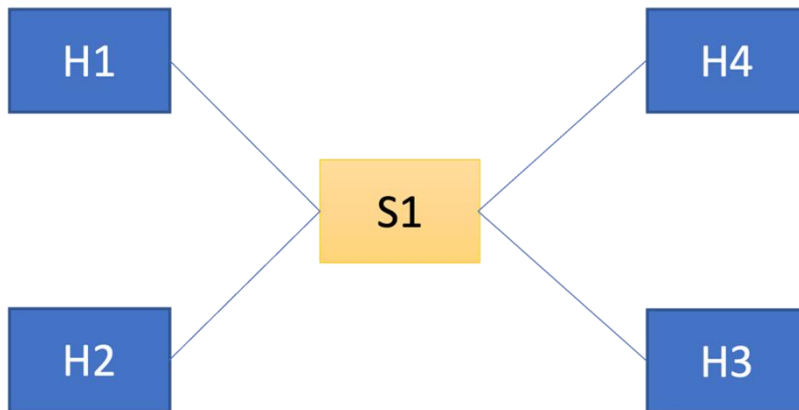


BW=250Mb/s



You need to build an OpenFlow controller that implements two types of applications: firewall and implementation of quality of service.

You need to complete the two files controller assignment1.py and topo_Assignment1.oy, as explained at the end of this text.

All rules need to be installed dynamically, i.e., when packet arrive (no rule hardcoding is allow), i.e. the controller operates in reactive mode.

The rules are exclusive, so if a packet does not match the rules below, it needs to be dropped. A rule should also be installed so that additional similar packets are dropped without going to the controller.

All installed rules should have a timeout of 40 seconds

The rules are the following:

- Data towards H3:
 - Data from H1 to H3 should be capped at 50 Mb/s
 - Data from H2 to H3 should be capped at 100 Mb/s
 - Data from H3 to H1 and H2 should not be capped
- Data towards H4:
 - Data from H1 to H4 should not be capped
 - Data from H2 to H4 should be capped at 500 Mb/s

- Data from H3 to H4 should not be capped
- Data from H3 to H1, H2 and H4 should not be capped
- Data from H1 to H2 and from H2 to H1 needs to be blocked.

To help you coding you will receive two template files, with some of the code already implemented. You only need to add your code where stated in these files (**you will see a =I sign**):

- topo_assignment1.py: this is the file where you can add code to define the topology above (notice that the link to node H1 has to be limited to 250Mb/s, all other links are unlimited), add the ovs-vsctl QoS rules. This file also runs the iperf tests, but these have already been coded for you.
- controller_assignment1.py: this is the file where you add your code to define the controller behaviour.

Additional notes:

- POX does not support meters, this is why we will use ovs-vsctl to install QoS rules (do this in the topo_assignment1.py file).

When you run the files, you now that your work is correct if the results you get match the capacity you get from iperf, as defined by the rules above. Notice that you will not get exact match, as it is an emulation.

The POX library is available at: <https://github.com/att/pox>

These classes will be particularly useful:

https://github.com/att/pox/blob/master/pox/openflow/libopenflow_01.py

<https://github.com/att/pox/tree/master/pox/lib/packet>