

Lab2

翁牧言 521030910007

1

hw1.py

```
#!/usr/bin/env python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                  build=False,
                  ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6633)

    info( '*** Add switches\n')
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch, dpid='0000000000000001')
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch, dpid='0000000000000002')
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch, dpid='0000000000000003')

    info( '*** Add hosts\n')
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)

    info( '*** Add links\n')
    net.addLink(h1, s1)
    net.addLink(s1, s2, bw=20)
    net.addLink(s2, h2)
    net.addLink(h4, s1)
    net.addLink(s1, s3, bw=20)
```

```

net.addLink(h3, s3)

info( '*** Starting network\n')
net.build()
info( '*** Starting controllers\n')
for controller in net.controllers:
    controller.start()

info( '*** Starting switches\n')
net.get('s1').start([c0])
net.get('s2').start([c0])
net.get('s3').start([c0])

info( '*** Post configure switches and hosts\n')

CLI(net)
net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()

```

The iperf3 outputs

```

wmy@wmy-VirtualBox: ~/mininet/examples
File Edit View Search Terminal Help
*** Starting network
*** Configuring hosts
h1 h2 h3 h4
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
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> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['56.0 Gbits/sec', '56.0 Gbits/sec']
mininet> iperf h1 h3
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['48.5 Gbits/sec', '48.7 Gbits/sec']
mininet> iperf h1 h4
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['50.9 Gbits/sec', '51.0 Gbits/sec']
mininet>

```

2

hw2.py

```

#!/usr/bin/env python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController

```

```

from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                  build=False,
                  ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                        controller=RemoteController,
                        ip='127.0.0.1',
                        protocol='tcp',
                        port=6633)

    info( '*** Add switches\n')
    s1 = net.addSwitch('s1', cls=OVSKernelSwitch, dpid='0000000000000001')
    s2 = net.addSwitch('s2', cls=OVSKernelSwitch, dpid='0000000000000002')
    s3 = net.addSwitch('s3', cls=OVSKernelSwitch, dpid='0000000000000003')

    info( '*** Add hosts\n')
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
    h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)
    h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)

    info( '*** Add links\n')
    net.addLink(h1, s1)
    net.addLink(s1, s2, bw=20, loss=5)
    net.addLink(s2, h2)
    net.addLink(h4, s1)
    net.addLink(s1, s3, bw=20, loss=5)
    net.addLink(h3, s3)

    info( '*** Starting network\n')
    net.build()
    info( '*** Starting controllers\n')
    for controller in net.controllers:
        controller.start()

    info( '*** Starting switches\n')
    net.get('s1').start([c0])
    net.get('s2').start([c0])
    net.get('s3').start([c0])

    info( '*** Post configure switches and hosts\n')

    CLI(net)
    net.stop()

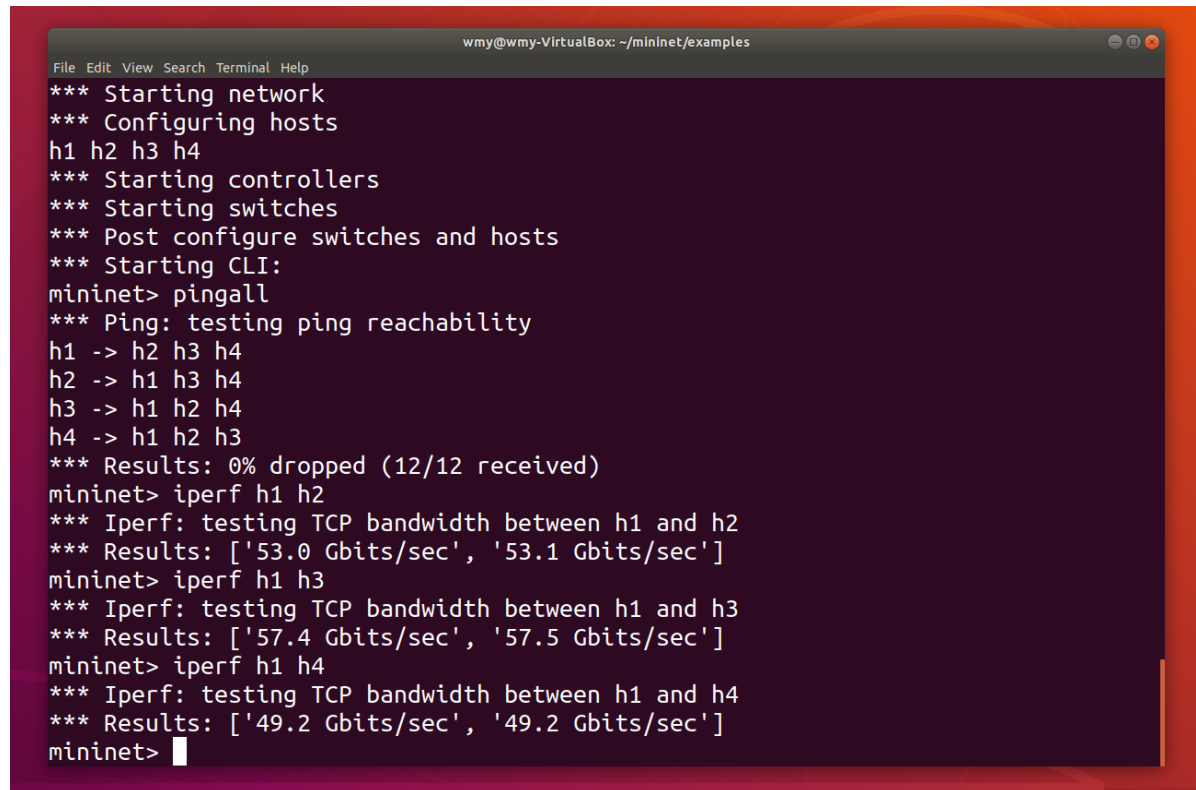
```

```

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()

```

the iperf3 outputs



```

wmy@wmy-VirtualBox: ~/mininet/examples
File Edit View Search Terminal Help
*** Starting network
*** Configuring hosts
h1 h2 h3 h4
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
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> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['53.0 Gbits/sec', '53.1 Gbits/sec']
mininet> iperf h1 h3
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['57.4 Gbits/sec', '57.5 Gbits/sec']
mininet> iperf h1 h4
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['49.2 Gbits/sec', '49.2 Gbits/sec']
mininet>

```

3

```

#!/usr/bin/env python

from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UsersSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call

def myNetwork():

    net = Mininet( topo=None,
                  build=False,
                  ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    c0=net.addController(name='c0',
                        controller=RemoteController,
                        ip='127.0.0.1',

```

```

        protocol='tcp',
        port=6633)

info( '*** Add switches\n')
s1 = net.addSwitch('s1', cls=OVSKernelSwitch, dpid='0000000000000001')
s2 = net.addSwitch('s2', cls=OVSKernelSwitch, dpid='0000000000000002')
s3 = net.addSwitch('s3', cls=OVSKernelSwitch, dpid='0000000000000003')

info( '*** Add hosts\n')
h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)
h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
h3 = net.addHost('h3', cls=Host, ip='10.0.0.3', defaultRoute=None)
h4 = net.addHost('h4', cls=Host, ip='10.0.0.4', defaultRoute=None)

info( '*** Add links\n')
net.addLink(h1, s1)
net.addLink(s1, s2, bw=20, loss=5)
net.addLink(s2, h2)
net.addLink(h4, s1)
net.addLink(s1, s3, bw=20, loss=5)
net.addLink(h3, s3)
net.addLink(s2, s3)
info( '*** Starting network\n')
net.build()
info( '*** Starting controllers\n')
for controller in net.controllers:
    controller.start()

info( '*** Starting switches\n')
net.get('s1').start([c0])
net.get('s2').start([c0])
net.get('s3').start([c0])

info( '*** Post configure switches and hosts\n')

CLI(net)
net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()

```

after using "pingall"

```
wmy@wmy-VirtualBox: ~/mininet/examples
File Edit View Search Terminal Help
controllers2.py  intfoptions.py  nat.py          treeping64.py
controllers.py   limit.py        numberedports.py  vlanhost.py
controlnet.py    linearbandwidth.py  popenpoll.py
wmy@wmy-VirtualBox:~/mininet/examples$ vim hw3.py
wmy@wmy-VirtualBox:~/mininet/examples$ sudo python hw3.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h1 h2 h3 h4
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
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)
mininet>
```

hw3.sh

```
#!/bin/sh
sudo ovs-ofctl add-flow s1 in_port=1,actions=2,3,4
sudo ovs-ofctl add-flow s1 in_port=2,actions=1,3,4
sudo ovs-ofctl add-flow s1 in_port=3,actions=2,1,4
sudo ovs-ofctl add-flow s1 in_port=4,actions=2,3,1
sudo ovs-ofctl add-flow s2 in_port=1,actions=2,3
sudo ovs-ofctl add-flow s2 in_port=2,actions=1,3
sudo ovs-ofctl add-flow s2 in_port=3,actions=2,1
sudo ovs-ofctl add-flow s3 in_port=1,actions=2,3
sudo ovs-ofctl add-flow s3 in_port=2,actions=1,3
sudo ovs-ofctl add-flow s3 in_port=3,actions=2,1
```

the outputs

```
File Edit View Search Terminal Help
wmy@wmy-VirtualBox: ~/mininet/examples
" actions=output:"s3-eth1",output:"s3-eth2"
cookie=0x0, duration=20.659s, table=0, n_packets=0, n_bytes=0, in_port="s3-eth2"
" actions=output:"s3-eth1",output:"s3-eth3"
cookie=0x0, duration=9.963s, table=0, n_packets=0, n_bytes=0, in_port="s3-eth1"
" actions=output:"s3-eth2",output:"s3-eth3"
wmy@wmy-VirtualBox:~$ sudo ovs-ofctl dump-flows s2
cookie=0x0, duration=63.795s, table=0, n_packets=0, n_bytes=0, in_port="s2-eth1"
" actions=output:"s2-eth2",output:"s2-eth3"
cookie=0x0, duration=54.975s, table=0, n_packets=0, n_bytes=0, in_port="s2-eth2"
" actions=output:"s2-eth1",output:"s2-eth3"
cookie=0x0, duration=48.118s, table=0, n_packets=0, n_bytes=0, in_port="s2-eth3"
" actions=output:"s2-eth1",output:"s2-eth2"
wmy@wmy-VirtualBox:~$ sudo ovs-ofctl dump-flows s1
cookie=0x0, duration=131.892s, table=0, n_packets=0, *** Starting switches
1" actions=output:"s1-eth2",output:"s1-eth3",output:"s *** Post configure switches and hosts
cookie=0x0, duration=124.882s, table=0, n_packets=0, *** Starting CLI:
2" actions=output:"s1-eth1",output:"s1-eth3",output:"s mininet> pingall
cookie=0x0, duration=117.224s, table=0, n_packets=0, *** Ping: testing ping reachability
3" actions=output:"s1-eth1",output:"s1-eth2",output:"s
cookie=0x0, duration=97.479s, table=0, n_packets=0, nh1 -> X X X
" actions=output:"s1-eth1",output:"s1-eth2",output:"s1 h2 -> X X X
wmy@wmy-VirtualBox:~$ cd mininet/examples h3 -> X X X
wmy@wmy-VirtualBox:~/mininet/examples$ sudo sh hw3.sh h4 -> X X X
wmy@wmy-VirtualBox:~/mininet/examples$ *** Results: 100% dropped (0/12 received)
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>
```

After we add another link between s2 and s3, the hosts' connectivity is broken.

So we use OVS commands to set up flow rules and restore the connectivity.