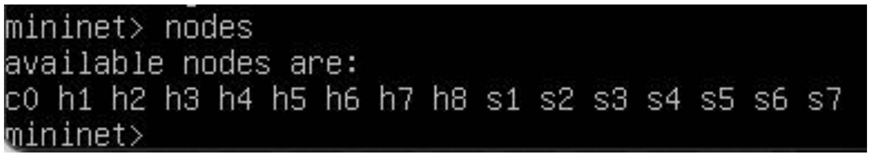
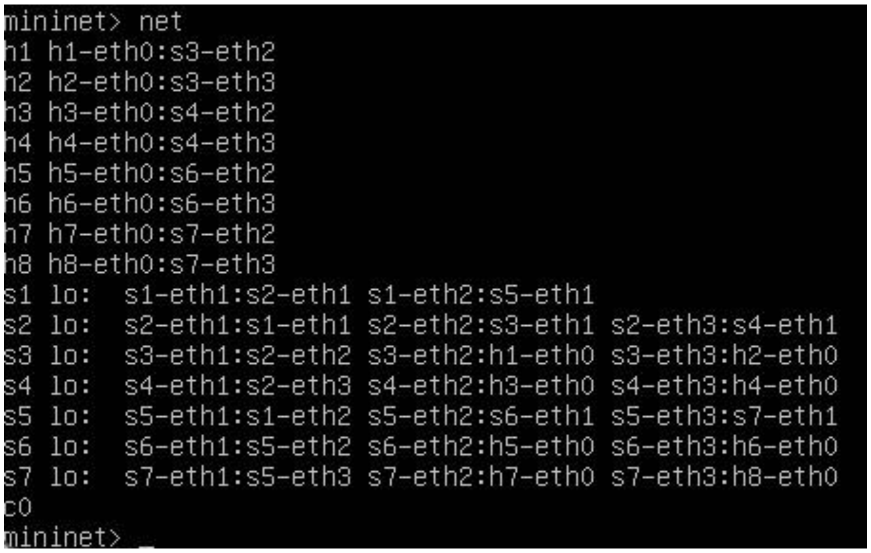
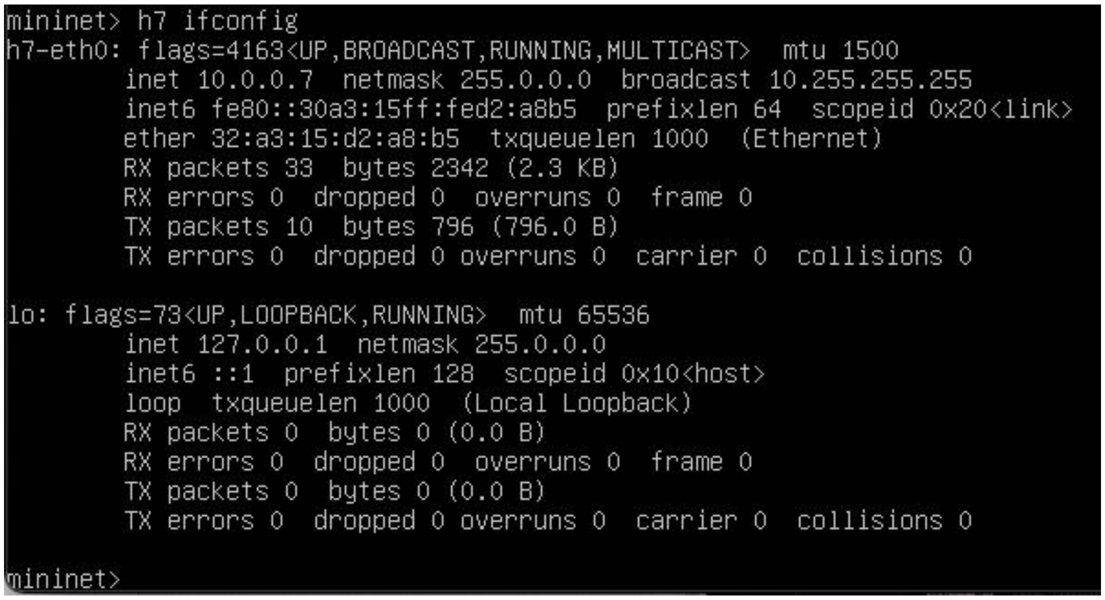
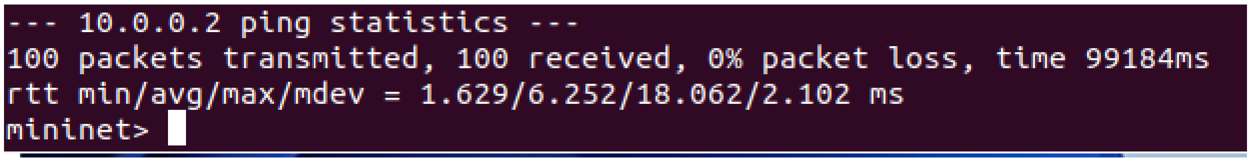
Task 1
Question 1
Nodes:

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

Question 2

```
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::30a3:15ff:fed2:a8b5  prefixlen 64  scopeid 0x20<link>
        ether 32:a3:15:d2:a8:b5  txqueuelen 1000  (Ethernet)
        RX packets 33  bytes 2342 (2.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 10  bytes 796 (796.0 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

mininet>
```

Task 2
Question 1

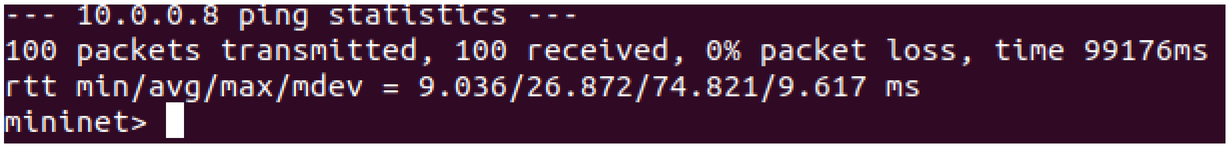
First of all, when packet comes in, _handle_PacketIn will be called first. If the packet is parsed successfully, act_like_hub will be called. Then act_like_hub will be called and flood the packet to all the ports except coming in port.

Question 2
For h1 ping h2
a. Avg rtt: 6.252 ms.
b. Min rtt: 1.629 ms. Max rtt: 18.062 ms



```
--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99184ms
rtt min/avg/max/mdev = 1.629/6.252/18.062/2.102 ms
mininet>
```

For h1 ping h8
a. Avg rtt: 26.872 ms.
b. Min rtt: 9.036 ms. Max rtt: 74.821 ms

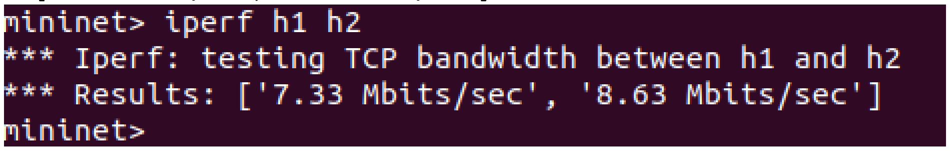


```
--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99176ms
rtt min/avg/max/mdev = 9.036/26.872/74.821/9.617 ms
mininet>
```

c. There are only one switch between h1 and h2. But for h1 and h8, there are 5 switches: s3, s2, s1, s5, s7. So h1 ping h8 will cost more delay.

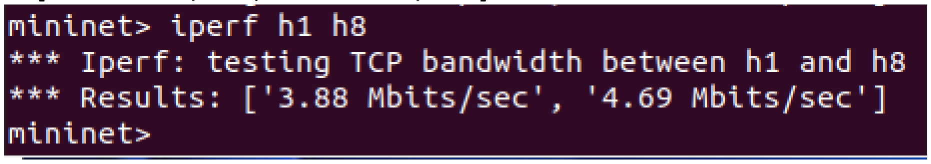
Question 3
a. Used for testing TCP bandwidth between two hosts.

For iperf h1 h2
b. ['7.33 Mbits/sec', '8.63 Mbits/sec']



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['7.33 Mbits/sec', '8.63 Mbits/sec']
mininet>
```

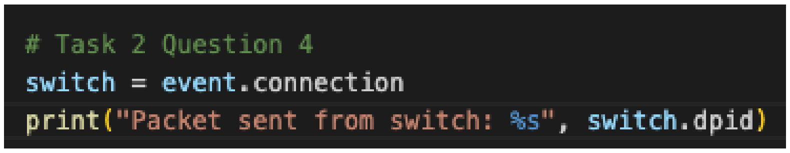
For iperf h1 h8
b. ['3.88 Mbits/sec', '4.69 Mbits/sec']



```
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['3.88 Mbits/sec', '4.69 Mbits/sec']
mininet>
```

c. There are only one switch between h1 and h2. But for h1 and h8, there are 5 switches: s3, s2, s1, s5, s7. So h1 ping h8 will cost more delay, which reduces bandwidth. Also since there are more traffic (flood packet to all port accept the one receive the packet), the bandwidth will also be lower.

Question 4.
We can do this simply by adding these two lines to _handle_PacketIn function.

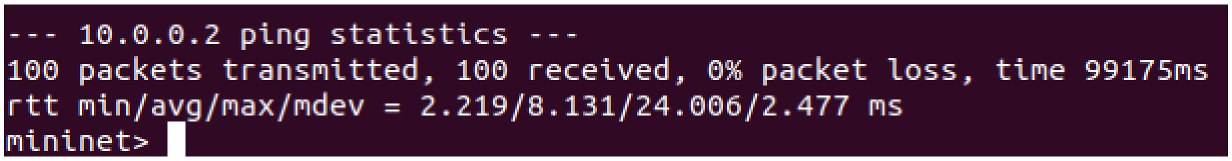


```
# Task 2 Question 4
switch = event.connection
print("Packet sent from switch: %s", switch.dpid)
```

_handle_PacketIn is called when new packet come in. So every time a packet came in, we will be able to know which switch does this packet came from.

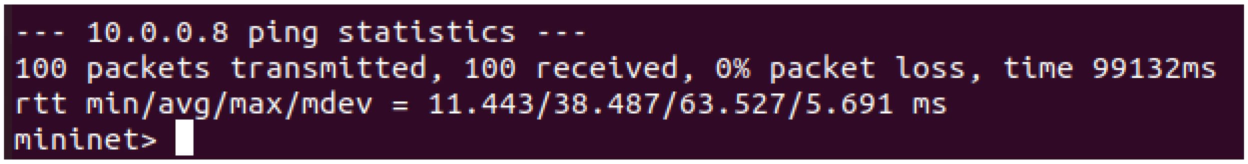
Task 3
Question 1
mac_to_port is a dictionary used to remember which port are associated to which mac address. So every time a packet comes in, it will learn the mac address and its port number if its own mac address is not in the mac_to_port. Then it will check if the dst address is in mac_to_port. If yes, it will simply send the packet to that port. Otherwise, it will flood the packet to all port except the port that receive packet.

Question 2
For h1 ping h2
a. Avg rtt: 8.131 ms
b. Min rtt: 2.219 ms. Max rtt: 24.006 ms



```
--- 10.0.0.2 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99175ms
rtt min/avg/max/mdev = 2.219/8.131/24.006/2.477 ms
mininet>
```

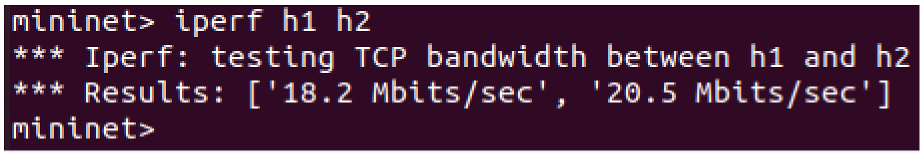
For h1 ping h8
a. Avg rtt: 38.487 ms
b. Min rtt: 11.443 ms. Max rtt: 63.627 ms



```
--- 10.0.0.8 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99132ms
rtt min/avg/max/mdev = 11.443/38.487/63.527/5.691 ms
mininet>
```

c. The rtt time is a little bit higher that task 2. The reason is that extra time is needed to learn and to check if the mac address exists. Compared to task 2, where the packet is simply flood to all port except the one accept the packet, more time is needed.

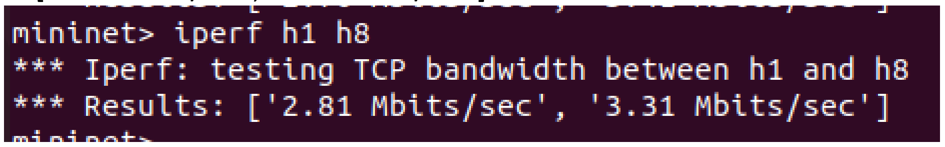
Question 3.
For iperf h1 h2
a. ['18.2 Mbits/sec', '20.5 Mbits/sec']



```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['18.2 Mbits/sec', '20.5 Mbits/sec']
mininet>
```

For iperf h1 h8

a. [‘2.81 Mbits/sec’, ‘3.31 Mbits/sec’]



```
mininet> iperf h1 h8
*** Iperf: testing TCP bandwidth between h1 and h8
*** Results: ['2.81 Mbits/sec', '3.31 Mbits/sec']
```

b. For the bandwidth between h1 and h2, it’s higher. But for h1 and h8, it’s a little bit lower. For h1 and h2, since the controller know which port to send the packet, there will be less traffic on the network. So bandwidth is higher. But for h1 and h8, it’s because the overhead of time which cause by learning and checking.