COEN 146: Computer Networks
Lab 7: Link state routing

Objectives

1. To develop link state (LS) routing algorithm

Network topology and data

A network topology is defined by a set of nodes N and a set of edges E, formally defined as:

- N = set of nodes (routers)
- E = set of edges (links between connecting nodes)

It is assumed that the following information will be available for a network.

- Router ID, which is its index into the tables below, is given at the command line.
- Number of nodes, N, in the topology will be given by the command line.
- Table with costs C, NxN, will be obtained from file1 (name given at the command line).
- Table with machines, names, IP addresses, and port numbers, Nx3, will be obtained from file2 (name given at the command line).

Your main data will be:

- Neighbor cost table – contains the cost from every node to every node, initially obtained from file1.
- Least cost array – obtained with the link state algorithm.

LS routing protocol

Each router iteratively runs LS to find its forwarding table to every other node in the network (i.e. the shortest path tree) using Dijkstra algorithm. The pseudo code as described in the class textbook, is given as follows:

```
Data:
        define N, E, C, and "empty" Nprime arrays
        set source node to u
        Define D(v) cost of path from source u to dest v
        Define p(v) predecessor node along path from u to v

Initialization:
        Nprime  ← u
        for each node v in N:
            If v adjacent to u
                D(v) ← C(u,v )
                p(v) ← u
          else
                D(v) ← INFINITY

        D(u) ← 0                // Distance from source to source
Loop:
        while all nodes not in Nprime:
            node ← node in N with min D(node)   // node with the least distance neighbor will be selected first
            remove node from N and add to Nprime
            update D(v) for all v adjacent to node and not in Nprime
                        D(v) ←  min (D(v), D(node)+C(node, v)(
                        p(v) ←  v (if D(v) is minimum, or node if D(node)+c(node, v) is minimum
        Repeat
```

LS Code at each router

You may build your code with 3 threads, each with the following task:

- Thread 1 loops forever. It receives messages from other nodes and updates the neighbor cost table. When receiving a new cost c from x to neighbor y, it should update the cost in both costs: x to y and y to x.
- Thread 2 reads a new change from the keyboard every 10 seconds, updates the neighbor cost table, and sends messages to the other nodes using UDP. It finishes 30 seconds after executing 2 changes. You may execute this part in the main thread.

- Thread 3 loops forever. It sleeps for a random number of seconds (10-20), run the algorithm to update the least costs. After the algorithm executes it outputs the current least costs.

Make sure to use a mutex lock to synchronize the access to the neighbor cost table.

The messages between the routers will have 3 integers:
<Routers' ID><neighbor ID><new cost>

The input for Thread 2, will have two integers:
<neighbor><space><new cost><new line>

The table with costs will look like this if N = 3:
<0,0> <0,1> <0,2>
<1,0> <1,1> <1,2>
<2,0> <2,1> <2,2>
where <i,j> represents the cost between node i and node j. If <i,j> is equal to infinite (defined as 1,000), nodes i and j are not neighbors.

The table with machines will look like this, if N = 3.
<machine0> <IP0> <port>
<machine1> <IP1> <port>
<machine2> <IP2> <port>

Requirements to complete the lab

Demonstrate to the TA LS routing for a network of your choice and upload source code to Camino.

Be sure to retain copies of your source code. You will want these for study purposes and to resolve any grading questions (should they arise)

Please start each program with a descriptive block that includes minimally the following information:

```
/*
* Name: <your name>
* Date:
* Title: Lab7 - ….
* Description: This program … <you should
* complete an appropriate description here.>
*/
```