

1.

```
mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b4:52:ec
          inet addr:192.168.56.100  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KB)  TX bytes:684 (684.0 B)

eth1      Link encap:Ethernet  HWaddr 08:00:27:44:fe:d6
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:352 errors:0 dropped:0 overruns:0 frame:0
          TX packets:339 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34999 (34.9 KB)  TX bytes:35556 (35.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1188 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:59400 (59.4 KB)  TX bytes:59400 (59.4 KB)

s3        Link encap:Ethernet  HWaddr ca:b4:ce:85:fb:45
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s4        Link encap:Ethernet  HWaddr fe:4c:cf:d3:50:4a
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

mininet@mininet-vm:~$
```

We have set up two network interfaces, one for connecting the VM to the outside through NAT, and one for the host only network. In this case, we can see eth0 is the interface for the latter.

The IP address is determined by ifconfig. As we can see here, the IP address is 192.168.56.100 because this is what we set for the host address earlier, and this is also the address at which the guests can access the host.

2.

```
mininet@mininet-vm: ~/mininet/custom
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$ sudo mn --custom topo-2sw-2host.py --topo mytopo --link tc,bw=10,delay=100ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(10.00Mbit 100ms delay) (h1, s3) (10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (s3, s4) (10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ... (10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (10.00Mbit 100ms delay)
*** Starting CLI:
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
```

On the picture above, we can see that a network is created. A controller is added. Two hosts are added, h1 and h2. Two switches are added, s3 and s4. Three links are created; the first link is created between h1 and s3 with bandwidth 10 Mbit and 100ms delay, the second link is created between s3 and s4 with bandwidth 10Mbit and 100ms delay, the third link is created between s4 and h2 with bandwidth 10Mbit and 100ms delay. The hosts, controller and switches launched.

```
mininet@mininet-vm: ~/mininet/custom
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> links
h1-eth0<->s3-eth1 (OK OK)
s3-eth2<->s4-eth1 (OK OK)
s4-eth2<->h2-eth0 (OK OK)
mininet> h1 ping -c 5 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=1216 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=605 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=601 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=601 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=601 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 601.137/725.356/1216.488/245.571 ms, pipe 2
mininet>
mininet> iperf h1 h2
*** Iperf: Testing TCP bandwidth between h1 and h2
*** Results: ['6.02 Mbits/sec', '8.80 Mbits/sec']
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
```

**Links**

Three links is created successfully

- One is created between h1 through eth0 and s3 through eth1
- One is created between s3 through eth2 and s4 through eth1
- One is created between s4 through eth2 and h2 through eth0

**H1 ping -c 5 h2**

The ping command is to verify the connectivity between host 1 and host 2 is successful. As we can see, 5 packets were sent to verify the connectivity, and it all went successfully and therefore the connectivity between h1 and h2 is successful created.

We can see a much lower ping time at the second try. A flow entry covering ping traffic was previously installed in the switch, so no control traffic was generated, and the packets immediately pass through the switch.

**Iperf h1 h2**

Iperf measured the maximum achieved bandwidth between h1 and h2. This indicates that the through put in the forward direction is 6.02Mbits/sec and the reverse direction is 8.80Mbits/sec.

```
mininet@mininet-vmx:~/mininet/custom
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$ sudo mn --custom topo-2sw-2host.py --topo mytopo --link tc
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(10.00Mbit 5ms delay) (h1, s3) (2.00Mbit 2ms delay) (2.00Mbit 2ms delay) (s3, s4) (10.00Mbit 5ms delay) (10.00Mbit 5ms delay) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ... (10.00Mbit 5ms delay) (2.00Mbit 2ms delay) (2.00Mbit 2ms delay) (10.00Mbit 5ms delay)
*** Starting CLI:
mininet>
```

```
mininet@mininet-vmx:~/mininet/custom
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$
mininet@mininet-vm:~/mininet/custom$ sudo mn --custom topo-2sw-2host.py --topo mytopo --link tc
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(10.00Mbit 5ms delay) (h1, s3) (2.00Mbit 2ms delay) (2.00Mbit 2ms delay) (s3, s4) (10.00Mbit 5ms delay) (10.00Mbit 5ms delay) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ... (10.00Mbit 5ms delay) (2.00Mbit 2ms delay) (2.00Mbit 2ms delay) (10.00Mbit 5ms delay)
*** Starting CLI:
mininet>
```

```
mininet@mininet-vm: ~/mininet/custom
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> links
h1-eth0<->s3-eth1 (OK OK)
s3-eth2<->s4-eth1 (OK OK)
s4-eth2<->h2-eth0 (OK OK)
mininet> h1 ping -c 5 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=64.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=26.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=25.4 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=25.6 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=26.3 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 25.460/33.657/64.462/15.407 ms
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['1.92 Mbits/sec', '2.36 Mbits/sec']
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
```

**Links**

Three links is created successfully

- One is created between h1 through eth0 and s3 through eth1
- One is created between s3 through eth2 and s4 through eth1
- One is created between s4 through eth2 and h2 through eth0

**H1 ping -c 5 h2**

The ping command is to verify the connectivity between host 1 and host 2 is successful. As we can see, 5 packets were sent to verify the connectivity, and it all went successfully and therefore the connectivity between h1 and h2 is successful created.

We can see a much lower ping time at the second try(26.3ms). A flow entry covering ping traffic was previously installed in the switch, so no control traffic was generated, and the packets immediately pass through the switch.

As we can see, the time is much lower compared to question 2. This happened because we have set the delay time a lot lower compared to Question 2.

**Iperf h1 h2**

Iperf measured the maximum achieved bandwidth between h1 and h2. This indicates that the through put in the forward direction is 1.92Mbits/sec and the reverse direction is 2.36Mbits/sec. As we can see

the throughput is slower in this case compared to Question 2 because we have set the bandwidth between switch 3 and switch 4 to 2Mbit, which is slower compared to Question2.

## Modified topo-2sw-2host.py

[illegible]