CSE205: Introduction to Networking

Project 2: Link state and distance vector routing algorithms

Name: Kai-Yu Lu Student ID: 1614649
Major: Computer Science and Technology (CST) Due date: Dec 18, 11:59 p.m.

1. Running instruction

Firstly, due to the requirement that the project should be tested programmatically, therefore, the file should be opened in cmd. Open cmd → input ‘cd’ to access to the aimed .py file called Project2 → input ‘C:\Python32\python.exe Project2.py nodeList.txt’ (here python.exe should depend on the root where the user install). In this case, the default starting node is 1, while the user could choose the desired starting node by inputting ‘C:\Python32\python.exe Project2.py nodeList.txt 3’ (this will start from node 3). Therefore, nodeList.txt and 3 are the afferent parameters.

2. Implementation

a. The program can read the graph in the txt file.

b. The second part is to implement the Dijkstra’s algorithm. The default node is the node in the first row in the txt file. The format of the file is the same as the example in the assignment script. In the file, it says “Node 1\n<nodename>\t<cost>”. The format and name of the input file used is called “nodelist.txt”. Finally, the result will be saved in “output.txt”.

c. The third is to implement the Bell-man Ford distance vector algorithm. The first step is to The client (current node) can obtain the table of server (neighbornode) by TCP connection. For every loop, the current node could call the function of Bell-man Ford and the it will update the current table. Due the property of Bell-man Ford algorithm, it can only obtain the information of neighbor nodes. It is noted that if the table of the node is updated (changed), it will call Bell-man Ford function again until all. In the end, the whole results would be saved in Bellman_node_<nodename>.txt.

3. Functions

1. server (node, port, nodelist, oldlist): the server is for the next node.
2. client (node, port, nodelist): the server is for the current node.
3. TCP (source, neigh, nodelist, oldlist): Multiprocessing for the client and the server.
4. bellman_ford (graph, source, nodelist): Bellman_Frod algorithm for obtaining the fewest cost.
5. update (nodelist, distancelist): If the oldlist is different from the newlist, the old one would be changed into new one.
6. ReadGraph (filename): It could read the graph from the file.
7. Dij (start, distancelist, nodelist): Implement Dijkatra’s algorithm to obtain the shortest path.

4. Limitations (bugs)

In summary, the requirements of the project have been all implemented. However, this program might run slowly and the paths are unidirectional.