Diego Vester

1001329342

04/23/2023

CSE 4344-001

Lab 2

Write up

**Command used to run the program:**

python3 main.py

The simulation will run without intervention and display a total time until the algorithm reaches a stable state.

**Notes:**

* Time is measured in steps, categorized as a “Hop”
* Each Node # has two columns that represent the DV table
  + The first column represents the destination node linked
  + The second column represents the cost of the link
  + Destination nodes not listed are not linked, same as the ones listed in the Lab 2 example with a “-“
* The state of stability is listed after the last node in each run

**Adjusting a link cost:**

When the cost of a link is set to infinity (16), the cost of the DV from 1 to 3 increased drastically. When the cost of a link from node 1 to node 2 is set to 16, as well as the cost of the link from node 2 to node 3 is set to 16, the cost of node 1 to node 3 becomes 32. Node 1 has no alternative means of travelling to node 3 before the programs finds stability because the nodes are generally not very interconnected.

When the cost is set back to what is was originally, the cost of node 1 to node 3 by the end of the program becomes 8.

**DVs for each step:**

\_\_\_ Hop: #0 \_\_\_

Node 1

1 0

2 7

5 1

Node 2

2 0

3 1

5 8

Node 3

3 0

Node 4

4 0

5 2

3 2

Node 5

5 0

Node 6

6 0

\_\_\_ Hop: #1 \_\_\_

Node 1

1 0

2 7

3 8

5 1

Node 2

2 0

3 1

5 8

Node 3

3 0

Node 4

4 0

5 2

3 2

Node 5

5 0

Node 6

6 0

### not stable ###

\_\_\_ Hop: #2 \_\_\_

Node 1

1 0

2 7

3 8

5 1

Node 2

2 0

3 1

5 8

Node 3

3 0

Node 4

4 0

5 2

3 2

Node 5

5 0

Node 6

6 0

### stable ###