CS3611 Computer Networks (Spring 2023)
Lab 2: Play with Mininet

Xiangyuan Xue (521030910387)

1. (a) Define the topology in Mininet with Python as follows.

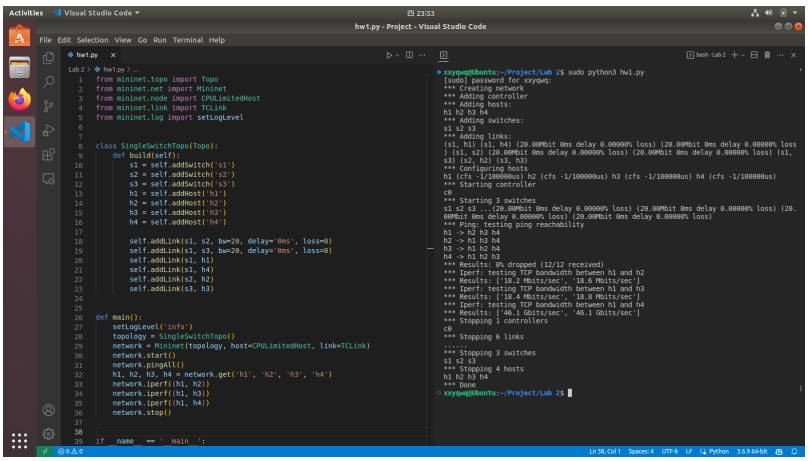
```python
class SingleSwitchTopo(Topo):
    def build(self):
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')

        self.addLink(s1, s2, bw=20, delay='0ms', loss=0)
        self.addLink(s1, s3, bw=20, delay='0ms', loss=0)
        self.addLink(s1, h1)
        self.addLink(s1, h4)
        self.addLink(s2, h2)
        self.addLink(s3, h3)
```

The topology contains 3 switches, 4 hosts and 6 links.

(b) Run the script and the results are as follows.

```
$ sudo python3 hw1.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0
   ms delay 0.00000% loss) (s1, s2) (20.00Mbit 0ms delay 0.00000%
    loss) (20.00Mbit 0ms delay 0.00000% loss) (s1, s3) (s2, h2) (
   s3, h3)
*** Configuring hosts
```

```
h1 (cfs -1/100000us) h2 (cfs -1/100000us) h3 (cfs -1/100000us) h4
    (cfs -1/100000us)
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms
    delay 0.00000% loss) (20.00Mbit 0ms delay 0.00000% loss)
    (20.00Mbit 0ms delay 0.00000% loss)
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['18.2 Mbits/sec', '18.6 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['18.4 Mbits/sec', '18.8 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['46.1 Gbits/sec', '46.1 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 6 links
......
*** Stopping 3 switches
s1 s2 s3
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
```



The results show that the TCP throughput between h_1 and h_2 is 18.4Mbits/s, between

h_1 and h_3 is 18.6Mbits/s and between h_1 and h_4 is 46.1Gbits/s, which is consistent with the parameters we set in the topology.

2. (a) Define the topology in Mininet with Python as follows.

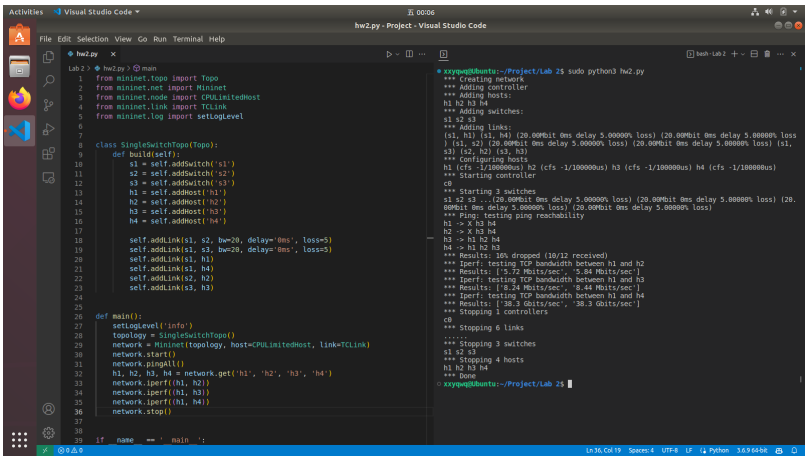
```
class SingleSwitchTopo(Topo):
    def build(self):
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')

        self.addLink(s1, s2, bw=20, delay='0ms', loss=5)
        self.addLink(s1, s3, bw=20, delay='0ms', loss=5)
        self.addLink(s1, h1)
        self.addLink(s1, h4)
        self.addLink(s2, h2)
        self.addLink(s3, h3)
```

The topology contains 3 switches, 4 hosts and 6 links. The packet loss rate of the link (s_1, s_2) and (s_1, s_3) is set to be 5%.

(b) Run the script and the results are as follows.

```
$ sudo python3 hw2.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (20.00Mbit 0ms delay 5.00000% loss) (20.00Mbit 0
   ms delay 5.00000% loss) (s1, s2) (20.00Mbit 0ms delay 5.00000%
    loss) (20.00Mbit 0ms delay 5.00000% loss) (s1, s3) (s2, h2) (
   s3, h3)
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us) h3 (cfs -1/100000us) h4
   (cfs -1/100000us)
*** Starting controller
c0
*** Starting 3 switches
```

```
s1 s2 s3 ...(20.00Mbit 0ms delay 5.00000% loss) (20.00Mbit 0ms
    delay 5.00000% loss) (20.00Mbit 0ms delay 5.00000% loss)
    (20.00Mbit 0ms delay 5.00000% loss)
*** Ping: testing ping reachability
h1 -> X h3 h4
h2 -> X h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 16% dropped (10/12 received)
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['5.72 Mbits/sec', '5.84 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['8.24 Mbits/sec', '8.44 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['38.3 Gbits/sec', '38.3 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 6 links
......
*** Stopping 3 switches
s1 s2 s3
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
```



The results show that the TCP throughput between h_1 and h_2 is 5.78Mbits/s, between h_1 and h_3 is 8.34Mbits/s and between h_1 and h_4 is 38.3Gbits/s, which drops significantly compared with the previous topology.

3. (a) Define the topology in Mininet with Python as follows.

```
class SingleSwitchTopo(Topo):
```

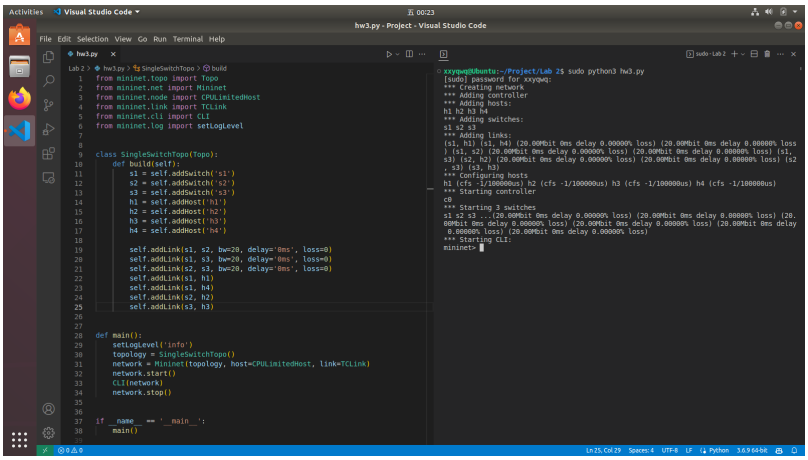
```
    def build(self):
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')

        self.addLink(s1, s2, bw=20, delay='0ms', loss=0)
        self.addLink(s1, s3, bw=20, delay='0ms', loss=0)
        self.addLink(s2, s3, bw=20, delay='0ms', loss=0)
        self.addLink(s1, h1)
        self.addLink(s1, h4)
        self.addLink(s2, h2)
        self.addLink(s3, h3)
```

The topology contains 3 switches, 4 hosts and 7 links, where another link (s_2, s_3) is added.

(b) Run the script and the results are as follows.

```
$ sudo python3 hw3.py
[sudo] password for xxyqwq:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0
    ms delay 0.00000% loss) (s1, s2) (20.00Mbit 0ms delay 0.00000%
     loss) (20.00Mbit 0ms delay 0.00000% loss) (s1, s3) (s2, h2)
    (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms delay
    0.00000% loss) (s2, s3) (s3, h3)
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us) h3 (cfs -1/100000us) h4
    (cfs -1/100000us)
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms
    delay 0.00000% loss) (20.00Mbit 0ms delay 0.00000% loss)
```

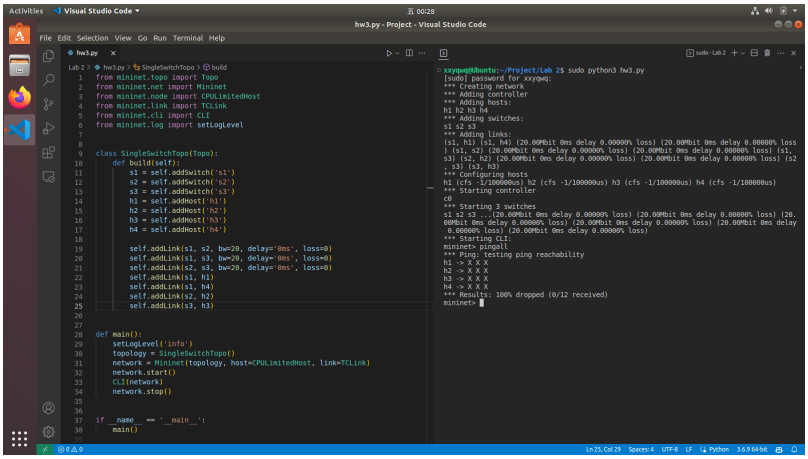
```
    (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms delay
    0.00000% loss) (20.00Mbit 0ms delay 0.00000% loss)
*** Starting CLI:
mininet>
```



The results show that the topology has been created.

(c) Run the command and the results are as follows.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X
h2 -> X X X
h3 -> X X X
h4 -> X X X
*** Results: 100% dropped (0/12 received)
```



The results show that all the packets are dropped and the hosts are disconnected. This problem comes from the loop in the topology, where the packets are broadcast back and forth among the switches, and will never be delivered correctly to the hosts.

(d) To solve the problem, we add 2 flow rules on the switches, where the packets from s_2 to s_3 are sent to h_3 only and the packets from s_3 to s_2 are sent to h_2 only. The shell script is written as follows

```
#!/bin/bash
sudo ovs-ofctl add-flow s2 in_port=s2-eth2,actions=output:s2-eth3
sudo ovs-ofctl add-flow s3 in_port=s3-eth2,actions=output:s3-eth3
echo Done
```

We expect these rules to solve the connection problem.

(e) First we create the topology as follows

```
$ sudo python3 hw3.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0
    ms delay 0.00000% loss) (s1, s2) (20.00Mbit 0ms delay 0.00000%
     loss) (20.00Mbit 0ms delay 0.00000% loss) (s1, s3) (s2, h2)
    (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms delay
    0.00000% loss) (s2, s3) (s3, h3)
*** Configuring hosts
h1 (cfs -1/100000us) h2 (cfs -1/100000us) h3 (cfs -1/100000us) h4
    (cfs -1/100000us)
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms
    delay 0.00000% loss) (20.00Mbit 0ms delay 0.00000% loss)
    (20.00Mbit 0ms delay 0.00000% loss) (20.00Mbit 0ms delay
    0.00000% loss) (20.00Mbit 0ms delay 0.00000% loss)
*** Starting CLI:
mininet>
```

Then we refer to the link information as follows

```
mininet> net
h1 h1-eth0:s1-eth3
h2 h2-eth0:s2-eth3
h3 h3-eth0:s3-eth3
h4 h4-eth0:s1-eth4
```

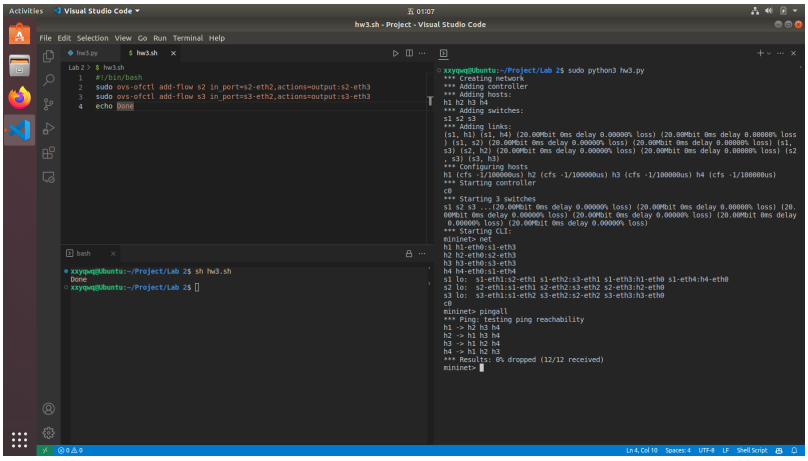
```
s1 lo:  s1-eth1:s2-eth1 s1-eth2:s3-eth1 s1-eth3:h1-eth0 s1-eth4:h4
   -eth0
s2 lo:  s2-eth1:s1-eth1 s2-eth2:s3-eth2 s2-eth3:h2-eth0
s3 lo:  s3-eth1:s1-eth2 s3-eth2:s2-eth2 s3-eth3:h3-eth0
c0
```

Then we run the shell script as follows

```
$ sh hw3.sh
Done
```

Finally we test the connectivity between all the hosts as follows

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
```



The results show that all the packets are delivered correctly and all the hosts are connected, which means that the problem is prefectly solved.

This experiment shows that the topology of switches should be carefully designed to avoid the loop problem. Flow rules should be correctly added to the switches when necessary, promising the connectivity between the hosts.