LAB 2: Testing performance in custom 4 node topology

After successful implementation and installation of the required tools for SDN lab, we move forward with the running a topology. The objective of this lab would be to run a simple topology with 4 nodes and testing the performance factors such as bandwidth. We will also make changes to the topology and vary the bandwidth of the links. I'll give you a sample topology with two nodes and you have to edit the file to the desired 4 node topology.

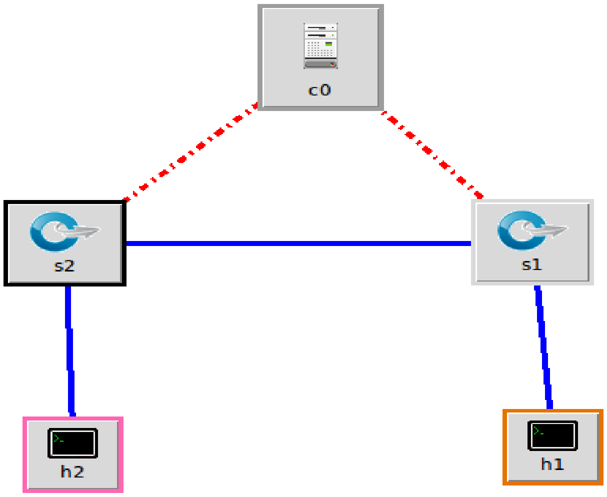
1. Start your Ubuntu virtual machine and open the text editor. Paste the following python code which is simple code for the two nodes switch. Save the file in MiniNAM folder and in '**.py'** format.

code.txt

2. To run the topology, you need to be in the MiniNAM folder in your terminal. So, open your terminal and direct to the MiniNAM directory. Then you can run the python file using following command:

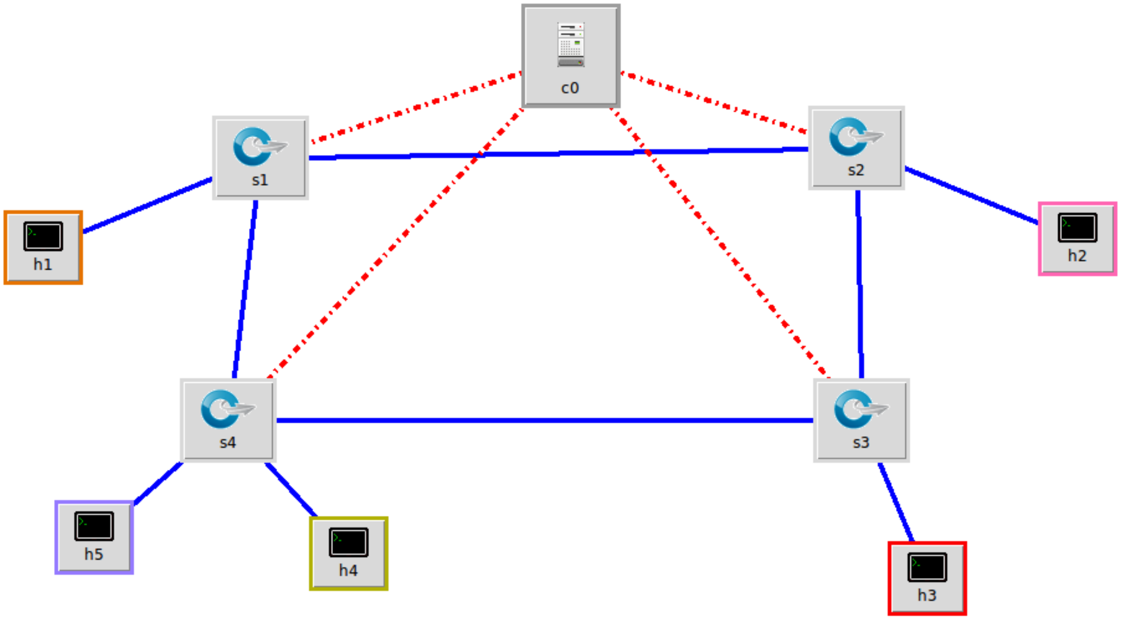
sudo python MiniNAM.py --custom "the name of the file" --topo mytopo --controller remote

Which will give you the following output on your MiniNAM window.



On the new window of the terminal, run the ryu controller (command given in the last lab). Once the ryu controller starts, go back to the Mininet terminal and issue a "**pingall"** command and check the connectivity. **(Paste the screenshot of the mininet window and the MiniNAM window)**

3. Now, we move to our custom topology i.e. ring topology with 4 switches and 5 hosts. Now, edit the same code given above so as to make the above topology as the following:



It’s the similar topology, we just have to add 2 more switches and 3 more hosts with the respective links.

4. Make the necessary changes and save the file in similar format and execute through the file with same MiniNAM command. Now, run ryu controller in new window and then try to ping. **Does the ping work? If yes, what command did you use, if no, why and how can you resolve it. (Paste the screenshot of the Mininet pingall and MiniNAM window)**

(Sometimes, when you try to run the topology after closing the previous topology, it doesn’t run and gives error. Then, make sure you clear cache using **<sudo mn -c>** command).

5. There is tool in the mininet called iperf, through which we can check the bandwidth of the link. Use **iperf** command to check the bandwidth between the host h1 and host h3. **(Paste the screenshot of it)**

Note: Iperf would only work if you get the controller running in the new window, so you must get the above answer first and then move on to the next one.

6. Now, once again edit the lab topology file and change the bandwidth of the link H1 and S1 to “3 Mb/s” and bandwidth of link H5 and S5 to “8 Mb/s”. Now run the mininet topology with same command just add **<--link tc>** parameter, otherwise **Now check bandwidth between H1 and H3 and between H5 and H2 and H4 and H3 and paste of the screenshot of it. Does the bandwidth vary according to the parameters you set?**

Thus, LAB 2 gave us idea of how we can manipulate the bandwidth and other parameter and how we can build and run the custom topology in Mininet.

Thank you!