**Running the Distance Vector Routing Algorithm with PyQt5**

This guide outlines how to run the distance vector routing algorithm using a Python program with PyQt5 and PyQt5.QtGui libraries. Follow these steps to execute the program and explore various functionalities of the routing algorithm.

**Setup and Execution**

1. Compiling the Program:

* Open your terminal or command prompt.
* Navigate to the directory containing display.py.
* Run the program by typing python display.py.

1. Loading the Input File:

* Upon running, you will be prompted to enter the name of your input file (e.g., inputfile.txt).

1. Starting Up Servers:

* The code initializes servers for each node listed in the input file, each running in a separate thread.
* Once you see the message "All servers are ready," the main menu will appear.

**Using the Main Menu**

The main menu offers several options for interacting with the distance vector algorithm:

* Option 1: Display Initial Distance Vector Table for Each Node
* Selecting this will open a GUI window showing the initial distance vector for each node.
* It's recommended to start here to familiarize yourself with the network's initial state.
* You can resize the GUI window for better visibility.
* Option 2: Run Algorithm in Single Step Mode
* This mode allows you to proceed through the algorithm step-by-step, manually advancing through each iteration.
* Please Note that if the Option 1 is chosen, then another GUI will appear, and you can have both GUI’s side by side.
* Option 3: Run Algorithm Without Intervention
* This mode runs the algorithm to completion without pausing, allowing you to observe the uninterrupted execution of the routing process.
* Option 4: Adjust Link Cost
* Here, you can modify the cost of any link between nodes.
* Enter the source ID, destination ID, and new cost in the specified format (e.g., 1 2 3 to set the cost from node 1 to node 2 as 3).
* After adjusting, a confirmation will display in the terminal detailing the updated link.
* A submenu will then ask how you'd like to run the algorithm: in single step mode or without intervention.

**Simulation of Line Repair**

To simulate a line repair:

* Restart the Program:
* Recompile and rerun the program without adjusting the previously modified link.
* This allows you to compare the network behavior with and without the link cost modification.

**Observations From Algorithm**

When simulating a line failure, the algorithm was able to update and reach stability in the same number of cycles as it did without the failure. This demonstrates the efficiency of the algorithm in adapting to changes in the network topology.

**Performance Metrics:**

Without any manual intervention, the algorithm completed its process with the following performance metrics:

***Total Number of Cycles: 4***

***Total Execution Time: 0.321 seconds***

**Impact of Changing Link Costs**

An adjustment was made to the link cost between nodes 2 and 5, reducing it from 8 to 5. This modification led to the algorithm stabilizing more quickly:

Reduced Number of Cycles: From 4 to 3

**Important Notes**

* After selecting an option from the main menu, the algorithm will run and then the servers will shut down. To test different functionalities or scenarios, you will need to restart the program each time.
* This manual restarting is necessary to reinitialize the environment and ensure accurate simulation results.