CSC450 – Computer Networks

Project: Link-State & Distance-Vector Routing **DUE DATE: 11/13/20**

In this project, you are to implement:

a. Dijkstra's algorithm to compute the shortest path tree and the cost of the least-cost paths for a given node in the network.
b. Distance-vector algorithm to compute distance vector for each node in the network.

Requirements:

1. First, your program must read the costs of all the links in the network from *topology.csv* file. In this file, the first row and the first column refer to the names of the nodes. Other cells show the cost of links between the row node and the column node (meaning two nodes are neighbors). The cost value "9999" indicates an infinity cost (meaning two nodes are not neighbors).

   Your program must take the name of the topology file as a command line argument.
   E.g. `python routing.py topology.csv`

2. Next, your program must take a source node as an input from a command line and print the **shortest path tree** and the **cost of the least-cost paths** for this node. The computation of the shortest path tree and the costs of the least-cost paths must be implemented using **Dijkstra's algorithm**.

3. Next, your program must calculate **distance vector** for each node in the network using distance-vector algorithm. Final converged distance vector for each node must be provided as an output. Distance vector estimates must be calculated using **Bellman-Ford equation.**

4. Comment your code thoroughly, explaining the steps of both: **link-state** and **distance-vector** algorithms and overall flow of your program (e.g. parsing input file, generating shortest path tree output, generating distance vectors outputs, etc.) In addition, make sure to properly specify any sources you have used.

5. Create a *readme.pdf* with the description of your group members responsibilities (if working in group), instructions of how to run your program, and screenshots of sample program runs.

6. Submit zipped folder with Python source code and the readme.pdf file on Moodle. Only one submission per group is required.

Sample run:

```
at@at:~$ python routing.py topology.csv
Please, provide the source node: u
Shortest path tree for node u:
uw, ux, uwv, uwvy, uwvyz
Costs of the least-cost paths for node u:
u:0, v:6, w:3, x:5, y:10, z:12

Distance vector for node u: 0 6 3 5 10 12
Distance vector for node v: 6 0 3 7 4 6
Distance vector for node w: 3 3 0 4 7 9
Distance vector for node x: 5 7 4 0 7 9
Distance vector for node y: 10 4 7 7 0 2
Distance vector for node z: 12 6 9 9 2 0
```