1. Should we only use Bellman-Ford to compute shortest paths?

- Though we have suggested (for generality) Bellman-Ford to compute shortest paths, you are free to choose whatever shortest path algorithms.
- Other possible algorithms: Djikstra, Floyd-Warshall

2. What is the metric for shortest path computation?

- Number of hops between source and destination.

3. What is the byte table data structure?

- The byte table contains switch rules that you supply to the controller. For this assignment, you will compute shortest paths between every pair of switches and you would then be invoking APIs in the SwitchCommands class to add/remove/modify rules in the table.

4. Why is my build (ant) taking so long to complete?

- The initial build is expected to take a longer time. However, subsequent builds should be faster. As we suggested in previous assignments, if your VM appears to be slow, find the unwanted zombie process and kill them. Following command will kill all instances of python.

```
sudo ps -aux | grep "/usr/bin/python" | awk '{print $2}' | xargs sudo kill -9
```

5. Should we compute the shortest path between hosts or between switches?

- For every pair of hosts (say h1 - h2), you should install rules for the shortest path at the switches in the path between h1 and h2. So, every host will have a rule for reaching itself in the shortest path from other hosts at every switch.

6. After the link between host and link is down, bringing back the link does not invoke device moved. How to handle this?

- After bringing the link back again, as mentioned in the assignment run the following command.

  `h1 arping -c 2 -A -I h1-eth0 10.0.0.1`

  Replace h1 and IP accordingly.

7. Which output port to forward the ARP packet?

- In your `receive()` callback, you have an instance of the `OFMessage` object passed to you. This contains information about the packet. You can obtain the port on which the ARP request packet was received and send your ARP reply on that same port.

8. How to test the load balancer application?

- Running the below command should download a page.

  `h1 curl http://<virtualIP>:<port>/index.html`

  `e.g: h1 curl http://10.0.100.1:80/index.html`

9. Can we add the same rules back again?

- We suggest a cleaner way of updating rules in the switch. Remove the older rule and install the new rule.

10. For connection specific rules, how to get the MACAdddress corresponding to the host IP?

- There is a function named `getHostMACAddress()`.

11. Do we need to send an ACK reply after the connection has been established using Load Balancer?

- You need not. But, please ensure that you are returning the right return value indicating that the SYN packet is processed.

12. Pingall works successfully sometimes, sometimes not. How to solve this?

- The issue could be because your shortest path algorithm takes too long to converge. Also, you need to consider how different event handlers are getting called and your code handles them correctly.

13. Should we forward all ARP requests to the controller or only ARP requests to Virtual IP?

- Match all ARP packets and send it to the controller. ARP replies to the ARP of Virtual IP will be handled by the code you have implemented. ARP replies to the other IPs will be handled by another ArpServer application running in the controller.

==========

Below are some useful tips:

1. Bellman-Ford Algorithm Sample implementation -- https://www.programiz.com/dsa/bellman-ford-algorithm

2. Quick tips on choosing shortest path algorithm

- For the lab, Djikstra should work as there are no negative weight edges / negative cycles.
- In this lab, the metric for shortest distance is the number of hops between hosts. Hence algorithms like BFS should also work. But, if the metric is a function of number of hops and link weight, it might not work.
- For the above reason, it is better to go with algorithms that specialize in measuring the shortest distance.
- Another factor you should consider is the time complexity for computing the shortest path as a function of the number of switches.

3. Unexpected errors while testing

For this lab, you may face some weird errors (Floodlight showing some random errors, TCP dump not showing expected packets, Build takes so long, etc). We suggest you follow the below mentioned solutions first.

- Make sure you patch the mininet and follow other prerequisites as mentioned in assignment description
- Cleanup any old running java/python/mininet instances. This may intrude in your testing and cause unexpected results.
- Clean mininet (`sudo mn -c`) before starting every testing.
- Make sure you are adding the Match rules with necessary mandatory fields set in them.