

# CS 7001-03: Report for GENI Lab 3 - QoS Configuration and Load Balancing using Software-Defined Networking

Chanmann Lim  
cl9p8@mail.mail.missouri.edu

April 14, 2015

## 1. Screenshots taken in Step 3.4:

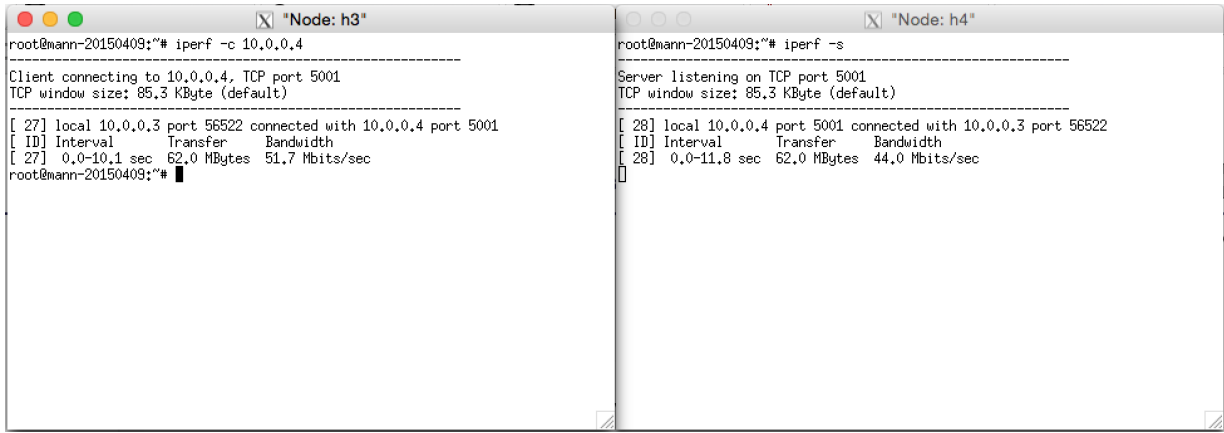


Figure 1: Iperf test from h3 to h4

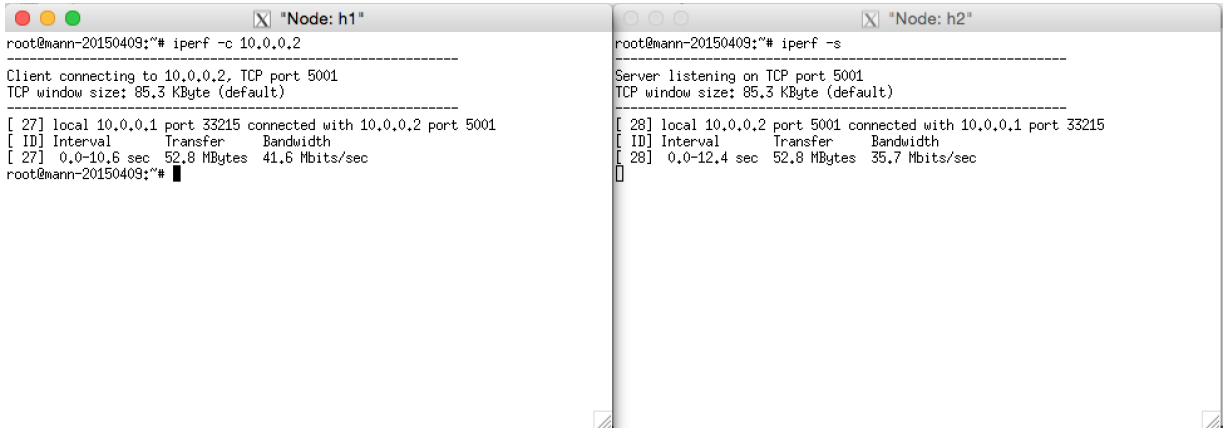


Figure 2: Iperf test from h1 to h2

The packets captured by Wireshark with "lo" interface and "of" filter in **step. 3.4** list only the OpenFlow control packet of the loopback interface, and when clicking on an "of\_aggregate\_stats\_request" packet, we obtain the detail OpenFlow headers in the packet such as *version*, *type*, *length*, *xid*, *stats\_type*, *flags*, *of\_match*, *table\_id*, and *out\_port*.

And the *of\_match* header contains *wildcards*: Wildcards fields, *in\_port*: Input switch port, *eth\_src*: Ethernet source address, *eth\_dst*: Ethernet destination address, *vlan\_vid*: Input VLAN id, *vlan\_pcp*: Input VLAN priority, *eth\_type*: Ethernet frame type, *id\_dscp*: IP ToS (actually DSCP field, 6 bits), *ip\_proto*: IP protocol or lower 8 bits of ARP opcode, *ipv4\_src*: IP source address, *ipv4\_dst*: IP destination address, *tcp\_src*: TCP source port, *tcp\_dst*: TCP destination port.

2. Screenshot of Iperf test from host4 to host1:

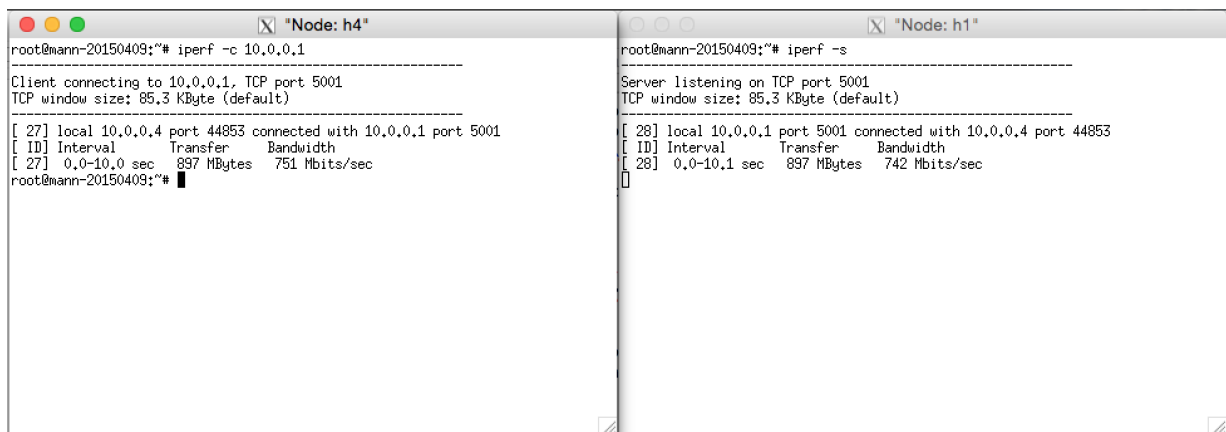


Figure 3: Iperf test from h4 to h1

By using the topology created in **step 3.3.1** and the QOS configuration in **step 3.3.2**, the traffic flow between host4 and host1 is conducted through queue "0" which had configured with the maximum bandwidth up to 1 Gbps, similarly since the traffic flow between host2 and host1 doesn't match with both queue "1" and queue "2" rules it will fallback to use queue "0" in the switches.

3. Consider the new topology shown in Figure 14 with 6 hosts, 7 switches and 1 controller. Describe the commands to establish a new queue q3 of 80 Mbps between host 'h5' as source and host 'h6' as destination (i.e., give the two commands for queue configuration and for adding the flows).

4. Attach the two screenshots taken in Step 3.5

5. This question requires that you extend the load\_balancer.sh script and run new Load Balancer Experimentation using the following steps:

i) Scale the load balancer to handle more requests by adding two new hosts h5 (10.0.0.5) and h6 (10.0.0.6) to the load balancer pool and adding the appropriate entries in the load\_balancer.sh script. Run the load\_balancer.sh script again to update the new flow rules.

ii) On the mininet CLI, start the xterm terminals for the new end-hosts h7 and h8 by giving the below command: \$mininet: xterm h7 h8

iii) Start the ping command from hosts h1 and h2 terminals to load balancer 10.0.0.100 and allow the ping to run continuously. Simultaneously, start the ping command from hosts h7 and h8 terminals to the load balancer 10.0.0.100.

What happens when you ping from new end-hosts h7 and h8 to the load balancer 10.0.0.100? Which hosts are responding to the new requests and what does this result suggest? Attach a single screenshot with all the four ping windows running simultaneously.