COEN241 HW-3

Mininet & OpenFlow

Name: Surya Kiran Udaya Kumar

TASK-1

1. **What is the output of "nodes" and "net"**
   nodes
   h1 h2 h3 h4 h5 h6 h7 h8 s1 s2 s3 s4 s5 s6 s7

   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:s5-eth1 s1-eth2:s2-eth1
   s2 lo: s2-eth1:s1-eth2 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-eth1 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

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::c44d:43ff:fe42:115d prefixlen 64 scopeid 0x20<link>
ether c6:4d:43:42:11:5d txqueuelen 1000 (Ethernet)
RX packets 81 bytes 6234 (6.2 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

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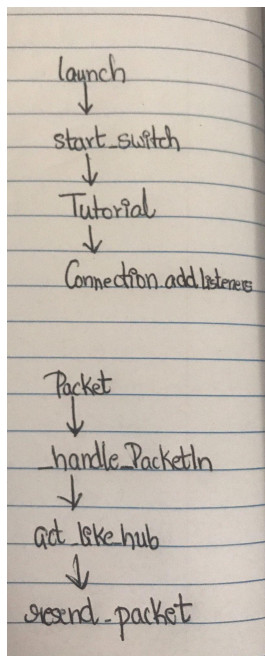
2. Have h1 ping h2, and h1 ping h8 for 100 times (e.g., h1 ping -c100 h2).

a) How long does it take (on average) to ping for each case?
Ans) h1 ping h2**:** average: 4.788 ms
       h1 ping h8 : average: 28.001 ms

b) What are the minimum and maximum ping you have observed?
Ans) **h1 ping -c 100 h2**

min/max: 2.043/ 14.680 ms

h1 ping -c 100 h8

min/max: 7.873/ 74.716 ms

c. What is the difference, and why?

Ans) (h1 ping -c 100 h2) - (h1 ping -c 100 h8):

Difference in average: 28.001 - 4.788 = 23.213 ms
Difference in min : 7.873 - 2.043 = 5.83 ms
Difference in max : 74.716 - 14.680 = 60.036 ms

It was seen that the packet communication between h1 and h2 was faster when compared to h1 and h8. This could be due to the number of switches in between hosts, h1 ping h8 would have to traverse through 7 switches whereas h1 ping h2 will have to traverse through only one switch.

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

a) What is "iperf" used for?
Ans) It is the throughput of the network. "iperf" tests the maximum network bandwidth, calculates the data that can be transferred between two nodes within a given time frame.

b) What is the throughput for each case?
Ans) **iperf h1 h2**

Results: ['9.71 Mbits/sec', '11.3 Mbits/sec']

iperf h1 h8
Results: ['4.14 Mbits/sec', '4.77 Mbits/sec']

c) **What is the difference, and explain the reasons for the difference.**
Ans) The throughput for iperf h1 h2 is better than iperf h1 h8 because of the number of switches it needs to traverse in order to successfully deliver the packets. Since the rtt between h1 and h2 will be less, it will allow more packets to flow, faster.

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)   All the switches observe the traffic s1,s2,s3,s4,s5,s6,s7.
Added a print line in the _handle_PacketIn(self, event) function to see the packets from the switch.
Eg: print('Incoming packet from: ',event.connection)

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) For MAC addresses that are not known until a packet is received from said MAC address, the Tutorial object creates a "MAC to Port" map for each switch, storing the MAC address as a key and the port as its value. If the packet destination MAC address and port are known, the switch can send packets directly to the ports it needs to send packets to.

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

a) **How long did it take (on average) to ping for each case?**
   Ans) h1 ping h2 :avg = 1.245 ms

h1 ping h8 :avg = 7.607 ms

b)   What is the minimum and maximum ping you have observed?
Ans) h1 ping h2 : min/ max = 0.965/1.677 ms
h1 ping h8 : min/max = 4.137/15.448 ms

c)   Any difference from Task 2 and why do you think there is a change if there is?
Ans) The rtt is much lower and the packets take a shorter time than task 2 because the switches know where exactly to send the packets after the first addition of the src address into the address dictionary. It would not flood all the ports with packets once the src and destination is known.

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

a. What is the throughput for each case?
  iperf h1 h2
  Results: ['29.9 Mbits/sec', '33.5 Mbits/sec']

  iperf h1 h8
  Results: ['4.83 Mbits/sec', '5.18 Mbits/sec']

```
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['4.83 Mbits/sec', '5.18 Mbits/sec']
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['29.9 Mbits/sec', '33.5 Mbits/sec']
mininet>
```

b. What is the difference from Task 2 and why do you think there is a change if there is?
Ans) We can observe that the throughput for h1 ping h2 is much better than in task2, this is because the destination port is detected and known by the sender, it would use this information to send the data only to the desired destination and thus the data transfer rate is much faster.

For h1 ping h8, there isn't a significant change in the throughput. This can be accounted for due to the fact that it needs to traverse a much longer distance(5 switches)s3, s2, s2, s5, s7 and a short testing time of iperf isn't sufficient.