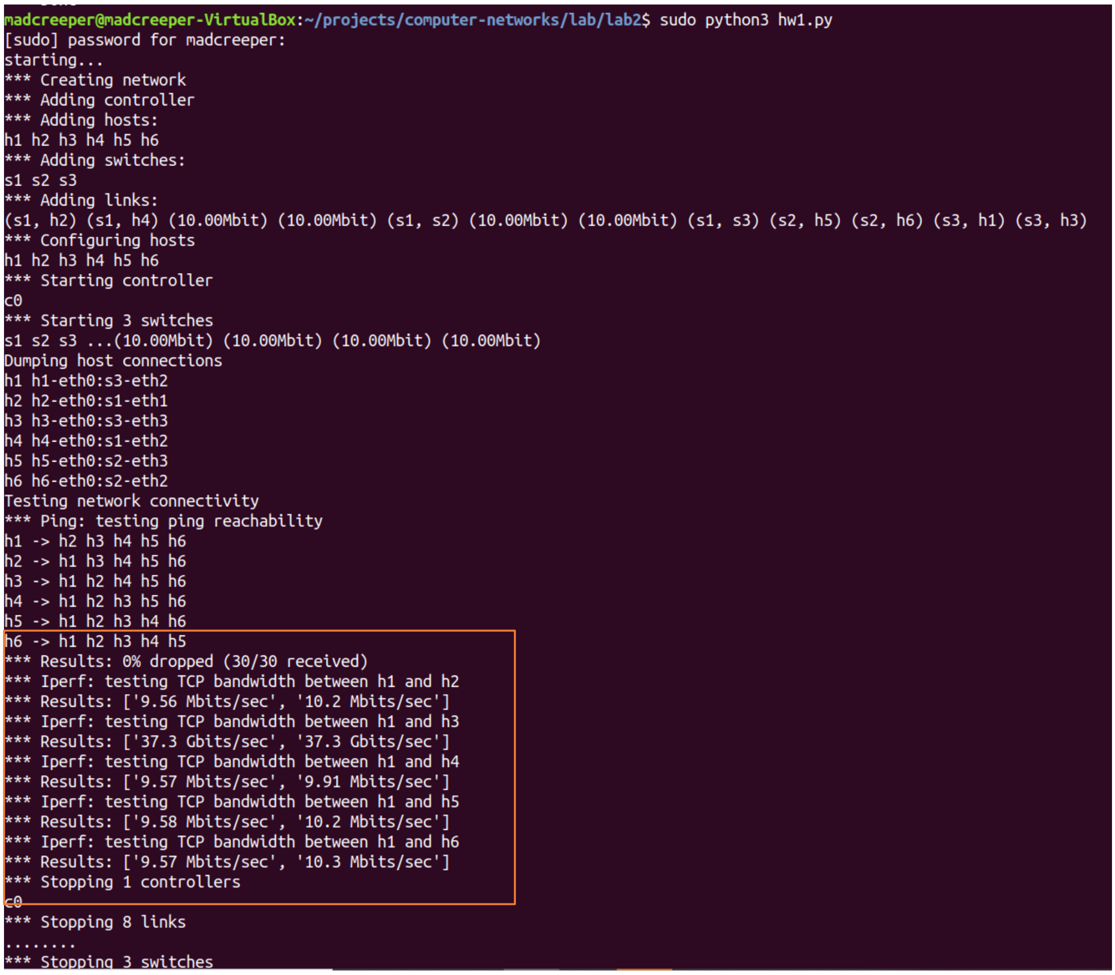
Lab2 Mininet

孙济宸 520030910016

1.

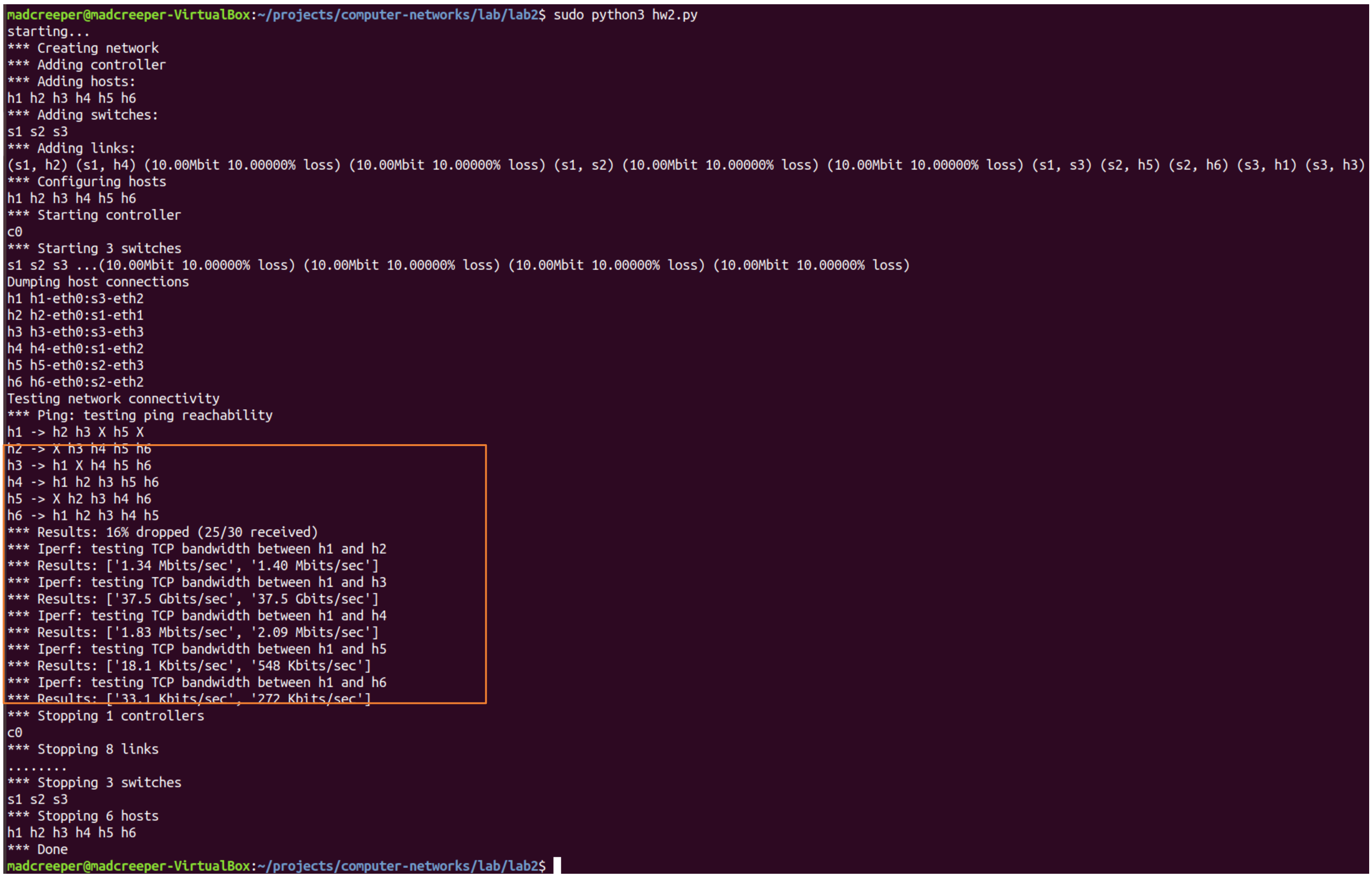
Every pair is limited by 10Mbps bandwidth, except h1-h3.



```
madcreeper@madcreeper-VirtualBox:~/projects/computer-networks/lab/lab2$ sudo python3 hw1.py
[sudo] password for madcreeper:
starting...
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h2) (s1, h4) (10.00Mbit) (10.00Mbit) (s1, s2) (10.00Mbit) (10.00Mbit) (s1, s3) (s2, h5) (s2, h6) (s3, h1) (s3, h3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(10.00Mbit) (10.00Mbit) (10.00Mbit) (10.00Mbit)
Dumping host connections
h1 h1-eth0:s3-eth2
h2 h2-eth0:s1-eth1
h3 h3-eth0:s3-eth3
h4 h4-eth0:s1-eth2
h5 h5-eth0:s2-eth3
h6 h6-eth0:s2-eth2
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.56 Mbits/sec', '10.2 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['37.3 Gbits/sec', '37.3 Gbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['9.57 Mbits/sec', '9.91 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['9.58 Mbits/sec', '10.2 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h6
*** Results: ['9.57 Mbits/sec', '10.3 Mbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 8 links
........
*** Stopping 3 switches
```

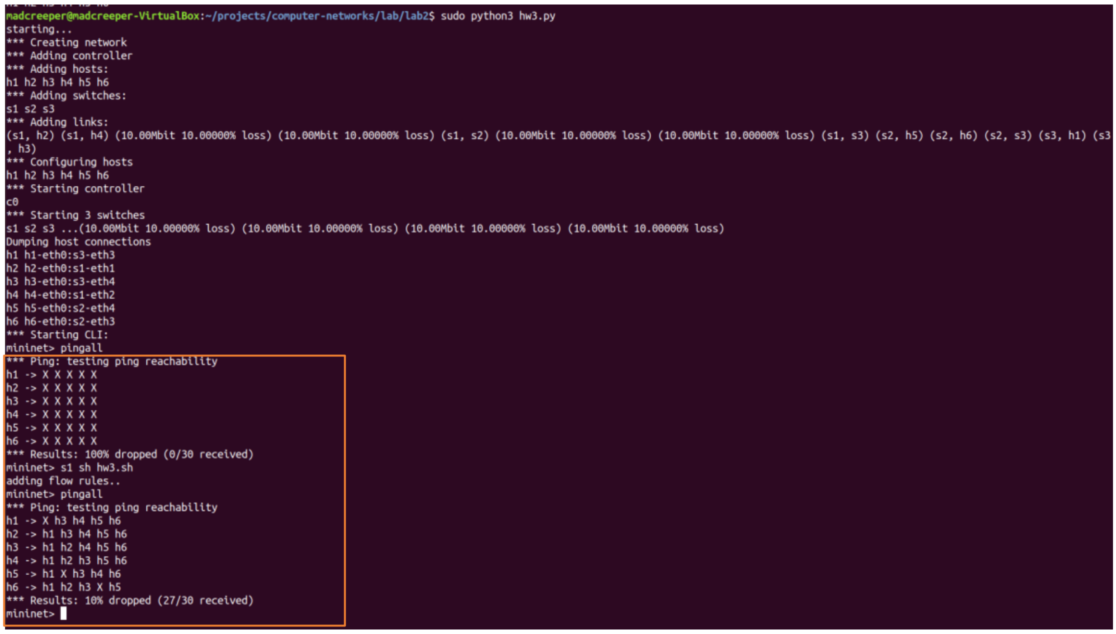
2.

Iperf shows data transfer passing links with packet loss is significantly slower, only h1-h3 is unaffected.



```
madcreeper@madcreeper-VirtualBox:~/projects/computer-networks/lab/lab2$ sudo python3 hw2.py
starting...
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h2) (s1, h4) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (s1, s2) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (s1, s3) (s2, h5) (s2, h6) (s3, h1) (s3, h3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss)
Dumping host connections
h1 h1-eth0:s3-eth2
h2 h2-eth0:s1-eth1
h3 h3-eth0:s3-eth3
h4 h4-eth0:s1-eth2
h5 h5-eth0:s2-eth3
h6 h6-eth0:s2-eth2
Testing network connectivity
*** Ping: testing ping reachability
h1 -> h2 h3 X h5 X
h2 -> X h3 h4 h5 h6
h3 -> h1 X h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> X h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 16% dropped (25/30 received)
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['1.34 Mbits/sec', '1.40 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['37.5 Gbits/sec', '37.5 Gbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['1.83 Mbits/sec', '2.09 Mbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h5
*** Results: ['18.1 Kbits/sec', '548 Kbits/sec']
*** Iperf: testing TCP bandwidth between h1 and h6
*** Results: ['33.1 Kbits/sec', '272 Kbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 8 links
........
*** Stopping 3 switches
s1 s2 s3
*** Stopping 6 hosts
h1 h2 h3 h4 h5 h6
*** Done
madcreeper@madcreeper-VirtualBox:~/projects/computer-networks/lab/lab2$ 
```

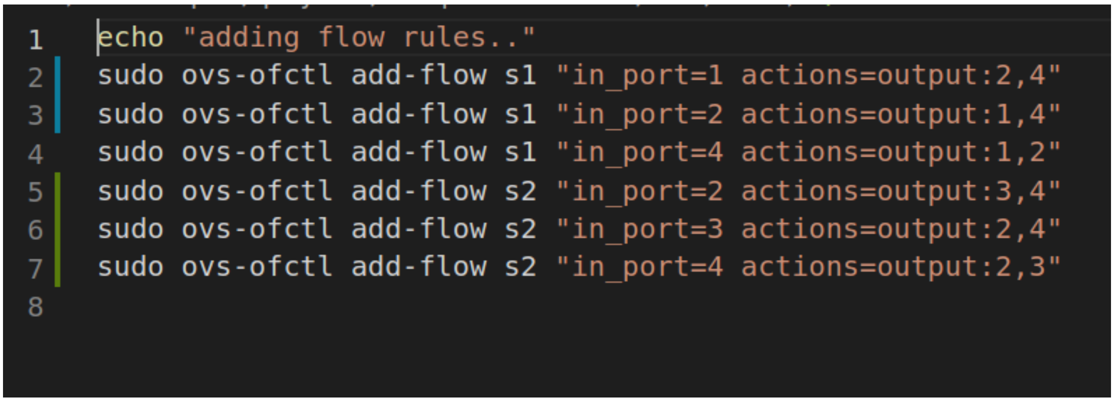
3.



```
madcreeper@madcreeper-VirtualBox:~/projects/computer-networks/lab/lab2$ sudo python3 hw3.py
starting...
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, h2) (s1, h4) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (s1, s2) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (s1, s3) (s2, h5) (s2, h6) (s2, s3) (s3, h1) (s3
, h3)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...(10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss) (10.00Mbit 10.00000% loss)
Dumping host connections
h1 h1-eth0:s3-eth3
h2 h2-eth0:s1-eth1
h3 h3-eth0:s3-eth4
h4 h4-eth0:s1-eth2
h5 h5-eth0:s2-eth4
h6 h6-eth0:s2-eth3
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X X X X
*** Results: 100% dropped (0/30 received)
mininet> s1 sh hw3.sh
adding flow rules..
mininet> pingall
*** Ping: testing ping reachability
h1 -> X h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 X h3 h4 h6
h6 -> h1 h2 h3 X h5
*** Results: 10% dropped (27/30 received)
mininet> 
```

After adding the new link, pingall command shows all host pairs are unreachable. The reason: The new link generates a loop in the network, and without flow control the traffic can get stuck in the loop forever and never reach destination.

The solution: I added flow rules to s1 and s2 so that the traffic no longer loops. You can see after adding flow rules, the network now works as intended (packet drop still make some seem unreachable, just like in 2.).



```
echo "adding flow rules.."
sudo ovs-ofctl add-flow s1 "in_port=1 actions=output:2,4"
sudo ovs-ofctl add-flow s1 "in_port=2 actions=output:1,4"
sudo ovs-ofctl add-flow s1 "in_port=4 actions=output:1,2"
sudo ovs-ofctl add-flow s2 "in_port=2 actions=output:3,4"
sudo ovs-ofctl add-flow s2 "in_port=3 actions=output:2,4"
sudo ovs-ofctl add-flow s2 "in_port=4 actions=output:2,3"
```