CSCI 39554/77300

Spring 2020

Final Project

Final Project: Implementing Distance Vector Routing on a Network

Instructor: *Prof. Saptarshi Debroy (saptarshi.debroy@hunter.cuny.edu)*

Lab Assistant: *Minh Nguyen (minh.nguyen@hunter.cuny.edu)*

Deliverables Due: May 24th via Blackboard

1. Goal & Example

In this project, we will execute Bellman–Ford algorithm find the shortest path using the same topology as Lab #3. Bellman–Ford algorithm is the core of any Distant Vector (DV) protocol. You will have to write a code using any programming language in your comfort zone.

Note: *First thing first, please read and understand how Bellman–Ford algorithm works. There are many examples on the Internet and YouTube that can help you.*

Your code must find the shortest path from source node to other nodes also has to keep track of the intermediate nodes.

E.g., run Bellman–Ford algorithm on this graph (source **S**):

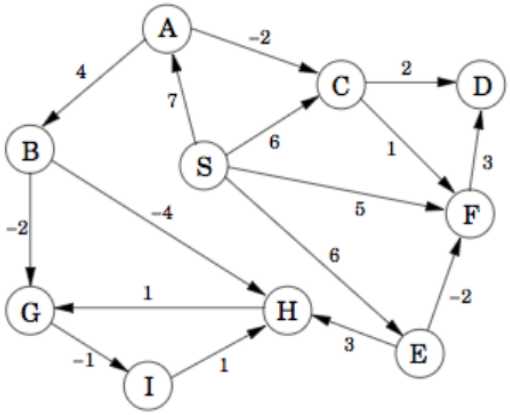

Figure 1.

The output should be a matrix like below:

	Iteration						
Node	0	1	2	3	4	5	6
S	0	0	0	0	0	0	0
A	∞	7	7	7	7	7	7
B	∞	∞	11	11	11	11	11
C	∞	6	5	5	5	5	5
D	∞	∞	8	7	7	7	7
E	∞	6	6	6	6	6	6
F	∞	5	4	4	4	4	4
G	∞	∞	∞	9	8	8	8
H	∞	∞	9	7	7	7	7
I	∞	∞	∞	∞	8	7	7

Table 1.

Note: *Theoretically, the algorithm produces 9 iterations for 10 vertices (nodes). However, from iteration 5 onward the outcome will remain the same.*

As I said, your code also has to keep track of the intermediate nodes. E.g., the shortest path from **S** to **I** is thru **A – B – H – G** and the length of that path is 7 (as in the table above).

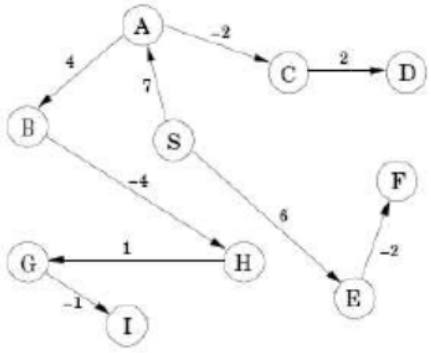

Figure 2.

2. Experiment Setup

Reserve the experiment using provided RSpec file on Blackboard. We now have a topology with six nodes with source node is ‘bellman-ford’.

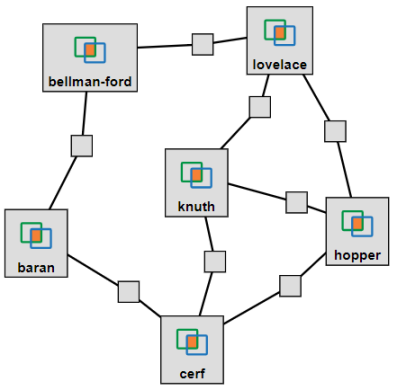

Figure 3.

Run your Bellman–Ford algorithm on this topology. With the result of your code and based on Lab#3, route the traffic from 'bellman-ford' to all other nodes.

Now, write a simple interface (shell/bash or Python script) on 'bellman-ford' node that prompt user to enter the name of destination node. The output is the 'mtr' from 'bellman-ford' to that destination.

Note: *Your algorithms should work with any graph, not just hard-coded for the graph in this project.*

What to turn in?

A report, your Bellman–Ford algorithm code, and your interface script. In the report:

- Include two pictures of IP addresses defined on each node and cost on each links.
- After running Bellman–Ford algorithm, include the screenshot of your output.
- Also include the picture of marking each network interface as being part of the shortest path tree if they are on a "pink" link, and not part of the shortest path tree if they are on a "grey" link. Please refer to Lab#3.
- Screenshot of interface script output.
- Some explanations for each screenshot or picture.
- Instruction of how to run your Bellman–Ford algorithm code and script. Please be specific, e.g., how to get the input or the command to compile/run the code, etc. If I can't run the codes, your grade might be affected.