

## ASSIGNMENT-6

Topology Code :

```
from mininet.net import Mininet
from mininet.node import Controller, OVSController
from mininet.cli import CLI
from mininet.link import TCLink
from mininet.log import setLogLevel, info

net = Mininet()

h1 = net.addHost( 'h1' )
h2 = net.addHost( 'h2' )
h3 = net.addHost( 'h3' )
h4 = net.addHost( 'h4' )

s1 = net.addSwitch( 's1' )
s2 = net.addSwitch( 's2' )

net.addLink( h1, s1, bw=20, delay='10ms' )
net.addLink( h2, s1, bw=20, delay='10ms' )
net.addLink( h3, s2, bw=20, delay='10ms' )
net.addLink( h4, s2, bw=20, delay='10ms' )
net.addLink( s1, s2, bw=50, delay='10ms' )

net.start()
CLI( net )
net.stop()
```

## Server Code :

```
import socket

HOST = '10.0.0.'
PORT = 5000

hostlist = ['1', '2']
hst = input("Which host to serve? [1, 2]    ")

while (hst not in hostlist):
    hst = input("Which host to serve? [1, 2]    ")

HOST += hst
print("Listening at ", HOST, ":", PORT)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        while True:
            data = conn.recv(1024).decode()
            print("\n[Client] : ", data, "\n")
            message = input("Type a message: ")
            conn.sendall(message.encode())
```

## Client Code :

```
import socket

HOST = '10.0.0.'
PORT = 5000

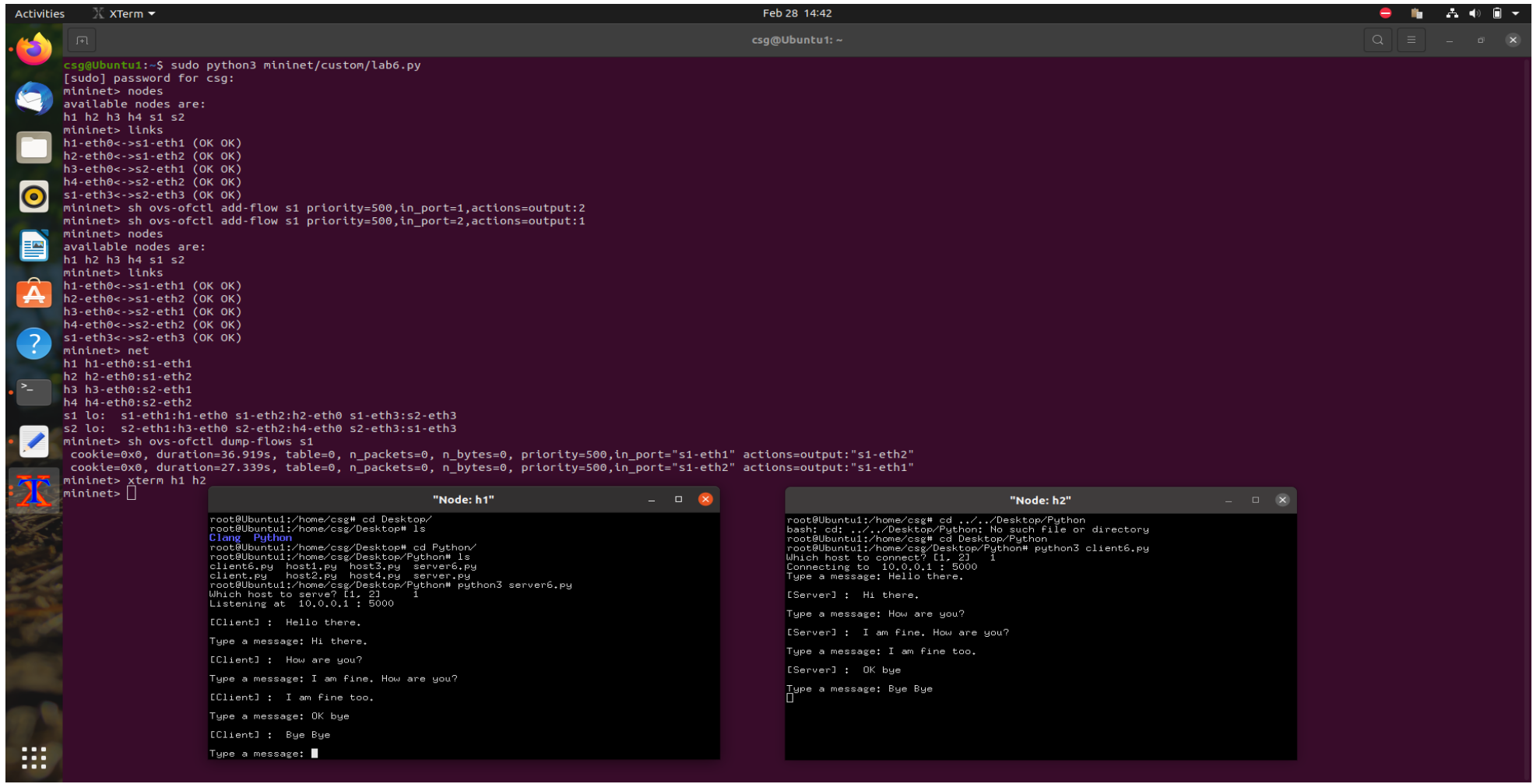
hostlist = ['1', '2']
hst = input("Which host to connect? [1, 2] ")

while (hst not in hostlist):
    hst = input("Which host to connect? [1, 2] ")

HOST += hst
print("Connecting to ", HOST, ":", PORT)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    while True:
        message = input("Type a message: ")
        s.sendall(message.encode())
        data = s.recv(1024).decode()
        print("\n[Server] : ", data, "\n")
```

## Output Screenshot with terminal code and Xterms for H1 and H2 :



The screenshot displays a terminal window and two xterm windows. The terminal window shows the execution of a Mininet script, which sets up a network topology with four hosts (h1, h2, h3, h4) and three switches (s1, s2, s3). It then adds OpenFlow rules to the switches and displays the network topology. The two xterm windows, titled "Node: h1" and "Node: h2", show a chat application running on these nodes. The application allows users to connect to a server and chat with other nodes.

```
csg@Ubuntu1:~$ sudo python3 mininet/custom/lab6.py
[sudo] password for csg:
mininet> nodes
available nodes are:
h1 h2 h3 h4 s1 s2
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s2-eth1 (OK OK)
h4-eth0<->s2-eth2 (OK OK)
s1-eth3<->s2-eth3 (OK OK)
mininet> sh ovs-ofctl add-flow s1 priority=500,in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 priority=500,in_port=2,actions=output:1
mininet> nodes
available nodes are:
h1 h2 h3 h4 s1 s2
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s2-eth1 (OK OK)
h4-eth0<->s2-eth2 (OK OK)
s1-eth3<->s2-eth3 (OK OK)
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s2-eth1
h4 h4-eth0:s2-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth3
s2 lo: s2-eth1:h3-eth0 s2-eth2:h4-eth0 s2-eth3:s1-eth3
mininet> sh ovs-ofctl dump-flows s1
cookie=0x0, duration=36.919s, table=0, n_packets=0, n_bytes=0, priority=500,in_port="s1-eth1" actions=output:"s1-eth2"
cookie=0x0, duration=27.339s, table=0, n_packets=0, n_bytes=0, priority=500,in_port="s1-eth2" actions=output:"s1-eth1"
mininet> xterm h1 h2
mininet>
```

**"Node: h1"**

```
root@Ubuntu1:/home/csg# cd Desktop/
root@Ubuntu1:/home/csg/Desktop# ls
Client Python
root@Ubuntu1:/home/csg/Desktop# cd Python/
root@Ubuntu1:/home/csg/Desktop/Python# ls
client6.py host1.py host3.py server6.py
client.py host2.py host4.py server.py
root@Ubuntu1:/home/csg/Desktop/Python# python3 server6.py
Which host to serve? [1, 2] 1
Listening at 10.0.0.1 : 5000

[Client] : Hello there.

Type a message: Hi there.

[Client] : How are you?

Type a message: I am fine. How are you?

[Client] : I am fine too.

Type a message: OK bye

[Client] : Bye Bye

Type a message:
```

**"Node: h2"**

```
root@Ubuntu1:/home/csg# cd ../../Desktop/Python
bash: cd: ../../Desktop/Python: No such file or directory
root@Ubuntu1:/home/csg# cd Desktop/Python
root@Ubuntu1:/home/csg/Desktop/Python# python3 client6.py
Which host to connect? [1, 2] 1
Connecting to 10.0.0.1 : 5000
Type a message: Hello there.

[Server] : Hi there.

Type a message: How are you?

[Server] : I am fine. How are you?

Type a message: I am fine too.

[Server] : OK bye

Type a message: Bye Bye
```