Link State Routing Protocol

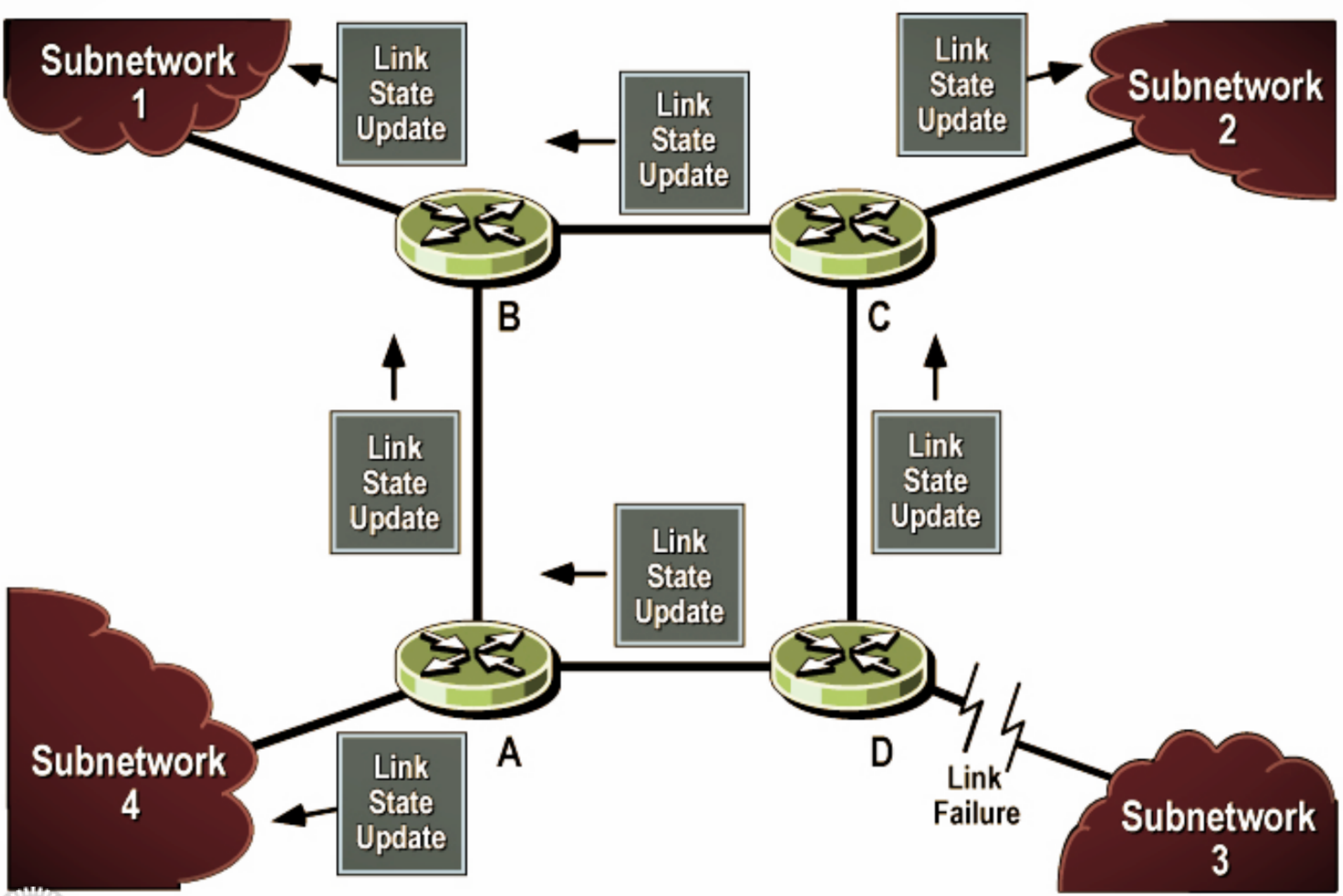
CS542-02 PROJECT

Link State Routing Protocol

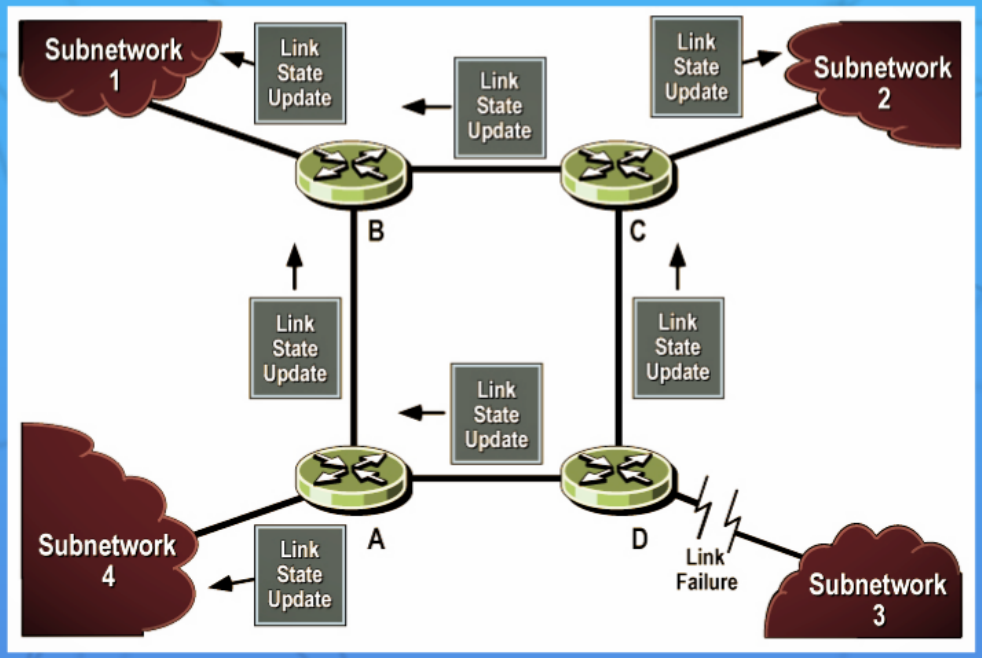
CS542-02 PROJECT

CS542-02 PROJECT

Link State Routing Protocol


Subnetwork 1
Link State Update
Link State Update
Link State Update
Subnetwork 2
B
C
Link State Update
Link State Update
Link State Update
Subnetwork 4
Link State Update
A
D
Link Failure
Subnetwork 3

-Dheeraj Prajwal B V (A20329437)
-Praninder Gupta V (A20330107)
-Sahith Kumar Andel (A20329614)
-Anilraj Chennuru (A20328762)



-Dheeraj Prajwal B V (A20329437)
-Praninder Gupta V (A20330107)
-Sahith Kumar Andel (A20329614)
-Anilraj Chennuru (A20328762)

INTRODUCTION

- 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
- Every switching node performs the Link state Routing Protocol.
- Every router has information about every other router in the neighborhood and passes the routing information to others without changing.
- Each Router calculates its own best path
- Eg: OSPF(Open Shortest Path First)

CODE IMPLEMENTATION

- The main purpose is to implement the Link-State routing protocol using a program. The program's main goals are as follows:
  1) Building a path from one router to another with the given topology matrix.
  2) Generating routing tables .
  3) Finding the optimal least cost path given source and destination
  4) Using Dijkstra's algorithm, obtain the shortest path as well as the direction .
- Java and Applets are used to obtain both the logical and graphical representation of this project.
- In the code, we use 2-dimension arrays to store original routing table, the distance between routers, values of distance during shortest path calculation, final table after calculation.
- Integer values are used for routers representation;
- the numbers of routers are limited to 100 routers.
- The program interface displays an applet through which output is shown

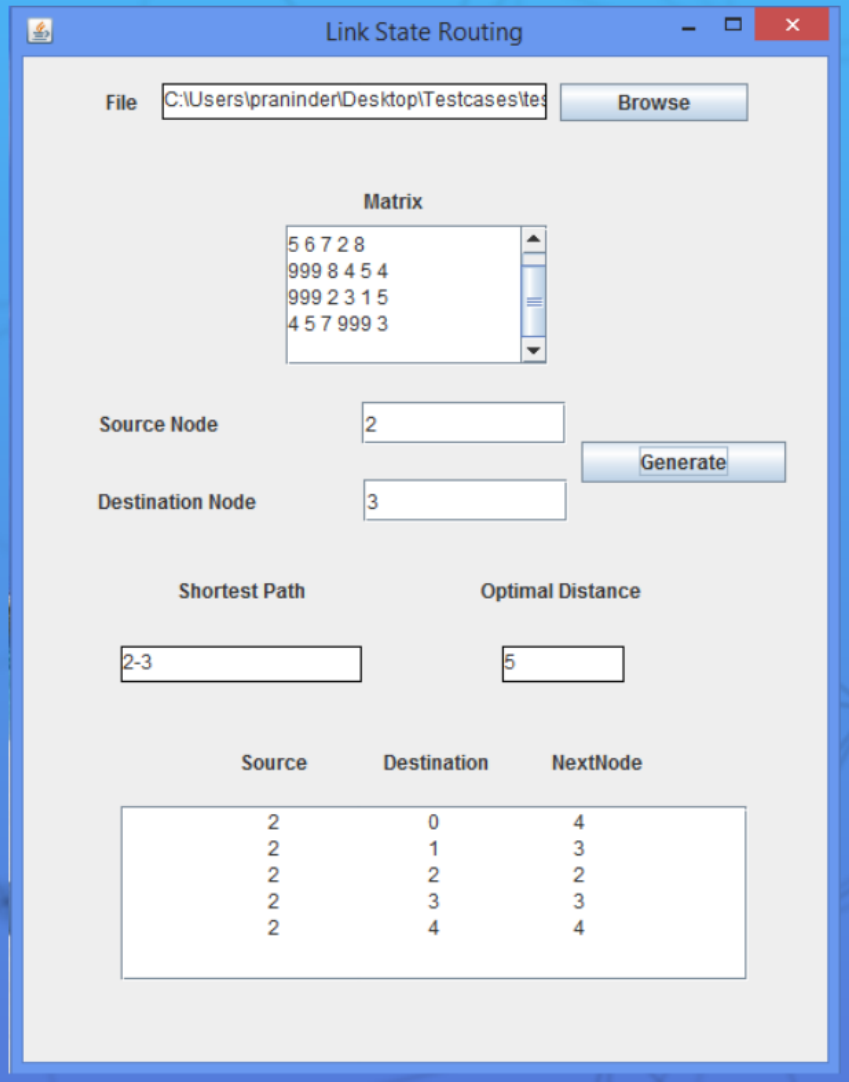
CODE IMPLEMENTATION

- The main purpose is to implement the Link-State routing protocol using a program. The program's main goals are as follows:
  1) Building a path from one router to another with the given topology matrix.
  2) Generating routing tables .
  3) Finding the optimal least cost path given source and destination
  4) Using Dijkstra's algorithm, obtain the shortest path as well as the direction .
- Java and Applets are used to obtain both the logical and graphical representation of this project.
- In the code, we use 2-dimension arrays to store original routing table, the distance between routers, values of distance during shortest path calculation, final table after calculation.
- Integer values are used for routers representation;
- the numbers of routers are limited to 100 routers.
- The program interface displays an applet through which output is shown

ALGORITHM

- Initially set min to a large value which is equivalent infinity
  min=999;
- Store the Source and Destination routers
  src=x; des=y;
- Check whether min is greater than Distance to a particular node and the node is not visited
  if(min>distance[k][j] && visited[k][j]!=1)
- If this is satisfied, min value is updated to the lesser cost distance
  min=distance[k][j];
- Minimum distance node is assigned to nextnode.
  nextnode=j;
- Visited from source to next node is changed to 1.
  visited[k][nextnode]=1;
- If the given node is not visited then check whether the min value and cost for nextnode is less than distance
  if(visited[k][c]!=1)
  if(min+matrix[nextnode][c]<distance[k][c])
- If the condition satisfies the distance is updated with min + cost of the nextnode and pred stores the path traversed(Shortest path).
  distance[k][c]=min+matrix[nextnode][c];
  pred[k][c]=nextnode;

Output



DEFAULT GATEWAY

Link-State Characteristics

- SPF algorithm - Link-State routing protocols are designed around Dijkstra's Shortest Path First Algorithm (SPF) in which the shortest path from point A to point B is build around a metric of cost.
- Cost metric - SPF algorithm 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 the quality of a link based mostly on bandwidth.
- Hello packets - Link-State routing protocols establish adjacencies with neighboring routers using hello packets.
- Link State Packets (LSP) - Initial flooding of link-states to all routers in the network.
- Topology or SPF Tree - Link-State routing protocols build and maintain a complete map or topology of the network area.

Link-State Advantages

- Faster Convergence - Unlike Distance Vector routing protocols which run algorithm calculations before sending updates, Link-State routing protocols send link-state updates to all routers in the network before running route calculations
- Triggered Updates - Unlike Distance Vector routing protocols which send periodic updates at regular intervals, Link-State routing protocols send LSPs during router startup (flooding) and when a link changes states like going up or down. If their are no changes in the network the protocol only sends hello packets to maintain adjacencies.
- Scalability - Link-State routing protocols support the ability to configure multiple routing "areas" which allows an administrator to segment a routing protocol processes to defined areas which supports the expansion and troubleshooting of much larger networks.

LInk State Disadvantages

- Greater Processing Requirements - Link-State routing protocols typically demand greater processing power and memory resources from the router.
- Greater Administrator Knowledge - Link-State routing protocols can demand advanced administrator knowledge to configure and troubleshoot the network area

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]MIT OpenCourseware, 6.046J Introduction to Algorithms.
< http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-046JFall-2005/CourseHome/> Accessed 4/25/09
[3]Department of Mathematics, University of Melbourne. Dijkstra's Algorithm.<http://www.ms.unimelb.edu.au/~moshe/620-261/dijkstra/dijkstra.html Accessed 4/25/09
[4]http://en.wikipedia.org/wiki/Linkstate_routing_protocol
[5]https://www.google.com/?gws_rd=ssl
[6]https://docs.oracle.com/javase/tutorial/deployment/applet/getStarted.html
[7] http://reference.wolfram.com/language/Combinatorica/ref/Dijkstra.html
[8]http://www.danscourses.com/CCNA-2/link-state-routing-protocols.html

Graphical Representation

CONNECTION.OPEN ( )

NODE Created( )

NODES CONNECTED( )

Shortest Path Algorithm ( )

Routing Table Generated()

SHORTEST PATH FOUND

OPTIMAL DISTANCE
CALCULTED

Shortest Path Algorithm ( )

Routing Table Generated()

SHORTEST PATH FOUND

OPTIMAL DISTANCE

SHORTEST PATH FOUND

OPTIMAL DISTANCE
CALCULTED

Connection.Close( )

Connection.Close( )