Design:

Topology chart creator designed in python 2.7.

Data structure:

I have a node class that has attributes that hold either single integer values, lists of integers, or lists of other node objects. Each node has an int ID, a list of only it’s immediate neighbors with a corresponding list of int weights, a different set of corresponding lists (3 total) that will contain the final chart data for each node (all nodes, the cost to get to them, and the next hop from the initial node), a package list to put the chart data (package will contain 3 chart lists), and a buffer list to accept neighbor packages.

Overall approach/procedure:

Split each line of the incoming file into a list of 3 numbers and create a node for each node that appears in column 1 or 2, saving the neighbor and the weight into the node object’s neighbor and weigh lists (not chart lists). Save each separate node object into a node list in main. Update topology chart data (3 lists per node) with initial neighbor and weight data for each node. Next hop is the initial neighbor itself. Each node then packages their current chart data (neighbors, weights, next hop) into a package (list), and the package is sent (appended) to their immediate neighbor’s buffer. Total packages are tracked in sending (appending) to buffer phase. Then, each node looks through their buffer and fills out their own chart by that new data, and searches for a lower cost to a destination if a path already exists. Once every node has been through it’s buffer data, buffers are cleared. This is repeated until the amount of changes made for all nodes in that updating round are 0. Last node changed round number are tracked in update phase. All charts are complete by this point and are printed to the outfile in order of node number. The rounds, last node changed, and total DV packets are displayed at the end of the output file.

Commandline Notes:

Can be used with a text file in the format of the given topology files as input. Command line argument 3 can be any integer. The program will simulate rounds of DV packet sending up to and including the round chosen.

Command line: program to execute, desired input topology file with extension, space, desired text output file with extension, space, desired rounds to simulate (int). Please input a number greater than 0. Any round below 1 will just display the initial charts with info about only a node’s immediate neighbor.

**Format: program.pyc input\_file.txt output\_file.txt 7**

**Ex:** **cs4310-project1.pyc topology1.txt results1.txt 4**

Results:

The charts for chosen round, last node to update for chosen round and after convergence, the totak number of packages sent up to the round chosen and until convergence, and the number of rounds before convergence will be printed in the output file after the program is run.

Attached result files

The included result text files are base files that display charts and convergence data. The packet traversal path for each topology file is in those results.