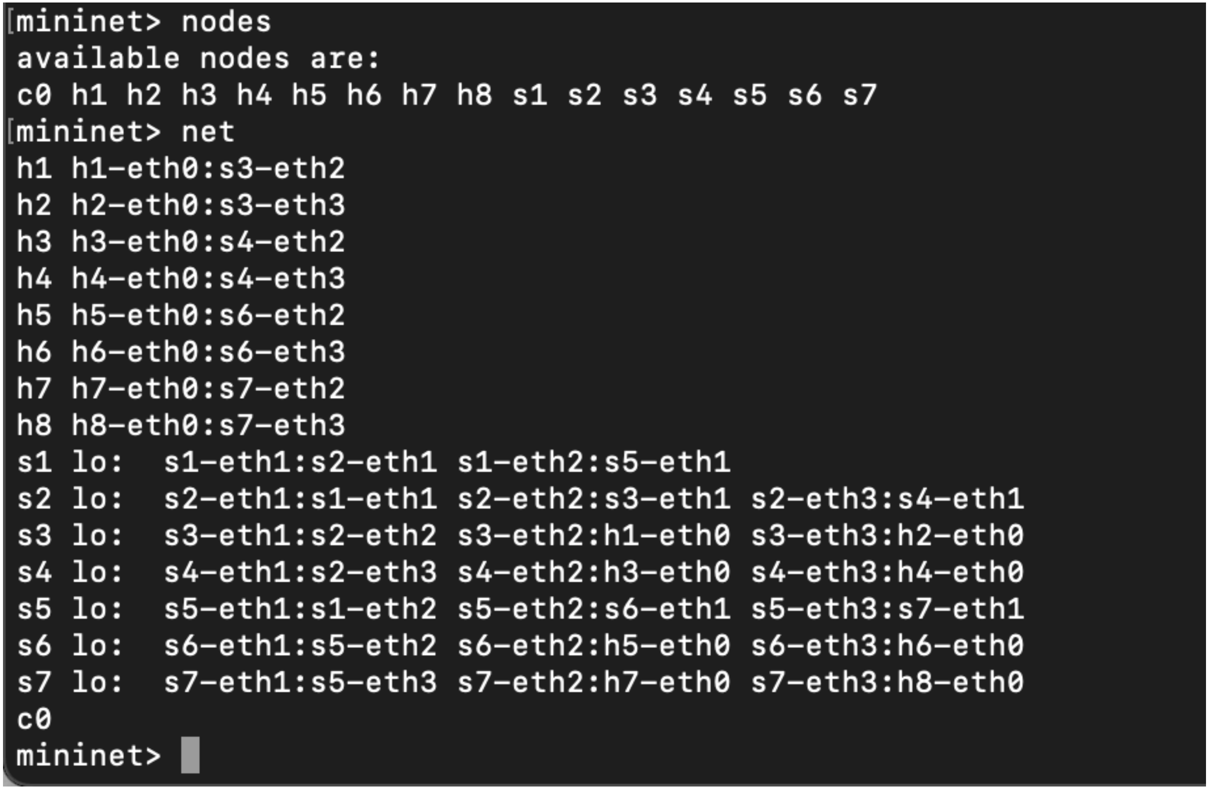
COEN 241: Homework 3

Name: Juhi Checker
SCU ID: 1605378

Task 1

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

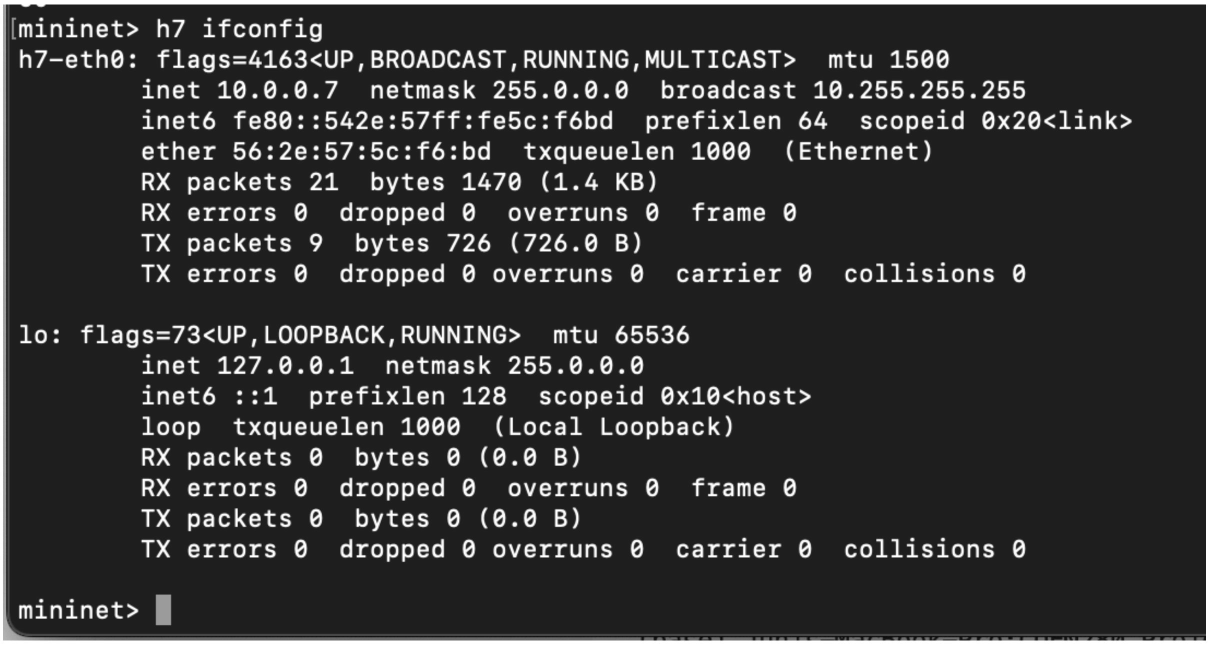
Ans. Output of nodes and net



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

2. What is the output of “h7 ifconfig”

Ans.



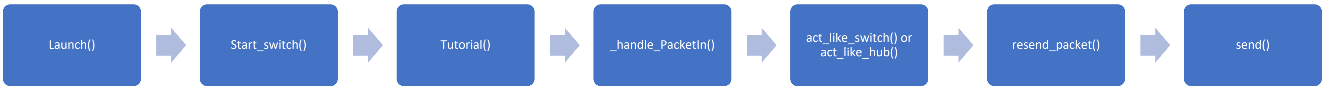
```
[mininet> h7 ifconfig
h7-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.7  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::542e:57ff:fe5c:f6bd  prefixlen 64  scopeid 0x20<link>
        ether 56:2e:57:5c:f6:bd  txqueuelen 1000  (Ethernet)
        RX packets 21  bytes 1470 (1.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 726 (726.0 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

mininet> 
```

Task 2

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. When the of_tutorial.py file runs the following steps happen:

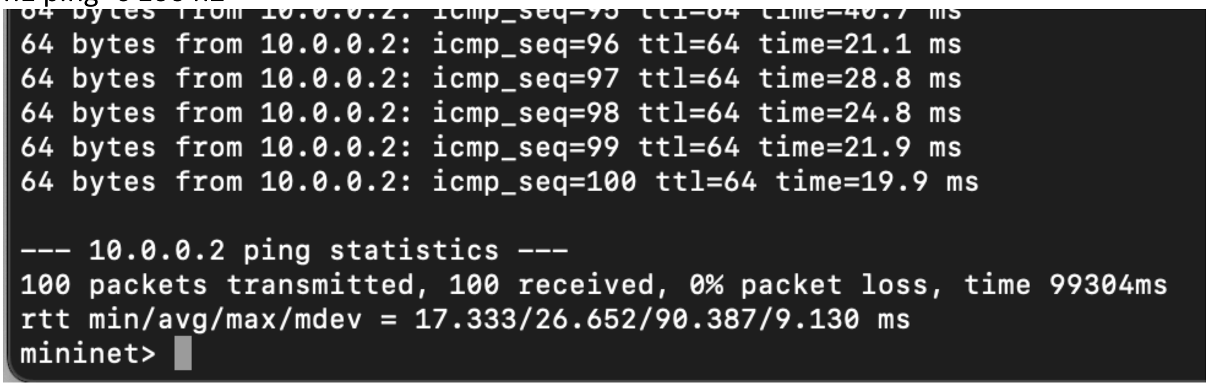
a. Launch() function is called and start_switch() component executes.
b. Start_switch() component calls Tutorial class and connection sets ups.
c. By doing addListernerByName() calls Tutorials' _handle_PacketIn() function
d. The _handle_PacketIn() function calls the act_like_hub or act_like_switch function is executed
e. The above function after performing the steps inside the function calls resend_packet function
f. The resend_packet function forwards/sends the message to the mentioned port



2. Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 p2).
   a. How long does it take (on average) to ping for each case?
   b. What is the minimum and maximum ping you have observed?
   c. What is the difference, and why?

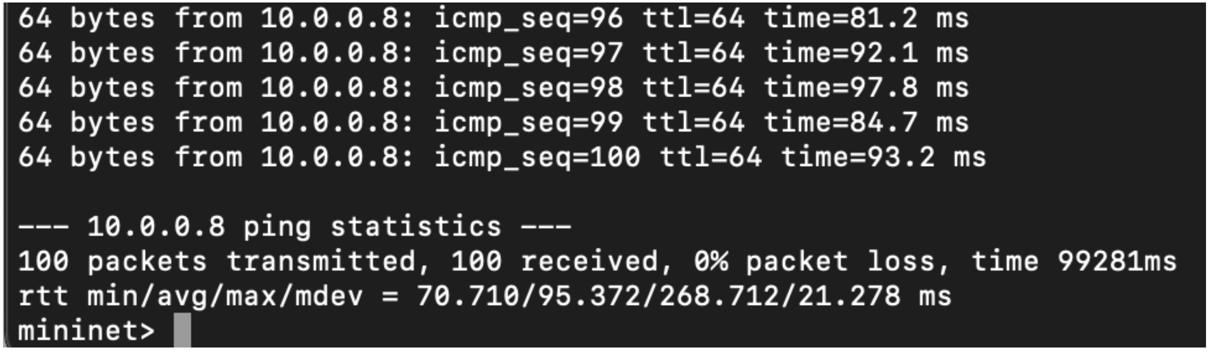
Ans.

h1 ping -c 100 h2



```
64 bytes from 10.0.0.2: icmp_seq=95 ttl=64 time=40.7 ms
64 bytes from 10.0.0.2: icmp_seq=96 ttl=64 time=21.1 ms
64 bytes from 10.0.0.2: icmp_seq=97 ttl=64 time=28.8 ms
64 bytes from 10.0.0.2: icmp_seq=98 ttl=64 time=24.8 ms
64 bytes from 10.0.0.2: icmp_seq=99 ttl=64 time=21.9 ms
64 bytes from 10.0.0.2: icmp_seq=100 ttl=64 time=19.9 ms

--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99304ms
rtt min/avg/max/mdev = 17.333/26.652/90.387/9.130 ms
mininet>
```

h1 ping -c 100 h8



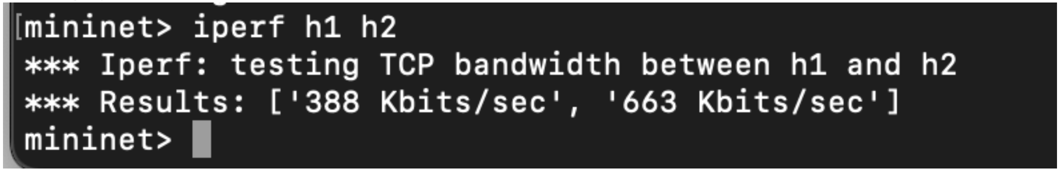
```
64 bytes from 10.0.0.8: icmp_seq=96 ttl=64 time=81.2 ms
64 bytes from 10.0.0.8: icmp_seq=97 ttl=64 time=92.1 ms
64 bytes from 10.0.0.8: icmp_seq=98 ttl=64 time=97.8 ms
64 bytes from 10.0.0.8: icmp_seq=99 ttl=64 time=84.7 ms
64 bytes from 10.0.0.8: icmp_seq=100 ttl=64 time=93.2 ms

--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99281ms
rtt min/avg/max/mdev = 70.710/95.372/268.712/21.278 ms
mininet>
```

a. Average for h1 ping h2 = 26.652 ms
   Average for h1 ping h8 = 95.372 ms
b. Minimum value for h1 ping h2 = 17.333 ms
   Minimum value for h1 ping h8 = 70.710 ms
   Maximum value for h1 ping h2 = 90.387 ms
   Maximum value for h1 ping h8 = 268.712 ms
c. It can be observed that average, minimum and maximum value for h1 ping h2 is lesser than h1 ping h8. This can be because for h1 has to only traverse through one switch which is s3 to reach the destination. Whereas, in second case where h1 pings h8, h1 has to traverse through s3, s2, s1, s5 and s7 to reach to h8.

3. Run “iperf h1 h2” and “iperf h1 h8”
   a. What is “iperf” used for?
   b. What is the throughput for each case?
   c. What is the difference, and explain the reasons for the difference.

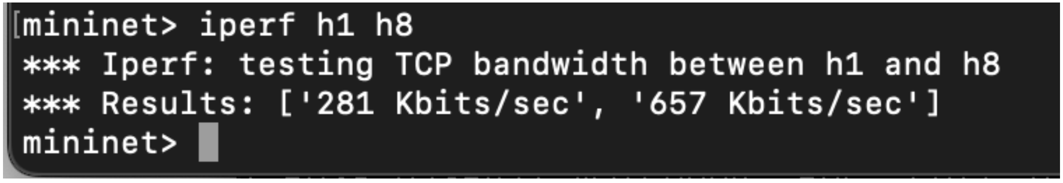
Ans.

Running iperf h1 h2



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['388 Kbits/sec', '663 Kbits/sec']
mininet>
```

Running iperf h1 h8



```
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['281 Kbits/sec', '657 Kbits/sec']
mininet>
```

a. Iperf command is used for testing the TCP bandwidth between two hosts for network performance and quality of network.
b. Throughput for iperf h1 h2 = 388 Kbits/sec and 663 Kbits/sec
   Throughput for iperf h1 h8 = 281 Kbits/sec and 657 Kbits/sec
c. It is observed that throughput is higher in both case in iperf h1 h2 than iperf h1 h8. This can be because since h1 has to traverse through only one switch to get to h2 and so the data can be passed at a higher rate. Whereas, it was

seen in the previous task that from h1 to h8 we have to traverse through 5 switches which makes the data transmission a bit slower.

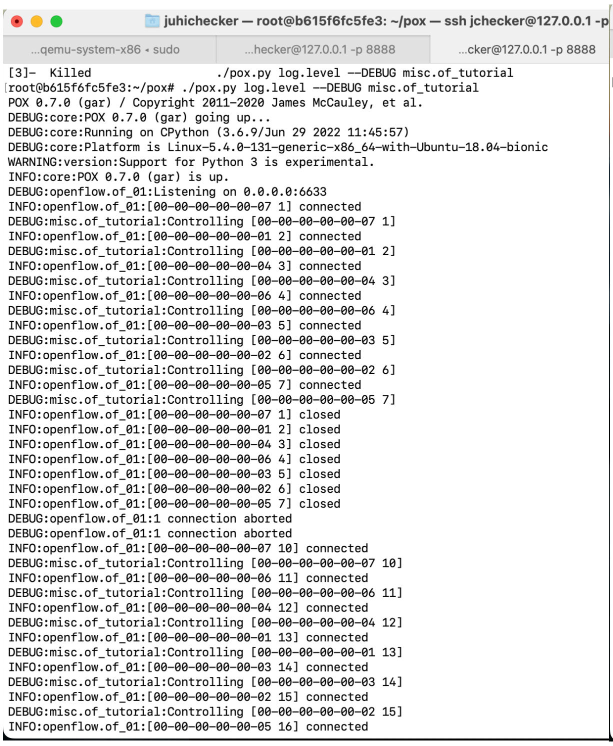
4. Which of the switches observe traffic? Please describe your way for observing such traffic on switches (e.g., adding some functions in the “of_tutorial” controller).

Ans. In order to observe the traffic on switches I add print statement in _handle_PacketIn() function of of_tutorial.py. On running the code again, it was observed that each switch sees the traffic following because every time someone wants to send a packet _handle_PacketIn function is called.

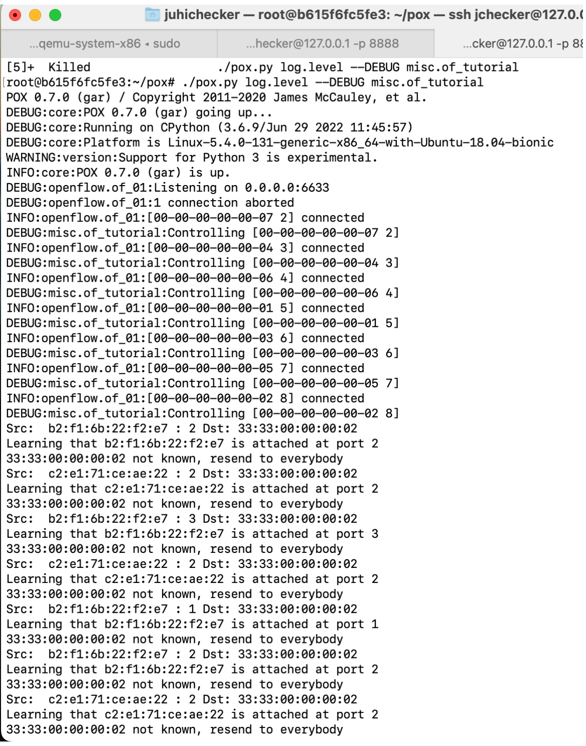
Task 3

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).

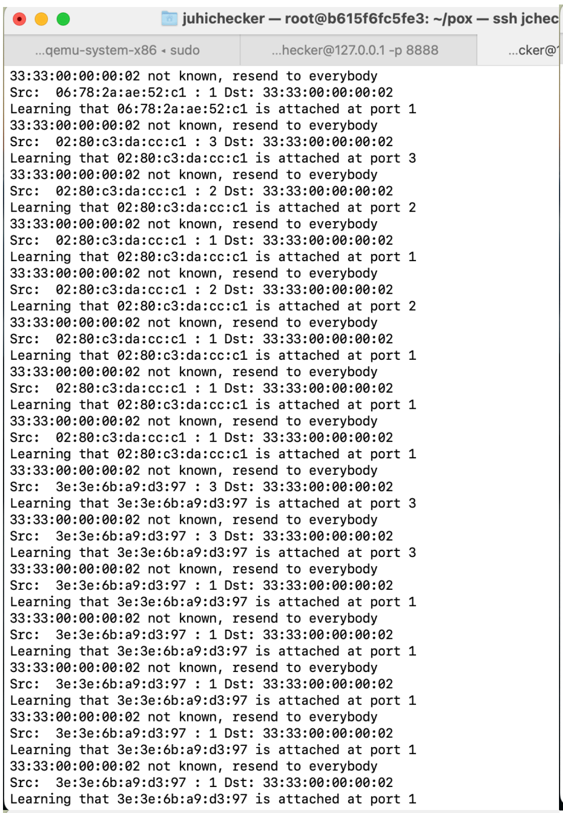
Ans. The new function act_like_switch when packets arrive the function maps the MAC addresses. Due to the mapping during the next ping the controller sends the packets to the know (mapped) addresses. Whereas, if the address is not mapped then it will send packets to all the destinations and learn it. Essentially, controller maps the MAC address to a port if MAC address is the one the sender wants to send the packet to. This process makes sending packets faster therefore increasing the throughput.


juhichecker — root@b615f6fc5fe3: ~/pox — ssh jchecker@127.0.0.1 -p
...qemu-system-x86 ◂ sudo
...hecker@127.0.0.1 -p 8888
...cker@127.0.0.1 -p 8888

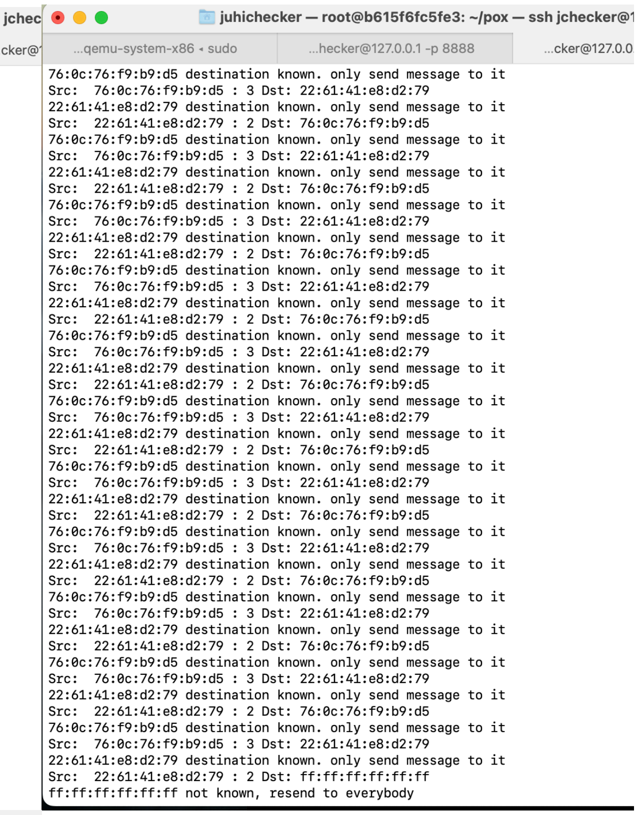
```
[3]-  Killed                  ./pox.py log.level --DEBUG misc.of_tutorial
root@b615f6fc5fe3:~/pox# ./pox.py log.level --DEBUG misc.of_tutorial
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
DEBUG:core:POX 0.7.0 (gar) going up...
DEBUG:core:Running on CPython (3.6.9/Jun 29 2022 11:45:57)
DEBUG:core:Platform is Linux-5.4.0-131-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
INFO:openflow.of_01:[00-00-00-00-00-07 1] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-07 1]
INFO:openflow.of_01:[00-00-00-00-00-01 2] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 2]
INFO:openflow.of_01:[00-00-00-00-00-04 3] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-04 3]
INFO:openflow.of_01:[00-00-00-00-00-06 4] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-06 4]
INFO:openflow.of_01:[00-00-00-00-00-03 5] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-03 5]
INFO:openflow.of_01:[00-00-00-00-00-02 6] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-02 6]
INFO:openflow.of_01:[00-00-00-00-00-05 7] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-05 7]
INFO:openflow.of_01:[00-00-00-00-00-07 1] closed
INFO:openflow.of_01:[00-00-00-00-00-01 2] closed
INFO:openflow.of_01:[00-00-00-00-00-04 3] closed
INFO:openflow.of_01:[00-00-00-00-00-06 4] closed
INFO:openflow.of_01:[00-00-00-00-00-03 5] closed
INFO:openflow.of_01:[00-00-00-00-00-02 6] closed
INFO:openflow.of_01:[00-00-00-00-00-05 7] closed
DEBUG:openflow.of_01:1 connection aborted
DEBUG:openflow.of_01:1 connection aborted
INFO:openflow.of_01:[00-00-00-00-00-07 10] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-07 10]
INFO:openflow.of_01:[00-00-00-00-00-06 11] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-06 11]
INFO:openflow.of_01:[00-00-00-00-00-04 12] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-04 12]
INFO:openflow.of_01:[00-00-00-00-00-01 13] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 13]
INFO:openflow.of_01:[00-00-00-00-00-03 14] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-03 14]
INFO:openflow.of_01:[00-00-00-00-00-02 15] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-02 15]
INFO:openflow.of_01:[00-00-00-00-00-05 16] connected
```


juhichecker — root@b615f6fc5fe3: ~/pox — ssh jchecker@127.0.(
...qemu-system-x86 ◂ sudo
...hecker@127.0.0.1 -p 8888
...cker@127.0.0.1 -p 88

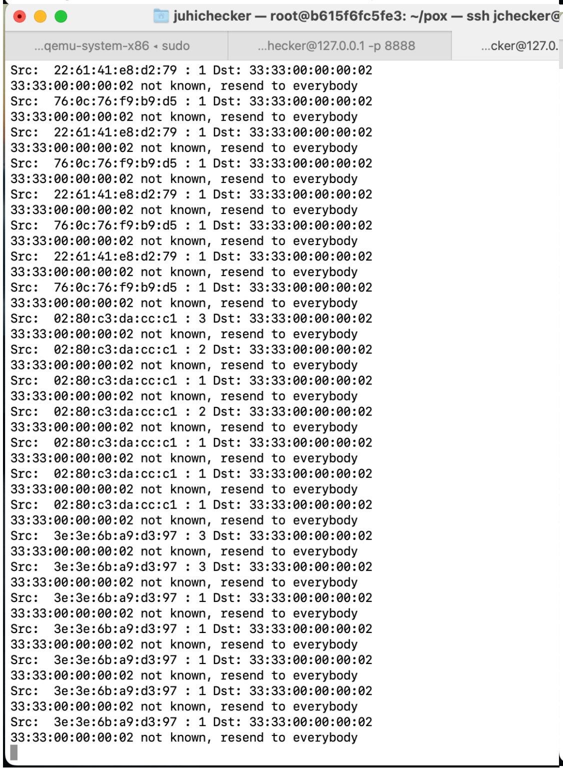
```
[5]+  Killed                  ./pox.py log.level --DEBUG misc.of_tutorial
root@b615f6fc5fe3:~/pox# ./pox.py log.level --DEBUG misc.of_tutorial
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
DEBUG:core:POX 0.7.0 (gar) going up...
DEBUG:core:Running on CPython (3.6.9/Jun 29 2022 11:45:57)
DEBUG:core:Platform is Linux-5.4.0-131-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
DEBUG:openflow.of_01:1 connection aborted
INFO:openflow.of_01:[00-00-00-00-00-07 2] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-07 2]
INFO:openflow.of_01:[00-00-00-00-00-04 3] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-04 3]
INFO:openflow.of_01:[00-00-00-00-00-06 4] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-06 4]
INFO:openflow.of_01:[00-00-00-00-00-01 5] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-01 5]
INFO:openflow.of_01:[00-00-00-00-00-03 6] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-03 6]
INFO:openflow.of_01:[00-00-00-00-00-05 7] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-05 7]
INFO:openflow.of_01:[00-00-00-00-00-02 8] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-00-02 8]
Src:  b2:f1:6b:22:f2:e7 : 2 Dst: 33:33:00:00:00:02
Learning that b2:f1:6b:22:f2:e7 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
Src:  c2:e1:71:ce:ae:22 : 2 Dst: 33:33:00:00:00:02
Learning that c2:e1:71:ce:ae:22 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
Src:  b2:f1:6b:22:f2:e7 : 3 Dst: 33:33:00:00:00:02
Learning that b2:f1:6b:22:f2:e7 is attached at port 3
33:33:00:00:00:02 not known, resend to everybody
Src:  c2:e1:71:ce:ae:22 : 2 Dst: 33:33:00:00:00:02
Learning that c2:e1:71:ce:ae:22 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
Src:  b2:f1:6b:22:f2:e7 : 1 Dst: 33:33:00:00:00:02
Learning that b2:f1:6b:22:f2:e7 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  b2:f1:6b:22:f2:e7 : 2 Dst: 33:33:00:00:00:02
Learning that b2:f1:6b:22:f2:e7 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
Src:  c2:e1:71:ce:ae:22 : 2 Dst: 33:33:00:00:00:02
Learning that c2:e1:71:ce:ae:22 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
```


juhichecker — root@b615f6fc5fe3: ~/pox — ssh jchec
...qemu-system-x86 ◂ sudo
...hecker@127.0.0.1 -p 8888
...cker@1

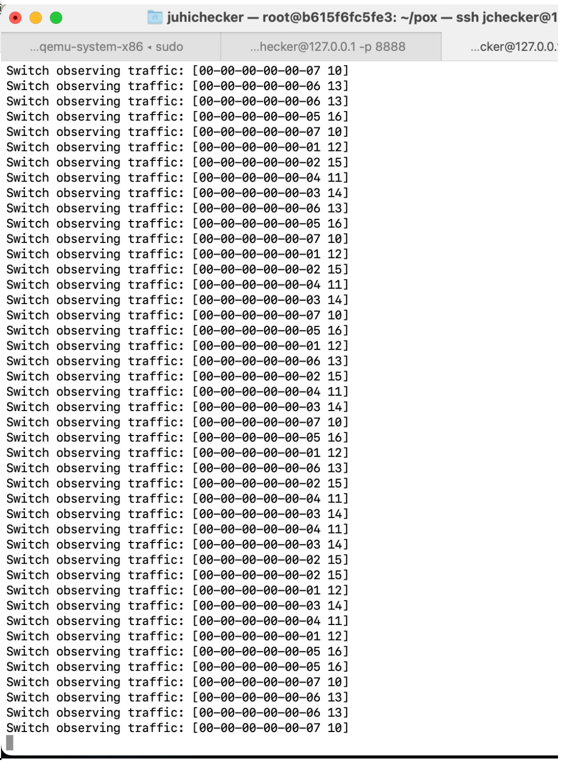
```
33:33:00:00:00:02 not known, resend to everybody
Src:  06:78:2a:ae:52:c1 : 1 Dst: 33:33:00:00:00:02
Learning that 06:78:2a:ae:52:c1 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 3 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 3
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 2 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 2 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 2
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
Learning that 02:80:c3:da:cc:c1 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 3 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 3
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 3 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 3
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 1
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
Learning that 3e:3e:6b:a9:d3:97 is attached at port 1
```


juhichecker — root@b615f6fc5fe3: ~/pox — ssh jchecker@1
...qemu-system-x86 ◂ sudo
...hecker@127.0.0.1 -p 8888
...cker@127.0.0.

```
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: 76:0c:76:f9:b9:d5
76:0c:76:f9:b9:d5 destination known. only send message to it
Src:  76:0c:76:f9:b9:d5 : 3 Dst: 22:61:41:e8:d2:79
22:61:41:e8:d2:79 destination known. only send message to it
Src:  22:61:41:e8:d2:79 : 2 Dst: ff:ff:ff:ff:ff:ff
ff:ff:ff:ff:ff:ff not known, resend to everybody
```


juhichecker — root@b615f6fc5fe3: ~/pox — ssh jchecker@
...qemu-system-x86 ◂ sudo
...hecker@127.0.0.1 -p 8888
...cker@127.0.

```
Src:  22:61:41:e8:d2:79 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  76:0c:76:f9:b9:d5 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  22:61:41:e8:d2:79 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  76:0c:76:f9:b9:d5 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  22:61:41:e8:d2:79 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  76:0c:76:f9:b9:d5 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  22:61:41:e8:d2:79 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  76:0c:76:f9:b9:d5 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 3 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 2 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 2 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  02:80:c3:da:cc:c1 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 3 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 3 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
Src:  3e:3e:6b:a9:d3:97 : 1 Dst: 33:33:00:00:00:02
33:33:00:00:00:02 not known, resend to everybody
```

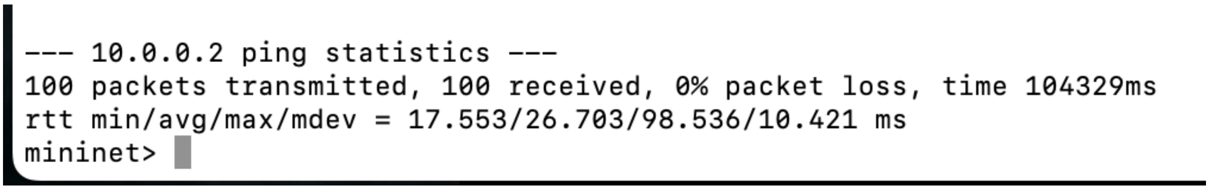

juhichecker — root@b615f6fc5fe3: ~/pox — ssh jchecker@1
...qemu-system-x86 ◂ sudo
...hecker@127.0.0.1 -p 8888
...cker@127.0.0.

```
Switch observing traffic: [00-00-00-00-00-07 10]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-05 16]
Switch observing traffic: [00-00-00-00-00-07 10]
Switch observing traffic: [00-00-00-00-00-01 12]
Switch observing traffic: [00-00-00-00-00-02 15]
Switch observing traffic: [00-00-00-00-00-04 11]
Switch observing traffic: [00-00-00-00-00-03 14]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-05 16]
Switch observing traffic: [00-00-00-00-00-07 10]
Switch observing traffic: [00-00-00-00-00-01 12]
Switch observing traffic: [00-00-00-00-00-02 15]
Switch observing traffic: [00-00-00-00-00-04 11]
Switch observing traffic: [00-00-00-00-00-03 14]
Switch observing traffic: [00-00-00-00-00-07 10]
Switch observing traffic: [00-00-00-00-00-05 16]
Switch observing traffic: [00-00-00-00-00-01 12]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-02 15]
Switch observing traffic: [00-00-00-00-00-04 11]
Switch observing traffic: [00-00-00-00-00-03 14]
Switch observing traffic: [00-00-00-00-00-07 10]
Switch observing traffic: [00-00-00-00-00-05 16]
Switch observing traffic: [00-00-00-00-00-01 12]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-02 15]
Switch observing traffic: [00-00-00-00-00-04 11]
Switch observing traffic: [00-00-00-00-00-03 14]
Switch observing traffic: [00-00-00-00-00-04 11]
Switch observing traffic: [00-00-00-00-00-03 14]
Switch observing traffic: [00-00-00-00-00-02 15]
Switch observing traffic: [00-00-00-00-00-02 15]
Switch observing traffic: [00-00-00-00-00-01 12]
Switch observing traffic: [00-00-00-00-00-03 14]
Switch observing traffic: [00-00-00-00-00-04 11]
Switch observing traffic: [00-00-00-00-00-01 12]
Switch observing traffic: [00-00-00-00-00-05 16]
Switch observing traffic: [00-00-00-00-00-05 16]
Switch observing traffic: [00-00-00-00-00-07 10]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-06 13]
Switch observing traffic: [00-00-00-00-00-07 10]
```

2. Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 p2).
   a. How long did it take (on average) to ping for each case?
   b. What is the minimum and maximum ping you have observed?
   c. Any difference from Task 2 and why do you think there is a change if there is?

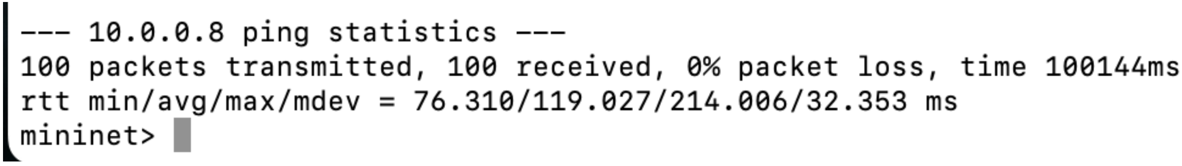
Ans.

h1 ping -c100 h2



```
--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 104329ms
rtt min/avg/max/mdev = 17.553/26.703/98.536/10.421 ms
mininet> 
```

h1 ping -c100 h8



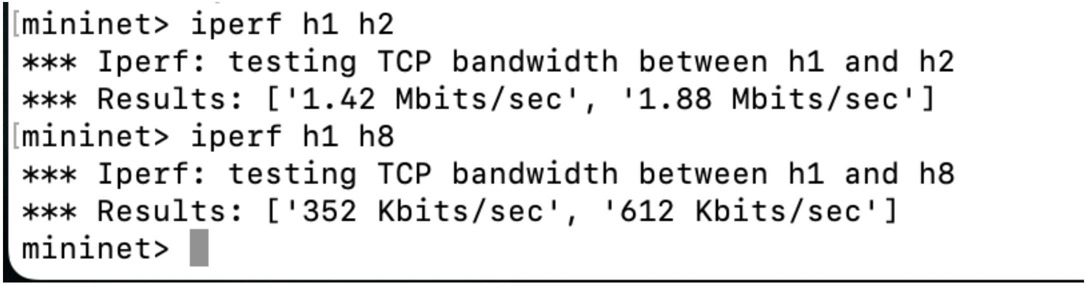
```
--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 100144ms
rtt min/avg/max/mdev = 76.310/119.027/214.006/32.353 ms
mininet> 
```

a. Average for h1 ping h2 = 26.703 ms
   Average for h1 ping h8 = 119.027 ms
b. Minimum value for h1 ping h2 = 17.533 ms
   Minimum value for h1 ping h8 = 76.310 ms
   Maximum value for h1 ping h2 = 98.536 ms
   Maximum value for h1 ping h8 = 214.006 ms
c. It can be observed that average, minimum and maximum value for h1 ping h2 is lesser than h1 ping h8. In terms of h1 ping h2 in Task 2 is almost same and slightly higher than Task 3. In h1 ping h3, the maximum value is significantly lower than in Task 3 than Task 2 and this is because in Task 3, it sends the packets first time and it saves the mac address. In all the later packets it directly maps mac address to port using mac_to_port and so it is faster.

3. Run “iperf h1 h2” and “iperf h1 h8”.
   a. What is the throughput for each case?
   b. What is the difference from Task 2 and why do you think there is a change if there is?

Ans.

a. Throughput in each of the following cases: Iperf h1 h2 and iperf h1 h8



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['1.42 Mbits/sec', '1.88 Mbits/sec']
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['352 Kbits/sec', '612 Kbits/sec']
mininet> 
```

b. The throughput in Task 3 for second case is more than Task 2 and this can be because once mac_to_port learns all the ports then it becomes faster to transmit the packets to the destination. This also causes less traffic on the network. It was also expected that in the first case Task 3 would perform better than Task 2 for the above mentioned reasons.