

Homework 3

COEN 241 : Cloud Computing

Name: Saurabh Thalkari
SCU ID: W1648455

[Github](#)

Task 1 : Defining custom topologies

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

a) nodes :

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

b) net :

```
mininet> net
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” ?

```
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::f04f:5ff:febe:b7c4 prefixlen 64 scopeid 0x20<link>
    ether f2:4f:05:be:b7:c4 txqueuelen 1000 (Ethernet)
    RX packets 58 bytes 4452 (4.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 796 (796.0 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 : Analyze the “of_tutorial” controller

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:

Initiating the POX Listener is done through the execution of the command:

‘./pox.py log.level –DEBUG misc.of_tutorial’

Executing this command activates the ‘**start switch**’, which triggers the execution of the **_handle_PacketIn()** function within the switch. This function, in turn, invokes the **act_like_hub()** method. The **act_like_hub()** method orchestrates a scenario where packets are dispatched to every port except the one they arrived on, emulating the behavior of a hub. Following this, the **resend_packet()** function is employed to append a packet to the message data and to undertake an action on it. The switch receives instructions to forward the packet to a designated port through the message.

The sequence of calls within the controller is depicted as follows:

start switch -> _handle_PacketIn() -> act_like_hub() -> resend_packet() -> send(message)

2.
h1 ping -c100 h2

```
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=2.42 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=1.20 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=1.25 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.971 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=1.14 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.937 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=1.20 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=1.98 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=1.26 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=1.41 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=1.14 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=1.31 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=1.21 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=1.22 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=1.15 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=1.43 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=1.37 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=1.37 ms
```

```
64 bytes from 10.0.0.2: icmp_seq=54 ttl=64 time=1.23 ms
64 bytes from 10.0.0.2: icmp_seq=55 ttl=64 time=1.23 ms
64 bytes from 10.0.0.2: icmp_seq=56 ttl=64 time=1.39 ms
64 bytes from 10.0.0.2: icmp_seq=57 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=58 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=59 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=60 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=61 ttl=64 time=1.29 ms
64 bytes from 10.0.0.2: icmp_seq=62 ttl=64 time=1.37 ms
64 bytes from 10.0.0.2: icmp_seq=63 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=1.14 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=1.38 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=1.19 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=1.16 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=1.39 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=1.35 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=1.07 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=1.39 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=1.34 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=1.15 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=1.19 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=1.28 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=1.50 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=1.20 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=1.59 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=1.35 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=1.16 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=1.25 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=1.23 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=1.17 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=1.40 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=1.14 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99115ms
rtt min/avg/max/mdev = 0.937/1.200/2.423/0.200 ms
```


h1 ping -c100 h8

```
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=8.91 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=4.05 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=5.32 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=4.14 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=4.00 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=3.99 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=3.79 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=3.89 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=5.42 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=4.05 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=3.82 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=4.14 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=4.03 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=3.45 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=4.86 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=3.76 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=4.21 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=3.72 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=3.82 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=3.73 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=5.17 ms
64 bytes from 10.0.0.8: icmp_seq=22 ttl=64 time=3.96 ms
64 bytes from 10.0.0.8: icmp_seq=23 ttl=64 time=4.00 ms
64 bytes from 10.0.0.8: icmp_seq=24 ttl=64 time=3.87 ms
64 bytes from 10.0.0.8: icmp_seq=25 ttl=64 time=3.76 ms
64 bytes from 10.0.0.8: icmp_seq=26 ttl=64 time=3.80 ms
64 bytes from 10.0.0.8: icmp_seq=27 ttl=64 time=5.47 ms
64 bytes from 10.0.0.8: icmp_seq=28 ttl=64 time=3.89 ms
64 bytes from 10.0.0.8: icmp_seq=29 ttl=64 time=4.15 ms
64 bytes from 10.0.0.8: icmp_seq=30 ttl=64 time=4.04 ms
64 bytes from 10.0.0.8: icmp_seq=31 ttl=64 time=3.91 ms
64 bytes from 10.0.0.8: icmp_seq=32 ttl=64 time=4.06 ms
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=5.08 ms
64 bytes from 10.0.0.8: icmp_seq=34 ttl=64 time=3.93 ms
64 bytes from 10.0.0.8: icmp_seq=35 ttl=64 time=3.64 ms
64 bytes from 10.0.0.8: icmp_seq=36 ttl=64 time=3.39 ms
64 bytes from 10.0.0.8: icmp_seq=37 ttl=64 time=3.85 ms
64 bytes from 10.0.0.8: icmp_seq=38 ttl=64 time=3.84 ms
64 bytes from 10.0.0.8: icmp_seq=39 ttl=64 time=5.43 ms
64 bytes from 10.0.0.8: icmp_seq=40 ttl=64 time=3.90 ms
64 bytes from 10.0.0.8: icmp_seq=41 ttl=64 time=3.90 ms
64 bytes from 10.0.0.8: icmp_seq=42 ttl=64 time=4.01 ms
64 bytes from 10.0.0.8: icmp_seq=43 ttl=64 time=4.03 ms
64 bytes from 10.0.0.8: icmp_seq=44 ttl=64 time=4.08 ms
64 bytes from 10.0.0.8: icmp_seq=45 ttl=64 time=4.99 ms
```

```

64 bytes from 10.0.0.8: icmp_seq=54 ttl=64 time=4.02 ms
64 bytes from 10.0.0.8: icmp_seq=55 ttl=64 time=3.54 ms
64 bytes from 10.0.0.8: icmp_seq=56 ttl=64 time=3.64 ms
64 bytes from 10.0.0.8: icmp_seq=57 ttl=64 time=4.17 ms
64 bytes from 10.0.0.8: icmp_seq=58 ttl=64 time=4.11 ms
64 bytes from 10.0.0.8: icmp_seq=59 ttl=64 time=5.67 ms
64 bytes from 10.0.0.8: icmp_seq=60 ttl=64 time=3.99 ms
64 bytes from 10.0.0.8: icmp_seq=61 ttl=64 time=4.10 ms
64 bytes from 10.0.0.8: icmp_seq=62 ttl=64 time=4.24 ms
64 bytes from 10.0.0.8: icmp_seq=63 ttl=64 time=4.28 ms
64 bytes from 10.0.0.8: icmp_seq=64 ttl=64 time=4.18 ms
64 bytes from 10.0.0.8: icmp_seq=65 ttl=64 time=5.51 ms
64 bytes from 10.0.0.8: icmp_seq=66 ttl=64 time=4.22 ms
64 bytes from 10.0.0.8: icmp_seq=67 ttl=64 time=4.13 ms
64 bytes from 10.0.0.8: icmp_seq=68 ttl=64 time=3.92 ms
64 bytes from 10.0.0.8: icmp_seq=69 ttl=64 time=4.01 ms
64 bytes from 10.0.0.8: icmp_seq=70 ttl=64 time=3.61 ms
64 bytes from 10.0.0.8: icmp_seq=71 ttl=64 time=5.14 ms
64 bytes from 10.0.0.8: icmp_seq=72 ttl=64 time=3.69 ms
64 bytes from 10.0.0.8: icmp_seq=73 ttl=64 time=3.70 ms
64 bytes from 10.0.0.8: icmp_seq=74 ttl=64 time=3.73 ms
64 bytes from 10.0.0.8: icmp_seq=75 ttl=64 time=3.57 ms
64 bytes from 10.0.0.8: icmp_seq=76 ttl=64 time=3.98 ms
64 bytes from 10.0.0.8: icmp_seq=77 ttl=64 time=5.08 ms
64 bytes from 10.0.0.8: icmp_seq=78 ttl=64 time=4.05 ms
64 bytes from 10.0.0.8: icmp_seq=79 ttl=64 time=4.18 ms
64 bytes from 10.0.0.8: icmp_seq=80 ttl=64 time=4.05 ms
64 bytes from 10.0.0.8: icmp_seq=81 ttl=64 time=4.10 ms
64 bytes from 10.0.0.8: icmp_seq=82 ttl=64 time=4.07 ms
64 bytes from 10.0.0.8: icmp_seq=83 ttl=64 time=5.21 ms
64 bytes from 10.0.0.8: icmp_seq=84 ttl=64 time=4.20 ms
64 bytes from 10.0.0.8: icmp_seq=85 ttl=64 time=3.80 ms
64 bytes from 10.0.0.8: icmp_seq=86 ttl=64 time=4.00 ms
64 bytes from 10.0.0.8: icmp_seq=87 ttl=64 time=4.06 ms
64 bytes from 10.0.0.8: icmp_seq=88 ttl=64 time=4.12 ms
64 bytes from 10.0.0.8: icmp_seq=89 ttl=64 time=5.53 ms
64 bytes from 10.0.0.8: icmp_seq=90 ttl=64 time=3.58 ms
64 bytes from 10.0.0.8: icmp_seq=91 ttl=64 time=3.86 ms
64 bytes from 10.0.0.8: icmp_seq=92 ttl=64 time=3.63 ms
64 bytes from 10.0.0.8: icmp_seq=93 ttl=64 time=3.79 ms
64 bytes from 10.0.0.8: icmp_seq=94 ttl=64 time=3.83 ms
64 bytes from 10.0.0.8: icmp_seq=95 ttl=64 time=5.26 ms
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=3.85 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=4.95 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=3.58 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=4.01 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=3.87 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99144ms
rtt min/avg/max/mdev = 3.395/4.241/8.911/0.834 ms

```

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

	Average Ping
h1 ping -c100 h2	1.991 ms
h1 ping -c100 h8	4.329 ms

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

	Minimum Ping	Maximum Ping
h1 ping -c100 h2	0.971 ms	2.393 ms
h1 ping -c100 h8	3.289 ms	8.889 ms

c) What is the difference, and why?

Ans:

The latency observed when pinging from h1 to h8 is greater, at 4.241 ms, in comparison to the ping from h1 to h2, which is 1.200 ms. This is due to the path taken by the data from h1 to reach h8, which involves traversing through five switches—namely s3, s2, s1, s5, and s7. On the other hand, the connection between h1 and h2 is facilitated by just a single switch. Therefore, the increased number of switches in the path from h1 to h8 results in a higher ping time.

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

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['11.4 Mbits/sec', '11.7 Mbits/sec']
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['2.90 Mbits/sec', '3.31 Mbits/sec']
```

a) What is “iperf” used for?

Ans:

iperf, a free open-source tool, assists administrators in evaluating network performance and line quality by determining bandwidth. It calculates the data transfer rate between any two nodes on a network connection.

b) What is the throughput for each case?

i) iperf: testing TCP bandwidth between h1 and h2
Results: ['11.4 Mbits/sec', '11.7 Mbits/sec']

ii) iperf: testing TCP bandwidth between h1 and h8
Results: ['2.90 Mbits/sec', '3.31 Mbits/sec']

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

Ans:

The throughput from h1 to h2 is greater than from h1 to h8, mirroring the reasons behind the slower ping time from h1 to h8. With fewer obstructions, reduced latency, and a shorter path between h1 and h2, data transmission occurs more swiftly. Consequently, the data transfer rate between h1 and h2 is superior to that between h1 and h8.

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 integrating loggers, such as `log.info(“Switch observing traffic : percent s” (self.connection))`, into line 107 of the “of_tutorial” file in the controller, we gain insights into traffic observation capabilities. This reveals that switches have the ability to track and scrutinize traffic flows, especially when they encounter a high volume of packets. The event handler function `_handle_PacketIn()` gets triggered in response to the reception of any packet.

Task 3 : MAC Learning Controller

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 facilitates the identification of MAC addresses and their corresponding locations. This function allows for efficient mapping of a MAC address to its specific port by the controller, using the stored metadata, whenever a message needs to be directed to that MAC address. This pre-existing knowledge about the port destination significantly accelerates the delivery process for messages aimed at the same address.

If the destination MAC address is not recognized, the `act_like_switch()` function defaults to distributing the packet across all possible addresses. The reduced likelihood of network flooding, courtesy of the MAC Learning Controller, contributes to enhanced pinging times and improved throughput.

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

h1 ping -c100 h2

```
*** Starting CLI:
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=3.00 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.13 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=1.19 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1.29 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=1.21 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=1.38 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.891 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=1.28 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=1.07 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=1.30 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=0.983 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=0.998 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=1.30 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=29 ttl=64 time=0.988 ms
64 bytes from 10.0.0.2: icmp_seq=30 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=31 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=32 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=33 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=34 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=35 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=36 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=37 ttl=64 time=1.36 ms
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=39 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=40 ttl=64 time=1.07 ms
64 bytes from 10.0.0.2: icmp_seq=41 ttl=64 time=1.18 ms
64 bytes from 10.0.0.2: icmp_seq=42 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=43 ttl=64 time=1.37 ms
64 bytes from 10.0.0.2: icmp_seq=44 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=45 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=46 ttl=64 time=1.12 ms
64 bytes from 10.0.0.2: icmp_seq=47 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=48 ttl=64 time=1.15 ms
64 bytes from 10.0.0.2: icmp_seq=49 ttl=64 time=1.33 ms
```

```
64 bytes from 10.0.0.2: icmp_seq=64 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=65 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=66 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=67 ttl=64 time=1.35 ms
64 bytes from 10.0.0.2: icmp_seq=68 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=69 ttl=64 time=1.17 ms
64 bytes from 10.0.0.2: icmp_seq=70 ttl=64 time=1.10 ms
64 bytes from 10.0.0.2: icmp_seq=71 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=72 ttl=64 time=1.18 ms
64 bytes from 10.0.0.2: icmp_seq=73 ttl=64 time=1.29 ms
64 bytes from 10.0.0.2: icmp_seq=74 ttl=64 time=0.971 ms
64 bytes from 10.0.0.2: icmp_seq=75 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=76 ttl=64 time=1.14 ms
64 bytes from 10.0.0.2: icmp_seq=77 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=78 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=79 ttl=64 time=1.22 ms
64 bytes from 10.0.0.2: icmp_seq=80 ttl=64 time=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=81 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=82 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=83 ttl=64 time=1.06 ms
64 bytes from 10.0.0.2: icmp_seq=84 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=85 ttl=64 time=1.29 ms
64 bytes from 10.0.0.2: icmp_seq=86 ttl=64 time=1.05 ms
64 bytes from 10.0.0.2: icmp_seq=87 ttl=64 time=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=88 ttl=64 time=1.09 ms
64 bytes from 10.0.0.2: icmp_seq=89 ttl=64 time=0.971 ms
64 bytes from 10.0.0.2: icmp_seq=90 ttl=64 time=1.07 ms
64 bytes from 10.0.0.2: icmp_seq=91 ttl=64 time=1.36 ms
64 bytes from 10.0.0.2: icmp_seq=92 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=93 ttl=64 time=1.03 ms
64 bytes from 10.0.0.2: icmp_seq=94 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=1.02 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=1.04 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=1.33 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=1.07 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=1.11 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=1.10 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99158ms
rtt min/avg/max/mdev = 0.890/1.147/3.007/0.238 ms
```

h1 ping -c100 h8

```
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=11.5 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=3.82 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=3.78 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=4.09 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=4.13 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=3.86 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=5.13 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=4.03 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=4.02 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=4.23 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=4.01 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=4.22 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=5.80 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=3.99 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=4.32 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=3.55 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=4.23 ms
64 bytes from 10.0.0.8: icmp_seq=18 ttl=64 time=3.91 ms
64 bytes from 10.0.0.8: icmp_seq=19 ttl=64 time=5.46 ms
64 bytes from 10.0.0.8: icmp_seq=20 ttl=64 time=4.11 ms
64 bytes from 10.0.0.8: icmp_seq=21 ttl=64 time=3.79 ms
64 bytes from 10.0.0.8: icmp_seq=22 ttl=64 time=3.84 ms
64 bytes from 10.0.0.8: icmp_seq=23 ttl=64 time=3.90 ms
64 bytes from 10.0.0.8: icmp_seq=24 ttl=64 time=3.75 ms
64 bytes from 10.0.0.8: icmp_seq=25 ttl=64 time=5.02 ms
64 bytes from 10.0.0.8: icmp_seq=26 ttl=64 time=3.59 ms
64 bytes from 10.0.0.8: icmp_seq=27 ttl=64 time=3.70 ms
64 bytes from 10.0.0.8: icmp_seq=28 ttl=64 time=3.86 ms
64 bytes from 10.0.0.8: icmp_seq=29 ttl=64 time=3.73 ms
64 bytes from 10.0.0.8: icmp_seq=30 ttl=64 time=4.26 ms
64 bytes from 10.0.0.8: icmp_seq=31 ttl=64 time=4.90 ms
64 bytes from 10.0.0.8: icmp_seq=32 ttl=64 time=3.95 ms
64 bytes from 10.0.0.8: icmp_seq=33 ttl=64 time=3.99 ms
64 bytes from 10.0.0.8: icmp_seq=34 ttl=64 time=4.20 ms
64 bytes from 10.0.0.8: icmp_seq=35 ttl=64 time=4.09 ms
64 bytes from 10.0.0.8: icmp_seq=36 ttl=64 time=4.17 ms
64 bytes from 10.0.0.8: icmp_seq=37 ttl=64 time=5.11 ms
64 bytes from 10.0.0.8: icmp_seq=38 ttl=64 time=3.39 ms
64 bytes from 10.0.0.8: icmp_seq=39 ttl=64 time=3.97 ms
64 bytes from 10.0.0.8: icmp_seq=40 ttl=64 time=3.57 ms
64 bytes from 10.0.0.8: icmp_seq=41 ttl=64 time=3.74 ms
64 bytes from 10.0.0.8: icmp_seq=42 ttl=64 time=3.68 ms
64 bytes from 10.0.0.8: icmp_seq=43 ttl=64 time=4.90 ms
64 bytes from 10.0.0.8: icmp_seq=44 ttl=64 time=3.69 ms
64 bytes from 10.0.0.8: icmp_seq=45 ttl=64 time=3.86 ms
```



```

64 bytes from 10.0.0.8: icmp_seq=54 ttl=64 time=4.20 ms
64 bytes from 10.0.0.8: icmp_seq=55 ttl=64 time=5.49 ms
64 bytes from 10.0.0.8: icmp_seq=56 ttl=64 time=4.20 ms
64 bytes from 10.0.0.8: icmp_seq=57 ttl=64 time=4.26 ms
64 bytes from 10.0.0.8: icmp_seq=58 ttl=64 time=4.21 ms
64 bytes from 10.0.0.8: icmp_seq=59 ttl=64 time=4.51 ms
64 bytes from 10.0.0.8: icmp_seq=60 ttl=64 time=4.06 ms
64 bytes from 10.0.0.8: icmp_seq=61 ttl=64 time=5.33 ms
64 bytes from 10.0.0.8: icmp_seq=62 ttl=64 time=4.07 ms
64 bytes from 10.0.0.8: icmp_seq=63 ttl=64 time=4.33 ms
64 bytes from 10.0.0.8: icmp_seq=64 ttl=64 time=4.33 ms
64 bytes from 10.0.0.8: icmp_seq=65 ttl=64 time=4.20 ms
64 bytes from 10.0.0.8: icmp_seq=66 ttl=64 time=4.24 ms
64 bytes from 10.0.0.8: icmp_seq=67 ttl=64 time=5.20 ms
64 bytes from 10.0.0.8: icmp_seq=68 ttl=64 time=4.18 ms
64 bytes from 10.0.0.8: icmp_seq=69 ttl=64 time=3.98 ms
64 bytes from 10.0.0.8: icmp_seq=70 ttl=64 time=3.96 ms
64 bytes from 10.0.0.8: icmp_seq=71 ttl=64 time=3.69 ms
64 bytes from 10.0.0.8: icmp_seq=72 ttl=64 time=3.99 ms
64 bytes from 10.0.0.8: icmp_seq=73 ttl=64 time=5.32 ms
64 bytes from 10.0.0.8: icmp_seq=74 ttl=64 time=3.73 ms
64 bytes from 10.0.0.8: icmp_seq=75 ttl=64 time=3.96 ms
64 bytes from 10.0.0.8: icmp_seq=76 ttl=64 time=4.00 ms
64 bytes from 10.0.0.8: icmp_seq=77 ttl=64 time=4.09 ms
64 bytes from 10.0.0.8: icmp_seq=78 ttl=64 time=3.80 ms
64 bytes from 10.0.0.8: icmp_seq=79 ttl=64 time=5.55 ms
64 bytes from 10.0.0.8: icmp_seq=80 ttl=64 time=4.36 ms
64 bytes from 10.0.0.8: icmp_seq=81 ttl=64 time=3.90 ms
64 bytes from 10.0.0.8: icmp_seq=82 ttl=64 time=3.89 ms
64 bytes from 10.0.0.8: icmp_seq=83 ttl=64 time=3.64 ms
64 bytes from 10.0.0.8: icmp_seq=84 ttl=64 time=4.04 ms
64 bytes from 10.0.0.8: icmp_seq=85 ttl=64 time=5.14 ms
64 bytes from 10.0.0.8: icmp_seq=86 ttl=64 time=4.43 ms
64 bytes from 10.0.0.8: icmp_seq=87 ttl=64 time=4.84 ms
64 bytes from 10.0.0.8: icmp_seq=88 ttl=64 time=4.09 ms
64 bytes from 10.0.0.8: icmp_seq=89 ttl=64 time=4.36 ms
64 bytes from 10.0.0.8: icmp_seq=90 ttl=64 time=4.55 ms
64 bytes from 10.0.0.8: icmp_seq=91 ttl=64 time=5.70 ms
64 bytes from 10.0.0.8: icmp_seq=92 ttl=64 time=4.44 ms
64 bytes from 10.0.0.8: icmp_seq=93 ttl=64 time=4.09 ms
64 bytes from 10.0.0.8: icmp_seq=94 ttl=64 time=4.16 ms
64 bytes from 10.0.0.8: icmp_seq=95 ttl=64 time=4.11 ms
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=3.92 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=5.32 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=4.17 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=3.79 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=3.97 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99138ms
rtt min/avg/max/mdev = 3.393/4.293/11.505/0.903 ms

```

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

	Average Ping
h1 ping -c100 h2	1.257 ms
h1 ping -c100 h8	4.343 ms

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

	Minimum Ping	Maximum Ping
h1 ping -c100 h2	0.890 ms	3.007 ms
h1 ping -c100 h8	3.393 ms	11.505 ms

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

Ans:

The duration required to execute operations in task 3 is shorter than in task 2, albeit by a slim margin. This variation in ping times arises in task 3 due to the approach of retransmitting packets only to those destinations already listed in the "mac to port" mapping, following their initial entry. Future packet transmissions to the same address are significantly expedited, as both the port and address are pre-identified, coupled with reduced network traffic congestion.

3. Run "iperf h1 h2" and "iperf h1 h8".

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['18.7 Mbits/sec', '19.2 Mbits/sec']
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['4.02 Mbits/sec', '4.75 Mbits/sec']
```

a) What is the throughput for each case?

i) iperf: testing TCP bandwidth between h1 and h2

Results: ['18.7 Mbits/sec', '19.2 Mbits/sec']

ii) iperf: testing TCP bandwidth between h1 and h8

Results: ['4.02 Mbits/sec', '4.75 Mbits/sec']

b) What is the difference and explain the reasons for the difference.

Ans:

Task 3 exhibits a higher throughput compared to task 2 in both scenarios, thanks to the efficiency of the mac to port mapping, which reduces network congestion. Since the system avoids flooding by utilizing pre-known address-port pairings established during initial packet transmission, it prevents the switches from being inundated with excessive requests. Specifically, the improvement in throughput for task 3 is more pronounced in the h1 -> h2 scenario due to decreased congestion, while the enhancement is less marked from h1 -> h8, attributed to packet losses and greater transmission distances.