2-D arrays to store original routing table, the distance between routers, values of distance during shortest path calculation, final table after calculation.
- Integers values are used for routers representation, and the number of routers for the connection are limited

Dijkstra’s Algorithm

- Initially, set the cost of the source node as 1 and other nodes as -1.
- Find out the neighboring nodes of the source node and calculate the cost to each of its neighboring node which is calculated by cost to the given current node + cost from current node to the neighbor.
- If the cost calculated is less than the existing cost, then change the cost with the newly calculated value.
- Make the current node as visited and remove it from the list of unvisited nodes.
- Repeat steps 2 and 3 until the destination node is visited.

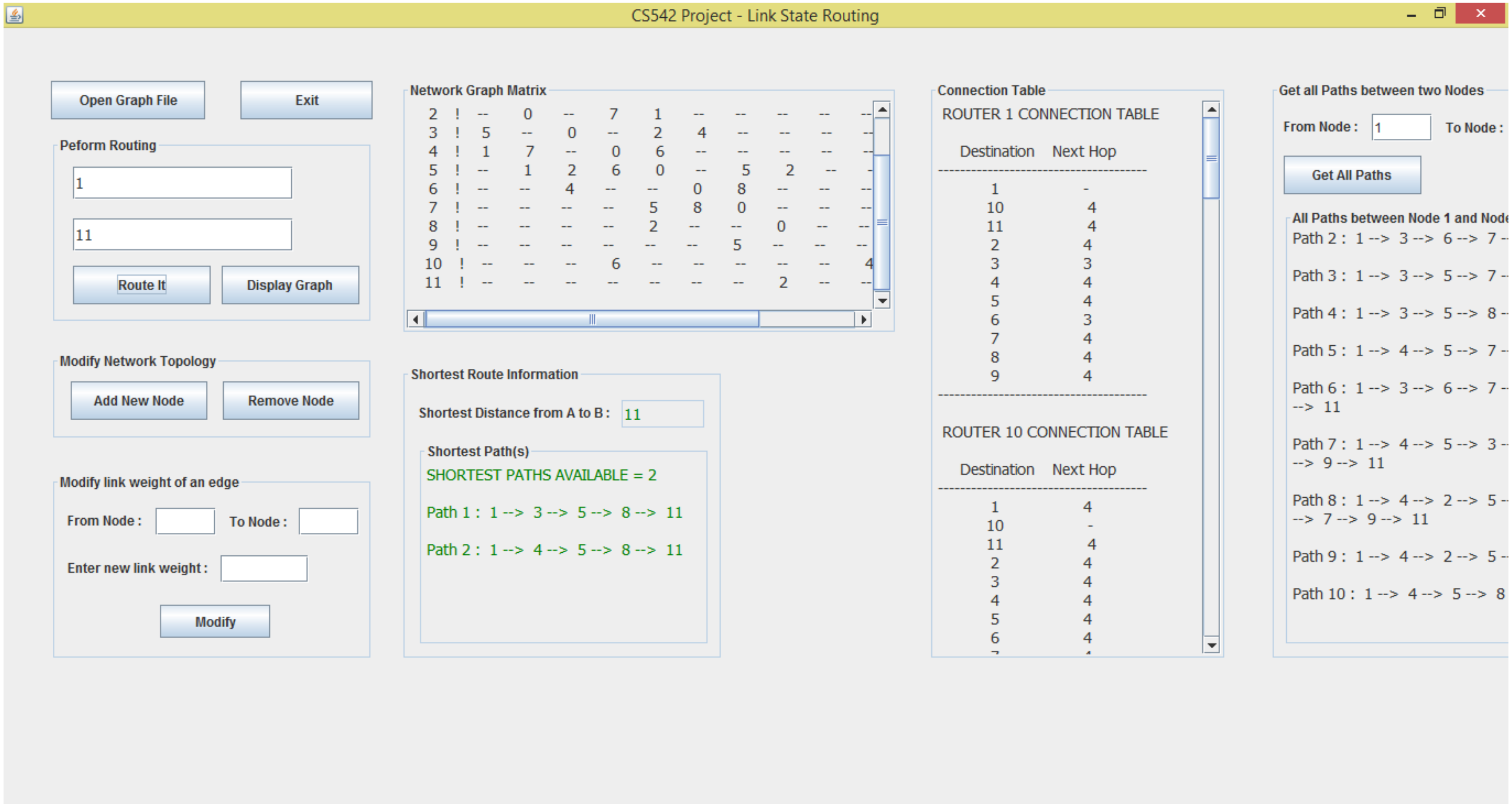
Input File

- □ **The 11*11 Matrix is passed for the input**

0	-1	5	1	-1	-1	-1	-1	-1	-1	-1
-1	0	-1	7	1	-1	-1	-1	-1	-1	-1
5	-1	0	-1	2	4	-1	-1	-1	-1	-1
1	7	-1	0	6	-1	-1	-1	-1	-1	-1
-1	1	2	6	0	-1	5	2	-1	-1	-1
-1	-1	4	-1	-1	0	8	-1	-1	-1	-1
-1	-1	-1	-1	5	8	0	-1	5	-1	-1
-1	-1	-1	-1	2	-1	-1	0	-1	-1	2
-1	-1	-1	-1	-1	-1	5	-1	0	4	-1
-1	-1	-1	6	-1	-1	-1	-1	4	0	-1
-1	-1	-1	-1	-1	-1	-1	2	-1	-1	0

GUI

Graphical User Interface



CS542 Project - Link State Routing

1

11

From Node : To Node :

Enter new link weight :

Network Graph Matrix

```
  2   !   --    0    --    7    1    --    --    --    --    --
  3   !    5    --    0    --    2    4    --    --    --    --
  4   !    1    7    --    0    6    --    --    --    --    --
  5   !   --    1    2    6    0    --    5    2    --    -
  6   !   --    --    4    --    --    0    8    --    --    --
  7   !   --    --    --    --    5    8    0    --    --    --
  8   !   --    --    --    --    2    --    --    0    --    --
  9   !   --    --    --    --    --    --    5    --    --    --
 10   !   --    --    --    6    --    --    --    --    --    4
 11   !   --    --    --    --    --    --    --    2    --    --
```

Shortest Route Information

Shortest Distance from A to B : 11

Shortest Path(s)

SHORTEST PATHS AVAILABLE = 2

Path 1 : 1 --> 3 --> 5 --> 8 --> 11

Path 2 : 1 --> 4 --> 5 --> 8 --> 11

Connection Table

```
ROUTER 1 CONNECTION TABLE

   Destination   Next Hop
---------------------------------------
       1            -
      10            4
      11            4
       2            4
       3            3
       4            4
       5            4
       6            3
       7            4
       8            4
       9            4
---------------------------------------

ROUTER 10 CONNECTION TABLE

   Destination   Next Hop
---------------------------------------
       1            4
      10            -
      11            4
       2            4
       3            4
       4            4
       5            4
       6            4
```

Get all Paths between two Nodes

From Node : 1 To Node :

All Paths between Node 1 and Node

Path 2 : 1 --> 3 --> 6 --> 7 -

Path 3 : 1 --> 3 --> 5 --> 7 -

Path 4 : 1 --> 3 --> 5 --> 8 -

Path 5 : 1 --> 4 --> 5 --> 7 -

Path 6 : 1 --> 3 --> 6 --> 7 -
--> 11

Path 7 : 1 --> 4 --> 5 --> 3 -
--> 9 --> 11

Path 8 : 1 --> 4 --> 2 --> 5 -
--> 7 --> 9 --> 11

Path 9 : 1 --> 4 --> 2 --> 5 -

Path 10 : 1 --> 4 --> 5 --> 8

Characteristics

- SPF algorithm – Link State routing Protocols are designed around Dijkstra’s Shortest Path First (SPF) in which the shortest path from point A to B is build w.r.t a metric of cost.
- Cost metric – SPF finds the shortest path based on a metric network link costs. Each router measures the cost of its own directly connected networks or “links”. Cost is a measure of quality of a link particularly based on the bandwidth.
- Hello Packets – Link State routing Protocol establish adjacencies with neighbouring routers using hello packets.
- LSP (Link State Packets) – Initial flooding of link-states to all routers in the network.
- Topology or SPF tree –Thus Protocols build and maintain a complete map or topology of network area.

Advantages

- Link state routing algorithms prove to be advantageous in case of changing networks by sending triggers which report back the changes that took place in the network.
- Link state algorithms provide major benefits when they are deployed in large, enterprise networks.
- It supports classless routing
- It uses multicasts to share routing information.
- Faster Convergence – Unlike Distance Vector routing protocols which run algorithm calculations before sending updates, they send link-state updates to all routers in the network before running route calculations.

Disadvantages

- Link state routing algorithms consume more CPU time and memory since the routing memory has to maintain more tables and the frequent changes in the network demands more CPU time.
- The link state routing algorithm requires the design of the network to be specific.
- Initial flooding of the network can significantly reduce the network performance.

References

1. Dijkstra’s original paper: E.W.Dijkstra.(1959) A note on Two Problems in Connection with Graphs. Numerische Mathematik,1.269-271.
2. http://en.wikepedia.org/wiki/Linkstate_routing_protocol
3. https://docs.oracle.com/javase/tutorial/deployment/swings/getstarted.html
4. https://reference.wolfam.com/language/Combinatorica/ref/Dijkstra.html
5. https://www.danscourses.com/CCNA-2/link-state-routing-protocols.html
6. http://graphstream-project.org/
7. http://www.gitta.info/Accessibiliti/en/html/Dijkstra_learningObject1.html

Thank you!!!