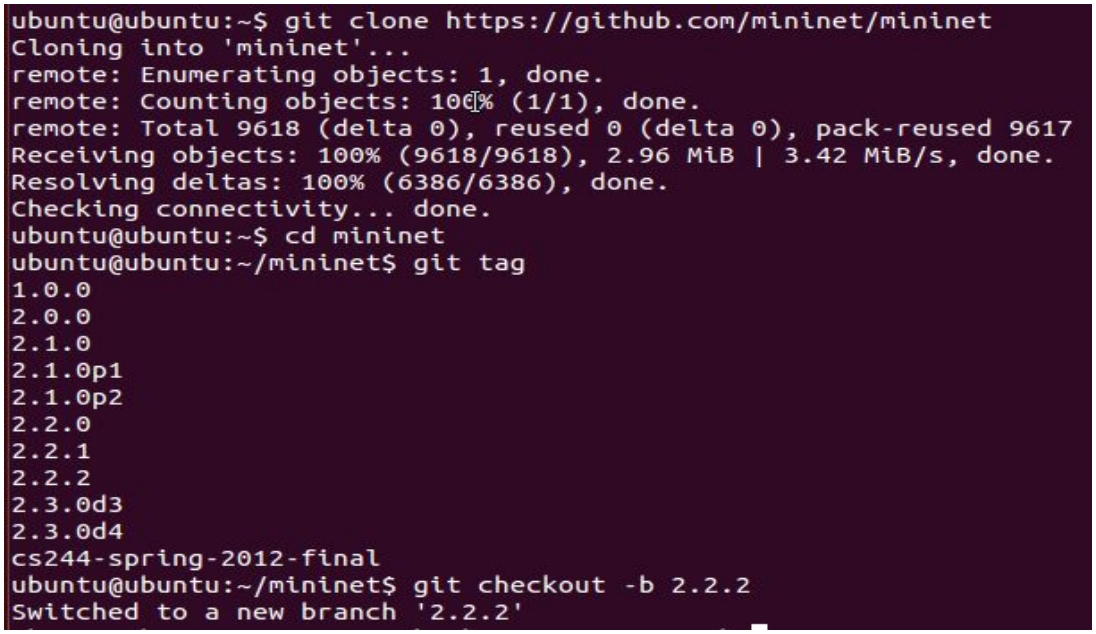
1.1.1 Installing mininet



```
ubuntu@ubuntu:~/mininet$ util/install.sh
Detected Linux distribution: Ubuntu 14.04 trusty amd64
python is version 2
Installing all packages except for -eix (doxypy, ivs, nox-classic)...
Install Mininet-compatible kernel if necessary
[sudo] password for ubuntu:
Ign http://us.archive.ubuntu.com trusty InRelease
Get:1 http://us.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:2 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Ign http://extras.ubuntu.com trusty InRelease
Hit http://us.archive.ubuntu.com trusty-backports InRelease
Hit http://us.archive.ubuntu.com trusty Release.gpg
Get:3 http://extras.ubuntu.com trusty Release.gpg [72 B]
Get:4 http://us.archive.ubuntu.com trusty-updates/main Sources [427 kB]
Hit http://extras.ubuntu.com trusty Release
Get:5 http://security.ubuntu.com trusty-security/main Sources [169 kB]
Hit http://extras.ubuntu.com trusty/main Sources
Get:6 http://us.archive.ubuntu.com trusty-updates/restricted Sources [6,322 B]
Get:7 http://us.archive.ubuntu.com trusty-updates/universe Sources [231 kB]
Hit http://extras.ubuntu.com trusty/main amd64 Packages
Get:8 http://us.archive.ubuntu.com trusty-updates/multiverse Sources [7,435 B]
Get:9 http://security.ubuntu.com trusty-security/restricted Sources [4,931 B]
Get:10 http://us.archive.ubuntu.com trusty-updates/main amd64 Packages [1,144 kB]
Hit http://extras.ubuntu.com trusty/main i386 Packages
Get:11 http://security.ubuntu.com trusty-security/universe Sources [101 kB]
Get:12 http://us.archive.ubuntu.com trusty-updates/restricted amd64 Packages [17.2 kB]
Get:13 http://security.ubuntu.com trusty-security/multiverse Sources [3,265 B]
Get:14 http://us.archive.ubuntu.com trusty-updates/universe amd64 Packages [516 kB]
Get:15 http://us.archive.ubuntu.com trusty-updates/multiverse amd64 Packages [14.6 kB]
Get:16 http://security.ubuntu.com trusty-security/main amd64 Packages [809 kB]
Get:17 http://us.archive.ubuntu.com trusty-updates/main i386 Packages [1,069 kB]
Get:18 http://us.archive.ubuntu.com trusty-updates/restricted i386 Packages [17.1 kB]
Get:19 http://us.archive.ubuntu.com trusty-updates/universe i386 Packages [499 kB]
Get:20 http://us.archive.ubuntu.com trusty-updates/multiverse i386 Packages [15.0 kB]
Get:21 http://security.ubuntu.com trusty-security/restricted amd64 Packages [14.2 kB]
```

```
Building dependency tree...
Reading state information...
linux-image-4.4.0-142-generic is already the newest version.
linux-image-4.4.0-142-generic set to manually installed.
The following package was automatically installed and is no longer required:
  linux-image-extra-4.4.0-134-generic
Use 'apt-get autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
Installing Mininet dependencies
Reading package lists...
Building dependency tree...
Reading state information...
ethtool is already the newest version.
gcc is already the newest version.
help2man is already the newest version.
make is already the newest version.
pep8 is already the newest version.
psmisc is already the newest version.
pyflakes is already the newest version.
pylint is already the newest version.
python-tk is already the newest version.
telnet is already the newest version.
xterm is already the newest version.
cgroup-bin is already the newest version.
iperf is already the newest version.
socat is already the newest version.
python-pexpect is already the newest version.
python-setuptools is already the newest version.
ssh is already the newest version.
The following package was automatically installed and is no longer required:
  linux-image-extra-4.4.0-134-generic
Use 'apt-get autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.
Reading package lists...
Building dependency tree...
Reading state information...
iproute2 is already the newest version.
The following package was automatically installed and is no longer required:
```

```
flag during linking and do at least one of the following:
   - add LIBDIR to the `LD_LIBRARY_PATH' environment variable
     during execution
   - add LIBDIR to the `LD_RUN_PATH' environment variable
     during linking
   - use the `-Wl,-rpath -Wl,LIBDIR' linker flag
   - have your system administrator add LIBDIR to `/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/home/ubuntu/oflops/example_modules/snmp_cpu'
make[2]: Leaving directory `/home/ubuntu/oflops/example_modules/snmp_cpu'
make[2]: Entering directory `/home/ubuntu/oflops/example_modules'
make[3]: Entering directory `/home/ubuntu/oflops/example_modules'
make[3]: Nothing to be done for `install-exec-am'.
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/home/ubuntu/oflops/example_modules'
make[2]: Leaving directory `/home/ubuntu/oflops/example_modules'
make[1]: Leaving directory `/home/ubuntu/oflops/example_modules'
Making install in cbench
make[1]: Entering directory `/home/ubuntu/oflops/cbench'
make[2]: Entering directory `/home/ubuntu/oflops/cbench'
 /bin/mkdir -p '/usr/local/bin'
  /bin/bash ../libtool   --mode=install /usr/bin/install -c cbench '/usr/local/b
in'
libtool: install: /usr/bin/install -c cbench /usr/local/bin/cbench
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/home/ubuntu/oflops/cbench'
make[1]: Leaving directory `/home/ubuntu/oflops/cbench'
Making install in doc
make[1]: Entering directory `/home/ubuntu/oflops/doc'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/home/ubuntu/oflops/doc'
Enjoy Mininet!
ubuntu@ubuntu:~/mininet$
```

1.1.2 To check options available:

```
ubuntu@ubuntu:~/mininet$ sudo mn -h
Usage: mn [options]
(type mn -h for details)

The mn utility creates Mininet network from the command line. It can create
parametrized topologies, invoke the Mininet CLI, and run tests.

Options:
  -h, --help            show this help message and exit
  --switch=SWITCH       default|ivs|lxbr|ovs|ovsbr|ovsk|user[,param=value...]
                        ovs=OVSSwitch default=OVSSwitch ovsk=OVSSwitch
                        lxbr=LinuxBridge user=UserSwitch ivs=IVSSwitch
                        ovsbr=OVSBridge
  --host=HOST           cfs|proc|rt[,param=value...]
                        rt=CPULimitedHost{'sched': 'rt'} proc=Host
                        cfs=CPULimitedHost{'sched': 'cfs'}
  --controller=CONTROLLER
                        default|none|nox|ovsc|ref|remote|ryu[,param=value...]
                        ovsc=OVSController none=NullController
                        remote=RemoteController default=DefaultController
                        nox=NOX ryu=Ryu ref=Controller
  --link=LINK           default|ovs|tc|tcu[,param=value...] default=Link
                        ovs=OVSLink tcu=TCULink tc=TCLink
  --topo=TOPO           linear|minimal|reversed|single|torus|tree[,param=value
                        ...] linear=LinearTopo torus=TorusTopo tree=TreeTopo
  -c, --clean           clean and exit
  --custom=CUSTOM       read custom classes or params from .py file(s)
  --test=TEST           none|build|all|iperf|pingpair|iperfudp|pingall
  -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!)
ubuntu@ubuntu:~/mininet$
```

1.1.3 Start a minimal topology and enter the CLI:

```
ubuntu@ubuntu:~$ sudo mn
[sudo] password for ubuntu:
*** 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>
```

1.1.4 Display nodes

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

1.1.5 Display links

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

1.1.6 Dump information about all nodes:

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2119>
<Host h2: h2-eth0:10.0.0.2 pid=2122>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=2127>
<Controller c0: 127.0.0.1:6653 pid=2112>
```

1.2 Mininet Networking

1.2.1 Check the network information of mininet host and switches

```
mininet> h1 ifconfig -a
h1-eth0   Link encap:Ethernet  HWaddr 4e:38:35:0d:b8:96
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::4c38:35ff:fe0d:b896/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 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:6703 (6.7 KB)  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:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

mininet>
```

```
mininet> s1 ifconfig -a
eth0      Link encap:Ethernet  HWaddr fa:16:3e:17:37:35
          inet addr:192.168.0.10  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe17:3735/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:101 errors:0 dropped:0 overruns:0 frame:0
          TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13329 (13.3 KB)  TX bytes:18221 (18.2 KB)

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:260 errors:0 dropped:0 overruns:0 frame:0
          TX packets:260 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:23184 (23.1 KB)  TX bytes:23184 (23.1 KB)

ovs-system Link encap:Ethernet  HWaddr ca:8f:22:bc:aa:45
          BROADCAST 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:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s1        Link encap:Ethernet  HWaddr 82:60:89:00:dc:44
          inet6 addr: fe80::9036:e7ff:fed5:cbf3/64 Scope:Link
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:780 (780.0 B)  TX bytes:648 (648.0 B)
```

```
s1-eth2   Link encap:Ethernet  HWaddr be:eb:f2:c2:ec:69
          inet6 addr: fe80::bceb:f2ff:fec2:ec69/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:648 (648.0 B)  TX bytes:12555 (12.5 KB)

mininet>
```

1.2.2 Check the processes running on the host

```
mininet> h1 ps -a
  PID TTY          TIME CMD
 2106 pts/0    00:00:00 sudo
 2107 pts/0    00:00:00 mn
 2165 pts/1    00:00:00 controller
 2219 pts/9    00:00:00 ps
```

```
mininet> s1 ps -a
  PID TTY          TIME CMD
 6400 pts/0    00:00:00 sudo
 6401 pts/0    00:00:00 mn
 6460 pts/4    00:00:00 controller
 6489 pts/15   00:00:00 ps
```

1.2.3 Test connectivity between hosts

```
mininet> h1 ping -c 1 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=3.37 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.374/3.374/3.374/0.000 ms
```

```
mininet> h1 ping -c 1 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=0.530 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.530/0.530/0.530/0.000 ms
```

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
```

1.2.4 Running simple web server and client

```
mininet> h1 python -m SimpleHTTPServer 80 &
mininet> h2 wget -O - h1
--2019-02-16 09:55:26--  http://10.0.0.1/
Connecting to 10.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1410 (1.4K) [text/html]
Saving to: 'STDOUT'

 0% [                                       ] 0           --.-K/s              <
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".compiz/">.compiz/</a>
<li><a href=".config/">.config/</a>
<li><a href=".dmrc">.dmrc</a>
<li><a href=".gconf/">.gconf/</a>
<li><a href=".ICEauthority">.ICEauthority</a>
<li><a href=".local/">.local/</a>
<li><a href=".mininet_history">.mininet_history</a>
<li><a href=".profile">.profile</a>
<li><a href=".rnd">.rnd</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".wireshark/">.wireshark/</a>
<li><a href=".Xauthority">.Xauthority</a>
<li><a href=".xsession-errors">.xsession-errors</a>
<li><a href=".xsession-errors.old">.xsession-errors.old</a>
```

```
<li><a href=".xsession-errors.old">.xsession-errors.old</a>
<li><a href="Desktop/">Desktop/</a>
<li><a href="Documents/">Documents/</a>
<li><a href="Downloads/">Downloads/</a>
<li><a href="examples.desktop">examples.desktop</a>
<li><a href="mininet/">mininet/</a>
<li><a href="Music/">Music/</a>
<li><a href="oflops/">oflops/</a>
<li><a href="oftest/">oftest/</a>
<li><a href="openflow/">openflow/</a>
<li><a href="Pictures/">Pictures/</a>
<li><a href="pox/">pox/</a>
<li><a href="Public/">Public/</a>
<li><a href="Templates/">Templates/</a>
<li><a href="Videos/">Videos/</a>
</ul>
<hr>
</body>
</html>
100%[======================================>] 1,410       --.-K/s   in 0s

2019-02-16 09:55:26 (62.4 MB/s) - written to stdout [1410/1410]

mininet> h1 kill %python
Serving HTTP on 0.0.0.0 port 80 ...
10.0.0.2 - - [16/Feb/2019 09:55:26] "GET / HTTP/1.1" 200 -
```

1.2.5 Mininet Cleanup

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 2.560 seconds
```

```
ubuntu@ubuntu:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflowd ovs-controllerovs-testcont
roller udpbwtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflowd ovs-controllerovs-testc
ontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
***  Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
```

1.3 Advanced Options with mininet and OpenFlow switch

1.3.1 Run a regression test

```
ubuntu@ubuntu:~$ 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 6.052 seconds
```

1.3.2 Another useful test is iperf

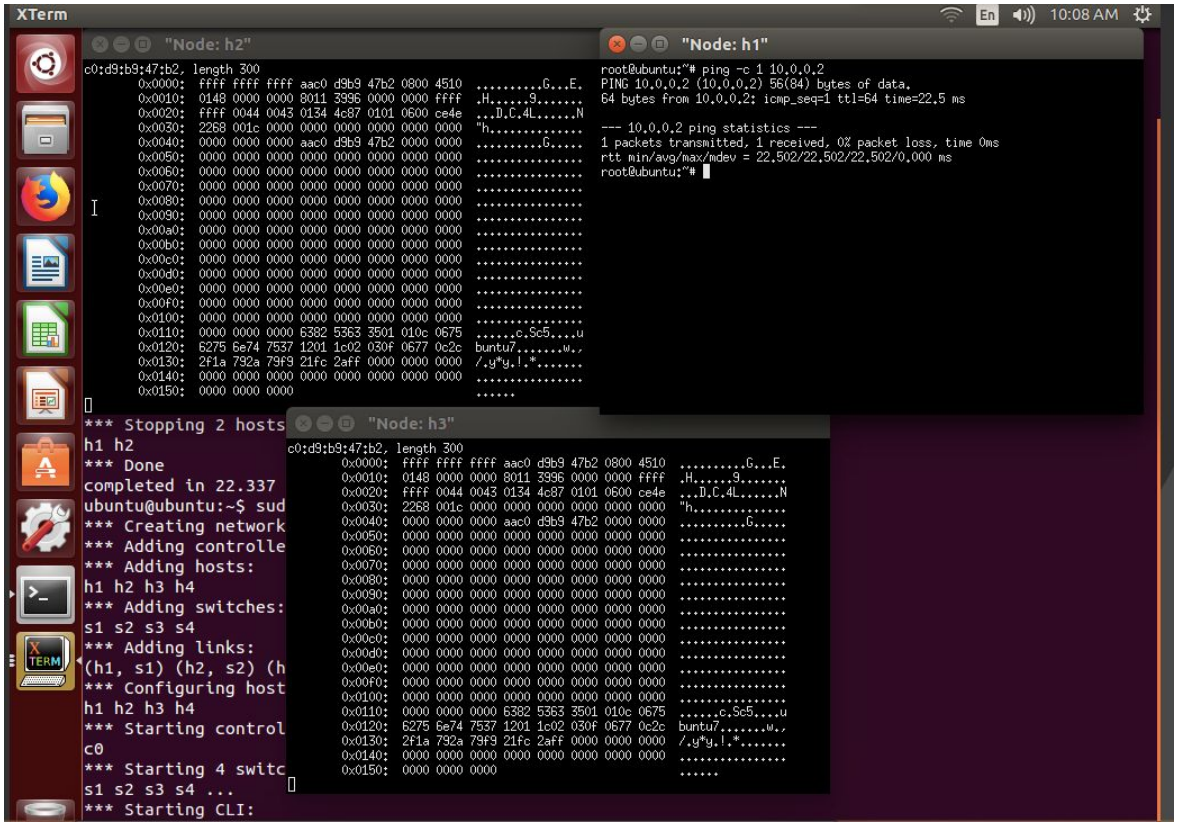
```
ubuntu@ubuntu:~$ sudo mn --test iperf
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Iperf: testing TCP bandwidth between h1 and h2
.*** Results: ['6.32 Gbits/sec', '6.33 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 11.670 seconds
```

1.3.3 Changing topology size

```
ubuntu@ubuntu:~$ 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 5.854 seconds
```

1.3.4 Changing topology type

```
ubuntu@ubuntu:~$ sudo mn --test pingall --topo linear,4
[sudo] password for ubuntu:
*** 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 7.747 seconds
```



1.3.5 Check ovs installation

```
ubuntu@ubuntu:~$ sudo ovs-vsctl show
978190d8-49be-4259-a872-1dbdd538d08f
    ovs_version: "2.0.2"
```

1.3.6 Check network bandwidth with ovs

```
ubuntu@ubuntu:~$ 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: ['6.51 Gbits/sec', '6.51 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 11.172 seconds
```

1.3.7 Check Link Variations

```
ubuntu@ubuntu:~$ 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> iperrf
*** Unknown command: iperrf
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['9.48 Mbits/sec', '11.9 Mbits/sec']
mininet> h1 ping -c 10 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=42.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=40.6 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=40.2 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=40.2 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=40.1 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=40.2 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=40.4 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=40.1 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=40.1 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 40.148/40.507/42.419/0.694 ms
```

1.3.8 Add verbosity to mininet output

```
ubuntu@ubuntu:~$ sudo mn -v debug
*** errRun: ['which', 'controller']
/usr/local/bin/controller
  0*** errRun: ['grep', '-c', 'processor', '/proc/cpuinfo']
1
  0*** Setting resource limits
*** Creating network
*** Adding controller
*** errRun: ['which', 'mnexec']
/usr/bin/mnexec
  0*** errRun: ['which', 'ifconfig']
/sbin/ifconfig
  0_popen ['mnexec', '-cd', 'env', 'PS1=\x7f', 'bash', '--norc', '--noediting', '-is', 'mininet:c0'] 3076*
** c0 : ('unset HISTFILE; stty -echo; set +m',)
unset HISTFILE; stty -echo; set +m
*** errRun: ['which', 'telnet']
/usr/bin/telnet
  0*** c0 : ('echo A | telnet -e A 127.0.0.1 6653',)
Telnet escape character is 'A'.
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
*** Adding hosts:
*** errRun: ['which', 'mnexec']
/usr/bin/mnexec
  0*** errRun: ['which', 'ifconfig']
/sbin/ifconfig
  0_popen ['mnexec', '-cdn', 'env', 'PS1=\x7f', 'bash', '--norc', '--noediting', '-is', 'mininet:h1'] 3083
*** h1 : ('unset HISTFILE; stty -echo; set +m',)
unset HISTFILE; stty -echo; set +m
h1 _popen ['mnexec', '-cdn', 'env', 'PS1=\x7f', 'bash', '--norc', '--noediting', '-is', 'mininet:h2'] 3086
*** h2 : ('unset HISTFILE; stty -echo; set +m',)
unset HISTFILE; stty -echo; set +m
h2
*** Adding switches:
*** errRun: ['which', 'ovs-vsctl']
```

```
ubuntu@ubuntu:~$ sudo mn -v output
mininet> exit
ubuntu@ubuntu:~$
```

1.4 Custom topologies in mininet

1.4.1 Example of a 2-host, 2-switch custom topology

```
ubuntu@ubuntu:~/mininet/custom$ sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --test p
ingall
*** 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 6.136 seconds
```

1.4.2 Custom Topology - File Attached in Canvas.

1.4.3 Testing Custom topology

```
ubuntu@ubuntu:~$ sudo mn --custom ~/mytopo.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s3) (h2, s3) (h2, s4) (h3, s4) (h4, s4) (h5, s4) (h5, s5) (h6, s5) (s3, s1) (s4, s1) (s4, s2) (s5, s2
)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h6
*** Results: ['4.74 Gbits/sec', '4.73 Gbits/sec']
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet>
```