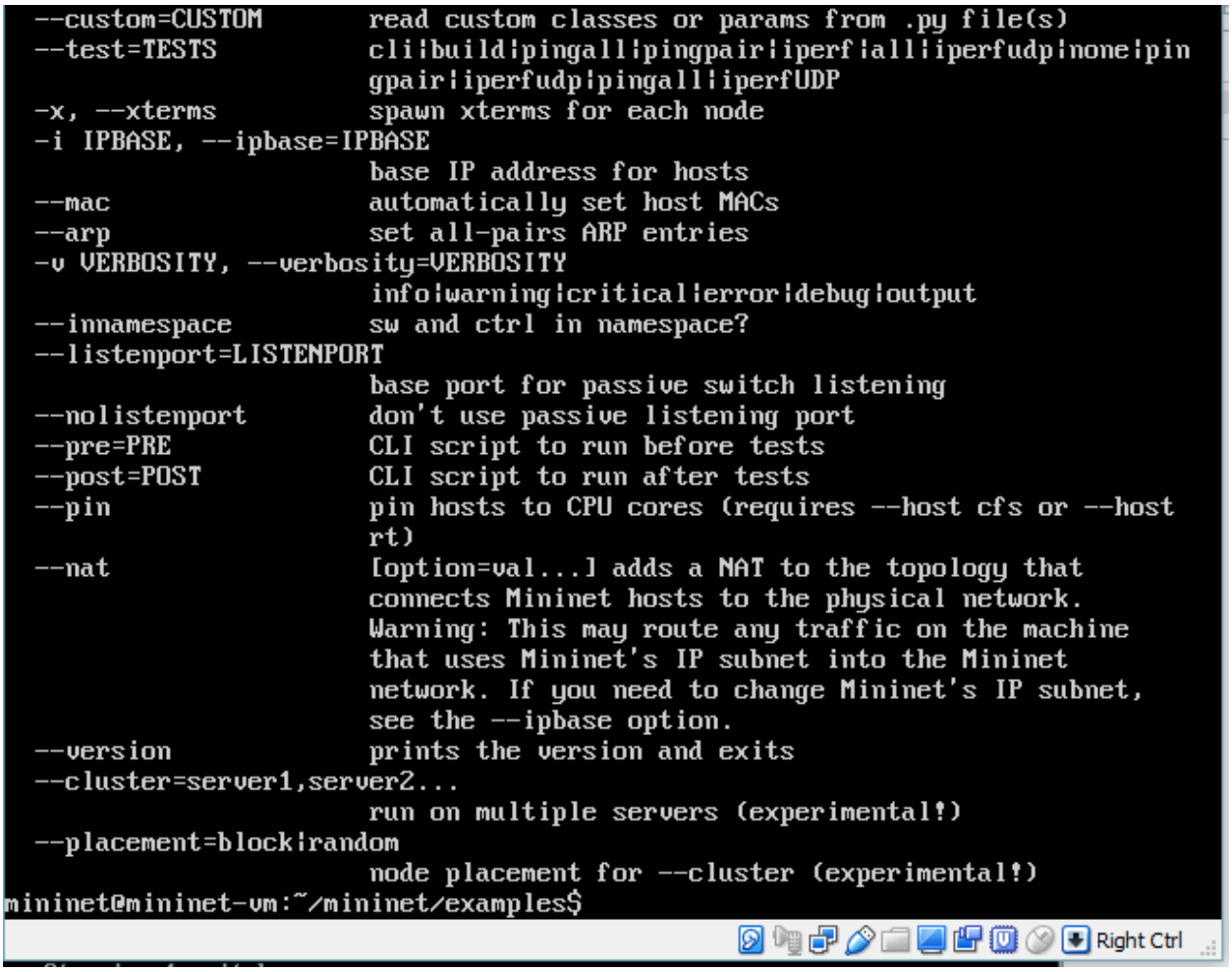
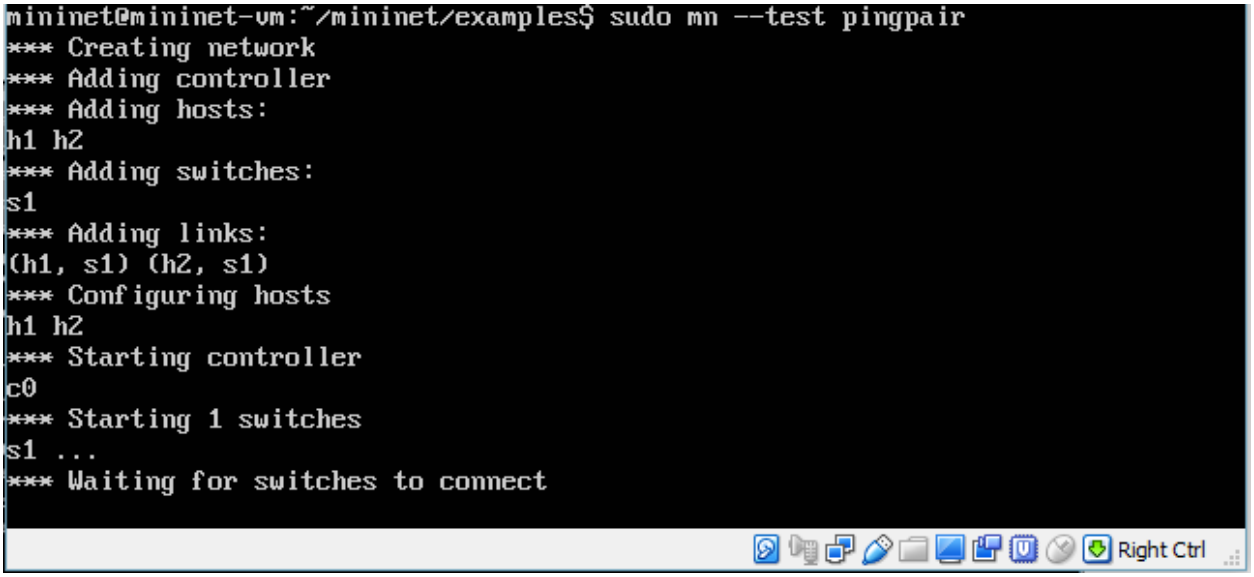
Notes

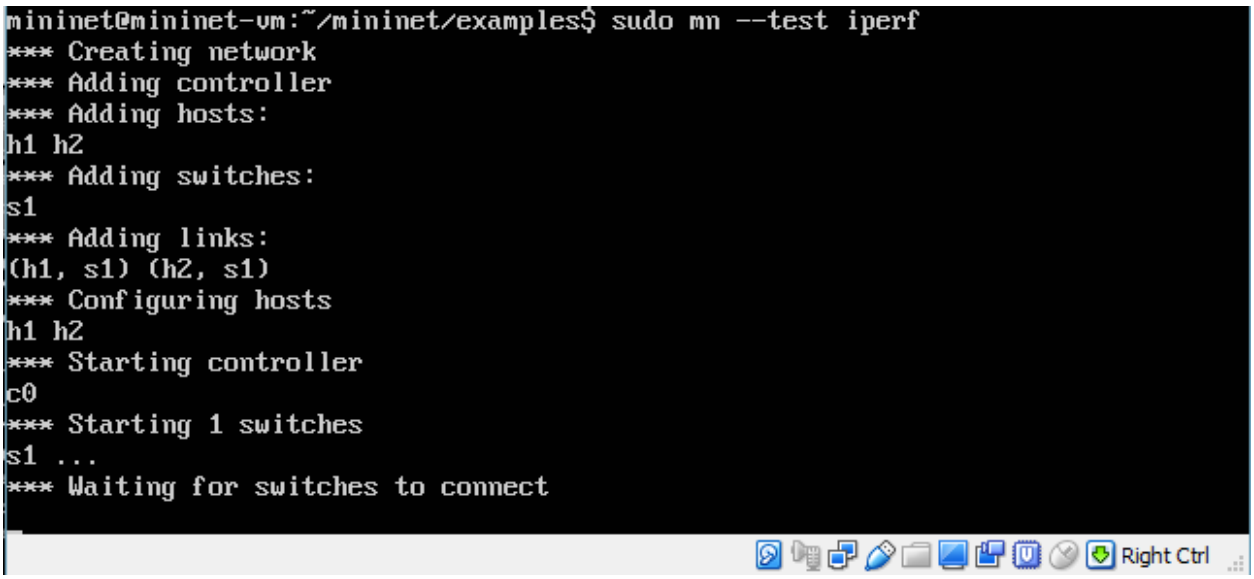
1.


--custom=CUSTOM read custom classes or params from .py file(s)
--test=TESTS cli|build|pingall|pingpair|iperf|all|iperfudp|none|pingpair|iperfudp|pingall|iperfUDP
-x, --xterms spawn xterms for each node
-i IPBASE, --ipbase=IPBASE
base IP address for hosts
--mac automatically set host MACs
--arp set all-pairs ARP entries
-v VERBOSITY, --verbosity=VERBOSITY
info|warning|critical|error|debug|output
--innamespace sw and ctrl in namespace?
--listenport=LISTENPORT
base port for passive switch listening
--nolistenport don't use passive listening port
--pre=PRE CLI script to run before tests
--post=POST CLI script to run after tests
--pin pin hosts to CPU cores (requires --host cfs or --host rt)
--nat [option=val...] adds a NAT to the topology that connects Mininet hosts to the physical network. Warning: This may route any traffic on the machine that uses Mininet's IP subnet into the Mininet network. If you need to change Mininet's IP subnet, see the --ipbase option.
--version prints the version and exits
--cluster=server1,server2...
run on multiple servers (experimental!)
--placement=block|random
node placement for --cluster (experimental!)
mininet@mininet-vm:~/mininet/examples$
Right Ctrl

2.


mininet@mininet-vm:~/mininet/examples$ 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
Right Ctrl

3.


mininet@mininet-vm:~/mininet/examples$ 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
Right Ctrl

4.

```
mininet@mininet-vm:~/mininet/examples$ sudo mn --link tc,bw=10,delay=10ms
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(10.00Mbit 10ms delay) (10.00Mbit 10ms delay) (h1, s1) (10.00Mbit 10ms delay) (1
0.00Mbit 10ms delay) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...(10.00Mbit 10ms delay) (10.00Mbit 10ms delay)
*** Starting CLI:
mininet>
```

5.

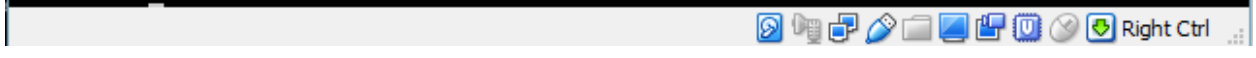
```
mininet@mininet-vm:~/mininet/examples$ sudo mn --test pingall --topo single,3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
```

6.

```
mn: error: no such option: --custom~/mininet/custom/topo-2sw-2host.py
mininet@mininet-vm:~/mininet/examples$ sudo mn --test pingall --topo mytopo --cu
stom ~/mininet/custom/topo-2sw-2host.py
*** 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
```

7.

```
mininet@mininet-vm:~/mininet/examples$ sudo mn --mac
*** 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 ...
*** Starting CLI:
mininet>
```



8.

```
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=4.23 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=1.47 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.108 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9009ms
rtt min/avg/max/mdev = 0.078/0.777/4.236/1.268 ms
mininet>
```

9.

```
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=4.23 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.113 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.078 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=1.47 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.084 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.108 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9009ms
rtt min/avg/max/mdev = 0.078/0.777/4.236/1.268 ms
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> h1 phyton -m SimpleHTTPServer 80 &
bash: phyton: command not found
mininet> h1 python -m SimpleHTTPServer 80 &
mininet> h2 wget -o - h1
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['22.0 Gbits/sec', '22.0 Gbits/sec']
mininet>
```

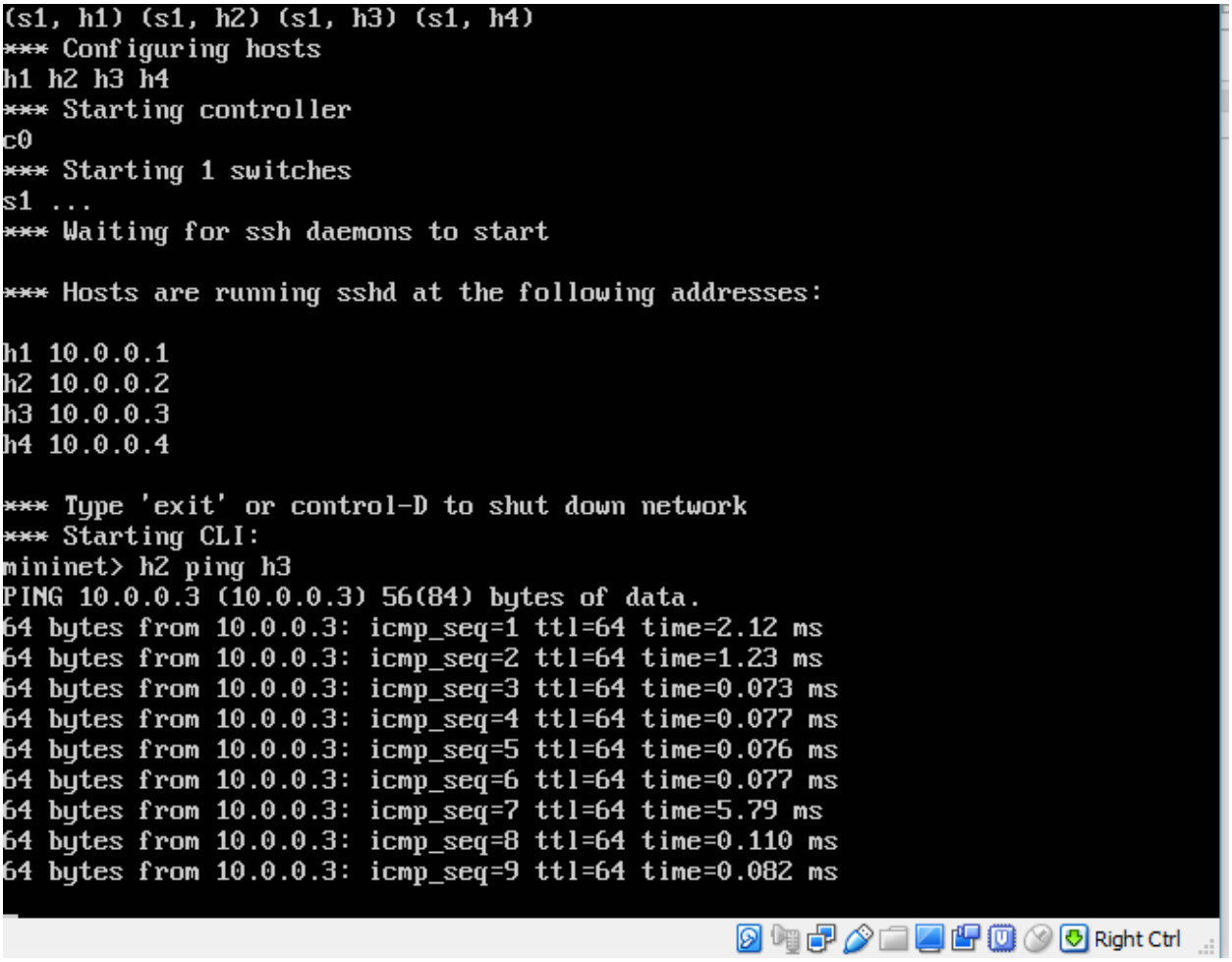
10.

```
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['22.0 Gbits/sec', '22.0 Gbits/sec']
mininet> link s1 h1 down
mininet> link s1 h1 up
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 309.639 seconds
mininet@mininet-vm:~/mininet/examples$
```

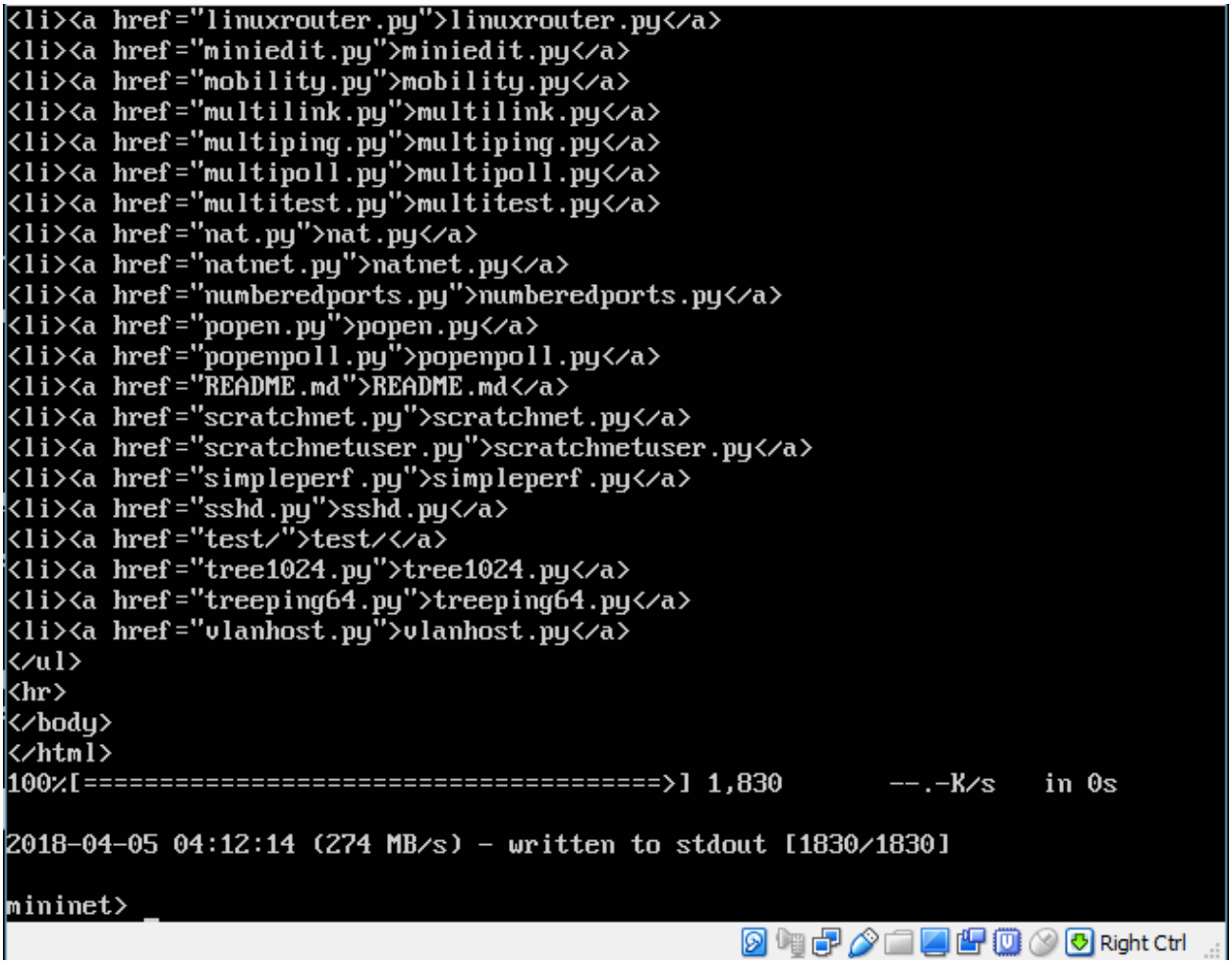
11.

```
mininet@mininet-vm:~/mininet/examples$ sudo ~/mininet/examples/sshd.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(s1, h1) (s1, h2) (s1, h3) (s1, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for ssh daemons to start

*** Hosts are running sshd at the following addresses:

h1 10.0.0.1
h2 10.0.0.2
h3 10.0.0.3
h4 10.0.0.4

*** Type 'exit' or control-D to shut down network
*** Starting CLI:
mininet>
```

12.


(s1, h1) (s1, h2) (s1, h3) (s1, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for ssh daemons to start
*** Hosts are running sshd at the following addresses:
h1 10.0.0.1
h2 10.0.0.2
h3 10.0.0.3
h4 10.0.0.4
*** Type 'exit' or control-D to shut down network
*** Starting CLI:
mininet> h2 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=2.12 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=1.23 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.073 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.077 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.076 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=0.077 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=5.79 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=0.110 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=0.082 ms
Right Ctrl

13.


linuxrouter.py
miniedit.py
mobility.py
multilink.py
multiping.py
multipoll.py
multitest.py
nat.py
natnet.py
numberedports.py
popen.py
popenpoll.py
README.md
scratchnet.py
scratchnetuser.py
simpleperf.py
sshd.py
test/
tree1024.py
treeping64.py
vlanhost.py

<hr>
</body>
</html>
100%[======================================>] 1,830 --.-K/s in 0s
2018-04-05 04:12:14 (274 MB/s) - written to stdout [1830/1830]
mininet>
Right Ctrl