Lab 2：Mininet

518030910283 王航宇

```
why@why-VirtualBox:/mnt/my_mnt/lab2$ sudo python hw1.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (s1, s2) (s1, s3) (s2, h2) (s3, h3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
Dumping host connections
h1 h1-eth0:s1-eth4
h2 h2-eth0:s2-eth2
h3 h3-eth0:s3-eth2
h4 h4-eth0:s1-eth3
Testing network connectivity
*** 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)
Testing bandwidth
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['24.4 Gbits/sec', '24.5 Gbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['25.1 Gbits/sec', '25.1 Gbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h4
.*** Results: ['21.5 Gbits/sec', '21.5 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 6 links
......
*** Stopping 3 switches
s1 s2 s3
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
why@why-VirtualBox:/mnt/my_mnt/lab2$
```

1.

2.

```
why@why-VirtualBox:/mnt/my_mnt/lab2$ sudo python hw2.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (s1, s2) (s1, s3) (s2, h2) (s3, h3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
Dumping host connections
h1 h1-eth0:s1-eth4
h2 h2-eth0:s2-eth2
h3 h3-eth0:s3-eth2
h4 h4-eth0:s1-eth3
Testing network connectivity
*** 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)
Testing bandwidth
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['17.9 Gbits/sec', '17.9 Gbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h3
.*** Results: ['25.2 Gbits/sec', '25.2 Gbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['18.6 Gbits/sec', '18.6 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 6 links
......
*** Stopping 3 switches
s1 s2 s3
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
why@why-VirtualBox:/mnt/my_mnt/lab2$
```

3.

Creat the topology:

```
why@why-VirtualBox:/mnt/my_mnt/lab2$ sudo python hw3.py
[sudo] why 的密码:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h1) (s1, h4) (s1, s2) (s1, s3) (s2, h2) (s2, s3) (s3, h3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth4
h2 h2-eth0:s2-eth2
h3 h3-eth0:s3-eth2
h4 h4-eth0:s1-eth3
s1 lo:  s1-eth1:s2-eth1 s1-eth2:s3-eth1 s1-eth3:h4-eth0 s1-eth4:h1-eth0
s2 lo:  s2-eth1:s1-eth1 s2-eth2:h2-eth0 s2-eth3:s3-eth3
s3 lo:  s3-eth1:s1-eth2 s3-eth2:h3-eth0 s3-eth3:s2-eth3
c0
```

"pingall" finds that the connections between all the hosts are disconnected.

```
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)
```

Use the command "sh hw3.sh" and pingall again:

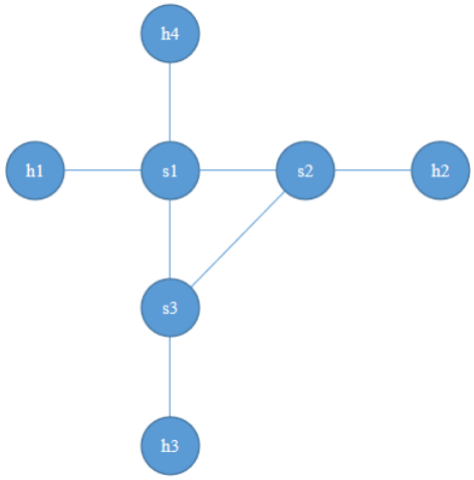
```
why@why-VirtualBox:/mnt/my_mnt/lab2$ sudo sh hw3.sh
why@why-VirtualBox:/mnt/my_mnt/lab2$
```

```
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)
mininet>
```

Then all connections between the hosts are connected.

The ovs-ofctl commands I used are as follows.

```
#!/bin/sh
ovs-ofctl add-flow s2 "in_port=s2-eth3,actions=output:s2-eth2"
ovs-ofctl add-flow s3 "in_port=s3-eth3,actions=output:s3-eth2"
```



As the picture shows, no pair of hosts can connect with each other after adding another link between s2 and s3. That's because the added link causes a loop(s1,s2,s3) in the network. When the switches are not configured, they will broadcast the packets to check the target hosts' connectivity. Because of the loop, these packets will continue in the loop and never reach the hosts.

So I use the commands to add flow rules on switches s2 and s3. When a packet goes through the added link, it will be transmitted to the neighbor host instead of being broadcast in the loop.

```
ovs-ofctl add-flow s2 "in_port=s2-eth3,actions=output:s2-eth2"
ovs-ofctl add-flow s3 "in_port=s3-eth3,actions=output:s3-eth2"
```