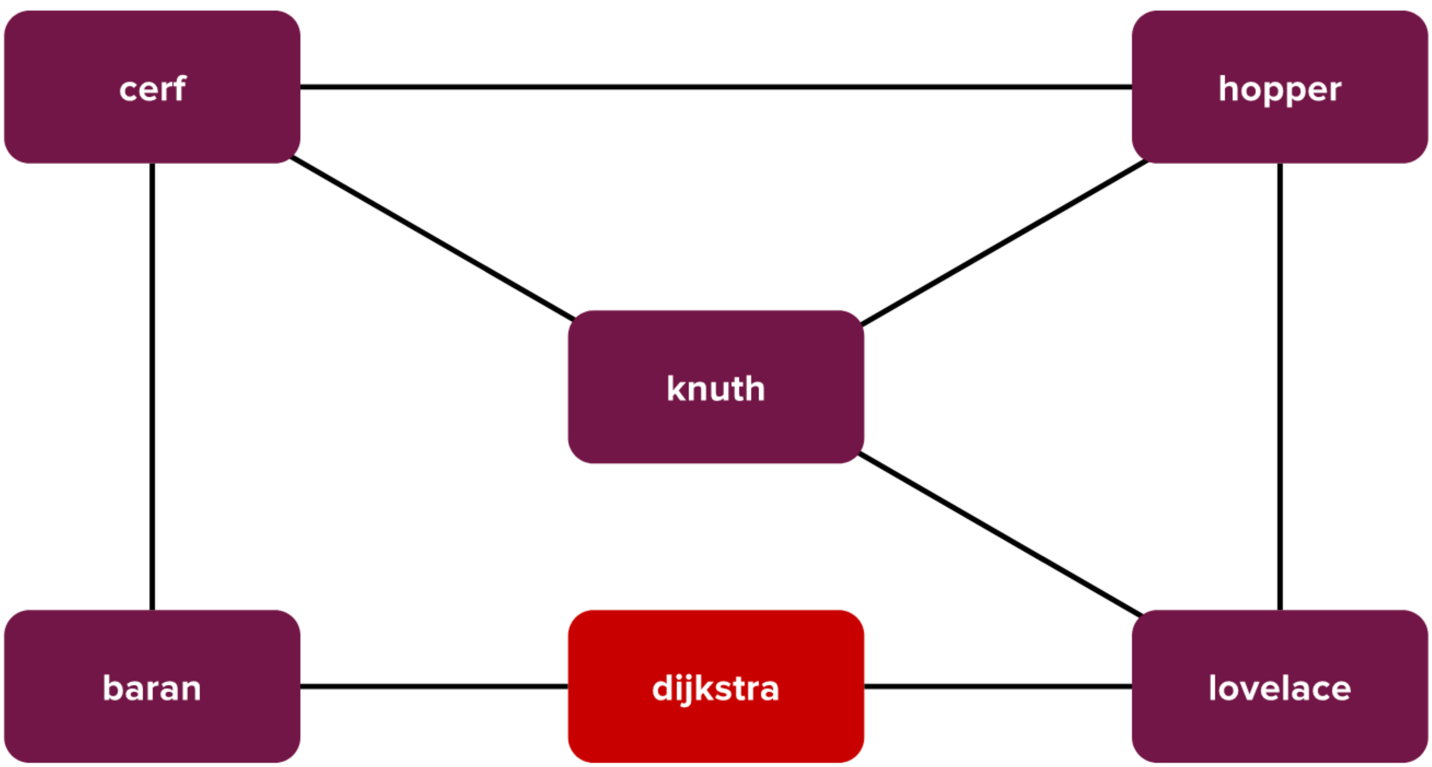
GENI Lab 7

Due: 3:30pm Nov 4, 2019

0. Introduction

In this lab, we are going to move down the network protocol hierarchy by one layer and arrive at the **Network Layer**.

In particular we will use **Dijkstra's algorithm** to find the minimum cost path (in terms of latency) from the node named "*dijkstra*" to each of the other nodes (all named after famous engineers or computer scientists) in this topology:



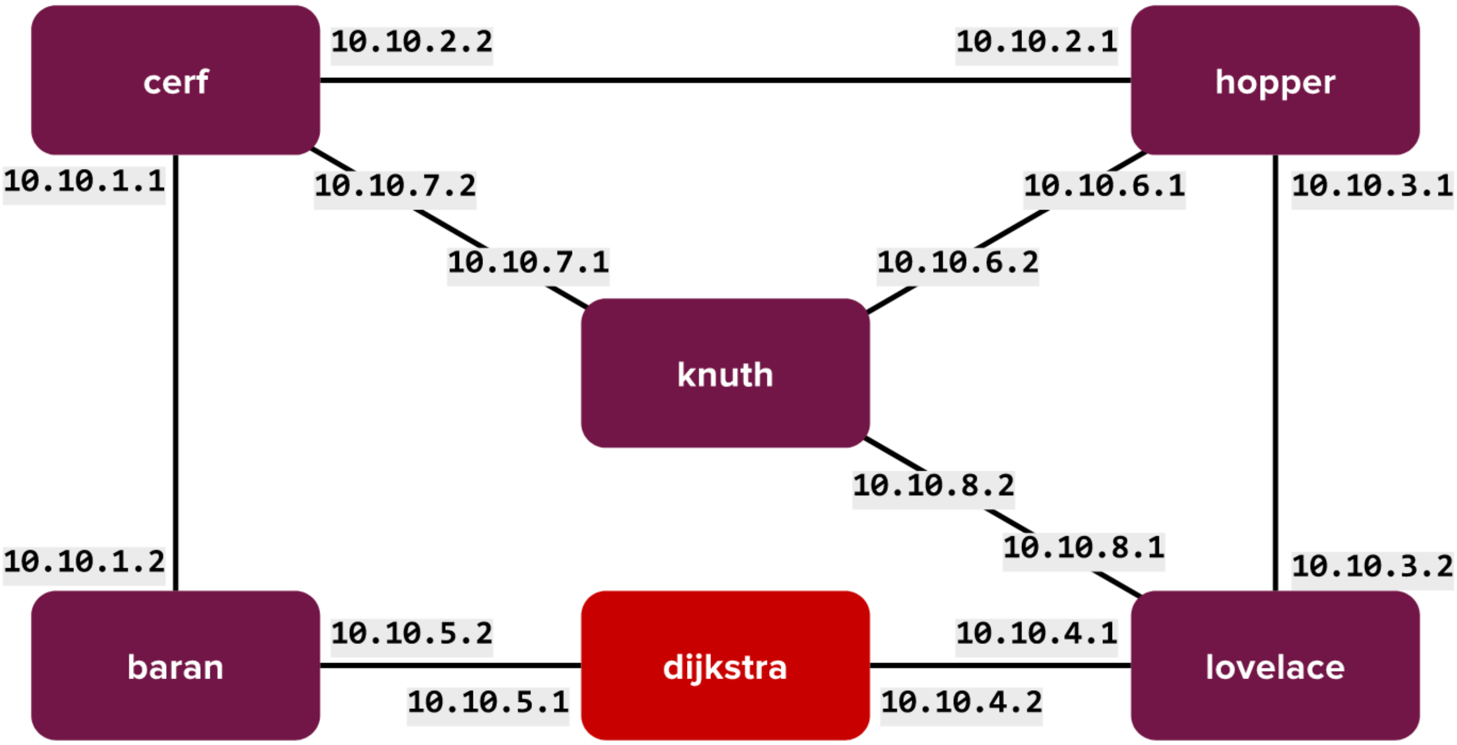
1. Lab Configurations

In the GENI Portal, create a new slice **"lab7-your-initial"**, and then click "Add Resources". Scroll down to the

"Choose RSpec" section and select "URL"; enter the URL https://git.io/vSCdF and choose "Select". This will load a six-node topology in your canvas.

Click on "Site 1" and select an InstaGENI site from the drop-down list on the left, then click "Reserve Resources". Wait until all of your nodes are up and ready to log in, then open six terminals and SSH into each node.

We now have a topology with six nodes, and IP addresses defined on each as shown in the following image:



2. Run Dijkstra's algorithm

EXTREMELY IMPORTANT NOTE:

In this section, I'll show how I would execute the Dijkstra's algorithm on my network with my link costs - **when you run this lab**, although the network topology, node names, and IP addresses stay the same, **you will have different link costs, and will have to modify the procedure accordingly (i.e. visiting nodes in a different order than I did)**.

To help you all, here is a Youtube video of how I executed Dijkstra's algorithm on this six-node topology:

Youtube

2.1. First iteration

In the first iteration of Dijkstra's algorithm, we will visit the "dijkstra" node. From the figure above, we can identify each of the direct neighbors of the "dijkstra" node and their IP address on the link that they share with "dijkstra":

- "lovelace" on 10.10.4.1
- "baran" on 10.10.5.2

To get the cost of the link from "dijkstra" to "lovelace", we will ping "lovelace" on its 10.10.4.1 IP address 25 times, and take the average latency as the cost. **From "dijkstra", run**:

```
ping -c 25 10.10.4.1
```

Note the average round trip time reported by the ping command, e.g. 32.742 ms in the example below:

```
--- 10.10.4.1 ping statistics ---
25 packets transmitted, 25 received, 0% packet loss, time 24045ms
rtt min/avg/max/mdev = 32.272/32.742/33.986/0.448 ms
```

Open the "lab7-worksheet" document, find "Step 2.1".

In the working table, fill in this value as the cost of the path from "dijkstra" to "lovelace", and note that the previous hop in this path is "dijkstra".

Then, repeat this step for "lovelace". **From the "dijkstra" node, run**:

```
ping -c 25 10.10.5.2
```

and record the result in the table.

The other entries in the table remain unchanged, so we can copy them from the previous row. Also, since we have already visited "dijkstra", we know that it will not change in future iterations, so we can copy its value all the way down the table.

For your reference, at the end of iteration 1, my table looks like this (your numbers should be different):

Iteration	Unvisited	Visited	Current	dijkstra	cerf	lovelace	hopper	baran	knuth
0	dijkstra, cerf, lovelace, hopper, baran knuth	-	-	0 -	∞ -	∞ -	∞ -	∞ -	∞ -
1	cerf, lovelace, hopper, baran, knuth	dijkstra	dijkstra	0 -	∞ -	32.742 dijkstra	∞ -	71.918 dijkstra	∞ -
2				0 -					
3				0 -					
4				0 -					
5				0 -					
6				0 -					

In the "lab7-worksheet" document, still under "Step 2.1",

- Insert a screenshot of the terminal for the “dijkstra” node. Make sure it includes both “ping” commands above and their outputs.
- Update the "Link costs figure".

2.2. Second iteration

AGAIN, THE NODES YOU CHOOSE MIGHT BE DIFFERENT FROM MINE DUE TO DIFFERENT LINK COSTS. DON'T BLINDLY COPY MINE.

To choose the next node to visit, we compare the costs to each of the remaining unvisited nodes. Here,

$$32.742 < 71.918 < \infty$$

so on the next iteration we visit "lovelace". Its direct unvisited neighbors (that we have not visited) are:

- "knuth" at 10.10.8.2
- "hopper" at 10.10.3.1

So from "lovelace", we will run `ping` commands to each of those nodes. In the table, we will record **the cost of the total path** from "dijkstra" to each of "knuth" and "hopper":

- I measure 39.809 from "lovelace" to "knuth", for a total cost of $32.742 + 39.809 = 72.551$. This is less than the previous (infinite) cost of the path to "knuth", so I update the table to note 72.551 as the cost to "knuth" and "lovelace" as the previous hop.
- I measure 24.254 ms from "lovelace" to "hopper", for a total cost of $32.742 + 24.254 = 56.996$. This is

less than the previous (infinite) cost of the path to "hopper", so I update the table to note 56.996 as the cost to "hopper" and "lovelace" as the previous hop.

The other entries in the table remain unchanged, so I copy them from the previous row. Also, now that I have visited "lovelace", its value will not change again and I can copy it all the way down the table. After iteration 2, my table looks like this:

Iteration	Unvisited	Visited	Current	dijkstra	cerf	lovelace	hopper	baran	knuth
0	dijkstra, cerf, lovelace, hopper, baran knuth	-	-	0	∞	∞	∞	∞	∞
				-	-	-	-	-	-
1	cerf, lovelace, hopper, baran, knuth	dijkstra	dijkstra	0	∞	32.742	∞	71.918	∞
				-	-	dijkstra	-	dijkstra	-
2	cerf, hopper, baran, knuth	dijkstra, lovelace	lovelace	0	∞	32.742	56.996	71.918	72.551
				-	-	dijkstra	lovelace	dijkstra	lovelace
3				0		32.742			
				-		dijkstra			
4				0		32.742			
				-		dijkstra			
5				0		32.742			
				-		dijkstra			
6				0		32.742			
				-		dijkstra			

In the "lab7-worksheet" document, find "Step 2.2",

- Update the working table
- Insert a screenshot of the terminal for your chosen node.
- Update the "Link costs figure".

2.3. Third iteration

Continue on to the next iteration.

In the "lab7-worksheet" document, find "Step 2.3",

- Update the working table
- Insert a screenshot of the terminal for your chosen node.
- Update the "Link costs figure".

2.4. Fourth iteration

Continue on to the next iteration.

In the "lab7-worksheet" document, find "Step 2.4",

- Update the working table
- Insert a screenshot of the terminal for your chosen node.
- Update the "Link costs figure".

2.5. Fifth iteration

Continue on to the next iteration.

In the "lab7-worksheet" document, find "Step 2.5",

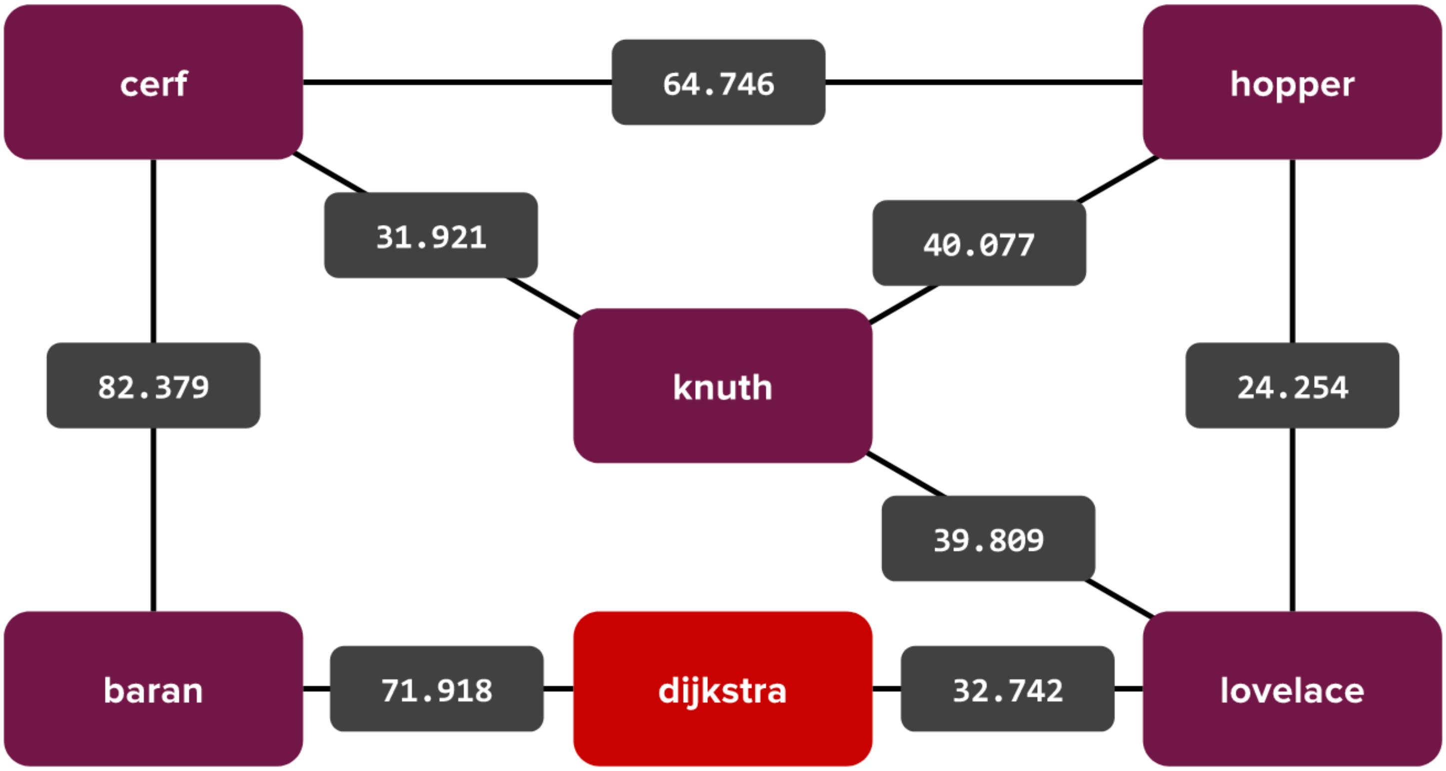
- Update the working table
- Insert a screenshot of the terminal for your chosen node.
- Update the "Link costs figure".

2.6. Sixth iteration

Continue on to the next iteration, which should be the last iteration because you will have visited all nodes. For your references, my final table looks like this:

Iteration	Unvisited	Visited	Current	dijkstra	cerf	lovelace	hopper	baran	knuth
				0	∞	∞	∞	∞	∞
0	dijkstra, cerf, lovelace, hopper, baran knuth	-	-	-	-	-	-	-	-
				0	∞	32.742	∞	71.918	∞
1	cerf, lovelace, hopper, baran, knuth	dijkstra	dijkstra	-	-	dijkstra	-	dijkstra	-
				0	∞	32.742	56.996	71.918	72.551
2	cerf, hopper, baran, knuth	dijkstra, lovelace	lovelace	-	-	dijkstra	lovelace	dijkstra	lovelace
				0	121.742	32.742	56.996	71.918	72.551
3	cerf, baran, knuth	dijkstra, lovelace, hopper	hopper	-	hopper	dijkstra	lovelace	dijkstra	lovelace
				0	121.742	32.742	56.996	71.918	72.551
4	cerf, knuth	dijkstra, lovelace, hopper, baran	baran	-	hopper	dijkstra	lovelace	dijkstra	lovelace
				0	103.842	32.742	56.996	71.918	72.551
5	cerf	dijkstra, lovelace, hopper, baran, knuth	knuth	-	knuth	dijkstra	lovelace	dijkstra	lovelace
				0	**103.842**	**32.742**	**56.996**	**71.918**	**72.551**
6	-	dijkstra, lovelace, hopper, baran, knuth, cerf	cerf	**-**	**knuth**	**dijkstra**	**lovelace**	**dijkstra**	**lovelace**

And my final "Link costs figure" looks like this:



In the "lab7-worksheet" document, find "Step 2.6",

- Update the working table
- Insert a screenshot of the terminal for your chosen node.
- Update the "Link costs figure".

3. The minimum-cost tree

Next, we will use the table to find the shortest-path tree rooted at the "dijkstra" node. This is a version of the topology that has no cycles (closed loops), and where the path from the "dijkstra" node to each other node is a minimum-cost path.

We can read the tree off the table as follows. For each destination node (each column), we refer to the final row to find the node that is "previous hop" to that destination node. The link between them is part of the shortest-path tree.

For example, in my topology (your experiment may come out different):

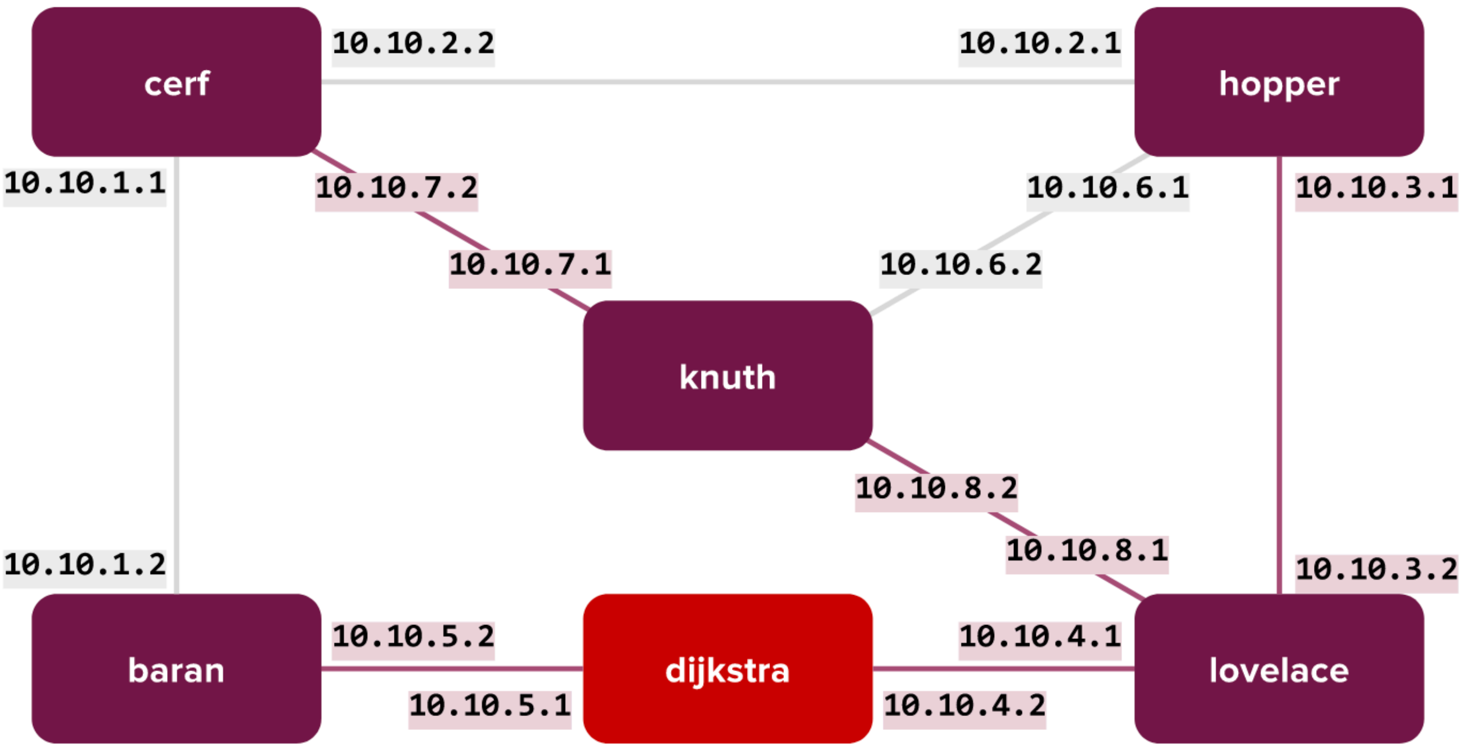
- The last hop before "cerf" is "knuth", so the link between "cerf" and "knuth" is in the shortest-path tree.
- The last hop before "lovelace" is "dijkstra", so the link between "lovelace" and "dijkstra" is in the shortest-path tree.
- The last hop before "hopper" is "lovelace", so the link between "hopper" and "lovelace" is in the shortest-

path tree.

- The last hop before "baran" is "dijkstra", so the link between "baran" and "dijkstra" is in the shortest-path tree.
- The last hop before "knuth" is "lovelace", so the link between "knuth" and "lovelace" is in the shortest-path tree.

We can now mark each link as either being part of the shortest path tree (red) or not (grey). Furthermore, we can mark each network interface as being part of the shortest path tree if they are on a "red" link, and not part of the shortest path tree if they are on a "grey" link.

My minimum-cost tree looks like this (yours could be different):



In the "lab7-worksheet" document, find "Step 3",

- Mark links and network interfaces on your minimum-cost tree with red and gray.

4. What to Turn in?

Submit the following file:

- lab7-worksheet.docx