TTM4180 - Applied Networking
Lab 3: Mininet

By S. Xiao Fernández Marín

T1: Find the command required to create a linear topology with 4 hosts, automatically set hosts' MAC addresses to more readable addresses, configure the controller in remote mode, and also set the required switches to ovsk type. (Hint: use mn manual to extract the required flags)

```
sudo mn --topo linear,4 --mac --controller remote --switch ovsk
```

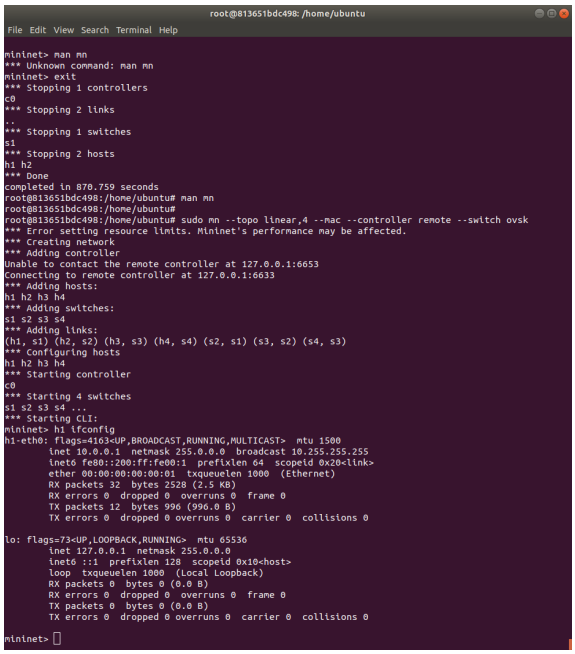


Figure 1

T2: Now start over the procedure by making a new linear topology with 4 hosts, bandwidth of 1 Mbps, and delay of 10ms. Draw the created topology with the assigned IP address on each interface, and include it in your report.
PS: There is no need to connect a controller in this lab. **So exclude** –controller and –switch, and **do not start the controller.**

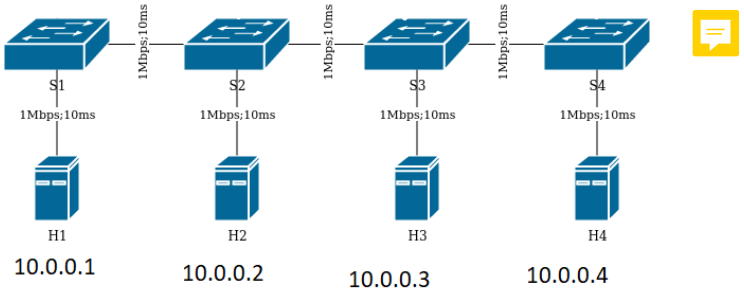


Figure 2

```
sudo mn --topo linear,4 --link tc,bw=1,delay=10ms
```

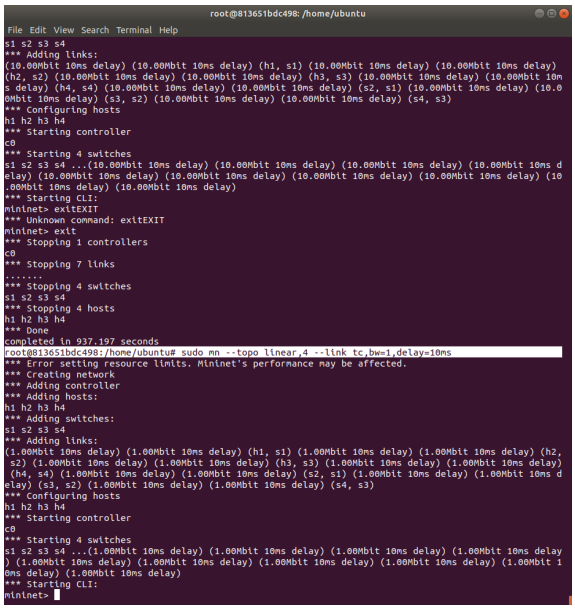


Figure 3

Q1: How can you verify that the bandwidth and the delay are set up correct?

The links created are represented in:

```
*** Adding links:
(1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (h1, s1) (1.00Mbit 10ms delay)
↪  (1.00Mbit 10ms delay) (h2, s2) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay)
↪  (h3, s3) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (h4, s4) (1.00Mbit 10ms
↪  delay) (1.00Mbit 10ms delay) (s2, s1) (1.00Mbit 10ms delay) (1.00Mbit 10ms
↪  delay) (s3, s2) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (s4, s3)
```

Q2: What is the Round-trip delay between h1 and h4?

It is 10ms h4-¿s4; 10ms s4-¿s3; 10ms s3-¿s2; 10ms s2-¿s1; 10ms s1-¿h1;
50ms in total from one host to another, 100ms for round trip.

T3: Set up a xterm window on h1, h2, h3 and h4.

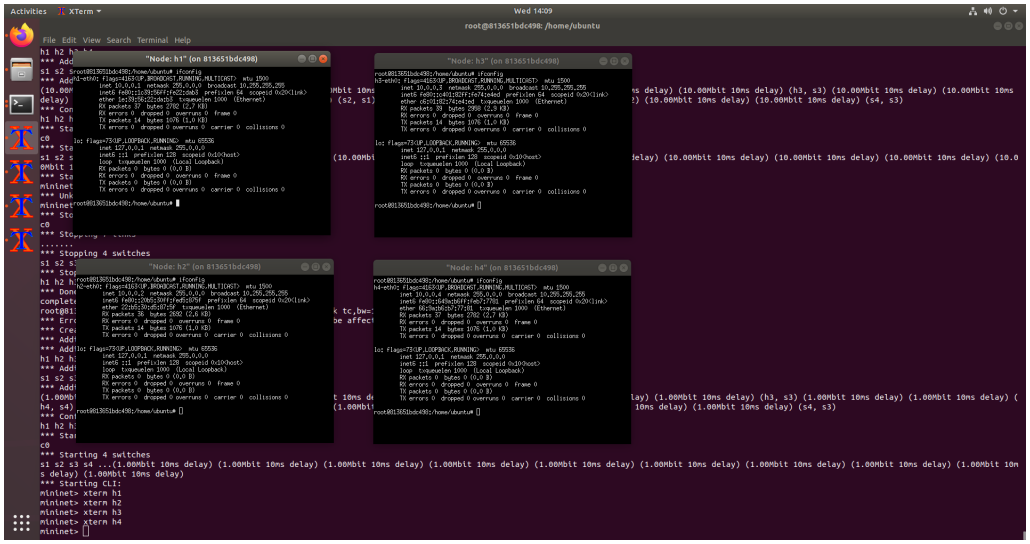


Figure 4

T4: Set up node h4 to be an iperf server.
T5: Start an iperf session between h1 and h4.

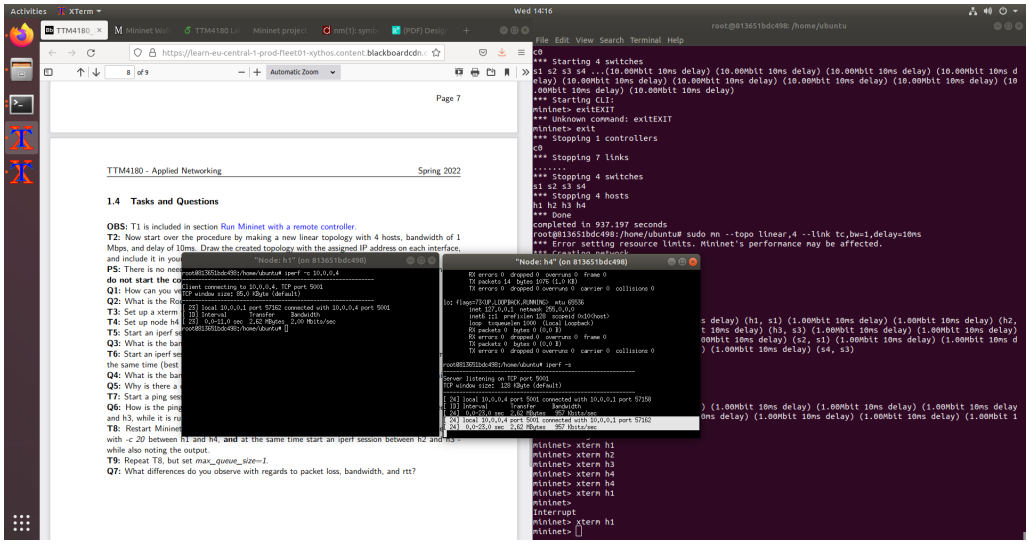


Figure 5

Q3: What is the bandwidth from h1 to h4?

957 Kbits/sec. The maximum bandwith is 1.00Mbps, so it makes sense.

T6: Start an iperf session between h1 and h4, ***and*** h2 and h3. This should be done at *approximately* the same time (best effort applies here).

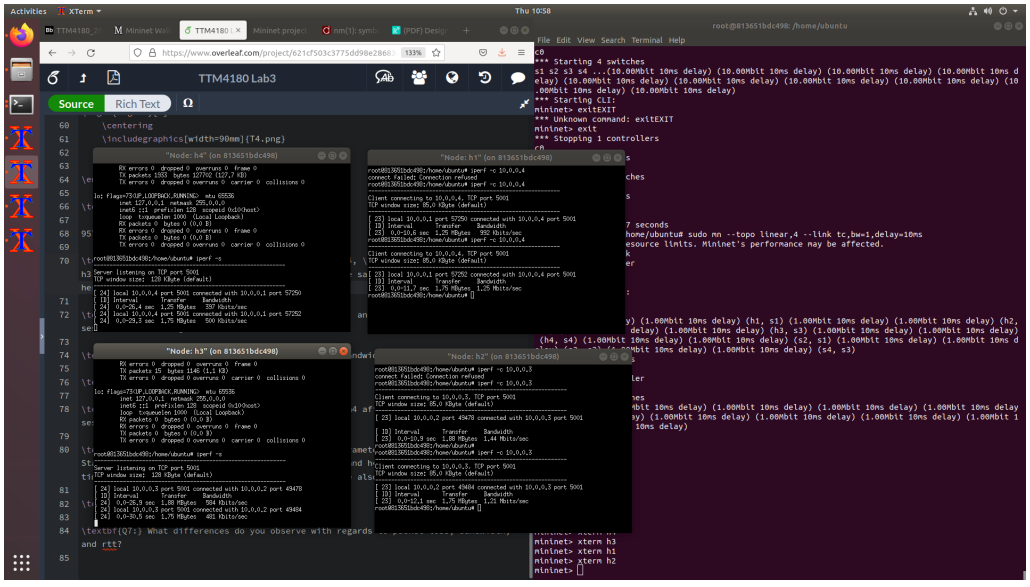


Figure 6

Q4: What is the bandwidth between h1 and h4, and h2 and h3, while both sessions are running?

h1 and h4: 500 Kbits/sec
h2 and h3: 481 Kbits/se
As the maximum bandwith is 1.00Mbps it makes sense as in total is 981Kbps.

Q5: Why is there a difference between the bandwidth in T6 and T5?

As it is linear, it is slower when both sessions are running, because they have to share the only link they have.

T7: Start a ping session between h2 and h4.

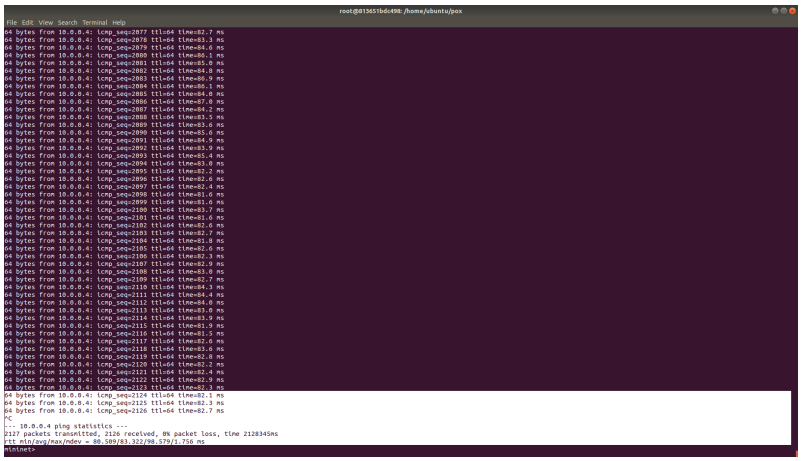
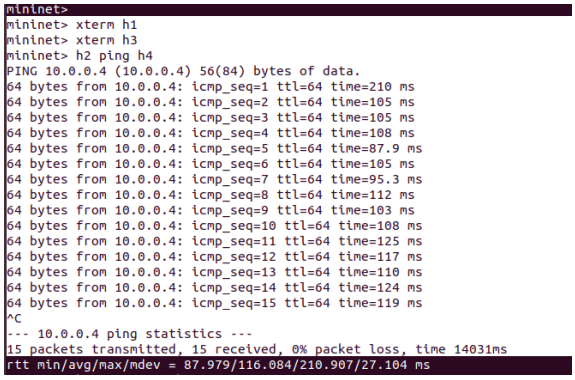


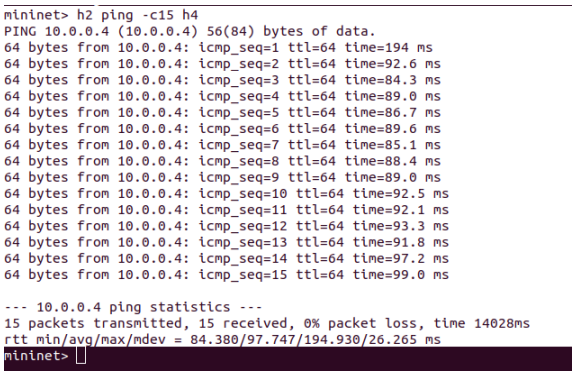
Figure 7

Q6: How is the ping session between h2 and h4 affected, if you start an iperf session between h1 and h3, while it is running?



```
mininet>
mininet> xterm h1
mininet> xterm h3
mininet> h2 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=210 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=108 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=87.9 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=95.3 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=112 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=103 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=108 ms
64 bytes from 10.0.0.4: icmp_seq=11 ttl=64 time=125 ms
64 bytes from 10.0.0.4: icmp_seq=12 ttl=64 time=117 ms
64 bytes from 10.0.0.4: icmp_seq=13 ttl=64 time=110 ms
64 bytes from 10.0.0.4: icmp_seq=14 ttl=64 time=124 ms
64 bytes from 10.0.0.4: icmp_seq=15 ttl=64 time=119 ms
^C
--- 10.0.0.4 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14031ms
rtt min/avg/max/mdev = 87.979/116.084/210.907/27.104 ms
```

Figure 8



```
mininet> h2 ping -c15 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=194 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=92.6 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=84.3 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=89.0 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=86.7 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=89.6 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=85.1 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=88.4 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=89.0 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=92.5 ms
64 bytes from 10.0.0.4: icmp_seq=11 ttl=64 time=92.1 ms
64 bytes from 10.0.0.4: icmp_seq=12 ttl=64 time=93.3 ms
64 bytes from 10.0.0.4: icmp_seq=13 ttl=64 time=91.8 ms
64 bytes from 10.0.0.4: icmp_seq=14 ttl=64 time=97.2 ms
64 bytes from 10.0.0.4: icmp_seq=15 ttl=64 time=99.0 ms

--- 10.0.0.4 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14028ms
rtt min/avg/max/mdev = 84.380/97.747/194.930/26.265 ms
mininet>
```

Figure 9

It takes more time in Figure 8 for the packages to arrive as there are also an iperf session.

T8: Restart Mininet and include the link parameter $max\_queue\_size = 100$. Start a ping session with *-c* 20 between h1 and h4, **and** at the same time start an iperf session between h2 and h3 - while also noting the output.

```
sudo mn --topo linear,4 --link tc,bw=1,delay=10ms,max_queue_size=100
```

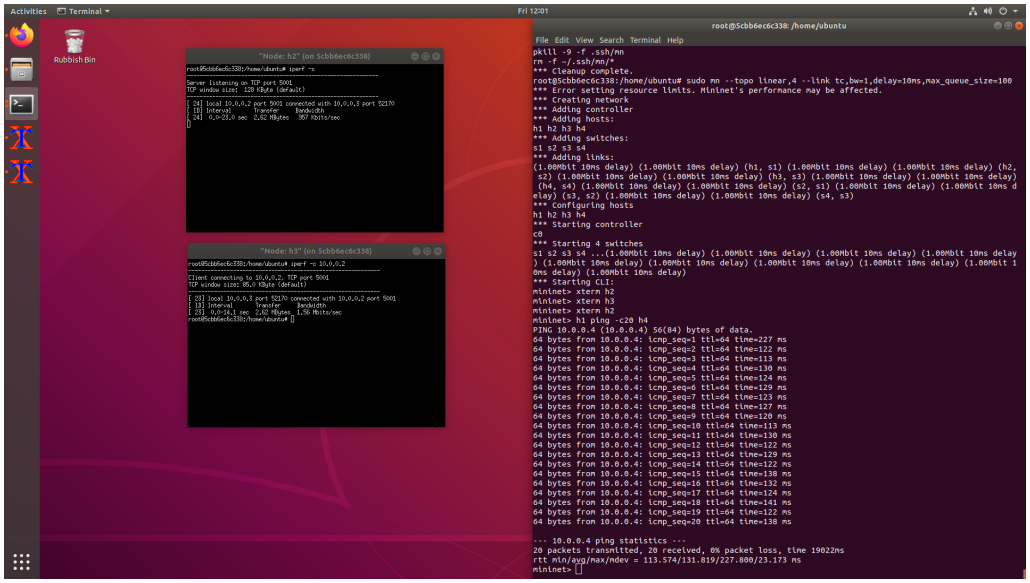


Figure 10

T9: Repeat T8, but set $max\_queue\_size = 1$.

```
sudo mn --topo linear,4 --link tc,bw=1,delay=10ms,max_queue_size=1
```

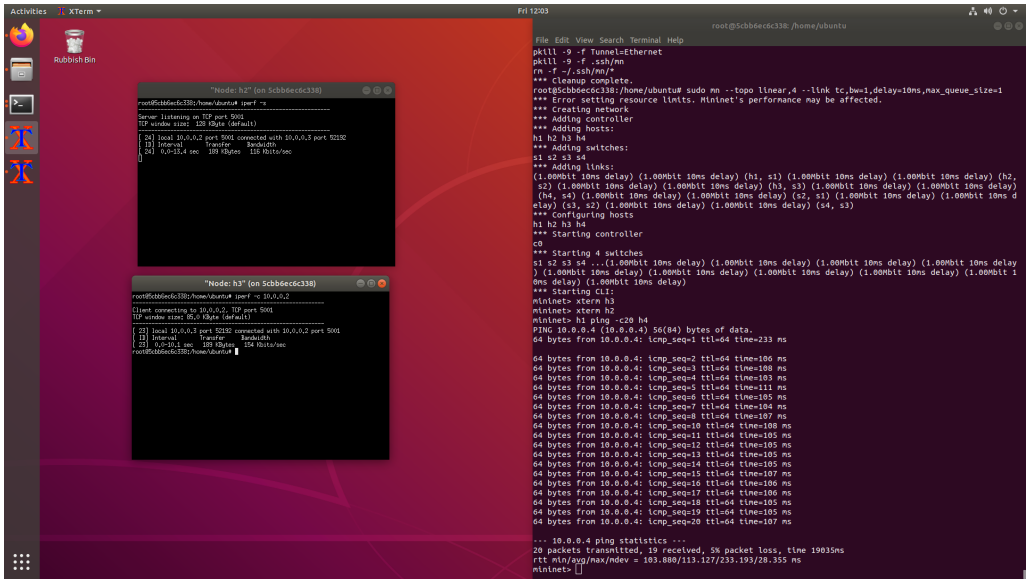


Figure 11

Q7: What differences do you observe with regards to packet loss, bandwidth, and rtt?

	Packet loss	Bandwith	RTT (min/avg/max/mdev)
T8	0%	957 Kbits/sec	113.574/131.819/227.800/23.173 ms
T9	5%	116 Kbits/sec	103.880/113.127/233.193/28.355 ms

Table 1: Caption

T9 is slower as it has a queue of 1 and the queue of T8 is of 100. This is why in the last one there are some packages missing and the bandwith is slower than the one of queue 100.