Project Description

The project implements all of the parts from the project requirements, which are as follows:

1) Read in the graph configuration file.
2) Implement the Dijkstra's (link state) algorithm and output the solution to a text file.
3) Implement the Bellman-Ford (distance vector) algorithm, which consists of the following sections:

   3.1) spawn the required subprocesses for each node respectively.

   3.2) establish the required connections for nodes to update their dv tables. The information sent to each node is local and was issued based on the program's protocol.

   3.3) update the nodes' dv tables until convergence of the algorithm.

   3.4) output the dv table for each node to their corresponding text files.
4) Recognize input arguments from the command line, where the file name is the compulsory argument and the starting node for the Dijkstra's algorithm is the optional argument.

Guide for program use

Firstly, place the three provided .py files inside a folder (project2.py, client.py and server.py). Accordingly, direct to the created directory using command line prompt and enter the following command: "python project2.py <filename.txt><startingNode>", where filename.txt and startingNode are the file name and the starting node for Dijkstra's algorithm chosen by the user respectively. It should be mentioned that the user can choose not to enter the <startingNode>. Consequently, the first node read from the file would be arbitrarily chosen.
Note that the mentioned command should be changed when using the computer systems available in the XJTLU computer lab. The acceptable command by the mentioned systems is as follows: C:\Python32\python.exe project2.py <filename.txt> <startingNode>"
Also note that the project must be run using python version 3.2 or higher.

Known bugs

The program spawns client subprocesses for all the nodes except for the last node and spawns server subprocesses for all the nodes except for the first node. Therefore, for n nodes, $2n - 2$ subprocesses would be spawned. If the number of nodes becomes rather large, considerable pressure will be put on the system's CPU, which is dissatisfactory. Moreover, with large number of nodes, the length of the transferred messages between servers and clients would increase, which would cause problems as the receiver only receives 1024 bytes.

Resources

As less than ten lines of code were implemented from each source, there are no resources.