Ashita Jayesh Shetty
Student Id: 00001607967

HW3:

Task 1

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

```
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-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
mininet> _
```

2. What is the output of “h7 ifconfig”

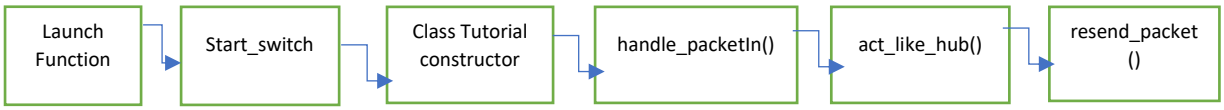
```
mininet> h7 ifconfig
h7-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.7  netmask 255.0.0.0  broadcast 10.255.255.255
        inet6 fe80::9c9c:ecff:fe20:fcd0  prefixlen 64  scopeid 0x20<link>
        ether 9e:9c:ec:20:fc:d0  txqueuelen 1000  (Ethernet)
        RX packets 72  bytes 5476 (5.4 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
```

Ashita Jayesh Shetty
Student Id: 00001607967

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?
   Solution:
   Here is the flow of the function call for the controller:



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 average time = 7.122 ms
      h1 ping h8 average time = 27.506 ms

   b. What is the minimum and maximum ping you have observed?
      h1 ping h2:
         min: 1.38 ms
         max: 17.739 ms
      h1 ping h8:
         min: 7.702 ms
         max: 55.482 ms
   c. What is the difference, and why?

      For h1 to ping h8 it has to go through series of switches, so the time difference is quite high when compared to h1 pinging h2.

3. Run “iperf h1 h2” and “iperf h1 h8”
   a. What is “iperf” used for?
      To test the TCP bandwidth between hosts.
   b. What is the throughput for each case?
      h1 – h2:
         Results: 6.14 Mbits/sec, 7.00 Mbits/sec
      h1 – h8:
         Results: 2.35 Mbits/sec, 2.84 Mbits/sec

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

      Looking at the figures we can deduce that the throughput between h1 and h2 is much higher than between host h1 and h8. The reasons can be due to congestion and latency. Congestion between hosts h1 and h8 will be higher because the packets have to travel through multiple switches. Also latency will also increase because TCP has to wait for ack from the first packet sent, so the time between sending a packet and waiting for an ACK will be more for h1 to h8.

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

    Solution:

    To observe the traffic, we can add a log statement in *handle_packetIn* method, which is the event listener for the switch and will keep track of each packets received.

    The log statement used:

    Log.debug(“Observing traffic on switch %s” % (self.connection))

    Self.connection will give the current switch number.

    On running the commands and looking at the debugs its clear that **all the switches** observe traffic.

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).
   Solution:
   The working of the code is as following:
   ➔ All the switches have their own dictionaries : self.mac_to_port, which maps the MAC address of the host with its corresponding port number.
   ➔ When this switch receives a packet, it checks if the source address is present in the dictionary and if not it will do the mapping. This is done in the method act_like_switch.
   ➔ Now it checks for the destination address, if present in the map it resends the packet, else will flood all the ports with the packet.

   Example:

   Following things will happen when h1 pings h2 multiple times:

    a. The first time h1 pings h2 :
        - H1 sends packet to s3 from port 1.
        - S3 does the mapping, P1 to the Mac address of H1.
        - Since there is no destination address, S3 will flood all ports, P2 and P3, with the packet.
        - H2 will get the packet from P2 and will send an ACK signal back.
        - S3 will map P2 with H2’s MAC address in the dictionary.
        - Since it already has H1’s MAC address and port details, it will send the packet through P1.
    b. The second time h1 pings h2.
        - Since S3 already has all the addresses, source and destination both, it will simply forward it to the respective ports.

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?
       h1 ping h2 average time = 4 .051 ms
       h1 ping h8 average time = 11.011 ms
    b. What is the minimum and maximum ping you have observed?
       h1 ping h2:
            min: 1.04 ms
            max: 9.29 ms
       h1 ping h8 :
            min: 6.812 ms

max: 22.404 ms

c. Any difference from Task 2 and why do you think there is a change if there is?
The time taken is less compared to task 3. Since the congestion and latency will be less as the addresses are calculated right at the start, within the 1st two pings. Later on there is no need to flood all the ports, as the addresses are already available. This makes the transmission faster.

3. Run "iperf h1 h2" and "iperf h1 h8".
   a. What is the throughput for each case?
   H1 – h2:
   Results: 61.01 Mbits/sec
   H1 – h8:
   Results: 12.35 Mbits/sec
   b. What is the difference from Task 2 and why do you think there is a change if there is?

   Throughput for both h1-h2 and h1-h8 has increased tremendously when compared to task2. This is because the ports are now not flooded with packets since the switches already have the address. So the congestion has reduced by a huge margin, thus increasing the throughput.