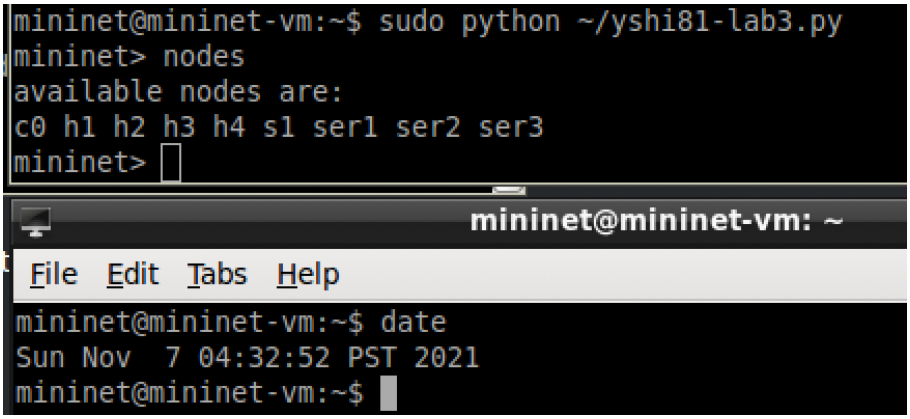
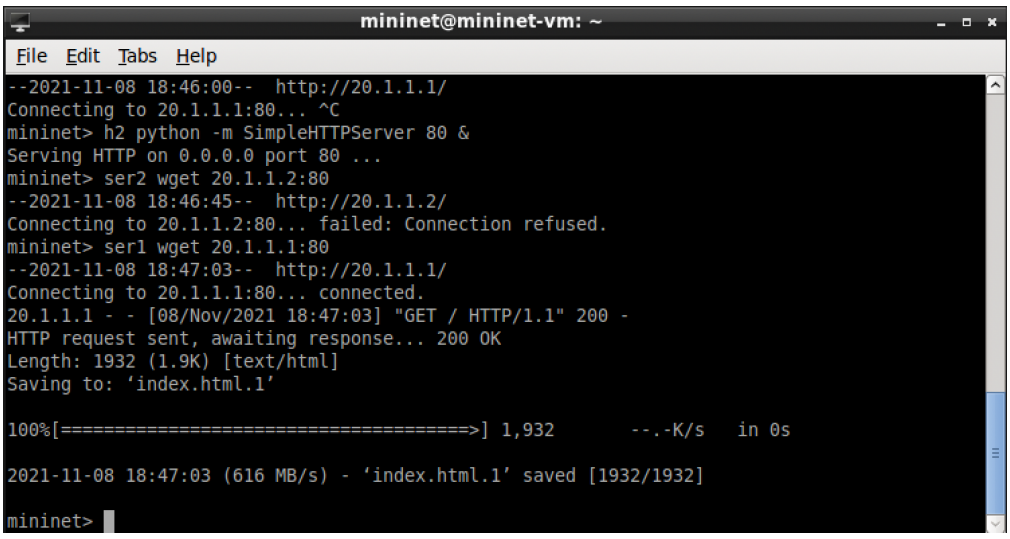
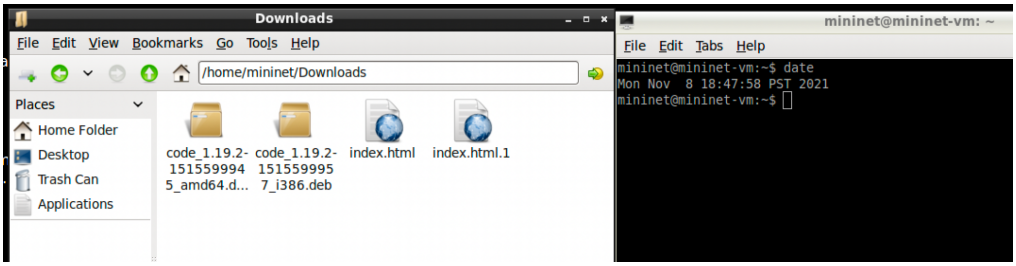
Yingkit Shi
CSE 150
Lab 3

1. See attached python file. Topology created with appropriate nodes.

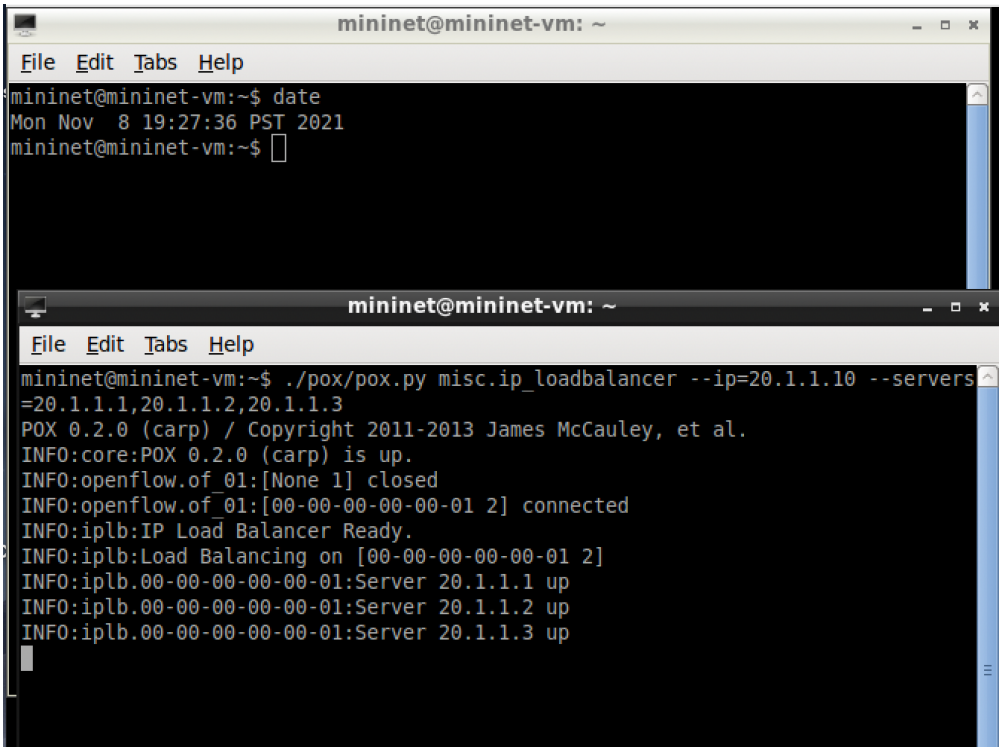


2.



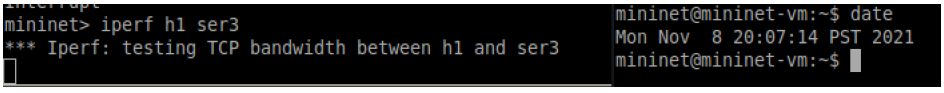


3. Remote controller terminal output.

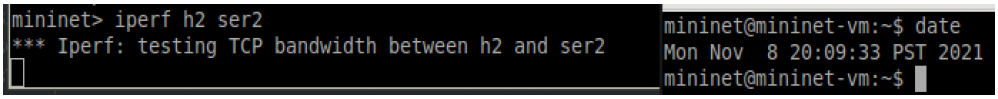


From this output after subsequently running “yshi81-lab3.py”, the load balance is up and ready. Load balancing is on the mac address “00-00-00-00-00-01 2” which distributes tasks onto the three servers. The last two lines indicate all three of the servers are up and running.

4. Iperf client 1 and server 3 command



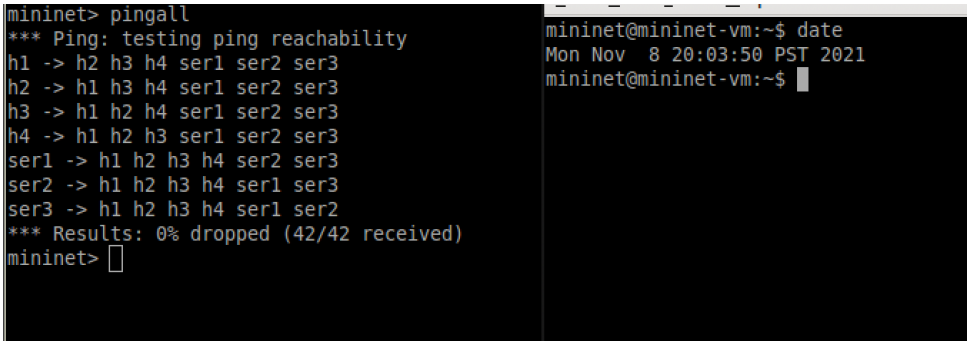
Iperf client 2 server 2 command



Both iperf commands ran indefinitely or never ended, so I manually ended the testing after 20 seconds. Iperf command measures the TCP bandwidth between hosts. In this case, the VM kept trying to test the bandwidth between client 1 and server 3 indefinitely and also between

client 2 and server 2 indefinitely. The first command should be running indefinitely because the table does not specify source with client 1 and destination with server 3. The second command should be also running indefinitely because the table does not specify source with client 2 and destination with server 2 (only source with server 2 and destination with client 2). Therefore, there are no TCP packets to be measured for the bandwidth for these two commands.

5. Pingall command



Pingall shows the connectivity between the clients and servers. They are all able to connect and reach each rest of the network hosts. To support this, no packets are dropped. If a host to host connection would be unreachable, there would be packets dropped.

6. The commands “msg.idle_timeout” and “msg.hard_timeout” help in installing the rules in a switch (l2_learning) and making them remembered. They can be modified by setting them to milliseconds for how long an entry stays in the switch. In my code for the firewall, I set them to 50ms.