Homework 3

COEN 241 Cloud Computing

Task 1

Questions

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

Output of nodes:

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

Output of 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”

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

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?
First the controller initializes with the following function calls

```
launch () -> start_switch (event) -> __init__ (self, connection)
```

Once a packet arrives, it follows the following graph based on the code

```
_handle_PacketIn (self, event)-> act_like_hub (self, packet, packet_in) ->
resend_packet (self, packet_in, out_port):

Or

_handle_PacketIn (self, event)-> act_like_switch (self, packet, packet_in)->
resend_packet (self, packet_in, out_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?
h1 ping h2 : 27.621ms

h1 ping h8 : 95.445ms

b. What is the minimum and maximum ping you have observed?
h1 ping h2 :
Minimum: 9.255ms
Maximum: 108.049ms
h1 ping h8:

Minimum: 59.295ms
Maximum: 199.414ms

c. What is the difference, and why?
Minimum ping difference: 50.04 ms
Maximum ping difference: 91.365

Since h1 and h2 are connected to the same switch, it will have lower ping time as packets have to travel between more switches in case of h1 and h8.

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

a. What is “iperf” used for?
Iperf is an open source tool for helping the administrators measure the bandwidth for the network performance and quality of a network line.

b. What is the throughput for each case?
iperf h1 h2
Throughput : 448 Kbits/sec, 802 Kbits/sec

iperf h1 h8
Throughput: 386 Kbits/sec , 667 Kbits/sec

c. What is the difference, and explain the reasons for the difference.
Difference in throughput: 62 Kbits/sec, 135 Kbits/sec

As the packet travel time is lesser for h1 and h2 compared to h1 and h8, more packets can be transmitted and hence has a higher throughput.

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).

I used tcpdump on the switches and hosts to observe traffic.

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).

When h1 pings h2, the controller will look in the switch's mac_to_port dictionary to see if the appropriate port is known. If it is, it uses self.resend_packet() to send packet_in to that port. If It is not, then send packet_in to of.OFPP_FLOOD (or ALL) is used to flood the packets.

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).

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

h1 ping h2 : 29.652 ms

h1 ping h8 : 83.243ms

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

h1 ping h2 :

Minimum: 9.915 ms

Maximum: 57.156 ms

h1 ping h8:

Minimum: 45.099 ms

Maximum: 241.015 ms

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

A slight increase in ping time is observed for h1 and h2. As there is a short traversal path between the hosts and as the controller is referring to the ports dictionary before sending the packets, it takes more time than flooding the packets.

In case of h1 and h8, a slight decrease is observed. This is due to the fact that h1 and h8 have a longer path between them. Once the ports are mapped, it will be quicker to send the packets directly to the destination compared to flooding the packets.

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

a. What is the throughput for each case?

iperf h1 h2

Throughput : 3.71 Mbits/sec, 4.21 Mbits/sec

iperf h1 h8

Throughput: 296 Kbits/sec , 711 Kbits/sec

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

The throughput is much higher than the throughput observed in Task 2. This might be due to the fact that once the mac address is mapped to the port, the packets are sent only to the required host instead of flooding to all hosts.