

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
```

```
root@813651bdc498:/home/ubuntu
File Edit View Search Terminal Help
mininet> man mn
*** Unknown command: man mn
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 870.759 seconds
root@813651bdc498:/home/ubuntu man mn
root@813651bdc498:/home/ubuntu
root@813651bdc498:/home/ubuntu sudo mn --topo linear,4 --mac --controller remote --switch ovsk
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::200:ff:fe00:1 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:01 txqueuelen 1000 (Ethernet)
    RX packets 32 bytes 2528 (2.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 996 (996.0 B)
    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 (Local Loopback)
    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

mininet>
```

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**.

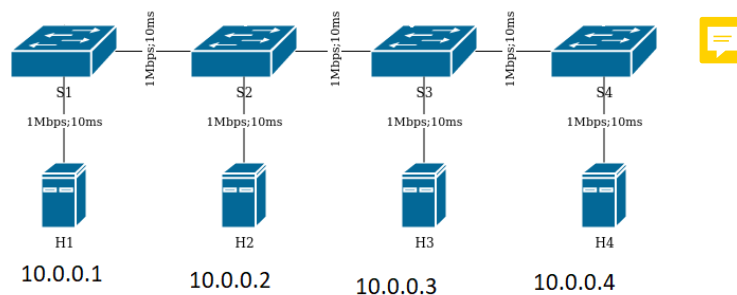


Figure 2

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

```

root@813651bdc498: /home/ubuntu
File Edit View Search Terminal Help
s1 s2 s3 s4
*** Adding links:
(10.00Mbit 10ms delay) (h1, s1) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay)
(h2, s2) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (h3, s3) (10.00Mbit 10ms delay) (10.00Mbit 10ms
delay) (h4, s4) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (s2, s1) (10.00Mbit 10ms delay) (10.0
0Mbit 10ms delay) (s3, s2) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c9
*** Starting 4 switches
s1 s2 s3 s4 ... (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (10.00Mbit 10ms d
elay) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (10
.00Mbit 10ms delay) (10.00Mbit 10ms delay)
*** Starting CLI:
mininet> exitEXIT
mininet> exit
Unknown command: exitEXIT
*** Stopping 1 controllers
c9
*** Stopping 7 links
.....
*** Stopping 4 switches
s1 s2 s3 s4
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in 937.197 seconds
root@813651bdc498: /home/ubuntu# sudo mn --topo linear,4 --link tc,bw=1,delay=10ms
*** Error setting resource limits. Mininet's performance may be affected.
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** 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 d
elay) (s3, s2) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c9
*** Starting 4 switches
s1 s2 s3 s4 ... (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay
) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (1.00Mbit 10ms delay) (1.00Mbit 1
0ms delay) (1.00Mbit 10ms delay)
*** Starting CLI:
mininet>

```

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- \rightarrow s4; 10ms s4- \rightarrow s3; 10ms s3- \rightarrow s2; 10ms s2- \rightarrow s1; 10ms s1- \rightarrow 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.

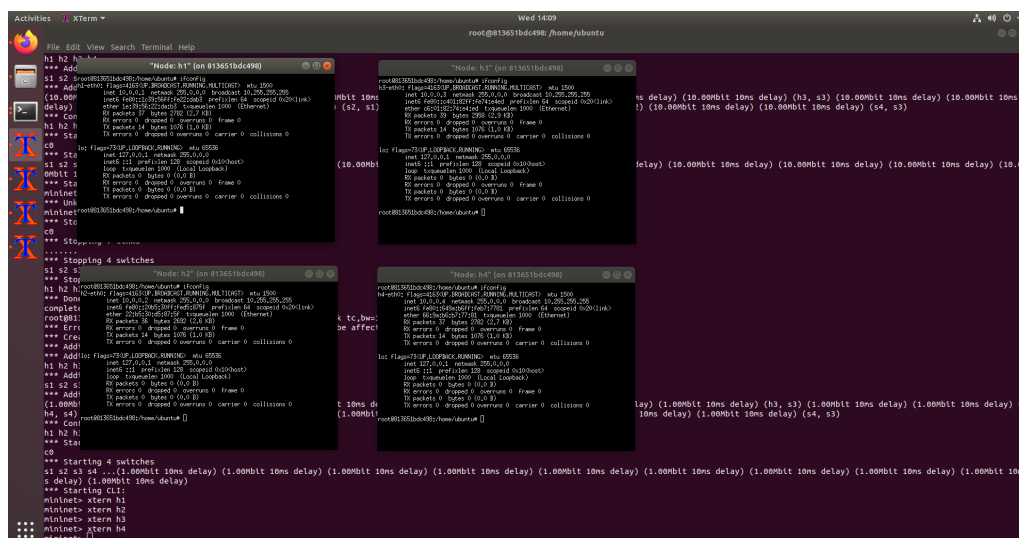


Figure 4

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

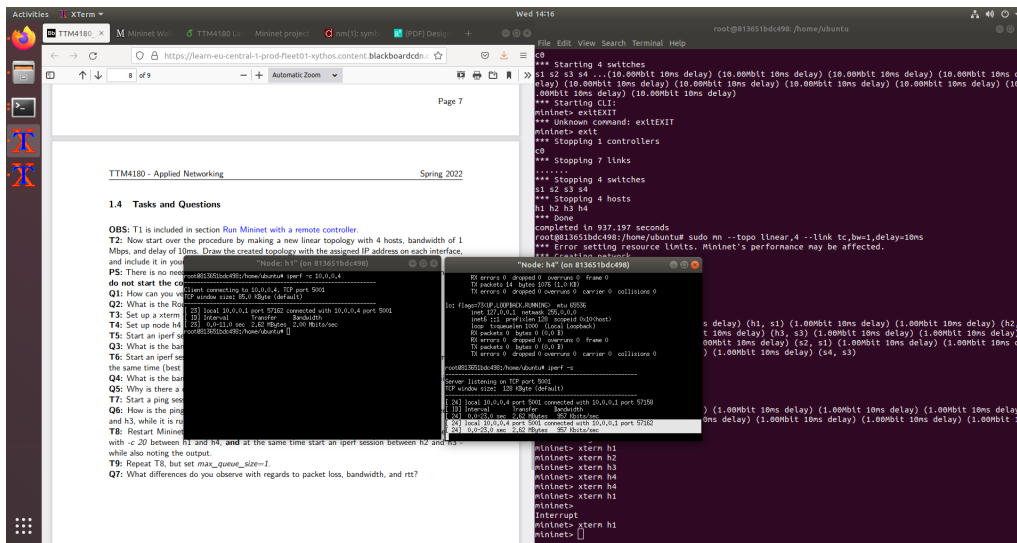


Figure 5

Q3: What is the bandwidth from h1 to h4?

957 Kbits/sec. The maximum bandwidth 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).

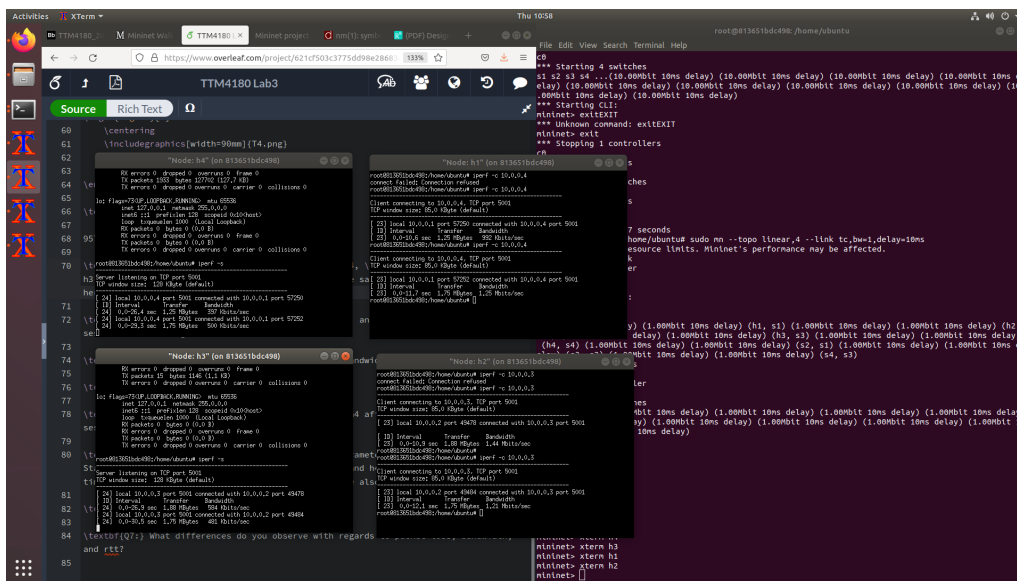


Figure 6

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

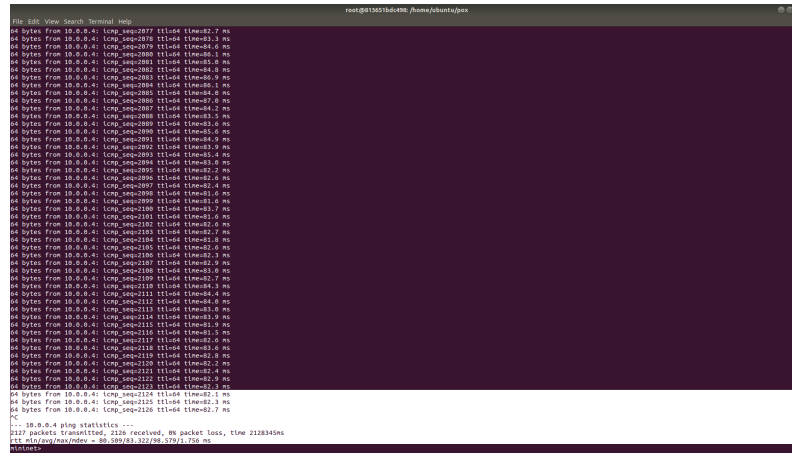
h1 and h4: 500 Kbits/sec

h2 and h3: 481 Kbits/sec

As the maximum bandwidth 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.



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?

Figure 8

Figure 9

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
```

```

root@Schibecdc338:/home/ubuntu# sudo mn --topo linear,4 --link tc,bw=1,delay=10ms,max_queue_size=100
*** Error setting resource limits. Mininet's performance may be affected.
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(1.000bit 10ms delay) (1.000bit 10ms delay) (h1, s1) (1.000bit 10ms delay) (1.000bit 10ms delay) (h2,
s2) (1.000bit 10ms delay) (1.000bit 10ms delay) (h3, s3) (1.000bit 10ms delay) (1.000bit 10ms delay)
(h4, s4) (1.000bit 10ms delay) (1.000bit 10ms delay) (s2, s1) (1.000bit 10ms delay) (1.000bit 10ms d
elay) (s3, s2) (1.000bit 10ms delay) (1.000bit 10ms delay) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ... (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay)
(1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 1
0ms delay) (1.000bit 10ms delay)
*** Starting CLI:
mininet> xterm h2
mininet> xterm h3
mininet> h1 ping -c20 h4
PING 10.0.0.4 (10.0.0.4) 56(64) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=227 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=122 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=113 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=130 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=124 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=123 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=123 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=127 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=129 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=113 ms
64 bytes from 10.0.0.4: icmp_seq=11 ttl=64 time=126 ms
64 bytes from 10.0.0.4: icmp_seq=12 ttl=64 time=122 ms
64 bytes from 10.0.0.4: icmp_seq=13 ttl=64 time=129 ms
64 bytes from 10.0.0.4: icmp_seq=14 ttl=64 time=132 ms
64 bytes from 10.0.0.4: icmp_seq=15 ttl=64 time=138 ms
64 bytes from 10.0.0.4: icmp_seq=16 ttl=64 time=133 ms
64 bytes from 10.0.0.4: icmp_seq=17 ttl=64 time=124 ms
64 bytes from 10.0.0.4: icmp_seq=18 ttl=64 time=141 ms
64 bytes from 10.0.0.4: icmp_seq=19 ttl=64 time=122 ms
64 bytes from 10.0.0.4: icmp_seq=20 ttl=64 time=138 ms
... 10.0.0.4 ping statistics ...
20 packets transmitted, 20 received, 0% packet loss, time 1902ms
rtt min/avg/max/mdev = 113.574/131.815/227.080/23.172 ms
mininet> [1]

```

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
```

```

root@Schibecdc338:/home/ubuntu# sudo mn --topo linear,4 --link tc,bw=1,delay=10ms,max_queue_size=1
*** Error setting resource limits. Mininet's performance may be affected.
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(1.000bit 10ms delay) (1.000bit 10ms delay) (h1, s1) (1.000bit 10ms delay) (1.000bit 10ms delay) (h2,
s2) (1.000bit 10ms delay) (1.000bit 10ms delay) (h3, s3) (1.000bit 10ms delay) (1.000bit 10ms delay)
(h4, s4) (1.000bit 10ms delay) (1.000bit 10ms delay) (s2, s1) (1.000bit 10ms delay) (1.000bit 10ms d
elay) (s3, s2) (1.000bit 10ms delay) (1.000bit 10ms delay) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ... (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay)
(1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 10ms delay) (1.000bit 1
0ms delay) (1.000bit 10ms delay)
*** Starting CLI:
mininet> xterm h2
mininet> xterm h3
mininet> h1 ping -c20 h4
PING 10.0.0.4 (10.0.0.4) 56(64) bytes of data:
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=233 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=106 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=108 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=103 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=111 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=104 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=107 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=108 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=11 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=12 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=13 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=14 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=15 ttl=64 time=107 ms
64 bytes from 10.0.0.4: icmp_seq=16 ttl=64 time=106 ms
64 bytes from 10.0.0.4: icmp_seq=17 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=18 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=19 ttl=64 time=105 ms
64 bytes from 10.0.0.4: icmp_seq=20 ttl=64 time=107 ms
... 10.0.0.4 ping statistics ...
20 packets transmitted, 19 received, 5% packet loss, time 1903ms
rtt min/avg/max/mdev = 103.880/113.127/233.193/28.355 ms
mininet> [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 bandwidth is slower than the one of queue 100.