

# CprE 489, Section 4

## Lab Experiment #7: Introduction to Mininet

Sean Gordon

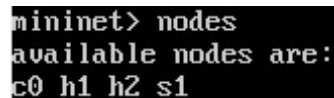
### **Experience:**

This lab was an introduction to virtual machines and mininet, running through some frankly sparse setup instructions and then some useful commands that can be run on the mininet VM. I have had only cursory experience with VMs before this lab, so after the several hours of agonizing setup, it was a nice, remarkably smooth experience that introduced me to the workings of a VM.

The use of some familiar functions in exploration of the VM was a nice tie to previous experience, with the extra optional concepts allowing further insight into its more complex workings. With this as a foundation, I will certainly be further exploring the uses of VMs.

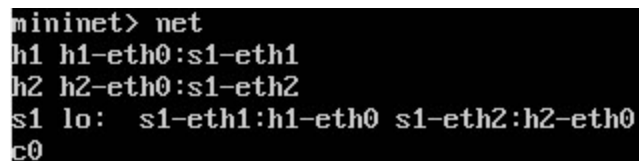
### **Part 1: Everyday Mininet Usage**

This section focused on starting up the VM with simple commands like 'sudo mn', as well as testing connection between hosts and exchanging basic information between them. This section laid the foundation for the rest of the lab, deepening understanding of the uses of the hosts.



```
mininet> nodes
available nodes are:
c0 h1 h2 s1
```

Image 1



```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
```

Image 2



## **Part 2: Advanced Startup Options**

This section focused on configuration options for mininet, and runs through some benchmarking for different configurations. This introduces several commands and config options from the command line and explains their uses.

```
mininet@mininet-vm:~$ sudo mn --test pingpair
**** Creating network
**** Adding controller
**** Adding hosts:
h1 h2
**** Adding switches:
s1
**** Adding links:
(h1, s1) (h2, s1)
**** Configuring hosts
h1 h2
**** Starting controller
c0
**** Starting 1 switches
s1 ...
**** Waiting for switches to connect
s1
h1 -> h2
h2 -> h1
**** Results: 0% dropped (2/2 received)
**** Stopping 1 controllers
c0
**** Stopping 2 links
..
**** Stopping 1 switches
s1
**** Stopping 2 hosts
h1 h2
**** Done
completed in 5.225 seconds
```

Image 7

```
mininet@mininet-vm:~$ sudo mn --test iperf
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Iperf: testing TCP bandwidth between h1 and h2
.*** Results: ['36.2 Gbits/sec', '36.3 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 10.738 seconds
```

Image 8

```
mininet@mininet-vm:~$ sudo mn --test pingall --topo single,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 1.703 seconds
```

Image 9

```

mininet@mininet-vm:~$ sudo mn --test pingall --topo linear,4
*** Creating network
*** Adding controller
*** 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 ...
*** Waiting for switches to connect
s1 s2 s3 s4
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
*** Stopping 1 controllers
c0
*** Stopping 7 links
.....
*** Stopping 4 switches
s1 s2 s3 s4
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in 5.582 seconds

```

Image 10

```

mininet> h1 ping -c10 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=42.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=40.0 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=40.9 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=40.9 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=40.9 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=41.9 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=42.2 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=41.8 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9003ms
rtt min/avg/max/mdev = 40.086/41.511/42.226/0.701 ms

```

Image 11

```

mininet@mininet-vm:~$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo
mytopo --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4
*** Adding links:
(h1, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s3 s4 ...
*** Waiting for switches to connect
s3 s4
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 3 links
...
*** Stopping 2 switches
s3 s4
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.325 seconds

```

Image 12

```

mininet> h1 ifconfig
h1-eth0  Link encap:Ethernet  HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

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: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)

```

Image 13

```
mininet@mininet-vm:~$ sudo mn --switch user --test iperf
**** Creating network
**** Adding controller
**** Adding hosts:
h1 h2
**** Adding switches:
s1
**** Adding links:
(h1, s1) (h2, s1)
**** Configuring hosts
h1 h2
**** Starting controller
c0
**** Starting 1 switches
s1
**** Waiting for switches to connect
s1
**** Iperf: testing TCP bandwidth between h1 and h2
**** Results: ['934 Mbits/sec', '936 Mbits/sec']
**** Stopping 1 controllers
c0
**** Stopping 2 links
..
**** Stopping 1 switches
s1 ..
**** Stopping 2 hosts
h1 h2
**** Done
completed in 5.711 seconds
```

Image 14



```
mininet@mininet-vm:~$ sudo mn --switch ovsk --test iperf
**** Creating network
**** Adding controller
**** Adding hosts:
h1 h2
**** Adding switches:
s1
**** Adding links:
(h1, s1) (h2, s1)
**** Configuring hosts
h1 h2
**** Starting controller
c0
**** Starting 1 switches
s1 ...
**** Waiting for switches to connect
s1
**** Iperf: testing TCP bandwidth between h1 and h2
**** Results: ['37.4 Gbits/sec', '37.4 Gbits/sec']
**** Stopping 1 controllers
c0
**** Stopping 2 links
..
**** Stopping 1 switches
s1
**** Stopping 2 hosts
h1 h2
**** Done
completed in 10.773 seconds
```

Image 15

```
mininet@mininet-vm:~$ sudo mn --test none
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 0.181 seconds
```

Image 16

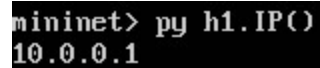
```
mininet@mininet-vm:~$ sudo mn --innamespace --switch user
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
c0 <-> s1
*** Testing control network
s1 -> c0
c0 -> s1
*** Results: 0% dropped (2/2 received)

*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1
*** Starting CLI:
```

Image 17

### **Part 3: Mininet CLI Commands**

This section showed the user where to find more information on what commands are usable with mininet, and ran through some rudimentary python as an example. This shows that the mininet command line is fully functional and can be used for much more than has already been done.

A terminal window with a black background and white text. The prompt 'mininet>' is followed by the command 'py h1.IP()' on the first line. The second line shows the output '10.0.0.1'.

```
mininet> py h1.IP()  
10.0.0.1
```

Image 18

### **Part 4: Python API Examples**

This section showed further uses of the mininet cli, putting focus on its python integration and what benefits it brings.

## Exercises:

- 1) The second ping is much faster as the first host now has the MAC address of the second cached, and does not need to spend the time to find it.

2)

```
Two directly connected switches plus a host for each switch:

    host --- switch --- switch --- host

Adding the 'topos' dict with a key/value pair to generate our newly defined
topology enables one to pass in '--topo=mytopo' from the command line.
'''

from mininet.topo import Topo

class MyTopo( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        leftHost = self.addHost( 'h1' )
        rightHost = self.addHost( 'h2' )
        leftSwitch = self.addSwitch( 's3' )
        midSwitch = self.addSwitch( 's4' )
        rightSwitch = self.addSwitch( 's5' )

        # Add links
        self.addLink( leftHost, leftSwitch )
        self.addLink( leftSwitch, midSwitch )
        self.addLink( midSwitch, rightSwitch )
        self.addLink( rightSwitch, rightHost )

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

```
mininet@mininet-vm:~$ sudo mn --custom ~/mininet/custom/new.py --topo mytopo --test pingall
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s3 s4 s5
*** Adding links:
(h1, s3) (s3, s4) (s4, s5) (s5, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 3 switches
s3 s4 s5 ...
*** Waiting for switches to connect
s3 s4 s5
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 4 links
....
*** Stopping 3 switches
s3 s4 s5
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.747 seconds
```