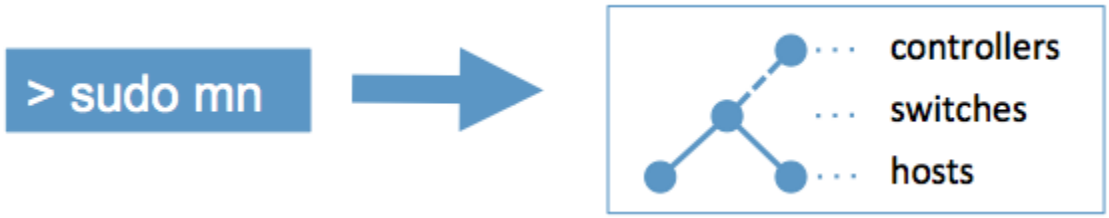
Introduction

- A network emulator which creates realistic virtual network
- Runs real kernel, switch and application code on a single machine



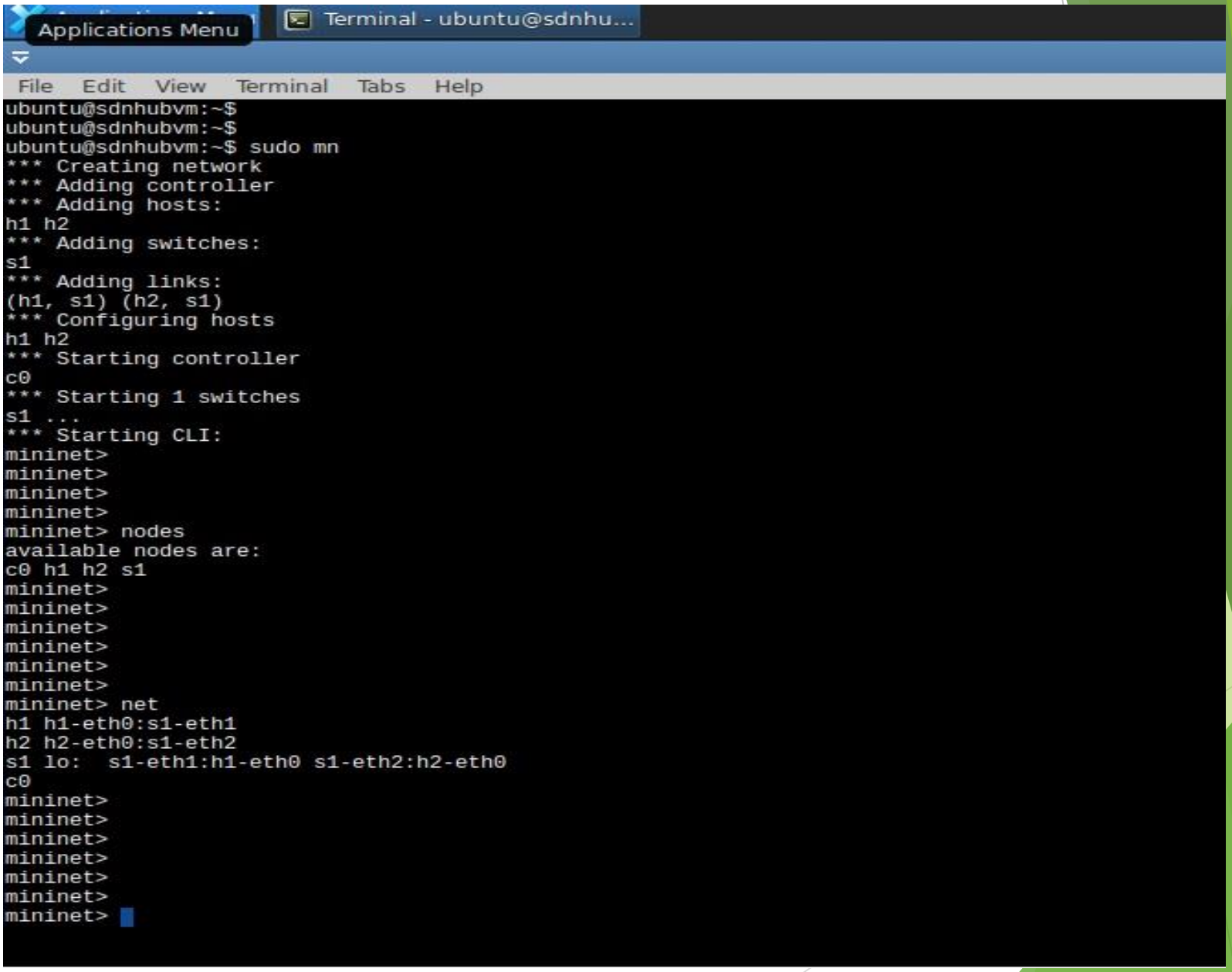
- Provides both Command Line Interface (CLI) and Application Programming Interface (API)
  - CLI: interactive commanding
  - API: automation
- Abstraction
  - Host: emulated as an OS level process
  - Switch: emulated by using software-based switch
    - E.g., Open vSwitch, SoftSwitch

Platforms for Network/Systems Teaching (and Research)

Platform	Advantages	Disadvantages
Hardware Testbed	fast accurate: "ground truth"	expensive shared resource? hard to reconfigure hard to change hard to download
Simulator	inexpensive, flexible detailed (or abstract!) easy to download virtual time (can be "faster" than reality)	may require app changes might not run OS code detail != accuracy may not be "believable" may be slow/non-interactive
Emulator	**inexpensive, flexible** **real code** **reasonably accurate** **easy to download** **fast/interactive usage**	slower than hardware experiments may not fit possible inaccuracy from multiplexing

Basic Commands

1) sudo mn //Start a minimal topology
2) nodes //Display nodes
3) net // Display links
4) dump // Dump information about all nodes
5) pingall //An all-pairs ping
6) If first string typed into the Mininet CLI is a host, switch or controller name, command is executed on that node
    1) h1 ifconfig –a
    2) h2 Ps -a
    3) h1 ping –c3 h2

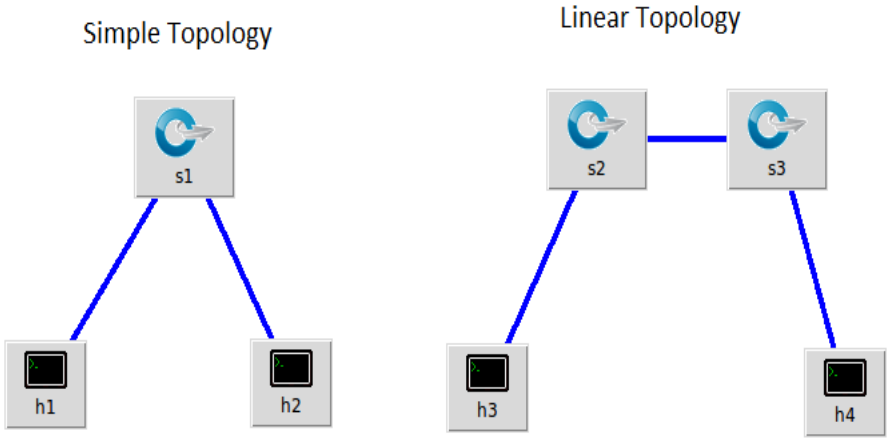

Applications Menu
Terminal - ubuntu@sdnhu...
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$
ubuntu@sdnhubvm:~$
ubuntu@sdnhubvm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
mininet>
mininet>
mininet>
mininet> nodes
available nodes are:
c0 h1 h2 s1
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
c0
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>
mininet>

```
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=5636>
<Host h2: h2-eth0:10.0.0.2 pid=5642>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=5649>
<Controller c0: 127.0.0.1:6633 pid=5629>
mininet>
mininet>
mininet>
mininet>
mininet> h1 ifconfig -a
h1-eth0   Link encap:Ethernet  HWaddr de:ac:f3:4b:f1:5e
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::dcac:f3ff:fe4b:f15e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:558 (558.0 B)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          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)

mininet>
mininet>
mininet>
mininet> h1 ps -a
  PID TTY          TIME CMD
 5621 pts/0    00:00:00 sudo
 5622 pts/0    00:00:00 mn
 5689 pts/1    00:00:00 controller
 5714 pts/3    00:00:00 ps
mininet>
mininet>
mininet>
mininet>
mininet> s1 ps -a
  PID TTY          TIME CMD
 5621 pts/0    00:00:00 sudo
 5622 pts/0    00:00:00 mn
 5689 pts/1    00:00:00 controller
 5716 pts/8    00:00:00 ps
mininet>
```

```
mininet> h1 ping -c3 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=4.68 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.696 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.101 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.101/1.825/4.680/2.033 ms
mininet>
mininet>
mininet>
mininet>
mininet>
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

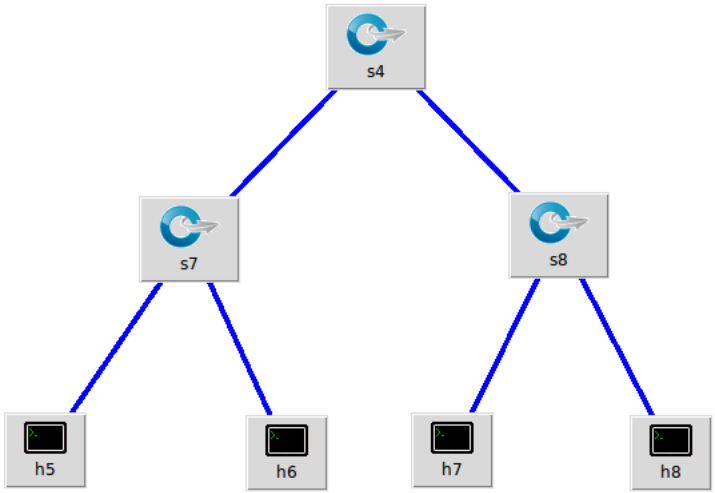
Advanced Commands

1) **Changing Topology Size and Type**
   - sudo mn --topo single, 2
   - sudo mn --topo linear, 2

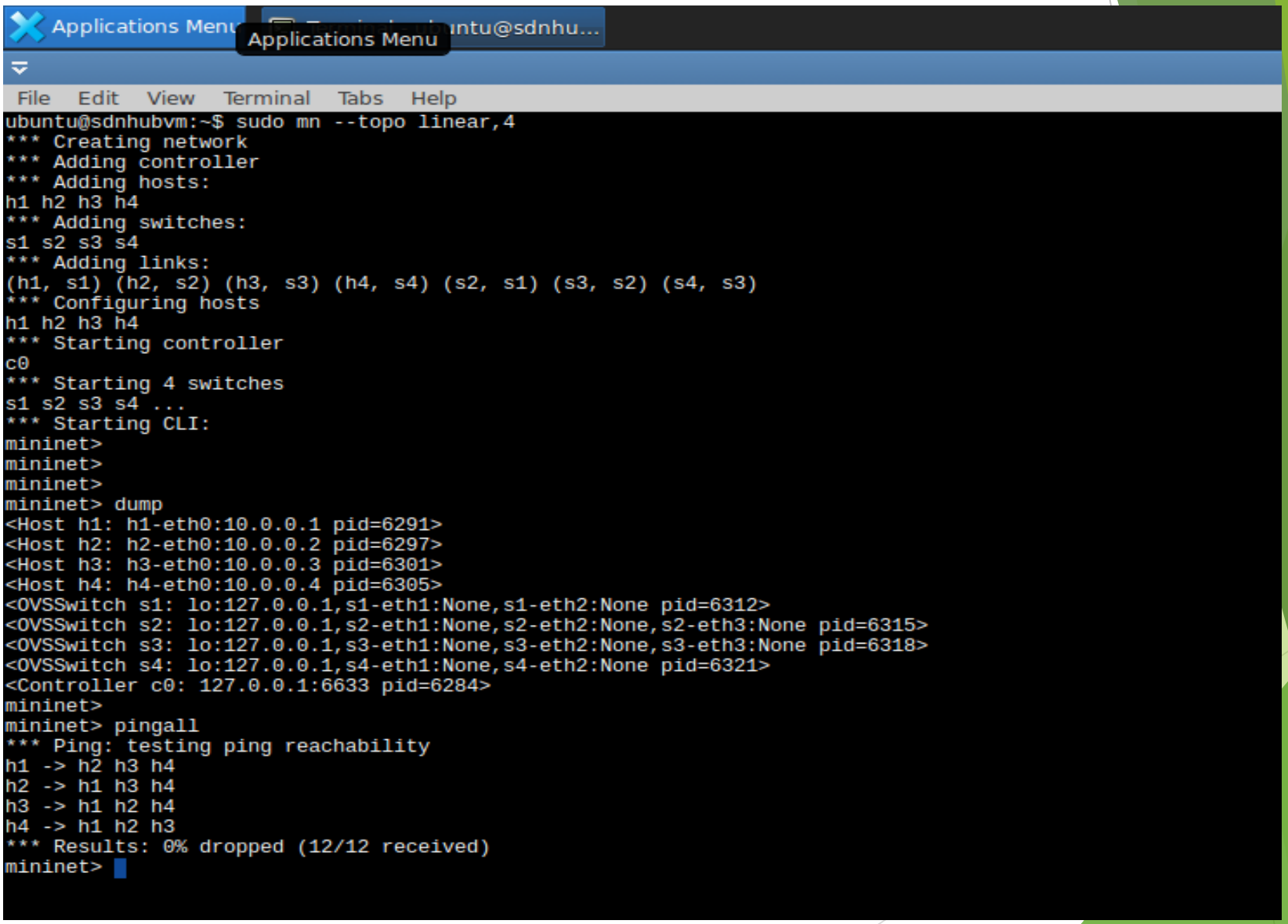


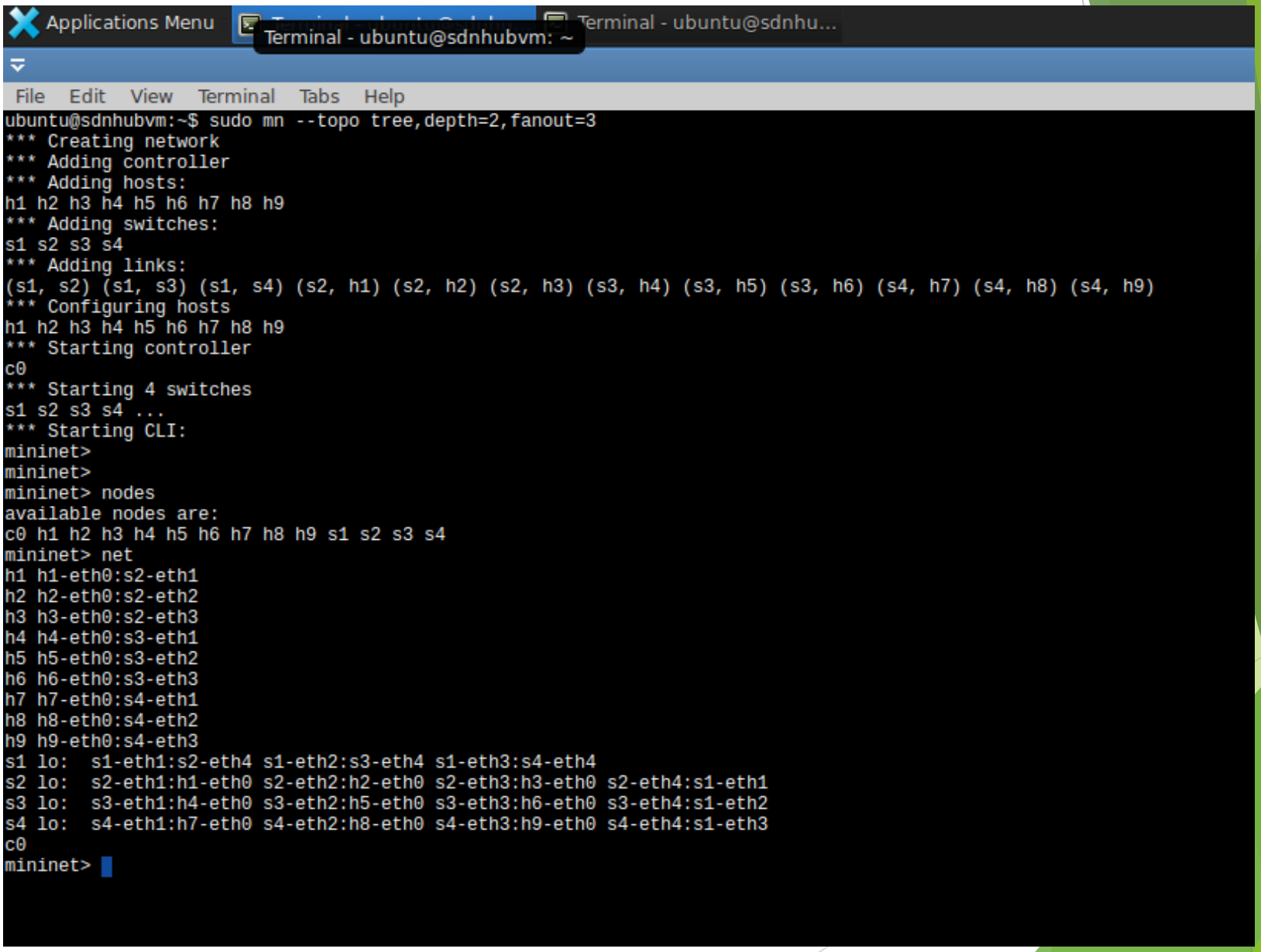
Advanced Commands

- sudo mn --topo tree,depth=2,fanout=2



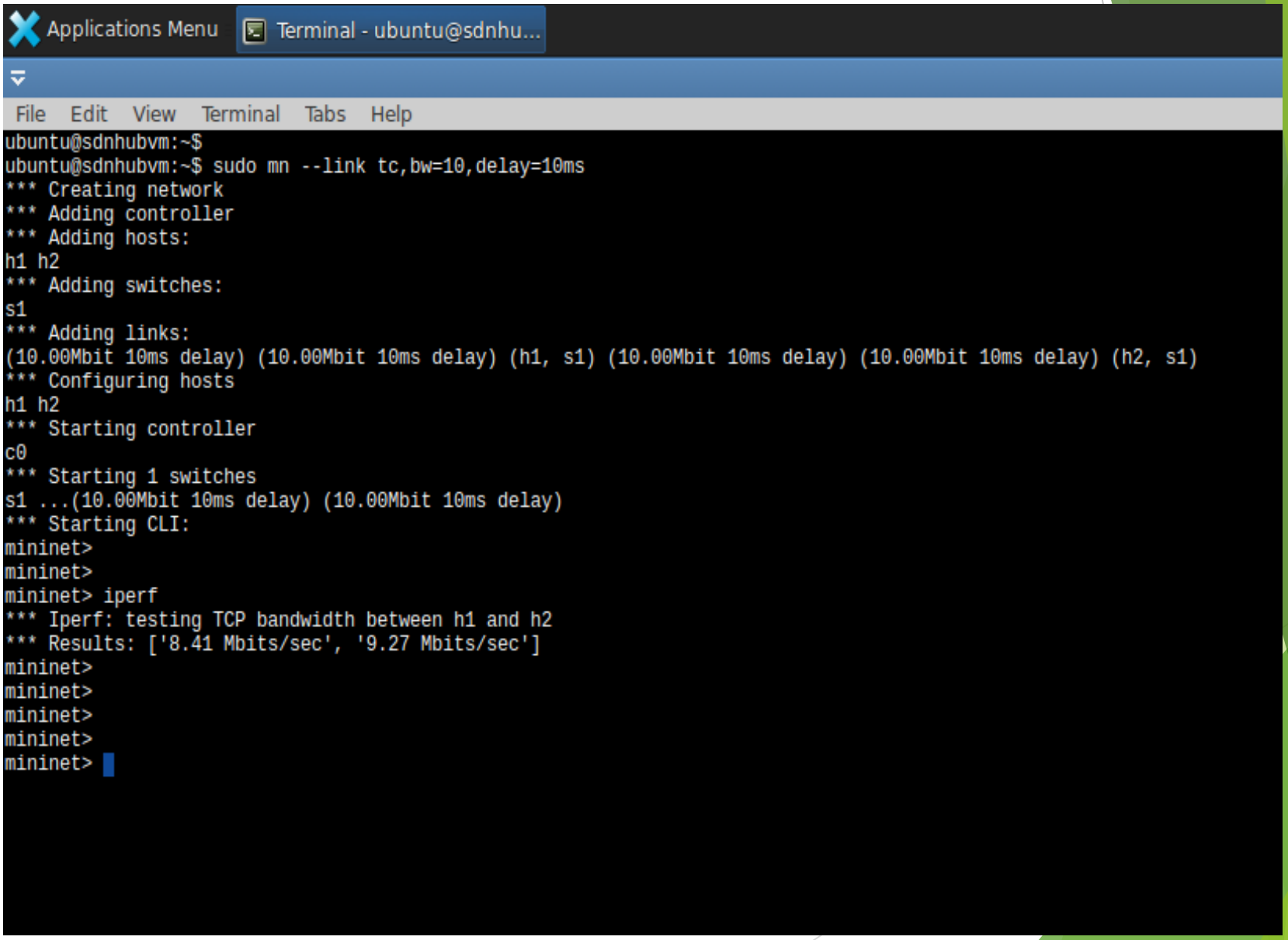
Tree Topology, depth=2, fanout=2

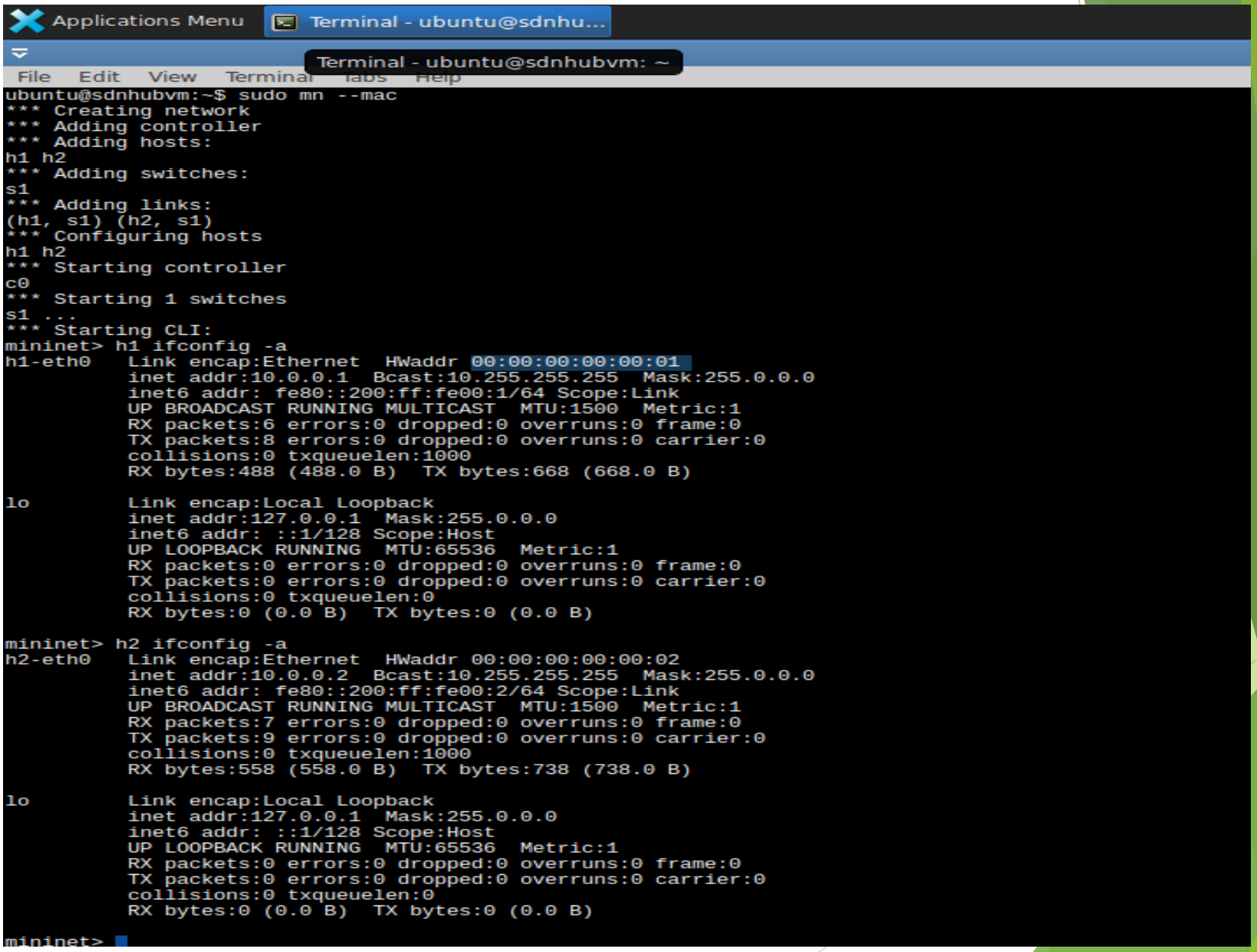

Applications Menu
Applications Menu
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$ sudo mn --topo linear,4
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet>
mininet>
mininet>
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=6291>
<Host h2: h2-eth0:10.0.0.2 pid=6297>
<Host h3: h3-eth0:10.0.0.3 pid=6301>
<Host h4: h4-eth0:10.0.0.4 pid=6305>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=6312>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=6315>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=6318>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=6321>
<Controller c0: 127.0.0.1:6633 pid=6284>
mininet>
mininet> pingall
*** 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)
mininet>


Applications Menu
Terminal - ubuntu@sdnhubvm: ~
Terminal - ubuntu@sdnhu...
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$ sudo mn --topo tree,depth=2,fanout=3
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(s1, s2) (s1, s3) (s1, s4) (s2, h1) (s2, h2) (s2, h3) (s3, h4) (s3, h5) (s3, h6) (s4, h7) (s4, h8) (s4, h9)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet>
mininet>
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 h7 h8 h9 s1 s2 s3 s4
mininet> net
h1 h1-eth0:s2-eth1
h2 h2-eth0:s2-eth2
h3 h3-eth0:s2-eth3
h4 h4-eth0:s3-eth1
h5 h5-eth0:s3-eth2
h6 h6-eth0:s3-eth3
h7 h7-eth0:s4-eth1
h8 h8-eth0:s4-eth2
h9 h9-eth0:s4-eth3
s1 lo: s1-eth1:s2-eth4 s1-eth2:s3-eth4 s1-eth3:s4-eth4
s2 lo: s2-eth1:h1-eth0 s2-eth2:h2-eth0 s2-eth3:h3-eth0 s2-eth4:s1-eth1
s3 lo: s3-eth1:h4-eth0 s3-eth2:h5-eth0 s3-eth3:h6-eth0 s3-eth4:s1-eth2
s4 lo: s4-eth1:h7-eth0 s4-eth2:h8-eth0 s4-eth3:h9-eth0 s4-eth4:s1-eth3
c0
mininet>

Advanced Commands

1) **Link variations**
    1) sudo mn --link tc,bw=10,delay=10ms
2) **Custom Topologies**
    1) sudo mn --custom [file_path] --topo mytopo
3) **Switch Types**
    1) sudo mn --switch ovsk
4) **Remote Controller**
    1) sudo mn --controller=remote,ip=127.0.0.1,port=6633


Applications Menu
Terminal - ubuntu@sdnhu...
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$
ubuntu@sdnhubvm:~$ 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) (10.00Mbit 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>
mininet>
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['8.41 Mbits/sec', '9.27 Mbits/sec']
mininet>
mininet>
mininet>
mininet>
mininet>


Applications Menu
Terminal - ubuntu@sdnhu...
Terminal - ubuntu@sdnhubvm: ~
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~$ 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> h1 ifconfig -a
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
inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:6 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:488 (488.0 B) TX bytes:668 (668.0 B)
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
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)
mininet> h2 ifconfig -a
h2-eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:02
inet addr:10.0.0.2 Bcast:10.255.255.255 Mask:255.0.0.0
inet6 addr: fe80::200:ff:fe00:2/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:7 errors:0 dropped:0 overruns:0 frame:0
TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:558 (558.0 B) TX bytes:738 (738.0 B)
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
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)
mininet>

Python API

1) Low-level API
   - base node and link classes
   - Switches, Host, Links
2) Mid-level API
   - Mininet object which serves as a container for nodes and links
   - addSwitch(), addLink(), addHost(), start(),stop()
3) High-level API
   - Topo class, to create parametrized topologies

```python
#!/usr/bin/python

from mininet.topo import Topo
from mininet.net import Mininet
from mininet.util import dumpNodeConnections
from mininet.log import setLogLevel

class SingleSwitchTopo(Topo):
    "Single switch connected to n hosts."
    def build(self, n=2):
        switch = self.addSwitch('s1')
        # Python's range(N) generates 0..N-1
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink(host, switch)

def simpleTest():
    "Create and test a simple network"
    topo = SingleSwitchTopo(n=4)
    net = Mininet(topo)
    net.start()
    print "Dumping host connections"
    dumpNodeConnections(net.hosts)
    print "Testing network connectivity"
    net.pingAll()
    net.stop()

if __name__ == '__main__':
    # Tell mininet to print useful information
    setLogLevel('info')
    simpleTest()
```

Topo:Base class
Mininet: main class
addSwitch(): adds a switch
addHost(): adds a host
addLink(): adds a bidirectional link
start(): starts your network
pingAll(): tests connectivity
stop(): stops network
net.hosts: all the hosts in a network
net.get(h1): retrieves a node by name
net.iperf((h1, h2)) : bandwidth
h1.cmd('ifconfig') : run a cmd in host
h1.IP() : get IP address
h1.MAC():get MAC address

Miniedit - GUI

- A GUI application which eases the Mininet topology generation
- Either save the topology or export as a Mininet python script
- sudo ~/mininet/examples/miniedit.py