

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

CSEN 241 HW 3 Mininet & OpenFlow

Task 1 Questions

1. What is the output of “nodes” and “net.”

Ans –

- a. The output of “nodes” is.

```
[mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 h7 h8 s1 s2 s3 s4 s5 s6 s7
```

- b. The output of “net” is.

```
[mininet> net
h1 h1-eth0:s3-eth1
h2 h2-eth0:s3-eth2
h3 h3-eth0:s4-eth1
h4 h4-eth0:s4-eth2
h5 h5-eth0:s6-eth1
h6 h6-eth0:s6-eth2
h7 h7-eth0:s7-eth1
h8 h8-eth0:s7-eth2
s1 lo:  s1-eth1:s2-eth3  s1-eth2:s5-eth3
s2 lo:  s2-eth1:s3-eth3  s2-eth2:s4-eth3  s2-eth3:s1-eth1
s3 lo:  s3-eth1:h1-eth0  s3-eth2:h2-eth0  s3-eth3:s2-eth1
s4 lo:  s4-eth1:h3-eth0  s4-eth2:h4-eth0  s4-eth3:s2-eth2
s5 lo:  s5-eth1:s6-eth3  s5-eth2:s7-eth3  s5-eth3:s1-eth2
s6 lo:  s6-eth1:h5-eth0  s6-eth2:h6-eth0  s6-eth3:s5-eth1
s7 lo:  s7-eth1:h7-eth0  s7-eth2:h8-eth0  s7-eth3:s5-eth2
c0
```

2. What is the output of “h7 ifconfig”

Ans – The output of “h7 ifconfig” is.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
[mininet> h7 ifconfig
h7-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.7 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::8426:f8ff:feac:1ab8 prefixlen 64 scopeid 0x20<link>
        ether 86:26:f8:ac:1a:b8 txqueuelen 1000 (Ethernet)
        RX packets 36 bytes 2492 (2.4 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 12 bytes 936 (936.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Task 2 Questions –

1. Draw the function call graph of this controller. For example, once a packet comes to the controller, which function is the first to be called, which one is the second, and so forth?

Ans –

- a. First, we need to start the POX listener using the following command:
`./pox.py log.level --DEBUG misc.of_tutorial`

The function call graph is –

start switch: `_handle_PacketIn()` -> `act_like_hub()` -> `resend_packet()` -> `send(msg)`

Firstly, ‘start switch’ command calls the method ‘`_handle_PacketIn()`’ which handles the packet received from the switch in the message. This method would call the ‘`act_like_hub()`’ method, that sends the packets to all the ports except the input port and simulates the hub environment. Next, the `resend_packet()` method is called. This method adds a packet to the message data and performs the action on it. The switch is then instructed to resend the packet to a specified port by this message.

2. Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 p2).

Ans –

Consider following two images for h1 ping h2 for 100 times.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
mininet> h1 ping -c100 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=462 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=148 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=119 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=80.3 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=178 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=147 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=152 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=202 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=73.9 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=98.7 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=149 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=115 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=141 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=72.8 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=83.1 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=83.7 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=239 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=83.5 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=148.8 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=184 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=314 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=146 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=112 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=88.3 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=69.2 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=69.4 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=67.5 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=175.7 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=69.6 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=233 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=127 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=89.7 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=91.3 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=151 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=116 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=131 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=186 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=153 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=74.6 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=153 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=497 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=286 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=141 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=127 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=287 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=439 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=171 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=129 ms
64 bytes from 10.0.0.2: icmp_seq=51 ttl=64 time=65.9 ms
64 bytes from 10.0.0.2: icmp_seq=52 ttl=64 time=177 ms
64 bytes from 10.0.0.2: icmp_seq=53 ttl=64 time=189 ms
64 bytes from 10.0.0.2: icmp_seq=54 ttl=64 time=176 ms
64 bytes from 10.0.0.2: icmp_seq=55 ttl=64 time=273 ms
64 bytes from 10.0.0.2: icmp_seq=56 ttl=64 time=128 ms
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=59.7 ms
64 bytes from 10.0.0.2: icmp_seq=58 ttl=64 time=175 ms
64 bytes from 10.0.0.2: icmp_seq=59 ttl=64 time=180 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=105 ms
64 bytes from 10.0.0.2: icmp_seq=61 ttl=64 time=49.7 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=130 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=264 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=234 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=84.2 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=72.4 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=124 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=138 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=117 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=62.3 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=115 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=88.8 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=150 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=388 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=126 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=75.3 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=117 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=112 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=77.1 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=78.5 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=200 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=117 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=398 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=137 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=80.4 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=78.8 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=107 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=92.2 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=67.6 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=47.5 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=73.4 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=165 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=312 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=74.5 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=139 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=92.0 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=165 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=99.2 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=84.0 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=196 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99353ms
rtt min/avg/max/mdev = 47.481/145.525/591.913/96.133 ms
```

Now, consider following two images for h1 ping h8 for 100 times.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
[mininet> h1 ping -c100 h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=844 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=1370 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=699 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=504 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=760 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=648 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=1553 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=1616 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=1288 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=1389 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=1121 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=1757 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=1176 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=820 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=667 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=627 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=556 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=578 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=619 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=559 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=541 ms
64 bytes from 10.0.0.8: icmp_seq=22 ttl=64 time=593 ms
64 bytes from 10.0.0.8: icmp_seq=23 ttl=64 time=1049 ms
64 bytes from 10.0.0.8: icmp_seq=24 ttl=64 time=761 ms
64 bytes from 10.0.0.8: icmp_seq=25 ttl=64 time=676 ms
64 bytes from 10.0.0.8: icmp_seq=26 ttl=64 time=695 ms
64 bytes from 10.0.0.8: icmp_seq=27 ttl=64 time=748 ms
64 bytes from 10.0.0.8: icmp_seq=28 ttl=64 time=868 ms
64 bytes from 10.0.0.8: icmp_seq=29 ttl=64 time=836 ms
64 bytes from 10.0.0.8: icmp_seq=30 ttl=64 time=810 ms
64 bytes from 10.0.0.8: icmp_seq=31 ttl=64 time=688 ms
64 bytes from 10.0.0.8: icmp_seq=32 ttl=64 time=739 ms
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=758 ms
64 bytes from 10.0.0.8: icmp_seq=34 ttl=64 time=697 ms
64 bytes from 10.0.0.8: icmp_seq=35 ttl=64 time=739 ms
64 bytes from 10.0.0.8: icmp_seq=36 ttl=64 time=802 ms
64 bytes from 10.0.0.8: icmp_seq=37 ttl=64 time=792 ms
64 bytes from 10.0.0.8: icmp_seq=38 ttl=64 time=1141 ms
64 bytes from 10.0.0.8: icmp_seq=39 ttl=64 time=1123 ms
64 bytes from 10.0.0.8: icmp_seq=40 ttl=64 time=866 ms
64 bytes from 10.0.0.8: icmp_seq=41 ttl=64 time=761 ms
64 bytes from 10.0.0.8: icmp_seq=42 ttl=64 time=828 ms
64 bytes from 10.0.0.8: icmp_seq=43 ttl=64 time=784 ms
64 bytes from 10.0.0.8: icmp_seq=44 ttl=64 time=744 ms
64 bytes from 10.0.0.8: icmp_seq=45 ttl=64 time=575 ms
64 bytes from 10.0.0.8: icmp_seq=46 ttl=64 time=674 ms
64 bytes from 10.0.0.8: icmp_seq=47 ttl=64 time=1006 ms
64 bytes from 10.0.0.8: icmp_seq=48 ttl=64 time=776 ms
64 bytes from 10.0.0.8: icmp_seq=49 ttl=64 time=864 ms
64 bytes from 10.0.0.8: icmp_seq=50 ttl=64 time=733 ms
64 bytes from 10.0.0.8: icmp_seq=51 ttl=64 time=689 ms
64 bytes from 10.0.0.8: icmp_seq=52 ttl=64 time=944 ms
64 bytes from 10.0.0.8: icmp_seq=53 ttl=64 time=2151 ms
64 bytes from 10.0.0.8: icmp_seq=54 ttl=64 time=1673 ms
64 bytes from 10.0.0.8: icmp_seq=55 ttl=64 time=1065 ms
64 bytes from 10.0.0.8: icmp_seq=56 ttl=64 time=1126 ms
64 bytes from 10.0.0.8: icmp_seq=57 ttl=64 time=1542 ms
64 bytes from 10.0.0.8: icmp_seq=58 ttl=64 time=998 ms
64 bytes from 10.0.0.8: icmp_seq=59 ttl=64 time=539 ms
64 bytes from 10.0.0.8: icmp_seq=60 ttl=64 time=750 ms
64 bytes from 10.0.0.8: icmp_seq=61 ttl=64 time=535 ms
64 bytes from 10.0.0.8: icmp_seq=62 ttl=64 time=729 ms
64 bytes from 10.0.0.8: icmp_seq=63 ttl=64 time=583 ms
64 bytes from 10.0.0.8: icmp_seq=64 ttl=64 time=528 ms
64 bytes from 10.0.0.8: icmp_seq=65 ttl=64 time=605 ms
64 bytes from 10.0.0.8: icmp_seq=66 ttl=64 time=718 ms
64 bytes from 10.0.0.8: icmp_seq=67 ttl=64 time=633 ms
64 bytes from 10.0.0.8: icmp_seq=68 ttl=64 time=798 ms
64 bytes from 10.0.0.8: icmp_seq=69 ttl=64 time=764 ms
64 bytes from 10.0.0.8: icmp_seq=70 ttl=64 time=788 ms
64 bytes from 10.0.0.8: icmp_seq=71 ttl=64 time=533 ms
64 bytes from 10.0.0.8: icmp_seq=72 ttl=64 time=1011 ms
64 bytes from 10.0.0.8: icmp_seq=73 ttl=64 time=777 ms
64 bytes from 10.0.0.8: icmp_seq=74 ttl=64 time=811 ms
64 bytes from 10.0.0.8: icmp_seq=75 ttl=64 time=730 ms
64 bytes from 10.0.0.8: icmp_seq=76 ttl=64 time=504 ms
64 bytes from 10.0.0.8: icmp_seq=77 ttl=64 time=667 ms
64 bytes from 10.0.0.8: icmp_seq=78 ttl=64 time=615 ms
64 bytes from 10.0.0.8: icmp_seq=79 ttl=64 time=547 ms
64 bytes from 10.0.0.8: icmp_seq=80 ttl=64 time=673 ms
64 bytes from 10.0.0.8: icmp_seq=81 ttl=64 time=408 ms
64 bytes from 10.0.0.8: icmp_seq=82 ttl=64 time=566 ms
64 bytes from 10.0.0.8: icmp_seq=83 ttl=64 time=867 ms
64 bytes from 10.0.0.8: icmp_seq=84 ttl=64 time=623 ms
64 bytes from 10.0.0.8: icmp_seq=85 ttl=64 time=630 ms
64 bytes from 10.0.0.8: icmp_seq=86 ttl=64 time=611 ms
64 bytes from 10.0.0.8: icmp_seq=87 ttl=64 time=809 ms
64 bytes from 10.0.0.8: icmp_seq=88 ttl=64 time=743 ms
64 bytes from 10.0.0.8: icmp_seq=89 ttl=64 time=664 ms
64 bytes from 10.0.0.8: icmp_seq=90 ttl=64 time=469 ms
64 bytes from 10.0.0.8: icmp_seq=91 ttl=64 time=443 ms
64 bytes from 10.0.0.8: icmp_seq=92 ttl=64 time=621 ms
64 bytes from 10.0.0.8: icmp_seq=93 ttl=64 time=602 ms
64 bytes from 10.0.0.8: icmp_seq=94 ttl=64 time=381 ms
64 bytes from 10.0.0.8: icmp_seq=95 ttl=64 time=421 ms
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=514 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=1614 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=1143 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=915 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=426 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99567ms
rtt min/avg/max/mdev = 381.041/811.544/2151.489/320.768 ms, pipe 3
```

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

- a. How long does it take (on average) to ping for each case?

Ans –

For h1 ping h2, 100 times, on average it takes about 145.525 ms to ping.

For h1 ping h8, 100 times, it takes about average of 811.544 ms to ping.

- b. What is the minimum and maximum ping you have observed?

Ans –

For h1 ping h2 100 times, the min ping takes 47.481 ms and max ping takes 591.913 ms.

For h1 ping h8 100 times, the min ping takes 381.041 ms and max ping takes 2151.489 ms.

- c. What is the difference, and why?

Ans – The ping time for h1 to h8 is higher than the ping time for h1 to h2 because of the number of hops present between h1 to h8 are more than the number of hops present between h1 to h2. Between h1 to h2 there is only one switch present, that is, s3. Whereas there are switches s3, s2, s1, s5 and s7 present between h1 to h8 and thus, the packet needs to traverse through all the switches to reach from h1 to h8.

3. Run “iperf h1 h2” and “iperf h1 h8”

- a. What is “iperf” used for?

Ans - Iperf is an open-source tool designed to assist administrators in gauging the bandwidth and assessing the performance and quality of a network connection between two hosts. It enables the measurement of throughput between any two nodes on a network line.

- b. What is the throughput for each case?

Ans –

Following is the output of the iperf command showcasing the throughput results for each case:

```
[mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['105 Kbytes/sec', '158 Kbytes/sec']
[mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['81.1 Kbytes/sec', '85.3 Kbytes/sec']
```

- c. What is the difference and explain the reasons for the difference.

Ans – The throughput of h1 to h2 is higher than the throughput of h1 to h8 since, the number of hops between h1 to h2 are less than the number of hops between h1 to h8. There is only one switch s3 present between h1 and h2, thus more data can be transferred in given time. While there are s3, s2, s1, s5, s7 switches present between h1 and h8, and thus the packets need to traverse through 5 switches to reach h8, resulting in less data transmission in given time.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

4. Which of the switches observe traffic? Please describe your way for observing such traffic on switches (e.g., adding some functions in the “of_tutorial” controller).

Ans - By adding log.info ("Switch observing traffic: %s" % (self.connection) in the line number 107 "of_tutorial" controller we can see the information which helps us to observe the traffic.

After viewing that, we can conclude that all the switches view the traffic, specifically when all are flooded with packets. The _handle_PacketIn function is the event listener, so it's called every time a packet is received.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

Task 3 Questions –

1. Describe how the above code works, such as how the "MAC to Port" map is established.

You could use a ‘ping’ example to describe the establishment process (e.g., h1 ping h2).

Ans – The ‘act_like_switch’ function helps us to determine where the MAC addresses are located. Thus, when a MAC address is determined to be the address a sender wishes to send a message to, the controller can map that MAC address to a port. It is very convenient as it helps the controller speed up the process of delivering packets to already known addresses, as it sends all the packets to that known port. In case the destination is not known, the function just floods the packets to all the destinations. Because flooding occurs less frequently, the MAC Learning Controller helps to improve ping times and throughput.

```
root@c32eb3499cf4:~/pox# ./pox.py log.level --DEBUG misc.of_tutorial
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
DEBUG:core:POX 0.7.0 (gar) going up...
DEBUG:core:Running on CPython (3.10.6/Mar 10 2023 10:55:28)
DEBUG:core:Platform is Linux-5.4.0-172-generic-x86_64-with-glibc2.35
WARNING:version:POX requires one of the following versions of Python: 3.6 3.7 3.8 3.9
WARNING:version:You're running Python 3.10.
WARNING:version:If you run into problems, try using a supported version.
INFO:core:POX 0.7.0 (gar) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[00-00-00-00-00-01 1] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 1]
INFO:openflow.of_01:[00-00-00-00-00-03 2] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-03 2]
INFO:openflow.of_01:[00-00-00-00-00-02 3] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-02 3]
INFO:openflow.of_01:[00-00-00-00-00-05 4] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-05 4]
INFO:openflow.of_01:[00-00-00-00-00-07 5] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-07 5]
INFO:openflow.of_01:[00-00-00-00-00-04 6] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-04 6]
INFO:openflow.of_01:[00-00-00-00-00-06 7] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-06 7]
Learning that 5a:b4:4d:54:eb:82 is attached at port 1
ff:ff:ff:ff:ff:ff not known, resent to everybody
Learning that 5a:b4:4d:54:eb:82 is attached at port 1
ff:ff:ff:ff:ff:ff not known, resent to everybody
Learning that 6e:42:a6:a0:6e:cc is attached at port 2
5a:b4:4d:54:eb:82 destination known. only send message to it
Learning that 5a:b4:4d:54:eb:82 is attached at port 1
ff:ff:ff:ff:ff:ff not known, resent to everybody
6e:42:a6:a0:6e:cc destination known. only send message to it
Learning that 5a:b4:4d:54:eb:82 is attached at port 3
ff:ff:ff:ff:ff:ff not known, resent to everybody
5a:b4:4d:54:eb:82 destination known. only send message to it
Learning that 5a:b4:4d:54:eb:82 is attached at port 3
ff:ff:ff:ff:ff:ff not known, resent to everybody
Learning that 5a:b4:4d:54:eb:82 is attached at port 3
ff:ff:ff:ff:ff:ff not known, resent to everybody
```

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

The output of h1 ping h2 is –

```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=476 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=108 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=88.4 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=143 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=121 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=78.2 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=180 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=186 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=252 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=154 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=117 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=128 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=175 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=103 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=192 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=115 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=92.8 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=132 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=100 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=83.5 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=99.2 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=149 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=142 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=268 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=137 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=124 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=109 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=128 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=120 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=146 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=200 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=134 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=151 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=156 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=94.0 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=128 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=130 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=205 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=174 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=113 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=128 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=144 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=202 ms
```

2. Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 p2).

Ans –

Consider the following two screenshots for h1 ping h2 for 100 times.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
mininet> h1 ping -c100 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=43.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=60.8 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=110 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=50.9 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=65.3 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=57.6 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=68.2 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=98.9 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=59.2 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=66.6 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=55.7 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=62.3 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=149 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=52.1 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=61.8 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=47.3 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=47.3 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=101 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=63.9 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=63.3 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=61.0 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=35.6 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=72.7 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=38.1 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=37.8 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=40.2 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=58.5 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=82.9 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=55.1 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=47.9 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=26.9 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=54.8 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=50.3 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=44.5 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=50.1 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=56.5 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=43.9 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=79.5 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=55.7 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=70.7 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=53.5 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=67.2 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=75.4 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=42.2 ms
```

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=42.2 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=61.9 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=95.9 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=36.4 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=62.6 ms
64 bytes from 10.0.0.2: icmp_seq=50 ttl=64 time=57.9 ms
64 bytes from 10.0.0.2: icmp_seq=51 ttl=64 time=65.4 ms
64 bytes from 10.0.0.2: icmp_seq=52 ttl=64 time=56.8 ms
64 bytes from 10.0.0.2: icmp_seq=53 ttl=64 time=58.8 ms
64 bytes from 10.0.0.2: icmp_seq=54 ttl=64 time=55.9 ms
64 bytes from 10.0.0.2: icmp_seq=55 ttl=64 time=38.1 ms
64 bytes from 10.0.0.2: icmp_seq=56 ttl=64 time=35.8 ms
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=41.7 ms
64 bytes from 10.0.0.2: icmp_seq=58 ttl=64 time=60.9 ms
64 bytes from 10.0.0.2: icmp_seq=59 ttl=64 time=111 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=37.8 ms
64 bytes from 10.0.0.2: icmp_seq=61 ttl=64 time=50.6 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=55.7 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=62.1 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=144 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=58.2 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=85.7 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=58.2 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=58.2 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=162 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=28.2 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=62.1 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=53.6 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=62.2 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=45.0 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=49.9 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=53.1 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=68.2 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=32.2 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=89.3 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=69.1 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=159 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=61.4 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=59.6 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=234 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=62.0 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=39.0 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=31.3 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=71.5 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=69.5 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=65.8 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=85.0 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=33.7 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=35.3 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=799 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=381 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=247 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=54.6 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=69.2 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=125 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=48.3 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99343ms
rtt min/avg/max/mdev = 26.892/77.227/799.114/86.761 ms
```

Now, consider the following two scenarios for h1 ping h8 100 times.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
mininet> h1 ping -c100 h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=663 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=393 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=245 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=266 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=271 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=263 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=457 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=291 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=264 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=303 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=255 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=388 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=310 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=304 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=203 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=240 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=327 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=214 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=190 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=240 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=213 ms
64 bytes from 10.0.0.8: icmp_seq=22 ttl=64 time=406 ms
64 bytes from 10.0.0.8: icmp_seq=23 ttl=64 time=231 ms
64 bytes from 10.0.0.8: icmp_seq=24 ttl=64 time=214 ms
64 bytes from 10.0.0.8: icmp_seq=25 ttl=64 time=173 ms
64 bytes from 10.0.0.8: icmp_seq=26 ttl=64 time=245 ms
64 bytes from 10.0.0.8: icmp_seq=27 ttl=64 time=427 ms
64 bytes from 10.0.0.8: icmp_seq=28 ttl=64 time=220 ms
64 bytes from 10.0.0.8: icmp_seq=29 ttl=64 time=264 ms
64 bytes from 10.0.0.8: icmp_seq=30 ttl=64 time=297 ms
64 bytes from 10.0.0.8: icmp_seq=31 ttl=64 time=294 ms
64 bytes from 10.0.0.8: icmp_seq=32 ttl=64 time=468 ms
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=259 ms
64 bytes from 10.0.0.8: icmp_seq=34 ttl=64 time=210 ms
64 bytes from 10.0.0.8: icmp_seq=35 ttl=64 time=254 ms
64 bytes from 10.0.0.8: icmp_seq=36 ttl=64 time=257 ms
64 bytes from 10.0.0.8: icmp_seq=37 ttl=64 time=450 ms
64 bytes from 10.0.0.8: icmp_seq=38 ttl=64 time=390 ms
64 bytes from 10.0.0.8: icmp_seq=39 ttl=64 time=271 ms
64 bytes from 10.0.0.8: icmp_seq=40 ttl=64 time=231 ms
64 bytes from 10.0.0.8: icmp_seq=41 ttl=64 time=252 ms
64 bytes from 10.0.0.8: icmp_seq=42 ttl=64 time=562 ms
64 bytes from 10.0.0.8: icmp_seq=43 ttl=64 time=382 ms
64 bytes from 10.0.0.8: icmp_seq=44 ttl=64 time=158 ms
64 bytes from 10.0.0.8: icmp_seq=45 ttl=64 time=151 ms
64 bytes from 10.0.0.8: icmp_seq=46 ttl=64 time=245 ms
64 bytes from 10.0.0.8: icmp_seq=47 ttl=64 time=493 ms
64 bytes from 10.0.0.8: icmp_seq=48 ttl=64 time=274 ms
64 bytes from 10.0.0.8: icmp_seq=49 ttl=64 time=263 ms
64 bytes from 10.0.0.8: icmp_seq=50 ttl=64 time=234 ms
64 bytes from 10.0.0.8: icmp_seq=51 ttl=64 time=205 ms
64 bytes from 10.0.0.8: icmp_seq=52 ttl=64 time=580 ms
64 bytes from 10.0.0.8: icmp_seq=53 ttl=64 time=181 ms
64 bytes from 10.0.0.8: icmp_seq=54 ttl=64 time=223 ms
64 bytes from 10.0.0.8: icmp_seq=55 ttl=64 time=214 ms
```

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

```
64 bytes from 10.0.0.8: icmp_seq=56 ttl=64 time=264 ms
64 bytes from 10.0.0.8: icmp_seq=57 ttl=64 time=423 ms
64 bytes from 10.0.0.8: icmp_seq=58 ttl=64 time=244 ms
64 bytes from 10.0.0.8: icmp_seq=59 ttl=64 time=295 ms
64 bytes from 10.0.0.8: icmp_seq=60 ttl=64 time=255 ms
64 bytes from 10.0.0.8: icmp_seq=61 ttl=64 time=201 ms
64 bytes from 10.0.0.8: icmp_seq=62 ttl=64 time=603 ms
64 bytes from 10.0.0.8: icmp_seq=63 ttl=64 time=201 ms
64 bytes from 10.0.0.8: icmp_seq=64 ttl=64 time=175 ms
64 bytes from 10.0.0.8: icmp_seq=65 ttl=64 time=225 ms
64 bytes from 10.0.0.8: icmp_seq=66 ttl=64 time=227 ms
64 bytes from 10.0.0.8: icmp_seq=67 ttl=64 time=371 ms
64 bytes from 10.0.0.8: icmp_seq=68 ttl=64 time=234 ms
64 bytes from 10.0.0.8: icmp_seq=69 ttl=64 time=224 ms
64 bytes from 10.0.0.8: icmp_seq=70 ttl=64 time=261 ms
64 bytes from 10.0.0.8: icmp_seq=71 ttl=64 time=221 ms
64 bytes from 10.0.0.8: icmp_seq=72 ttl=64 time=451 ms
64 bytes from 10.0.0.8: icmp_seq=73 ttl=64 time=174 ms
64 bytes from 10.0.0.8: icmp_seq=74 ttl=64 time=269 ms
64 bytes from 10.0.0.8: icmp_seq=75 ttl=64 time=197 ms
64 bytes from 10.0.0.8: icmp_seq=76 ttl=64 time=252 ms
64 bytes from 10.0.0.8: icmp_seq=77 ttl=64 time=392 ms
64 bytes from 10.0.0.8: icmp_seq=78 ttl=64 time=186 ms
64 bytes from 10.0.0.8: icmp_seq=79 ttl=64 time=318 ms
64 bytes from 10.0.0.8: icmp_seq=80 ttl=64 time=312 ms
64 bytes from 10.0.0.8: icmp_seq=81 ttl=64 time=267 ms
64 bytes from 10.0.0.8: icmp_seq=82 ttl=64 time=469 ms
64 bytes from 10.0.0.8: icmp_seq=83 ttl=64 time=243 ms
64 bytes from 10.0.0.8: icmp_seq=84 ttl=64 time=224 ms
64 bytes from 10.0.0.8: icmp_seq=85 ttl=64 time=306 ms
64 bytes from 10.0.0.8: icmp_seq=86 ttl=64 time=258 ms
64 bytes from 10.0.0.8: icmp_seq=87 ttl=64 time=453 ms
64 bytes from 10.0.0.8: icmp_seq=88 ttl=64 time=249 ms
64 bytes from 10.0.0.8: icmp_seq=89 ttl=64 time=447 ms
64 bytes from 10.0.0.8: icmp_seq=90 ttl=64 time=280 ms
64 bytes from 10.0.0.8: icmp_seq=91 ttl=64 time=251 ms
64 bytes from 10.0.0.8: icmp_seq=92 ttl=64 time=388 ms
64 bytes from 10.0.0.8: icmp_seq=93 ttl=64 time=258 ms
64 bytes from 10.0.0.8: icmp_seq=94 ttl=64 time=192 ms
64 bytes from 10.0.0.8: icmp_seq=95 ttl=64 time=169 ms
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=223 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=410 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=289 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=257 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=268 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99251ms
rtt min/avg/max/mdev = 151.376/291.558/662.777/102.563 ms
```

- a. How long did it take (on average) to ping for each case?

Ans –

For h1 ping h2, 100 times, it takes on average of 77.227 ms for each ping.

For h1 ping h8, 100 times, it takes on average of 291.558 ms for each ping.

Student Name – Avani Sanjay Vaidya

Student Id - 007700005517

- b. What is the minimum and maximum ping you have observed?

Ans –

For h1 ping h2, 100 times, the min ping takes 26.892 ms and max ping takes 799.114 ms.

For h1 ping h8, 100 times, the min ping takes 151.376 ms and max ping takes 662.777 ms.

- c. Any difference from Task 2 and why do you think there is a change if there is?

Ans - Comparing task 2 and task 3, pinging from h1 to h2 is faster in task 3. When pinging from h1 to h8, task 3 is significantly faster because it involves fewer switches. Task 3 is quicker because it only floods a few initial packets. Once the destination MAC address is learned, switches forward packets directly to the mapped port, reducing congestion. So, subsequent pings are faster due to less network congestion.

3. Q.3 Run “iperf h1 h2” and “iperf h1 h8”.

- a. What is the throughput for each case?

Ans - Following screenshot shows the output of “iperf h1 h2” and “iperf h1 h8” and their corresponding throughputs.

```
[mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['498 Kbits/sec', '546 Kbits/sec']
[mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['132 Kbits/sec', '124 Kbits/sec']
```

- b. What is the difference from Task 2 and why do you think there is a change if there is?

Ans - Task 3 has better throughput than Task 2 in both cases. This happens because Task 3 has less congestion since it stops flooding packets once it learns all the ports through the mac_to_port map. This means switches aren't overloaded. This improvement comes from pre-planned and dynamically learned routes when the controller changes. However, for h1 and h8, the improvement is small because there are more hops and sometimes packets are dropped.