Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 04
Report Name: Introduction to Mininet.
Course Title: Network Planning and Design Lab.
Course Code: ICT-3208

Submitted By	Submitted To
Name: **S.M. Hazrat Ali** ID: **IT-17016** Session: 2016-17 3rd Year 2nd Semester Dept. of Information & Communication Technology, MBSTU.	**Nazrul Islam** **Assistant Professor** Dept. of Information & Communication Technology, MBSTU.

Objective: In this lab we will learn about installation process of Mininet in Linux. After completion of installation .Apply some mininet command from Mininet Workthrough.

1. Installation process:

$ sudo apt-get install git

```
[sudo] password for binodon:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.7).
The following packages were automatically installed and are no longer required:
  efibootmgr gir1.2-geocodeglib-1.0 libfwup1 libllvm9 libpython-all-dev
  libpython-dev libpython2.7-dev python2.7-dev ubuntu-web-launchers
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n]
```

2. $ 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>
```

3. mininet> help

```
mininet> help

Documented commands (type help <topic>):
========================================
EOF    gterm  iperfudp  nodes        pingpair      py      switch
dpctl  help   link      noecho       pingpairfull  quit    time
dump   intfs  links     pingall      ports         sh      x
exit   iperf  net       pingallfull  px            source  xterm

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
```

4. mininet> nodes

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

5. mininet> net

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

6. mininet> net

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

7. mininet> h1 ifconfig -a

```
mininet> h1 ifconfig -a
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::5825:dbff:fe24:71e8  prefixlen 64  scopeid 0x20<link>
        ether 5a:25:db:24:71:e8  txqueuelen 1000  (Ethernet)
        RX packets 59  bytes 7140 (7.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1006 (1.0 KB)
        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>
```

8. mininet> s1 ifconfig -a

```
mininet> s1 ifconfig -a
enp0s25: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether d4:c9:ef:e9:da:ca  txqueuelen 1000  (Ethernet)
        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
        device interrupt 17  memory 0xd4700000-d4720000

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 22003  bytes 2774203 (2.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22003  bytes 2774203 (2.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether d6:fe:27:af:0d:9a  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
```

9. mininet> h1 ps -a

```
mininet> h1 ps -a
  PID TTY          TIME CMD
 1151 tty1     00:00:00 gnome-session-b
 1201 tty1     00:00:08 gnome-shell
 1399 tty1     00:00:00 Xwayland
 1444 tty1     00:00:00 ibus-daemon
 1447 tty1     00:00:00 ibus-dconf
 1449 tty1     00:00:00 ibus-x11
 1474 tty1     00:00:00 gsd-xsettings
 1477 tty1     00:00:00 gsd-a11y-settin
 1478 tty1     00:00:00 gsd-clipboard
 1481 tty1     00:00:01 gsd-color
 1482 tty1     00:00:00 gsd-datetime
 1483 tty1     00:00:00 gsd-housekeepin
 1484 tty1     00:00:00 gsd-keyboard
 1485 tty1     00:00:00 gsd-media-keys
 1489 tty1     00:00:00 gsd-mouse
 1490 tty1     00:00:00 gsd-power
 1493 tty1     00:00:00 gsd-print-notif
 1496 tty1     00:00:00 gsd-rfkill
 1499 tty1     00:00:00 gsd-screensaver
 1503 tty1     00:00:00 gsd-sharing
 1507 tty1     00:00:00 gsd-smartcard
 1511 tty1     00:00:00 gsd-sound
 1516 tty1     00:00:00 gsd-wacom
 1527 tty1     00:00:00 ibus-engine-sim
 1693 tty2     00:05:08 Xorg
```

10. mininet> s1 ps -a

```
mininet> s1 ps -a
  PID TTY          TIME CMD
 1151 tty1     00:00:00 gnome-session-b
 1201 tty1     00:00:08 gnome-shell
 1399 tty1     00:00:00 Xwayland
 1444 tty1     00:00:00 ibus-daemon
 1447 tty1     00:00:00 ibus-dconf
 1449 tty1     00:00:00 ibus-x11
 1474 tty1     00:00:00 gsd-xsettings
 1477 tty1     00:00:00 gsd-a11y-settin
 1478 tty1     00:00:00 gsd-clipboard
 1481 tty1     00:00:01 gsd-color
 1482 tty1     00:00:00 gsd-datetime
 1483 tty1     00:00:00 gsd-housekeepin
 1484 tty1     00:00:00 gsd-keyboard
 1485 tty1     00:00:00 gsd-media-keys
 1489 tty1     00:00:00 gsd-mouse
 1490 tty1     00:00:00 gsd-power
 1493 tty1     00:00:00 gsd-print-notif
 1496 tty1     00:00:00 gsd-rfkill
 1499 tty1     00:00:00 gsd-screensaver
 1503 tty1     00:00:00 gsd-sharing
 1507 tty1     00:00:00 gsd-smartcard
 1511 tty1     00:00:00 gsd-sound
 1516 tty1     00:00:00 gsd-wacom
 1527 tty1     00:00:00 ibus-engine-sim
 1693 tty2     00:05:10 Xorg
 1708 tty2     00:00:00 gnome-session-b
 1838 tty2     00:06:59 gnome-shell
 1878 tty2     00:00:09 ibus-daemon
 1886 tty2     00:00:00 ibus-dconf
```

11. mininet> h1 ping -c 1 h2

```
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=18.5 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 18.542/18.542/18.542/0.000 ms
mininet>
```

12. mininet> pingall

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

13. mininet> h1 python -m SimpleHTTPServer 80 &

```
mininet>  h1 python -m SimpleHTTPServer 80 &
mininet> h2 wget -O - h1
--2020-09-06 10:53:45--  http://10.0.0.1/
Connecting to 10.0.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1106 (1.1K) [text/html]
Saving to: 'STDOUT'

-                   0%[                    ]       0  --.-KB/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=".git/">.git/</a>
<li><a href=".gitattributes">.gitattributes</a>
<li><a href=".github/">.github/</a>
<li><a href=".gitignore">.gitignore</a>
<li><a href=".pylint">.pylint</a>
<li><a href=".travis.yml">.travis.yml</a>
<li><a href="bin/">bin/</a>
<li><a href="build/">build/</a>
<li><a href="CONTRIBUTORS">CONTRIBUTORS</a>
<li><a href="custom/">custom/</a>
<li><a href="debian/">debian/</a>
<li><a href="dist/">dist/</a>
<li><a href="doc/">doc/</a>
<li><a href="examples/">examples/</a>
<li><a href="INSTALL">INSTALL</a>
```

14. mininet> h1 kill %python

```
mininet> h1 kill %python
Serving HTTP on 0.0.0.0 port 80 ...
10.0.0.2 - - [06/Sep/2020 10:53:45] "GET / HTTP/1.1" 200 -
mininet>
```

15. mininet> exit

```
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1280.251 seconds
```

16. $ sudo mn -c

```
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openflowd
 ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_corelt-nox_core ovs-openfl
owd ovs-controllerovs-testcontroller udpbwtest mnexec ivs ryu-manager 2> /dev/n
ull
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.
```

17. $ 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 5.883 seconds
```

18. $ 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: ['25.0 Gbits/sec', '25.0 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 10.864 seconds
```

19. $ sudo mn --test pingall --topo single,3

```
gle,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
```

20. $ sudo mn --test pingall --topo linear,4

```
ear,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
```

21. $ sudo mn --link tc,bw=10,delay=10ms

```
s
*** 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> iperf
*** Unknown command: mininet> iperf
mininet>  ...
```

22. $ sudo mn -v debug

```
*** errRun: ['which', 'controller']
/usr/local/bin/controller
  0*** errRun: ['grep', '-c', 'processor', '/proc/cpuinfo']
4
  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', '--noeditin
 '-is', 'mininet:c0'] 21774*** 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', '--noediti
, '-is', 'mininet:h1'] 21781*** h1 : ('unset HISTFILE; stty -echo; set +m',)
unset HISTFILE; stty -echo; set +m
h1 _popen ['mnexec', '-cdn', 'env', 'PS1=\x7f', 'bash', '--norc', '--noediti
, '-is', 'mininet:h2'] 21783*** h2 : ('unset HISTFILE; stty -echo; set +m',)
```

23. mininet> exit

```
  0mininet> exit
*** Stopping 1 controllers
c0 *** c0 : ('kill %controller',)
*** c0 : ('wait %controller',)
bash: wait: %controller: no such job

*** Stopping 2 links
.*** h1 : ('ip link del h1-eth0',)
*** s1 : ('ip link del s1-eth1',)
Cannot find device "s1-eth1"
.*** h2 : ('ip link del h2-eth0',)
*** s1 : ('ip link del s1-eth2',)
Cannot find device "s1-eth2"

*** Stopping 1 switches
*** errRun: ['ovs-vsctl', '--if-exists', 'del-br', 's1']
  0*** errRun: ['kill', '-HUP', '21788']
  0s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 89.623 seconds
```

24. $sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --test pingall

```
m/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 6.467 seconds
```