

# lab 1 Report and Assignment

## Environment Setting

using following command to install the tools

1. `sudo apt install net-tools`
2. `sudo apt install openssh-server`
3. `sudo apt install traceroute`
4. `sudo apt install git`
5. `sudo apt-get install -y iperf3`
6. `sudo apt install wireshark`

## Usage

### net-tools

This includes `arp`, `ifconfig`, `netstat`, `route`.

1. `arp` command manipulates or displays the the `MAC` table

```
austinguish@austinguish-GL502VSK ~-> arp
地址          类型      硬件地址      标志  Mask      接口
192.168.31.18  ether     9c:28:f7:96:72:19  C      Mask      enp4s
0
XiaoQiang     ether     50:64:2b:e0:36:92  C      enp4s
0
```

2. `ifconfig` command is used to configure the kernel-resident network interfaces. This command is also used to assign the IP address and netmask to an interface or to enable or disable a given interface.

```
austinguish@austinguish-GL502VSK ~-> ifconfig
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.31.27 netmask 255.255.255.0 broadcast 192.168.31.255
    inet6 fe80::fbb5:586b:7844:d80d prefixlen 64 scopeid 0x20<link>
    ether 60:45:cb:2e:54:92 txqueuelen 1000 (以太网)
    RX packets 1647043 bytes 2278594730 (2.2 GB)
    RX errors 0 dropped 3 overruns 0 frame 0
    TX packets 844454 bytes 72300169 (72.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 2696113 bytes 2237681290 (2.2 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2696113 bytes 2237681290 (2.2 GB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether f4:8c:50:6c:c9:9c txqueuelen 1000 (以太网)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. `netstat` In computing, `netstat` (**\*network statistics\***) is a command-line network utility that displays network connections for Transmission Control Protocol (both incoming and outgoing), routing tables], and a number of network interface (network interface controller and network protocol statistics.

some useful commands:

```
1 netstat -natu | grep 'ESTABLISHED'
2 # Display all ESTABLISHED TCP connections
3 netstat -an | grep 'LISTEN'
4 # Display all listen IP:PORT
5 netstat -anlp | grep PID9
6 # Port Number used by PID
7 netstat -r
8 # Equal to "route" list route table
9 netstat -i
10 # Network interfaces
11 netstat -c
12 # Continuous listening
13
```

```
austinguish@austinguish-GL502VSK ~-> netstat -an | grep 'LISTEN'
tcp        0      0 127.0.0.1:15490      0.0.0.0:*             LISTEN
tcp        0      0 127.0.0.53:53        0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:22           0.0.0.0:*             LISTEN
tcp        0      0 127.0.0.1:631        0.0.0.0:*             LISTEN
tcp        0      0 127.0.0.1:8888       0.0.0.0:*             LISTEN
tcp        0      0 127.0.0.1:1088       0.0.0.0:*             LISTEN
tcp6       0      0 :::1716              :::*                   LISTEN
tcp6       0      0 :::1717              :::*                   LISTEN
tcp6       0      0 :::22                :::*                   LISTEN
tcp6       0      0 :::1:631             :::*                   LISTEN
```

## SSH

using `ssh -port username@host` to establish ssh connection between local and remote host.

Tips:

1. The default port for SSH is 22.
2. If want to access the computer in dorm. Should set the DMZ service in the router.

using JuiceSSH App on Android to access the laptop:

```
下午3:54 [icons] 4.1K/s [icons] 100%
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Sep 15 15:53:55 2020 from 192.168.31.18
unicode_start skipped on /dev/pts/2
Welcome to fish, the friendly interactive shell
Type 'help' for instructions on how to use fish
austinguish@austinguish-GL502VSK ~-> uname -r
5.4.0-47-generic
austinguish@austinguish-GL502VSK ~-> █
```

## Traceroute

```
traceroute [options] host_address
```

It will print the route a data packet takes to reach the host address.

eg:

The `-I` option means using ICMP protocol

```
austinguish@austinguish-GL502VSK ~ [2]> traceroute -I www.baidu.com
traceroute to www.baidu.com (182.61.200.6), 30 hops max, 60 byte packets
 1 XiaoQiang (192.168.31.1)  0.401 ms  0.427 ms  0.509 ms
 2 59.78.20.254 (59.78.20.254)  2.081 ms  2.395 ms  2.796 ms
 3 10.21.17.249 (10.21.17.249)  1.350 ms  1.361 ms  1.489 ms
 4 10.3.2.5 (10.3.2.5)  2.069 ms  2.639 ms  3.429 ms
 5 10.255.38.58 (10.255.38.58)  3.621 ms  3.630 ms  3.625 ms
 6 10.255.38.1 (10.255.38.1)  2.426 ms  1.765 ms  1.839 ms
 7 10.255.38.254 (10.255.38.254)  1.216 ms  1.221 ms  1.218 ms
 8 * * *
 9 101.4.115.105 (101.4.115.105)  2.226 ms  2.243 ms  2.429 ms
10 101.4.117.30 (101.4.117.30)  21.041 ms  20.857 ms  20.867 ms
11 * * *
12 101.4.112.69 (101.4.112.69)  27.366 ms  27.168 ms  27.653 ms
13 219.224.103.38 (219.224.103.38)  28.307 ms  28.333 ms  28.558 ms
14 101.4.130.34 (101.4.130.34)  62.974 ms  67.182 ms  63.297 ms
15 182.61.252.210 (182.61.252.210)  58.143 ms  59.105 ms  57.915 ms
16 * * *
17 * * *
18 * * *
19 * * *
20 182.61.200.6 (182.61.200.6)  63.790 ms  57.727 ms  58.024 ms
```

## iPerf3

used to test the bandwidth

```
iperf3 -s -p establish the server with selected port
```

```
iperf3 -c hostIP -p connect to the server and running perf
```

```
austinguish@austinguish-GL502VSK ~-> iperf3 -c localhost
Connecting to host localhost, port 5201
[ 5] local 127.0.0.1 port 57808 connected to 127.0.0.1 port 5201
[ ID] Interval           Transfer     Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec   5.89 GBytes  50.6 Gbits/sec    0   1.37 MBytes
[ 5]  1.00-2.00    sec   5.90 GBytes  50.7 Gbits/sec    0   1.37 MBytes
[ 5]  2.00-3.00    sec   5.92 GBytes  50.9 Gbits/sec    4   1.06 MBytes
[ 5]  3.00-4.00    sec   5.94 GBytes  51.0 Gbits/sec    0   1.19 MBytes
[ 5]  4.00-5.00    sec   5.94 GBytes  51.0 Gbits/sec    0   1.50 MBytes
[ 5]  5.00-6.00    sec   5.90 GBytes  50.6 Gbits/sec    0   1.50 MBytes
[ 5]  6.00-7.00    sec   5.89 GBytes  50.6 Gbits/sec    0   1.50 MBytes
[ 5]  7.00-8.00    sec   5.87 GBytes  50.4 Gbits/sec    0   1.50 MBytes
[ 5]  8.00-9.00    sec   5.88 GBytes  50.5 Gbits/sec    0   1.62 MBytes
[ 5]  9.00-10.00   sec   5.88 GBytes  50.5 Gbits/sec    0   1.62 MBytes
- - - - -
[ ID] Interval           Transfer     Bitrate      Retr
[ 5]  0.00-10.00    sec   59.0 GBytes  50.7 Gbits/sec    4
[ 5]  0.00-10.00    sec   59.0 GBytes  50.7 Gbits/sec

sender
receiver
```

## Question in lab1

1. What protocol does "ping" and "traceroute" use?

A: ICMP protocol

```
30 7.6070... 192.168.31.27 182.61.200.6 ICMP 98 Echo (ping) request id=0x0001, se...
31 7.6355... 182.61.200.6 192.168.31.27 ICMP 98 Echo (ping) reply id=0x0001, se...
32 8.6088... 192.168.31.27 182.61.200.6 ICMP 98 Echo (ping) request id=0x0001, se...
33 8.6369... 182.61.200.6 192.168.31.27 ICMP 98 Echo (ping) reply id=0x0001, se...
34 9.6094... 192.168.31.27 182.61.200.6 ICMP 98 Echo (ping) request id=0x0001, se...
35 9.6378... 182.61.200.6 192.168.31.27 ICMP 98 Echo (ping) reply id=0x0001, se...
36 10.611... 192.168.31.27 182.61.200.6 ICMP 98 Echo (ping) request id=0x0001, se...
37 10.639... 182.61.200.6 192.168.31.27 ICMP 98 Echo (ping) reply id=0x0001, se...
38 11.612... 192.168.31.27 182.61.200.6 ICMP 98 Echo (ping) request id=0x0001, se...
39 11.640... 182.61.200.6 192.168.31.27 ICMP 98 Echo (ping) reply id=0x0001, se...
```

2. What is the average round trip time between your laptop and [www.baidu.com](http://www.baidu.com)?

A: using command `ping url -c times` to measure the RTT

```
100 packets transmitted, 97 received, 3% packet loss, time 99214ms
rtt min/avg/max/mdev = 87.502/95.226/105.022/3.251 ms
```

In the pic, the avg rtt is 95.226ms

3. What is the TCP throughput between your two VMs?

A: Using the iperf3, the throughput is `50.7Gbits/sec`

4. What is the IP address of [www.sjtu.edu.cn](http://www.sjtu.edu.cn)?

A: `ping` command returns the IP address `202.120.2.119`

```
PING www.sjtu.edu.cn (202.120.2.119) 56(84)
64 bytes from 202.120.2.119 (202.120.2.119):
64 bytes from 202.120.2.119 (202.120.2.119):
```

5. Try to find out what protocols are used for WeChat app when sending text and when video/audio chatting?

using `Ettcap` to set an ARP poison attack on the device in LAN. Captured the wechat app traffic on Android device. Monitored the transportation with Wireshark.

When typing in wechat, it uses TCP and TLS. When video/audio, it uses the UDP protocol.

## Mininet Lab

In installation, I found that, if the present directory include 'space ', the install script will throw a exception. So the correct way is to remove the space in the directory.

1. `sudo mn` open the main program of mininet

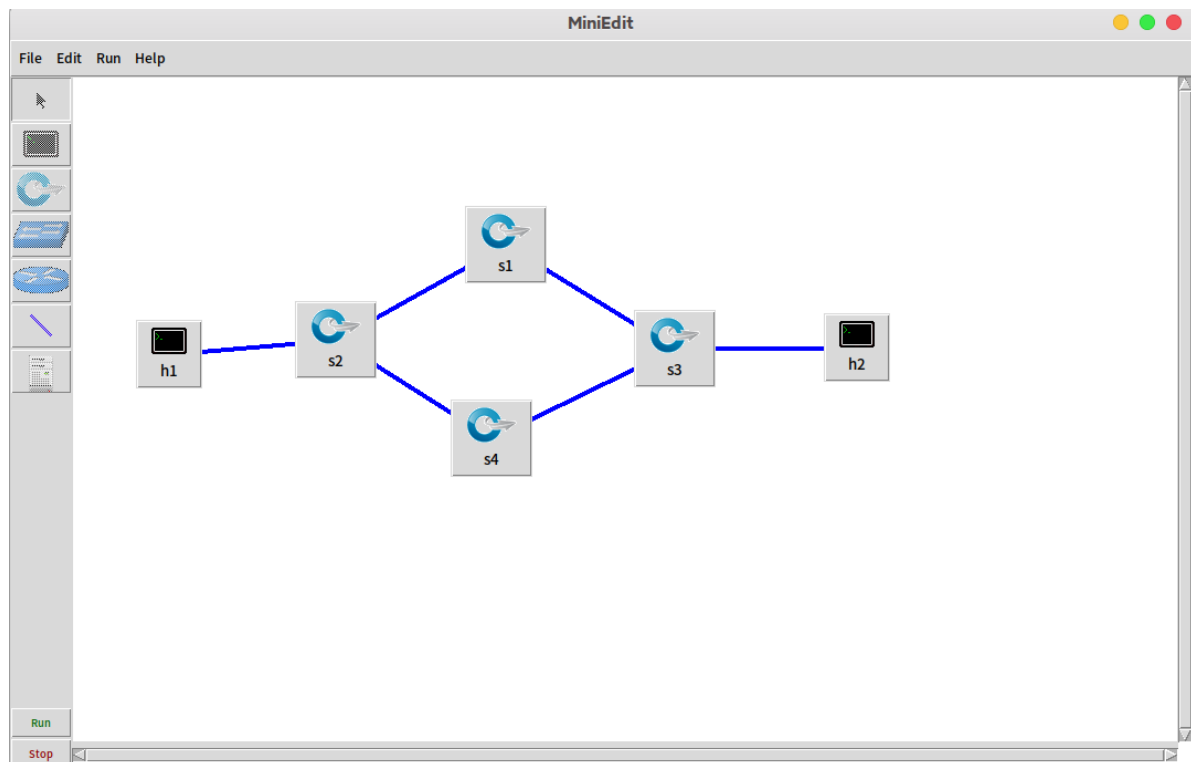
s1-eth1	_____
s1-eth2	_____
Loopback: lo	~~~~~
any	~~~~~
wlp3s0	_____
bluetooth-monitor	_____
nflog	_____
nfqueue	_____
bluetooth0	_____
ovs-system	_____
s1	_____

the `wireshark` show the interface of s1 and s2.

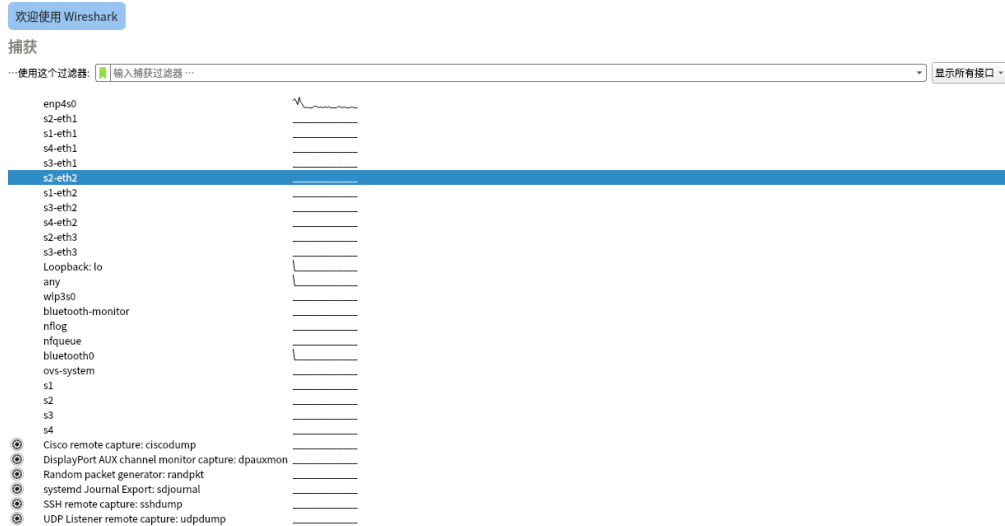
2. use GUI to setup a topology

`sudo python ./examples/miniedit.py` to start up the window. Just drag and drop and link, we can create a net topology.

Click `run` to running the emulation.



Monitoring in the wireshark



### 3. Commands in CLI

`help` display the CLI commands

`nodes` display the available nodes

`net` show the links between the devices

`device_name + command` execute the command on specify device.

`xterm hostname` open the xterm of the device

### 4. using Python script to set a topology

to set a topology like the GUI made above.

```

1  #!/usr/bin/python
2
3  """
4  Simple example of setting network and CPU parameters
5  """
6
7
8  from mininet.topo import Topo
9  from mininet.net import Mininet
10 from mininet.node import OVSBridge
11 from mininet.node import CPULimitedHost
12 from mininet.link import TCLink
13 from mininet.util import dumpNodeConnections
14 from mininet.log import setLogLevel, info
15 from mininet.cli import CLI
16
17 from sys import argv
18
19 # It would be nice if we didn't have to do this:
20 # pylint: disable=arguments-differ
21
22 class SingleSwitchTopo( Topo ):
23     def build( self ):
24         switch1 = self.addSwitch('s1',stp=True)
25         switch2 = self.addSwitch('s2',stp=True)
26         switch3 = self.addSwitch('s3',stp=True)
27         switch4 = self.addSwitch('s4',stp=True)
28         host1 = self.addHost('h1', cpu=.25)

```

```

29         host2 = self.addHost('h2', cpu=.25)
30         self.addLink(host1, switch1, bw=10, delay='5ms', loss=0,
use_htb=True)
31         self.addLink(host2, switch2, bw=10, delay='5ms', loss=0,
use_htb=True)
32         self.addLink(switch1, switch3, bw=10, delay='5ms', loss=0,
use_htb=True)
33         self.addLink(switch1, switch4, bw=10, delay='5ms', loss=0,
use_htb=True)
34         self.addLink(switch3, switch2, bw=10, delay='5ms', loss=0,
use_htb=True)
35         self.addLink(switch4, switch2, bw=10, delay='5ms', loss=0,
use_htb=True)
36
37     def Test():
38         "Create network and run simple performance test"
39         topo = SingleSwitchTopo()
40         net = Mininet( topo=topo,
41                       host=CPULimitedHost, link=TCLink,
42                       autoStaticArp=False )
43         net.start()
44         info( "Dumping host connections\n" )
45         dumpNodeConnections(net.hosts)
46         info( "Testing bandwidth between h1 and h4\n" )
47         h1, h2 = net.getNodeByName('h1', 'h2')
48         #s1, s2 = net.getNodeByName('s1', 's2')
49         CLI(net)
50         net.stop()
51
52 if __name__ == '__main__':
53     setLogLevel( 'info' )
54     # Prevent test_simpleperf from failing due to packet loss
55     Test()

```

## Homework

1. Simulate the following topology in Mininet. Set the link bandwidth for (s1,s2) and (s1,s3) as 10Mbps. Use Iperf to test the TCP throughput between every host pair.

```

h1--s1--s2--h2
  |
  s3
  |
  h3

```

iperf test result:



```

"Node: h1"
Server listening on 5201
Accepted connection from 10.0.0.2, port 45210
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 45212
[ ID] Interval      Transfer      Bitrate
[ 7] 0.00-1.00 sec  1.05 MBytes  8.79 Mbits/sec
[ 7] 1.00-2.00 sec  1.14 MBytes  9.56 Mbits/sec
[ 7] 2.00-3.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 3.00-4.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 4.00-5.00 sec  1.14 MBytes  9.56 Mbits/sec
[ 7] 5.00-6.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 6.00-7.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 7.00-8.00 sec  1.14 MBytes  9.56 Mbits/sec
[ 7] 8.00-9.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 9.00-10.00 sec 1.14 MBytes  9.56 Mbits/sec
[ 7] 10.00-10.68 sec 792 KBytes  9.58 Mbits/sec
[ ID] Interval      Transfer      Bitrate
[ 7] 0.00-10.68 sec 12.1 MBytes  9.49 Mbits/sec
Server listening on 5201

"Node: h2"
iperf Done.
root@austinguish-GL502VSK:/home/austinguish/Desktop/Computer Networks/Chapter1/
mininet/examples# iperf3 -c 10.0.0.1
Connecting to host 10.0.0.1, port 5201
[ 7] local 10.0.0.2 port 45212 connected to 10.0.0.1 port 5201
[ ID] Interval      Transfer      Bitrate      Retr      Cwnd
[ 7] 0.00-1.00 sec  2.42 MBytes  20.3 Mbits/sec  0         266 KBytes
[ 7] 1.00-2.00 sec  1.18 MBytes  9.90 Mbits/sec  0         324 KBytes
[ 7] 2.00-3.00 sec  1.43 MBytes  12.0 Mbits/sec  0         383 KBytes
[ 7] 3.00-4.00 sec  1.68 MBytes  14.1 Mbits/sec  0         441 KBytes
[ 7] 4.00-5.00 sec  954 KBytes  7.82 Mbits/sec  0         499 KBytes
[ 7] 5.00-6.00 sec  2.11 MBytes  17.7 Mbits/sec  0         557 KBytes
[ 7] 6.00-7.00 sec  1.18 MBytes  9.90 Mbits/sec  0         617 KBytes
[ 7] 7.00-8.00 sec  1.25 MBytes  10.5 Mbits/sec  0         675 KBytes
[ 7] 8.00-9.00 sec  1.25 MBytes  10.5 Mbits/sec  0         732 KBytes
[ 7] 9.00-10.00 sec 1.25 MBytes  10.5 Mbits/sec  0         792 KBytes
[ ID] Interval      Transfer      Bitrate      Retr
[ 7] 0.00-10.00 sec 14.7 MBytes  12.3 Mbits/sec  0
[ 7] 0.00-10.68 sec 12.1 MBytes  9.49 Mbits/sec
sender
receiver
iperf Done.
root@austinguish-GL502VSK:/home/austinguish/Desktop/Computer Networks/Chapter1/
mininet/examples#

```

the bandwidth is 12.3Mbits/s for sender and 9.49Mbits/s for receiver between the h1 and h2

```

"Node: h1"
Server listening on 5201
Accepted connection from 10.0.0.3, port 33790
[ 7] local 10.0.0.1 port 5201 connected to 10.0.0.3 port 33792
[ ID] Interval      Transfer      Bitrate
[ 7] 0.00-1.00 sec  1.05 MBytes  8.79 Mbits/sec
[ 7] 1.00-2.00 sec  1.14 MBytes  9.56 Mbits/sec
[ 7] 2.00-3.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 3.00-4.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 4.00-5.00 sec  1.14 MBytes  9.56 Mbits/sec
[ 7] 5.00-6.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 6.00-7.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 7.00-8.00 sec  1.14 MBytes  9.56 Mbits/sec
[ 7] 8.00-9.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 9.00-10.00 sec 1.14 MBytes  9.54 Mbits/sec
[ 7] 10.00-10.68 sec 793 KBytes  9.59 Mbits/sec
[ ID] Interval      Transfer      Bitrate
[ 7] 0.00-10.68 sec 12.1 MBytes  9.49 Mbits/sec
Server listening on 5201

"Node: h3"
root@austinguish-GL502VSK:/home/austinguish/Desktop/Computer Networks/Chapter1/
mininet/examples# iperf3 -c 10.0.0.1
Connecting to host 10.0.0.1, port 5201
[ 7] local 10.0.0.3 port 33792 connected to 10.0.0.1 port 5201
[ ID] Interval      Transfer      Bitrate      Retr      Cwnd
[ 7] 0.00-1.00 sec  2.42 MBytes  20.3 Mbits/sec  0         266 KBytes
[ 7] 1.00-2.00 sec  1.18 MBytes  9.90 Mbits/sec  0         324 KBytes
[ 7] 2.00-3.00 sec  1.43 MBytes  12.0 Mbits/sec  0         383 KBytes
[ 7] 3.00-4.00 sec  1.68 MBytes  14.1 Mbits/sec  0         441 KBytes
[ 7] 4.00-5.00 sec  954 KBytes  7.82 Mbits/sec  0         499 KBytes
[ 7] 5.00-6.00 sec  2.11 MBytes  17.7 Mbits/sec  0         557 KBytes
[ 7] 6.00-7.00 sec  1.18 MBytes  9.90 Mbits/sec  0         617 KBytes
[ 7] 7.00-8.00 sec  1.25 MBytes  10.5 Mbits/sec  0         675 KBytes
[ 7] 8.00-9.00 sec  1.25 MBytes  10.5 Mbits/sec  0         732 KBytes
[ 7] 9.00-10.00 sec 1.25 MBytes  10.5 Mbits/sec  0         792 KBytes
[ ID] Interval      Transfer      Bitrate      Retr
[ 7] 0.00-10.00 sec 14.7 MBytes  12.3 Mbits/sec  0
[ 7] 0.00-10.68 sec 12.1 MBytes  9.49 Mbits/sec
sender
receiver
iperf Done.
root@austinguish-GL502VSK:/home/austinguish/Desktop/Computer Networks/Chapter1/
mininet/examples#

```

the bandwidth is 12.3Mbits/s for sender and 9.49Mbits/s for receiver between the h1 and h3

```

"Node: h2"
mininet/examples#
root@austinguish-GL502VSK:/home/austinguish/Desktop/Computer Networks/Chapter1/
mininet/examples# iperf3 -c 10.0.0.3
Connecting to host 10.0.0.3, port 5201
[ 7] local 10.0.0.2 port 42476 connected to 10.0.0.3 port 5201
[ ID] Interval      Transfer      Bitrate      Retr      Cwnd
[ 7] 0.00-1.00 sec  1.94 MBytes  16.3 Mbits/sec  0         257 KBytes
[ 7] 1.00-2.00 sec  1.74 MBytes  14.6 Mbits/sec  0         317 KBytes
[ 7] 2.00-3.00 sec  1.43 MBytes  12.0 Mbits/sec  0         375 KBytes
[ 7] 3.00-4.00 sec  827 KBytes  6.77 Mbits/sec  0         433 KBytes
[ 7] 4.00-5.00 sec  1.80 MBytes  15.1 Mbits/sec  0         491 KBytes
[ 7] 5.00-6.00 sec  1018 KBytes  8.34 Mbits/sec  0         550 KBytes
[ 7] 6.00-7.00 sec  2.30 MBytes  19.3 Mbits/sec  0         608 KBytes
[ 7] 7.00-8.00 sec  1.25 MBytes  10.5 Mbits/sec  0         666 KBytes
[ 7] 8.00-9.00 sec  1.25 MBytes  10.5 Mbits/sec  0         725 KBytes
[ 7] 9.00-10.00 sec 1.25 MBytes  10.5 Mbits/sec  0         783 KBytes
[ ID] Interval      Transfer      Bitrate      Retr
[ 7] 0.00-10.00 sec 14.8 MBytes  12.4 Mbits/sec  0
[ 7] 0.00-10.67 sec 12.0 MBytes  9.45 Mbits/sec
send
rece
iperf Done.
root@austinguish-GL502VSK:/home/austinguish/Desktop/Computer Networks/Chapter1/
mininet/examples#

"Node: h3"
Server listening on 5201
Accepted connection from 10.0.0.2, port 42474
[ 7] local 10.0.0.3 port 5201 connected to 10.0.0.2 port 42476
[ ID] Interval      Transfer      Bitrate
[ 7] 0.00-1.00 sec  1015 KBytes  8.32 Mbits/sec
[ 7] 1.00-2.00 sec  1.14 MBytes  9.58 Mbits/sec
[ 7] 2.00-3.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 3.00-4.00 sec  1.14 MBytes  9.54 Mbits/sec
[ 7] 4.00-5.00 sec  1.14 MBytes  9.58 Mbits/sec
[ 7] 5.00-6.00 sec  1.14 MBytes  9.55 Mbits/sec
[ 7] 6.00-7.00 sec  1.14 MBytes  9.57 Mbits/sec
[ 7] 7.00-8.00 sec  1.14 MBytes  9.58 Mbits/sec
[ 7] 8.00-9.00 sec  1.14 MBytes  9.55 Mbits/sec
[ 7] 9.00-10.00 sec 1.14 MBytes  9.57 Mbits/sec
[ 7] 10.00-10.67 sec 788 KBytes  9.59 Mbits/sec
[ ID] Interval      Transfer      Bitrate
[ 7] 0.00-10.67 sec 12.0 MBytes  9.45 Mbits/sec
receiver
Server listening on 5201

```

the bandwidth is 12.4Mbits/s for sender and 9.45Mbits/s for receiver between the h2 and h3

- Now let us set the packet loss rate of the link (s1,s2) and (s1,s3) as 5%. Use Iperf to test the TCP throughput again.

the bandwidth is 1.94Mbits/s for sender and 1.85Mbits/s for receiver between the h1 and h2

the bandwidth is 2.08Mbits/s for sender and 1.91Mbits/s for receiver between the h1 and h3

the bandwidth is 1.44Mbits/s for sender and 1.13Mbits/s for receiver between the h2 and h3



3. Let us add another link between s2 and s3. Try pinging h2 from h1. What would happen? How would you solve the problem? (Hint: Use ovs-ofctl command to add flow rules.)

h1--s1--s2--h2

| /

s3

|

h3

```
mininet> h1 ping -c 4 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- 10.0.0.2 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3066ms
```

The result is all pings missed.

`links` display the links between the interfaces.

```
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
h3-eth0<->s3-eth1 (OK OK)
s1-eth3<->s2-eth2 (OK OK)
s1-eth2<->s3-eth2 (OK OK)
s2-eth3<->s3-eth3 (OK OK)
```

using `sudo ovs-ofctl add-flow` to set the flow rules.

```
1 sudo ovs-ofctl add-flow s2 "in_port=2 actions=output:1"
2 sudo ovs-ofctl add-flow s2 "in_port=1 actions=output:2"
3 sudo ovs-ofctl add-flow s1 "in_port=1 actions=output:3"
4 sudo ovs-ofctl add-flow s1 "in_port=1 actions=output:3"
```

test:

```
mininet> h1 ping -c 4 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=30.2 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=30.2 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=30.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=30.2 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 30.178/30.201/30.230/0.019 ms
```

```
mininet> h2 ping -c 4 h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=30.2 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=30.2 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=30.2 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=30.2 ms

--- 10.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 30.166/30.192/30.208/0.015 ms
```