partC

1

Log generated by POX when doing h1 ping h5:

```
POX> 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
```

Explanation of the logs:

The first 4 lines indicate that the controller is learning about the host ip(e.g. 10.0.0.1 for h1 and 10.0.0.5 for h5) and matches it with MAC address.

The following 10 lines indicate that the controller is installing flows for the path taken for h1 ping h5.

Since the path from h1 to h5 needs to go through 5 switches(s1 s2 s3 s5 s6), and for each of them it need to set both directions for the path. So there are total 10 logs generated.

2

Screen shot for h1 ping h5

```
mininet> h1 ping h5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=17.8 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=0.140 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.088 ms
64 bytes from 10.0.0.5: icmp_seq=4 ttl=64 time=0.048 ms
64 bytes from 10.0.0.5: icmp_seq=5 ttl=64 time=0.065 ms
^C
```

We can see that there is a significant difference between the RTT of the first ping message and the subsequent ones. The first ping message took a lot longer because the controller installed the flow rules to the switches at the same time when the ping message was initially sent. It took time for the controller to figure out how that translates to low-level OpenFlow commands and

deploy it on the network switching devices. After installing the flows, the following ping messages wouldn't need to wait anymore, so it's faster.

3

Screenshots before h1 ping p5

```
mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=64.414s, table=0, n_packets=28, n_bytes=1148, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=64.376s, 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:~$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=83.540s, table=0, n_packets=53, n_bytes=2173, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=83.502s, 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:~$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=87.863s, table=0, n_packets=19, n_bytes=779, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=87.825s, 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:~$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=91.483s, table=0, n_packets=19, n_bytes=779, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=91.445s, 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:~$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=93.697s, table=0, n_packets=56, n_bytes=2296, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=93.659s, 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:~$ sudo ovs-ofctl dump-flows s6
 cookie=0x0, duration=100.638s, 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=100.600s, 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:~$ sudo ovs-ofctl dump-flows s7
 cookie=0x0, duration=103.643s, 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=103.598s, 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

```
mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=16.333s, table=0, n_packets=9, n_bytes=770, 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=16.319s, table=0, n_packets=8, n_bytes=728, 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=173.620s, table=0, n_packets=72, n_bytes=2952, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=173.582s, 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:~$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=23.119s, table=0, n_packets=9, n_bytes=770, 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=23.117s, table=0, n_packets=8, n_bytes=728, 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=180.409s, table=0, n_packets=111, n_bytes=4551, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=180.371s, 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:~$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=27.516s, table=0, n_packets=9, n_bytes=770, 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=27.515s, table=0, n_packets=8, n_bytes=728, 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=184.807s, table=0, n_packets=38, n_bytes=1558, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=184.769s, 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:~$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=191.215s, table=0, n_packets=39, n_bytes=1599, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=191.177s, 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:~$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=37.522s, table=0, n_packets=9, n_bytes=770, 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=37.506s, table=0, n_packets=8, n_bytes=728, 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=194.808s, table=0, n_packets=117, n_bytes=4797, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=194.770s, 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:~$ sudo ovs-ofctl dump-flows s6
 cookie=0x0, duration=40.905s, table=0, n_packets=9, n_bytes=770, 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=40.889s, table=0, n_packets=8, n_bytes=728, 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=198.191s, table=0, n_packets=41, n_bytes=1681, priority=65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=198.153s, 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:~$ sudo ovs-ofctl dump-flows s7
 cookie=0x0, duration=202.235s, 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=202.190s, 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:~$
```

What do you think the initial rules are for?

The initial rules installed on the switches are for the controller to discover the topology of the network and do MAC mapping to ip address.

For example, the first rule matching dl_type=0x88cc means this packet is a Link Layer Discovery Protocol(LLDP) packet and it should be sent to controler(action=CONTROLLER:65535)

Similarly, the second rule with the key word arp means this packet is an Address Resolution Protocol packet and it should also be sent to controller.

Do all switches have newly installed flow rules after the ping?

We can see that after h1 ping h5, the switches on the path between h1 and h5(namely s1 s2 s3 s5 s6) all install new flows on the switch. The other switches don't install new flow rules. Because the flow rules are dynamically controlled by the controller and if there is not packet passing through, the switches won't be installed with new rules.

Observe how the OVS rules are different from the rules you defined in part A.

Compared to the rules we installed in part A

Manually installed rules:
in_port=1,ip,nw_src=10.0.0.2,nw_dst=10.0.2.2,actions=mod_dl_src:0A:00:0B:01:00:03,mod_dl_dst:0A:00:0B:FE:00:02,output=3

Controller auto installed rules:

cookie=0x0, duration=<time>, table=0, n_packets=<count>, n_bytes=<count>, priority=<priority>, dl_dst=<destination MAC>, dl_type=<EtherType> actions=<actions>

We can see that the auto installed rules have fields like cookie, duration, table, n_packets, n_types, priority, dl_type which have the following meanings:

Cookie: acts as a unique identification for controller to monitor and manage flow rules.

Duration: time since this flow has been activated.

Table: table 0 means this is the initial table.

n_packets: the number of packets has matched this flow rule.

n_bytes: the number of bytes has matched this flow rule.

Priority: the priority level of flow rule, higher value means higher priority

dl_type: etherType of the flow rule, particular value represents particular protocol.

In conclusion, controller auto installed rules have many fields for controller's overall management and statistic record over the network.