Part C: Introducing the Controller

1) Try pinging h5 from h1 (Type h1 ping h5 in mininet). The ping should succeed, even though you have not installed any OVS rules on the switches yourself. Study the output in the POX controller terminal. What has the controller done? Include a screenshot of the output as well as your interpretation of it. Make sure that the screenshot is readable. Hint: To interpret the effect of the controller, consider the path ping packets must take from h1 to h5. Now consider the relationship of the nodes depicted in the topology illustrated in Figure 2 and the dpid of the switches in the switch settings and the output of the POX controller.

```
mininet@mininet-vm:~/pox$ ./pox.py --verbose py openflow.of_01 --port=6633 openflow.discover
y forwarding.l2_learning host_tracker
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
INFO:host_tracker:host_tracker ready
DEBUG:core:POX 0.7.0 (gar) going up...
DEBUG:core:Running on CPython (3.6.9/Jan 26 2021 15:33:00)
DEBUG:core:Platform is Linux-4.15.0-112-generic-x86_64-with-Ubuntu-18.04-bionic
WARNING:version:Support for Python 3 is experimental.
INFO:core:POX 0.7.0 (gar) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
Ready.
POX> INFO:openflow.of_01:[00-00-00-00-00-03 4] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-03
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-03 4]
INFO:openflow.of_01:[00-00-00-00-00-02 5] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-02
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-02 5]
INFO:openflow.of_01:[00-00-00-00-00-07 6] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-07
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-07 6]
INFO:openflow.of_01:[00-00-00-00-00-04 7] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-04
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-04 7]
INFO:openflow.of_01:[00-00-00-00-00-05 8] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-05
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-05 8]
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-01
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-01 2]
INFO:openflow.of_01:[00-00-00-00-00-06 3] connected
DEBUG:openflow.discovery:Installing flow for 00-00-00-00-00-06
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-06 3]
INFO:openflow.discovery:link detected: 00-00-00-00-00-02.2 -> 00-00-00-00-00-04.3
INFO:openflow.discovery:link detected: 00-00-00-00-00-02.3 -> 00-00-00-00-00-01.1
INFO:openflow.discovery:link detected: 00-00-00-00-00-05.3 -> 00-00-00-00-00-01.2
INFO:openflow.discovery:link detected: 00-00-00-00-00-05.1 -> 00-00-00-00-00-06.3
INFO:openflow.discovery:link detected: 00-00-00-00-00-02.1 -> 00-00-00-00-00-03.3
INFO:openflow.discovery:link detected: 00-00-00-00-00-01.1 -> 00-00-00-00-00-02.3
INFO:openflow.discovery:link detected: 00-00-00-00-00-03.3 -> 00-00-00-00-00-02.1
INFO:openflow.discovery:link detected: 00-00-00-00-00-04.3 -> 00-00-00-00-00-02.2
INFO:openflow.discovery:link detected: 00-00-00-00-00-05.2 -> 00-00-00-00-00-07.3
INFO:openflow.discovery:link detected: 00-00-00-00-00-01.2 -> 00-00-00-00-00-05.3
INFO:openflow.discovery:link detected: 00-00-00-00-00-06.3 -> 00-00-00-00-00-05.1
INFO:openflow.discovery:link detected: 00-00-00-00-00-07.3 -> 00-00-00-00-00-05.2
DEBUG:openflow.of_01:1 connection aborted
INFO:host_tracker:Learned 3 1 00:00:00:00:00:01
INFO:host_tracker:Learned 3 1 00:00:00:00:00:01 got IP 10.0.0.1
INFO:host_tracker:Learned 6 1 00:00:00:00:00:05
INFO:host_tracker:Learned 6 1 00:00:00:00:00:05 got IP 10.0.0.5
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:05.1 -> 00:00:00:00:00:01.3
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:05.1 -> 00:00:00:00:00:01.3
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:05.2 -> 00:00:00:00:00:01.1
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:05.3 -> 00:00:00:00:00:01.1
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:05.3 -> 00:00:00:00:00:01.1
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:01.1 -> 00:00:00:00:00:05.3
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:01.1 -> 00:00:00:00:00:05.3
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:01.1 -> 00:00:00:00:00:05.2
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:01.3 -> 00:00:00:00:00:05.1
DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:01.3 -> 00:00:00:00:00:05.1
```

POX controller output

sudo mn --controller=remote,ip=127.0.0.1,port=6633 --topo=tree,depth=3 --mac --switch ovs

Creates tree topology and connects POX controller to it and sets the port number to 6633

INFO:openflow.of_01:[00-00-00-00-00-06 2] connected

Indicates controller and switches are connected. The controller connects to all the switches in the topology.

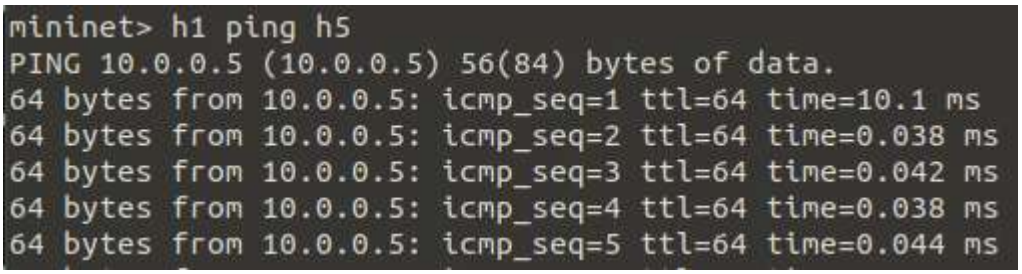
INFO:openflow.discovery:link detected: 00-00-00-00-00-01.1 -> 00-00-00-00-00-02.3

Indicates that links are formed between switches.

DEBUG:forwarding.l2_learning:installing flow for 00:00:00:00:00:05.1 -> 00:00:00:00:00:01.3

Indicates the flow rules are being installed by the controller on the switches to forward message.

2) Take a screenshot of the ping RTT times for the first 5 ping messages. Compare the RTT of the first ping message with the subsequent ones. Is there a difference? Why or why not?



RTT times for first 5 ping messages

The first ping message takes significantly longer than the subsequent ones. The first has an RTT of 10.1ms. This likely due to the controller needing time to compute the flow entries and install them on the necessary switches such that a message can be forwarded from h1 to h5. After the first ping message, since all the flow entries that are necessary for this action have been calculated and installed, they can be reused for the subsequent identical actions. The RTT times for the subsequent pings are much shorter and will remain shorter as they will continue to reuse the previously calculated and installed flow entries without having to repeat that work.

3) Open a new terminal in the CS456 VM and dump the flow rules installed on switch s1 using the command, sudo ovs-ofctl dump-flows s1. Dump the flow rules on all the switches, make sure you change s1 in the command above to the appropriate switch name. Explore the flow rules installed in the switches before and after the initiating ping between hosts. Include a screenshot of the terminal showing all the dump commands and their outputs in your answer, use multiple screenshots, if necessary. The screenshots should be readable and clearly show which output pertains to which switch. Are the OVS rules like the rules you defined in part A? Do some switches have empty flow rule tables, even after the ping? If so, why? Briefly explain what kind of packet forwarding the switches achieve collectively.

```
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=4.306s, table=0, n_packets=4, n_bytes=164, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=4.263s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=7.334s, table=0, n_packets=7, n_bytes=287, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=7.293s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=9.118s, table=0, n_packets=3, n_bytes=123, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=9.076s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=11.199s, table=0, n_packets=3, n_bytes=123, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=11.159s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=13.408s, table=0, n_packets=10, n_bytes=410, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=13.369s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s6
 cookie=0x0, duration=19.528s, table=0, n_packets=5, n_bytes=205, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=19.484s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s7
 cookie=0x0, duration=22.776s, table=0, n_packets=6, n_bytes=246, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=22.735s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
```

Before h1 ping h5

```
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=18.329s, table=0, n_packets=7, n_bytes=574, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=18.326s, table=0, n_packets=6, n_bytes=532, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s1-eth2"
 cookie=0x0, duration=102.865s, table=0, n_packets=42, n_bytes=1722, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=102.822s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=19.988s, table=0, n_packets=7, n_bytes=574, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s2-eth1"
 cookie=0x0, duration=19.986s, table=0, n_packets=6, n_bytes=532, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s2-eth3"
 cookie=0x0, duration=104.522s, table=0, n_packets=63, n_bytes=2583, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=104.481s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=21.652s, table=0, n_packets=7, n_bytes=574, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s3-eth1"
 cookie=0x0, duration=21.651s, table=0, n_packets=6, n_bytes=532, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s3-eth3"
 cookie=0x0, duration=106.188s, table=0, n_packets=21, n_bytes=861, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=106.146s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=109.763s, table=0, n_packets=22, n_bytes=902, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=109.723s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=27.925s, table=0, n_packets=7, n_bytes=574, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s5-eth3"
 cookie=0x0, duration=27.920s, table=0, n_packets=6, n_bytes=532, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s5-eth1"
 cookie=0x0, duration=112.456s, table=0, n_packets=67, n_bytes=2747, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=112.417s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s6
 cookie=0x0, duration=32.112s, table=0, n_packets=7, n_bytes=574, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=output:"s6-eth3"
 cookie=0x0, duration=32.107s, table=0, n_packets=6, n_bytes=532, hard_timeout=1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=output:"s6-eth1"
 cookie=0x0, duration=116.648s, table=0, n_packets=24, n_bytes=984, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=116.604s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
mininet@mininet-vm:~/cs456-a3/part-C$ sudo ovs-ofctl dump-flows s7
 cookie=0x0, duration=119.996s, table=0, n_packets=25, n_bytes=1025, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=119.955s, table=0, n_packets=0, n_bytes=0, priority=32769,arp,dl_dst=02:00:00:00:be:ef actions=CONTROLLER:65535
```

After h1 ping h5

The OVS rules are like the rules I defined in part A. The rules define a source and destination MAC addresses that match, and an in and out port, like what was done in part A. Unlike what was done in part A, we now have a hard_timeout field which is set to 1800.

After the ping, s4 and s7 both have empty flow rule tables. If we look at the tree topology, we notice that to send a message from h1 to h5, we only need flow rules son switches s1, s2, s3, s5, and s6 because these are the switches along the path from h1 to h5 using switches. At no point are s4 and s7 used in the forwarding of messages between h1 and h5.

Collectively, the flow rules allow for messages to be forwarded from h1 to h5 and from h5 to h1 through the switches. Notice that the rules allow for bi-directional forwarding like for s1 we have: dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 on one line

and

dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 on the other.

This allows for bi-directional forwarding which can be applied to the rest of the switches with non-empty flow rule tables. So, h1 can ping h5 and h5 can ping h1.