Andrew Knaus
COEN 241
HW 3

Task 1:

Questions:

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

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

    b. Output of “net”

```
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
```

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::4096:e2ff:fee6:e927  prefixlen 64  scopeid 0x20<link>
        ether 42:96:e2:e6:e9:27  txqueuelen 1000  (Ethernet)
        RX packets 61  bytes 4402 (4.4 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?
    a. Starts with the launch() function
    b. Starts up a listener that calls the start_switch() function when a connection is made
    c. It creates an instance of class Tutorial for each of the switches, and the constructor is called
    d. The _handle_PacketIn() function then handles the packets in msg from each switch
    e. Finally act_like_hub() (OR act_like_switch()) function is called which in turn calls resend_packet()
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?
        i. h1 ping h2
            - avg: 36.020 ms
        ii. h1 ping h8:
            - avg: 95.879 ms
    b. What is the minimum and maximum ping you have observed?
        i. h1 ping h2
            - min: 11.139 ms
            - max: 59.079 ms
        ii. h1 ping h8:
            - min: 54.809 ms
            - max: 144.199 ms
    c. What is the difference, and why?
        i. Ping between h1 and h2 is faster than the ping between h1 and h8. This is because the ping between h1 and h2 only has to go through 1 switch (h1 -> s3 -> h2) while the ping between h1 and h8 has to go through 5 switches (h1 -> s3 -> s2 -> s1 -> s5 -> s7 -> h8).
3. Run “iperf h1 h2” and “iperf h1 h8”
    a. What is “iperf” used for?
        i. iperf is used to test the tcp bandwidth between 2 hosts.
    b. What is the throughput for each case?
        i. Case: iperf h1 h2
            - Results 1: [‘729 Kbits/sec’, ‘1.32 Mbits/sec’]

- Results 2: [‘701 Kbits/sec’, ‘1.17 Mbits/sec’]
- Results 3: [‘733 Kbits/sec’, ‘1.14 Mbits/sec’]
- Results 4: [‘625 Kbits/sec’, ‘1.07 Mbits/sec’]
- Results 5: [‘589 Kbits/sec’, ‘1.15 Mbits/sec’]
- Results Avg: [‘675.4 Kbits/sec’, ‘1.17 Mbits/sec’]
- The average throughput of **h1 is 675.4 Kbits/sec** and the average throughput of **h2 is 1.17 Mbits/sec**

ii. Case: iperf h1 h8

- Results 1: [‘491 Kbits/sec’, ‘953 Kbits/sec’]
- Results 2: [‘482 Kbits/sec’, ‘940 Kbits/sec’]
- Results 3: [‘514 Kbits/sec’, ‘865 Kbits/sec’]
- Results 4: [‘468 Kbits/sec’, ‘809 Kbits/sec’]
- Results 5: [‘481 Kbits/sec’, ‘782 Kbits/sec’]
- Results Avg: [‘487.2 Kbits/sec’, ‘869.8 Kbits/sec’]
- The average throughput of **h1 is 487.2 Kbits/sec** and the average throughput of **h2 is 869.8 Kbits/sec**

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

i. The throughput for the h1 -> h2 was slightly better than the throughput for h1 -> h8. This can be expected because h1 is closer (with fewer switches to go through) to h2 than h8.

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).
    a. All of the switches are observing traffic. This is because when a packet is sent, it is sent to all of the switches and not just the ones between the two communicating hosts. If we wish to observe this traffic, we can add print statements to the _handle_PacketIn() function with the switch ID, gotten with the self.connection.dpid command, to show when traffic goes through the switch.

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).
    a. When sending a message, if the destination MAC already exists in the "MAC to Port" map, it can be directly sent to the destination. Otherwise, it will flood the message to every port except the incoming port.
    b. Example: h1 ping h2

i. The map is initially empty
ii. When h1 first pings h2, we check if the mapping of h1 to port exists. Since it does not exist, it gets inserted into the map.
iii. Next, the destination MAC address to port mapping is checked. Since the mapping does not exist yet, we send the packet to all ports.
iv. When h2 receives the packet, it sends back an acknowledgment.
v. We check if mapping exists for h2. Since it does not, it gets inserted into the map.
vi. We then check if the port of h1 is available in the map. It is, so we send the packet directly to h1.
vii. For every subsequent ping, we are able to directly send the packet from h1 to h2 and directly send the acknowledgment from h2 to h1.

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?
        i. h1 ping -c 100 h2
            - Avg: 43.101 ms
        ii. h1 ping -c 100 h8
            - Avg: 103.748 ms
    b. What is the minimum and maximum ping you have observed?
        i. h1 ping -c 100 h2
            - Min: 8.462 ms
            - Max: 70.511 ms
        ii. h1 ping -c 100 h8
            - Min: 80.330 ms
            - Max: 144.880 ms
    c. Any difference from Task 2 and why do you think there is a change if there is?
        i. There was no improvement, in fact, it appears that on average the pings in Task 2 were slightly better. This makes some sense considering the act_like_hub function is shorter than the act_like_switch function that we implemented. Additionally, the implementation of act_like_switch does not shorten the path between hubs, it just minimizes the packets sent to switches that are not between the two communicating hosts.
3. Run “iperf h1 h2” and “iperf h1 h8”.
    a. What is the throughput for each case?

i. iperf h1 h2
    - Results: [‘4.55 Mbits/sec’, ‘5.02 Mbits/sec’]
    - Results: [‘4.48 Mbits/sec’, ‘5.12 Mbits/sec’]
    - Results: [‘4.42 Mbits/sec’, ‘5.02 Mbits/sec’]
    - Results: [‘4.36 Mbits/sec’, ‘5.17 Mbits/sec’]
    - Results: [‘3.91 Mbits/sec’, ‘4.38 Mbits/sec’]
    - Average Results: [‘4.344 Mbits/sec’, ‘4.942 Mbits/sec’]
    - The average throughput of **h1 is 4.344 Mbits/sec** and the average throughput of **h2 is 4.942 Mbits/sec**

ii. iperf h1 h8
    - Results: [‘696 Kbits/sec’, ‘1.05 Mbits/sec’]
    - Results: [‘735 Kbits/sec’, ‘1.14 Mbits/sec’]
    - Results: [‘784 Kbits/sec’, ‘1.22 Mbits/sec’]
    - Results: [‘730 Kbits/sec’, ‘1.10 Mbits/sec’]
    - Results: [‘751 Kbits/sec’, ‘1.11 Mbits/sec’]
    - Average Results: [‘739.2 Kbits/sec’, ‘1.124 Mbits/sec’]
    - The average throughput of **h1 is 739.2 Kbits/sec** and the average throughput of **h2 is 1.124 Mbits/sec**

b. What is the difference from Task 2 and why do you think there is a change if there is?
    i. The throughput was massively better than the results from Task 2 when looking at iperf h1 h2, especially for h1. When looking at iperf h1 h8 we can still seed substantial improvements with its throughput when compared to Task 2 results. These improvements make sense because once the port for a given MAC is known, the packets are able to be directly sent to their destination instead of being flooded to all ports.