CS-542 PROJECT REPORT

LINK STATE ROUTING PROTOCOL

By-Bhargavi Deshpande

A20395720

Contents

1.Introduction

In the seven-layer OSI model of computer networking, the **network layer** is **layer 3**. The network layer is responsible for packet forwarding including routing through intermediate routers.

The network layer provides the means of transferring variable-length network packets from a source to a destination host via one or more networks. Within the service layering semantics of the OSI network architecture, the network layer responds to service requests from the transport layer and issues service requests to the data link layer.

Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface. *Routing* refers to the network-wide process that determines the end-to-end paths that packets take from source to destination. The network layer must determine the route or path taken by packets as they flow from a sender to a receiver. The algorithms that calculate these paths are referred to as **routing algorithms**.

Every router has a **forwarding table**. A router forwards a packet by examining the value of a field in the arriving packet's header, and then using this header value to index into the router's forwarding table. The value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded. Depending on the network-layer protocol, the header value could be the destination address of the packet or an indication of the connection to which the packet belongs

Functions of the network layer include:

Connectionless communication

For example, IP is connectionless, in that a data packet can travel from a sender to a recipient without the recipient having to send an acknowledgement. Connection-oriented protocols exist at other, higher layers of the OSI model.

Host addressing

Every host in the network must have a unique address that determines where it is. This address is normally assigned from a hierarchical system. On the Internet, addresses are known as IP addresses.

Message forwarding

Since many networks are partitioned into subnetworks and connect to other networks for wide-area communications, networks use specialized hosts, called gateways or routers, to forward packets between networks.

2.Link State Routing Protocols

Link-state routing protocols are one of the two main classes of routing protocols used in packet switching networks for computer communications, the other being distance-vector routing protocols.

Distance Vector router tells ONLY neighbors about ALL routes.

Link-State router tells ALL other routers about ONLY its neighbors and links Examples of link-state routing protocols include: Open Shortest Path First (OSPF) for IP, The ISO's Intermediate System to Intermediate System (IS-IS) for CLNS (Connectionless-mode Network Service) and IP DEC's DNA Phase V ,Novell's NetWare Link Services Protocol (NLSP).

Traditional Distance Vector Routing Protocols (DVRP) relay information regarding their relative distance to a destination. Link State Protocols relay specific link characteristics and state information. Only changes or updates are sent across the network. Each router uses that information to build a routing table on its own.

The link-state protocol is performed by every switching node in the network. The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. Each collection of best paths will then form each node's routing table.

Each router creates a Link State Packet (LSP). LSPs are flooded reliably within the network. The LSPs are collected by each router to form a Link State Database (LSDB) or complete "picture" of the network. Link state Protocol uses Shortest Path First (SPF) algorithm to put the pieces together.

Advantages of a Link State Protocol

- It uses metrics (costs) to calculate path
- Typically displays faster convergence than distance vector routing protocols

- Typically, more scalable due to hierarchical nature

Disadvantages of a Link State Protocol

- Require more memory to store state information, i.e. databases
- Increased complexity in designing networks

3.Application Design

The objective of the application is to performs the following tasks

- To simulate the process of generating connection table for each router in given network.
- To compute the optimal path with least cost between any two specific routers

The application is designed in python2.7.

The application provides the following options to the user

(1) Create a Network Topology

(2) Build a Forward Table

(3) Shortest Path to Destination Router

(4) Modify a Topology (Change the status of the Router)

(5) Best Router for Broadcast

(6) Exit

- The Algorithm used in the project is Dijkstra's Shortest Path Algorithm.
- The Algorithm used is a modified version of the original algorithm which takes one source and calculates the shortest path for all the remaining vertices.
- The algorithm I've used accepts a matrix from the user, validates if the matrix is of N*N.
- Then creates a routing table of all the nodes present in the matrix. The weight to be entered if a path does not exist between two nodes in -1 and the cost of the path from the node to the same node is 0. Below is the example of a matrix with 4 nodes.

	R1	R2	R3	R4	R5
R1	0	2	5	1	-1
R2	2	0	8	7	9
R3	5	8	0	-1	4
R4	1	7	-1	0	2
R5	-1	9	4	2	0

Table: A sample original topology matrix

- Once the matrix is updated then a routing table which shows the next hop from a source to all destinations are shown. This is done for all the given nodes.
- This option can be chosen separately by selecting a source router where in the table is shown for only the router chosen but also selects the node as a source.
- The destination router can be then selected and an optimal cost and the shortest route will be displayed.
- There is an option to delete a node once the source and destination have been set if a node is deleted which forms the path of the algorithm then an updated table for the selected router, updated matrix, and a new shortest path and cost will be displayed.
- If a node is deleted then it is considered as down then the matrix will be updated such that the node's row and column will be -1. If 5 is deleted then it looks as follows.

	R1	R2	R3	R4	R5
R1	0	2	5	1	-1
R2	2	0	8	7	-1
R3	5	8	0	-1	-1
R4	1	7	-1	0	-1
R5	-1	-1	-1	-1	-1

- Dijkstra's algorithm is used to calculate the shortest path and the cost for that route.

4.Dijkstra's Algorithm:

Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree.

For a given source node in the graph, the algorithm finds the shortest path between that node and every other. It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path algorithm is widely used in network routing protocols, most notably IS-IS (Intermediate System to Intermediate System) and Open Shortest Path First (OSPF).

Pseudo Code:

```
function Dijkstra's(Graph, source, dest):

for each vertex v in Graph:              // Initialization

dist[v] := infinity                      // initial distance from source
                                            to vertex   v is set to infinite

previous[v] := undefined                 //Previous node in optimal path from
                                            source.

Dist[source]:=-1                         //Distance from source to source

Q:=the set of all nodes in graph         //All nodes in the graph are un optimized
                                           thus are in Q.

while Q is not empty:                    //main loop

    u:= node in Q with smallest dist[]

    remove u from Q
```

```
  for each v of u:                    //where v has not yet been removed from Q
      alt:= dist[u] +dist_between(u,v)
      if alt<dist[v]                          //Relax (u,v)
         dist[v]:=alt
         previous[v]:=u
return previous[ ]
```

- Initialize all other nodes with distance "infinite"; initialize the starting node with 0
- Mark the distance of the starting node as permanent, all other distances as temporary.
- Set start node as active.
- Calculate the temporary distances of all neighbor nodes from the active node by summing up its distance with the weights of the edges.
- If such a calculated distance of a node is smaller than the current one, update the distance and set the current node as antecessor. This step is also called update and is Dijkstra's central idea.
- Setting of the node with the minimal temporary distance as active. Mark its distance as permanent.
- Repeating of steps 4 to 7 until there aren't any nodes left with a permanent distance, which neighbors still have temporary distances.

5.Test Cases:

Choice 1:

- When option1 is selected the file name should be entered if not found No such directory or file exists message is prompted. If the file is found the topology matrix is displayed.

```
Command Prompt - python cn.py
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\venkatesh>cd pythonprog

C:\Users\venkatesh\pythonprog>python cn.py
Please enter 1 of the following options
 1: To create topology
 2: get the instances
 3: To get the shortest pasth
 4: To delete a router
 5: To find the most accisable router
 6: Exit
 Here:1
Please enter the file name
Here:topology.txt
Review original topology matrix:
0 2 4 1 5 -1 -1 -1 -1 -1
1 0 1 1 1 1 1 1 1 -1
6 2 0 4 4 2 8 4 5 1
7 4 5 0 9 4 -1 4 8 9
5 4 8 -1 0 8 4 7 4 -1
-1 4 -1 5 9 0 9 4 3 1
-1 4 3 -1 -1 5 0 9 3 1
-1 3 4 2 -1 9 -1 0 3 1
1 2 3 4 5 6 7 8 0 -1
-1 -1 -1 -1 3 4 1 4 9 0
```

Choice2:

- This choice allows the user to enter the source router and select an router from existing list of routers. Shows error if router is not in list or is down.
- The connection table for the selected source router is displayed.

```
Please enter 1 of the following options
 1: To create topology
 2: get the instances
 3: To get the shortest pasth
 4: To delete a router
 5: To find the most accisable router
 6: Exit
 Here:2
Enter source router: 1
Destination      Interface
===========================
         1           None
         2           2
         3           2
         4           4
         5           2
         6           2
         7           2
         8           2
         9           2
         10          2
```

Choice 3:

- The choice 3 will allow you to enter a destination router

- Error will be shown if router selected is not in the list of routers or is down
- If choice 2 is not selected before choice 3 then prompts to enter choice 2 first.
- If a valid destination router is entered then shortest path from source and destination is calculated and cost is displayed
- Press ‘Y’ if you wish to test other router or press any other key to return to the main menu.

```
Please enter 1 of the following options
 1: To create topology
 2: get the instances
 3: To get the shortest pasth
 4: To delete a router
 5: To find the most accisable router
 6: Exit
 Here:3
Enter end router: 8
8
2
1 -> 2 -> 8 -> and the total cost is 3

Press 'Y' to test other router (or) type anything for main menu
```

Choice 4:

- This choice will allow user to delete a router.
- The router which is selected is made down
- If a source and destination is selected then the updated routing table for source is shown and the updated cost and path is calculated.

```
Press 'Y' to test other router (or) type anything for main menuf
Please enter 1 of the following options
 1: To create topology
 2: get the instances
 3: To get the shortest pasth
 4: To delete a router
 5: To find the most accisable router
 6: Exit
 Here:4
Source is :  1  End is  8
Please select the router to make it down: 5
New topology
0 2 4 1 -1 -1 -1 -1 -1 -1
1 0 1 1 -1 1 1 1 1 -1
6 2 0 4 -1 2 8 4 5 1
7 4 5 0 -1 4 -1 4 8 9
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 4 -1 5 -1 0 9 4 3 1
-1 4 3 -1 -1 5 0 9 3 1
-1 3 4 2 -1 9 -1 0 3 1
1 2 3 4 -1 6 7 8 0 -1
-1 -1 -1 -1 -1 4 1 4 9 0
8
2
1 --> 2 --> 8 --> And the total is 3
```

Choice 5:

- Show the router which has shortest paths to all the other routers in the network along with the cost.
- The sum of the costs from this router to all other routers is calculated which is the minimum. If more than one router satisfies the criteria, choose one arbitrarily. This choice will cause the system to exit.

```
Please enter 1 of the following options
 1: To create topology
 2: get the instances
 3: To get the shortest pasth
 4: To delete a router
 5: To find the most accisable router
 6: Exit
 Here:5
router is 2 And total is 9
```

Choice 6:

- Exits the program.

```
Please enter 1 of the following options
 1: To create topology
 2: get the instances
 3: To get the shortest pasth
 4: To delete a router
 5: To find the most accisable router
 6: Exit
 Here:6
thanks for the business

C:\Users\venkatesh\pythonprog>
```

References:

- https://www.techopedia.com/definition/24204/network-layer
- https://electronicspost.com/forwarding-and-routing-in-network-layer/

- https://en.wikipedia.org/wiki/Link-state_routing_protocol
- https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- https://www.webopedia.com/TERM/R/routing.html
- http://www.ciscopress.com/articles/article.asp?p=24090&seqNum=4
- http://www.gitta.info/Accessibiliti/en/html/Dijkstra_learningObject1.html