Munem Rastgir, Glyne Gittens, Jakub Taraska

5/26/18

CSCI 395 Final Project

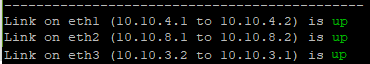
Bellman-ford IPs and link costs

Baran IPs and link costs







Lovelace IPs and link costs





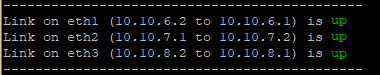


Cerf IPs and link costs







Knuth IPs and link costs







Hopper IPs and link costs







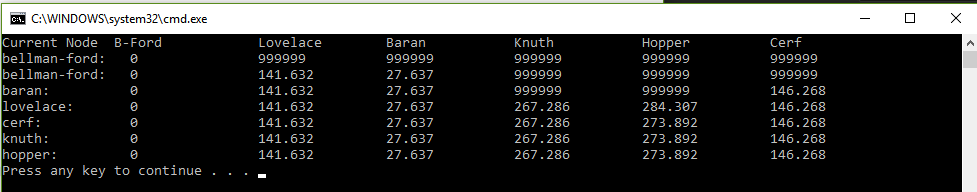
The final distances that we used are:

* Bellman-ford to lovelace: 141.632
* Bellman-ford to baran- 27.637
* Lovelace to knuth- 125.654
* Lovelace to hopper- 142.675
* Baran to cerf- 118.631
* Cerf to knuth- 196.626
* Cerf to hopper- 127.624
* Hopper to knuth- 105.593

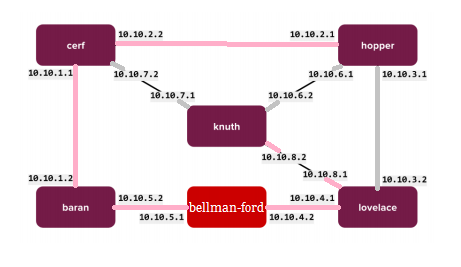
We also used these link costs in both directions; For example, hopper to knuth and knuth to hopper have the same link cost.

Table of shortest path:



Output of C++ program with the above values:

The left column represents the current node we are on and can be thought of as the iteration, we start at bellman-ford and continue to baran, lovelace, cerf, knuth, and hopper with our final iteration. We then update the current distances in the 6 right side columns. 999999 can be thought of as infinity. Also, the final shortest path distances are found in the last row. One note is that the table pictured here is slightly different from the matrix example in the lab instructions because the rows and columns are swapped.

Picture of the shortest path tree on the network topology:

We marked the shortest path in pink and then we used these commands to route the topology

lovelace:

sudo ifconfig eth3 down

hopper:

sudo ifconfig eth2 down

sudo ifconfig eth1 down

sudo route add -net 10.10.5.0/24 gw 10.10.2.2

knuth:

sudo ifconfig eth1 down

sudo ifconfig eth2 down

sudo route add -net 10.10.4.0/24 gw 10.10.8.1

cerf:

sudo ifconfig eth1 down

sudo route add -net 10.10.5.0/24 gw 10.10.1.2

bford:

sudo route add -net 10.10.8.0/24 gw 10.10.4.1

sudo route add -net 10.10.1.0/24 gw 10.10.5.2

sudo route add -net 10.10.2.0/24 gw 10.10.5.2

baran:

sudo route add -net 10.10.2.0/24 gw 10.10.1.1

Bellman-ford c++:

Compile command: g++ -std=c++11 Source.cpp -o Source

Running command: ./Source

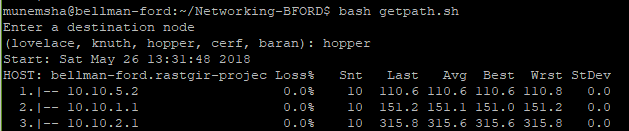
Input 1 to run the code using our values OR press 2 to input 8 numbers for the 8 link costs if you want to adjust the values of the nodes in this topology. Also, after the shortest path matrix is printed, you can enter the name of a node to see its intermediate nodes get displayed.

Bash Script:

Upload the bash script getpath.sh onto the bellman-ford shell after the routing is finished.

Running command: bash getpath.sh

After we enter the command to run the script, we will get prompted to enter the destination node name that we want to reach, and the output will be the mtr of that destination.

To test it out, we cloned our github repository onto the node using the command “git clone https://github.com/munemrastgir/Networking-BFORD.git” and then we cd into Networking-BFORD and then we ran the shell script from that directory. But if you have the shell script file, then you can simply upload it to the node and run it from there.