

HW 3 (COEN 241)

By: Aanchal Rai

Task 1

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

- Output of “nodes” in mininet is as follows:

c0 h1 h2 h3 h4 h5 h6 h7 h8 s1 s2 s3 s4 s5 s6 s7

-Output of “net” in mininet is as follows:

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”?

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::9484:13ff:fec1:47d1 prefixlen 64 scopeid 0x20<link>

ether 96:84:13:c1:47:d1 txqueuelen 1000 (Ethernet)

RX packets 51 bytes 3542 (3.5 KB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 13 bytes 1006 (1.0 KB)

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

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?
 - 1.1. **lauch()** , to launch the component
 - 1.2. **start_switch (event)** , to start the switch
 - 1.3. **__init__ (self, connection)**, function is called through the Tutorial class and
The packet arrives.
 - 1.4. **_handle_PacketIn (self, event)**, function will handle the packet and will send
The packet to either **act_like_switch** or
act_like_hub function depending on how the
Script is written

If it sends the packet to act_like_switch:

- 1.5 **act_like_switch (self, packet, packet_in)**, in this function, it will learn the port
For the mac address and try to
Resend the packet
- 1.6 **resend_packet (self, packet_in, out_port)**, instructs the switch to resend the
packet

If it sends the packet to act_like_hub:

- 1.5 **act_like_hub (self, packet, packet_in)**, implements hub-like behaviour and
Sends all packets to all ports besides input
Port
- 1.6 **resend_packet (self, packet_in, out_port)**, instructs the switch to resend the
packet

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

H1 Ping H2:-

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

Avg: 29.319ms

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

Min: 9.719ms

Max: 62.940ms

Mdev: 9.828ms

H1 Ping H8:-

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

Avg: 91.262ms

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

Min: 57.861ms

Max: 313.408ms

Mdev: 27.245ms

c. What is the difference, and why?

Difference in minimum time: 48.142ms

Difference in maximum time: 250.468ms

Difference in Avg time: 61.943

The **H1 ping H2 is faster** than H1 ping H8 because the packet from H1 to H2 has to traverse through only one switch - S3.

The H1 ping H8 takes more time because the packet from H1 has to traverse through many switches to get to H8 (i.e. S3, S2, S1, S5, S7)

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

a. What is "iperf" used for?

Iperf is a tool that is open source. It is primarily used to help admins measure the Bandwidth for network performance and quality of a network line.

b. What is the throughput for each case?

A: Iperf h1 h2:

Throughput : 478 Kbits/sec, 696 Kbits/sec

Iperf h1 h8:

Throughput: 413 Kbits/sec, 667 Kbits/sec

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

A: Difference in Throughput: 65 Kbits/sec, 29 Kbits/sec.

When the packets travel from H1 to H2, they have to travel through only one switch. This leads to increase in throughput.

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).

A: When H1 ping H2 command is run on mininet, the controller searches the mac_to_port Dictionary to find a port that is attached to the MAC address. If it finds it, resend_packet() Function is called to send the packet (packet_in) to that port.

If it does not find the port, then the packet (packet_in) is sent to of.OFPP_ALL port for flooding.

2.(Comment out all prints before doing this experiment)

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

H1 ping H2:

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

A: Avg: 29.650ms

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

A: Min: 9.805ms

Max: 56.156ms

Mdev: 8.050ms

H1 ping H8:

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

A: Avg:82.240ms

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

A: Min:44.099ms
Max:240.015ms
Mdev:20.362ms

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

A: For H1 ping H2, there is a minor increase in Task 3 compared to task 2, this may be because it takes more time for the controller to add both items into the dictionary port And then send the packets (this is a short travel path)

For H1 ping H8, there is a slight decrease in Task 3 compared to task 2, this may be because The traversal path from h1 to h8 is longer, the ports might already be mapped in the dictionary And therefore flooding may not be required as much.

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

a.What is the throughput for each case?

Iperf h1 h2:

- Throughput : 3.70 Mbits/sec, 4.42 Mbits/sec

Iperf h1 h8:

- Throughput: 295 Kbits/sec, 713 Kbits/sec

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

A: Throughput in Task 3 is higher than Task 2. This may be because in Task 3, the Controller uses the port dictionary to map the MAC address to the correct port. This would Mean that it is spending less time finding the correct port through flooding and therefore Providing higher throughput.