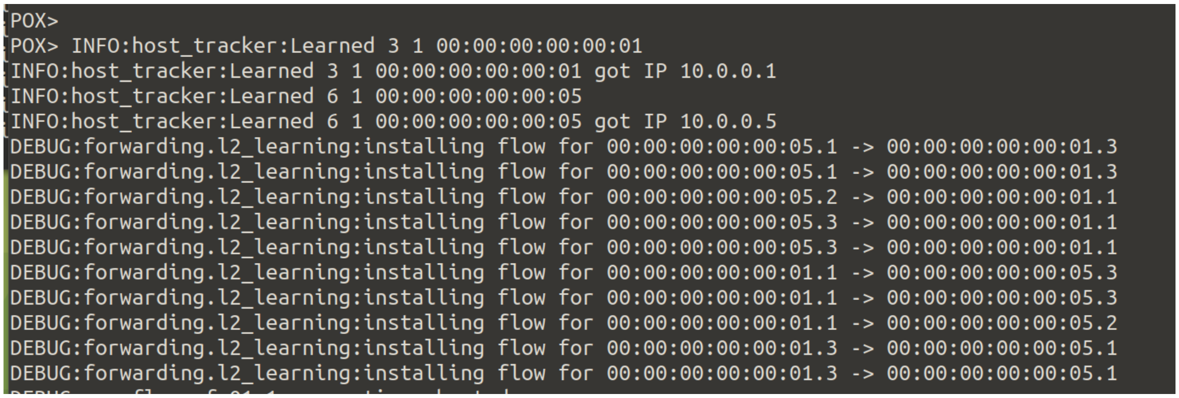
Part C

1. **Pinging from h1 to h5**



```
POX>
POX> 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
```

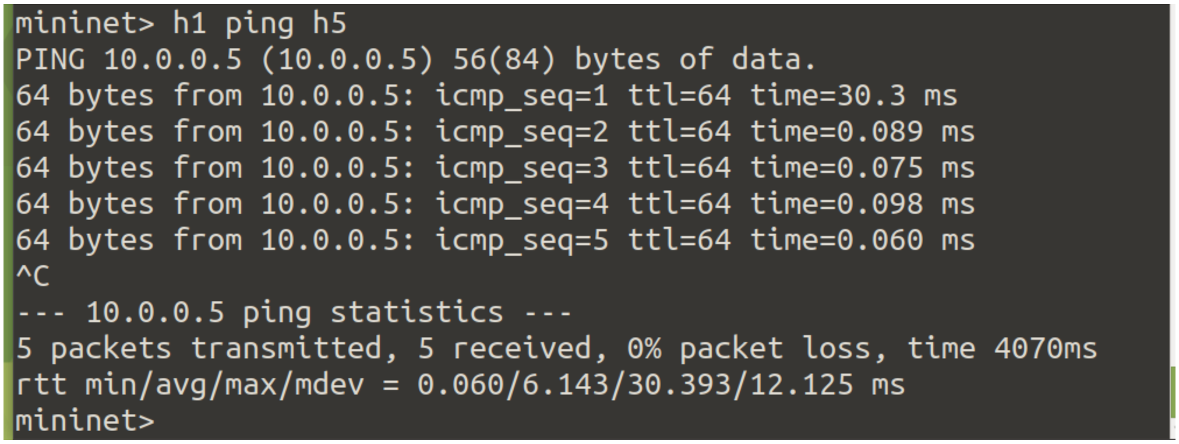
To get from h1 to h5, the packet must pass through s3, s2, s1, s5, and finally s6.

The controller installs the correct flow rules needed so that packets can pass from h1 to h5 along this path.

We first see the controller getting the IP addresses for the two hosts we are pinging and the respective MAC addrs of the switches they are attached to. Then the controller begins to install flows that allow forwarding along the s3 =¿ s2 =¿ s1 =¿ s5 =¿ s6 path.

The l2_learning logs only show the MAC addresses being matched to and the in / out ports to use if the packet matches. We see 5 entries for the MAC of h1, and 5 entries for the MAC of h5. Each entry is adding a flow rule setting the output port if it matches to the input port and src/dst MACs.

2. **RTT for first 5 pings**



```
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=30.3 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=0.089 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 10.0.0.5: icmp_seq=4 ttl=64 time=0.098 ms
64 bytes from 10.0.0.5: icmp_seq=5 ttl=64 time=0.060 ms
^C
--- 10.0.0.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4070ms
rtt min/avg/max/mdev = 0.060/6.143/30.393/12.125 ms
mininet>
```

The first ping is significantly slower because it needs to wait for the controller to install the flow rules.

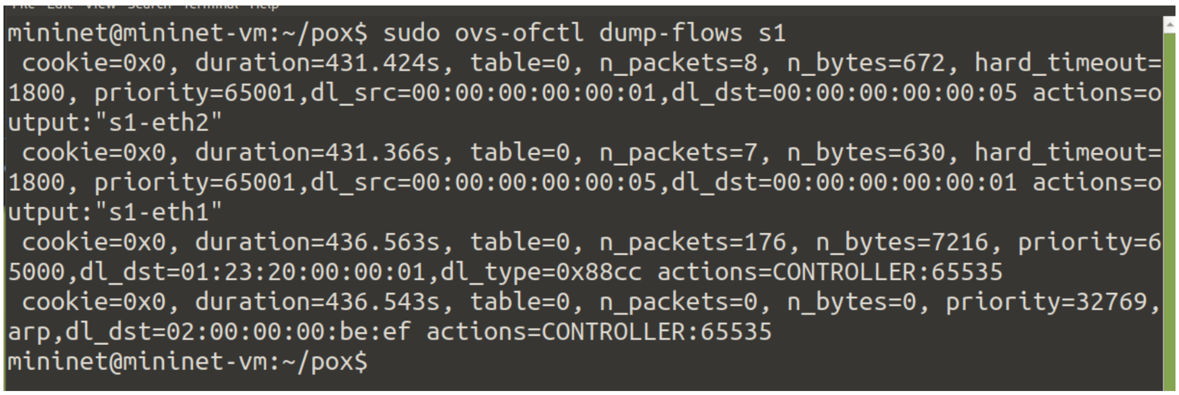
3. **Flow rules for all switches**

Screenshots taken after pinging h1 to h5.

Before the ping, all the flow tables started off with two entries which pointed to the controller. After the ping, we can see entries in the tables of all the switches that the packet passed through.

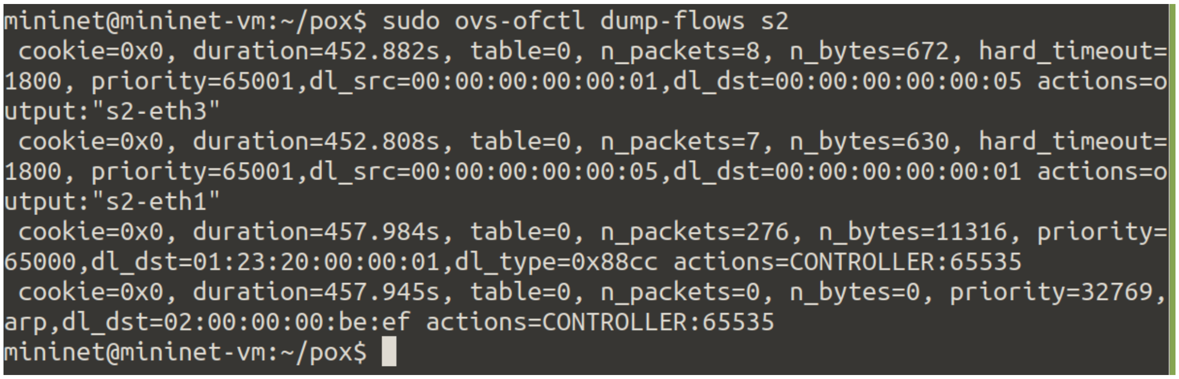
After the ping, s4 and s7 still don't have any flow rules defined in them (besides the ones pointing to the controller). This is because the packets going from h1 to h5 did not need to pass through these switches, so they were left out of the communication.

The collective effect of these flows is destination-based forwarding, since it's allowing any and all traffic to flow between h1 and h5 without specific matches being required. It differs from partA and partB because there we restricted our forwarding to specific matches at the IP and MAC levels.



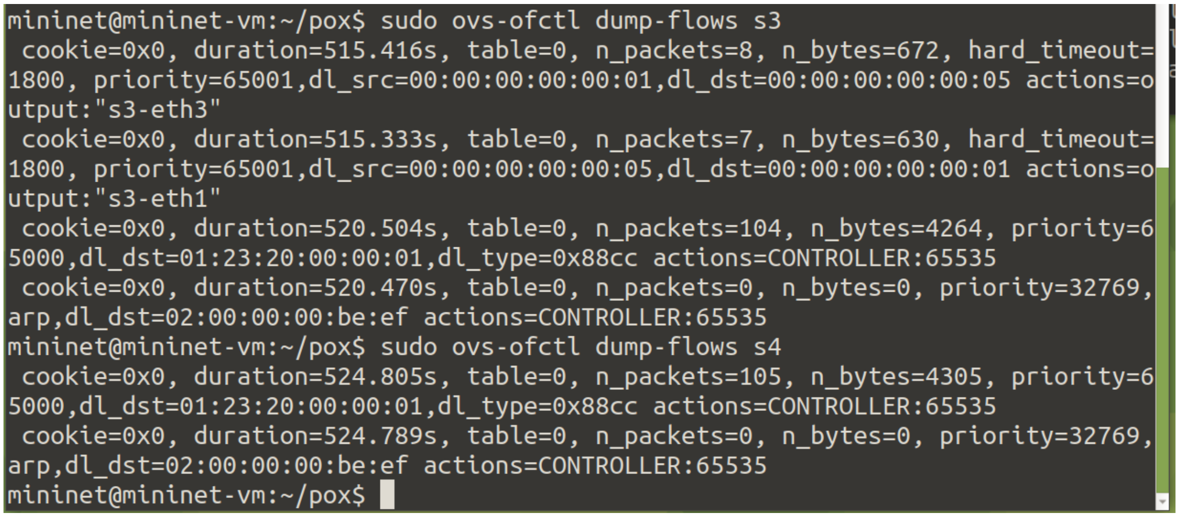
```
mininet@mininet-vm:~/pox$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=431.424s, table=0, n_packets=8, n_bytes=672, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=o
utput:"s1-eth2"
 cookie=0x0, duration=431.366s, table=0, n_packets=7, n_bytes=630, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=o
utput:"s1-eth1"
 cookie=0x0, duration=436.563s, table=0, n_packets=176, n_bytes=7216, priority=6
5000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=436.543s, 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:~/pox$
```

Figure 1: s1



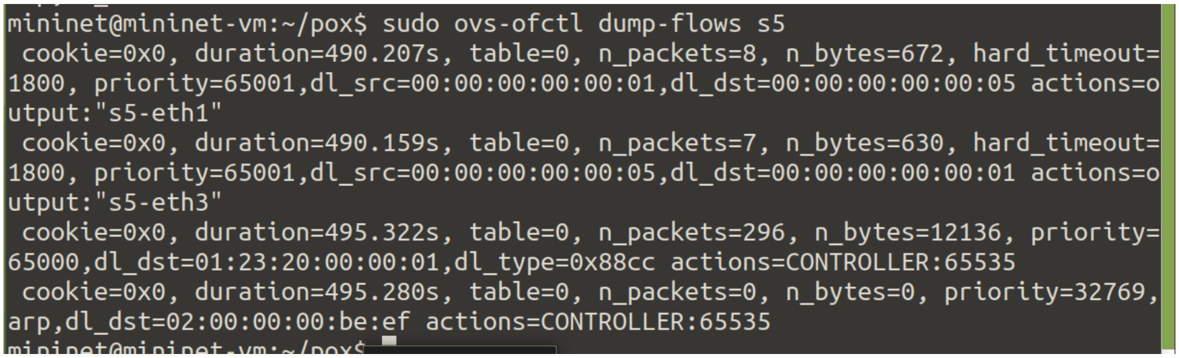
```
mininet@mininet-vm:~/pox$ sudo ovs-ofctl dump-flows s2
 cookie=0x0, duration=452.882s, table=0, n_packets=8, n_bytes=672, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=o
utput:"s2-eth3"
 cookie=0x0, duration=452.808s, table=0, n_packets=7, n_bytes=630, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=o
utput:"s2-eth1"
 cookie=0x0, duration=457.984s, table=0, n_packets=276, n_bytes=11316, priority=
65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=457.945s, 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:~/pox$ 
```

Figure 2: s2



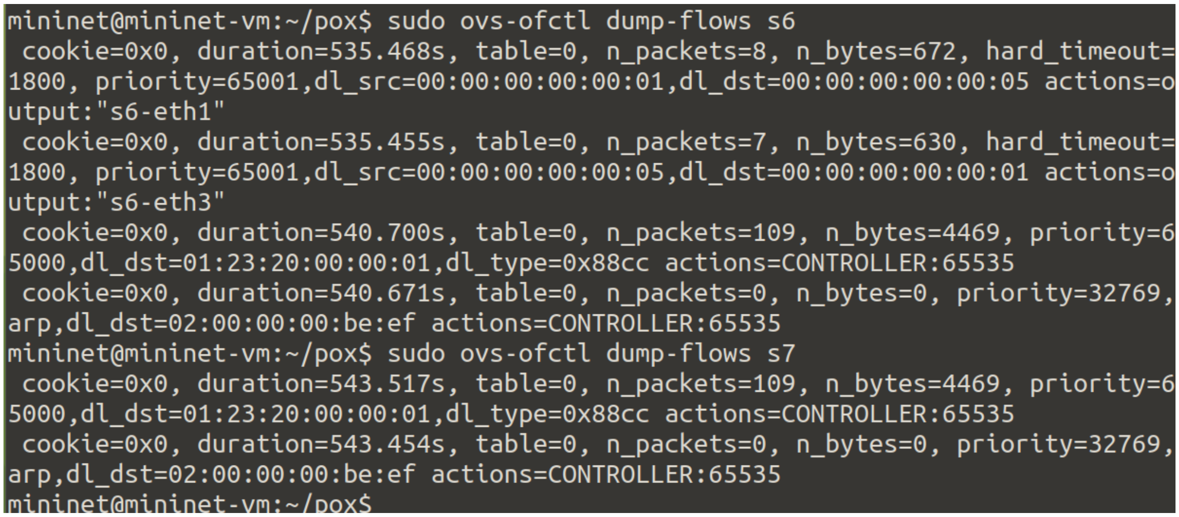
```
mininet@mininet-vm:~/pox$ sudo ovs-ofctl dump-flows s3
 cookie=0x0, duration=515.416s, table=0, n_packets=8, n_bytes=672, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=o
utput:"s3-eth3"
 cookie=0x0, duration=515.333s, table=0, n_packets=7, n_bytes=630, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=o
utput:"s3-eth1"
 cookie=0x0, duration=520.504s, table=0, n_packets=104, n_bytes=4264, priority=6
5000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=520.470s, 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:~/pox$ sudo ovs-ofctl dump-flows s4
 cookie=0x0, duration=524.805s, table=0, n_packets=105, n_bytes=4305, priority=6
5000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=524.789s, 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:~/pox$
```

Figure 3: s3 and s4



```
mininet@mininet-vm:~/pox$ sudo ovs-ofctl dump-flows s5
 cookie=0x0, duration=490.207s, table=0, n_packets=8, n_bytes=672, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=o
utput:"s5-eth1"
 cookie=0x0, duration=490.159s, table=0, n_packets=7, n_bytes=630, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=o
utput:"s5-eth3"
 cookie=0x0, duration=495.322s, table=0, n_packets=296, n_bytes=12136, priority=
65000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=495.280s, 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:~/pox$
```

Figure 4: s5



```
mininet@mininet-vm:~/pox$ sudo ovs-ofctl dump-flows s6
 cookie=0x0, duration=535.468s, table=0, n_packets=8, n_bytes=672, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05 actions=o
utput:"s6-eth1"
 cookie=0x0, duration=535.455s, table=0, n_packets=7, n_bytes=630, hard_timeout=
1800, priority=65001,dl_src=00:00:00:00:00:05,dl_dst=00:00:00:00:00:01 actions=o
utput:"s6-eth3"
 cookie=0x0, duration=540.700s, table=0, n_packets=109, n_bytes=4469, priority=6
5000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=540.671s, 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:~/pox$ sudo ovs-ofctl dump-flows s7
 cookie=0x0, duration=543.517s, table=0, n_packets=109, n_bytes=4469, priority=6
5000,dl_dst=01:23:20:00:00:01,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=543.454s, 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:~/pox$
```

Figure 5: s6 and s7