



MawlanaBhashani Scienceand Technology University

Lab-Report

Report No: 01

Course code:ICT-3208

Course title:Computer Network Lab

Date of Performance: 8 / 11 /2020

Date of Submission: 9 / 11 / 2020

Submitted by

Name: Afra Ibnat Tethye

ID:IT-18055

3rd year 2nd semester

Session: 2017-2018

Dept. of ICT

Submitted To

Nazrul Islam,

Assistant Professor,

Dept. of ICT,

MBSTU.

Experiment No: 01

Experiment Name : Basic mininet commands

Create Virtual Network :

To manage our virtual network we will use CLI (sudom command) . Two hosts (h1, h2), openflow switch(s1) and openflow controller (c0) are includes in the default topology.

Interact with Hosts and Switches :

Start a minimal topology and enter the CLI :

```
File Edit View Search Terminal Help
Tethye@tethye:~$ sudo mn
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> █
```

When issuing the sudomn command, Mininet initializes the topology and launches its command line interface which looks like this:

mininet>

AgainDisplay Mininet CLI commands:

mininet> help

```
File Edit View Search Terminal 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
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
```

To display the available nodes, type the following command:

mininet> nodes

```
File Edit View Search Terminal Help
mininet> nodes
available nodes are:
h1 h2 s1
mininet> █
```

Display links:

mininet> net

```
File Edit View Search Terminal Help
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> █
```

Dump information about all nodes:

mininet> dump

```
File Edit View Search Terminal Help
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=3848>
<Host h2: h2-eth0:10.0.0.2 pid=3850>
<OVSBridge s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=3855>
mininet>
```

If the first string typed into the Mininet CLI is a host, switch or controller name, the command is executed on that node. Run a command on a host process:

mininet> h1 ifconfig -a

```
File Edit View Search Terminal Help
mininet> s1 ifconfig -a
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 503 bytes 43054 (43.0 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 503 bytes 43054 (43.0 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 96:6c:8e:0f:a1:a6 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

s1: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 3a:e0:f1:fd:64:49 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 20 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

This command executes the ifconfig Linux command on host h1. The command shows host h1's interfaces. The display indicates that host h1 has an interface h1-eth0 configured with IP address 10.0.0.1, and another interface lo configured with IP address 127.0.0.1

Test connectivity :

Mininet's default topology assigns the IP addresses 10.0.0.1/8 and 10.0.0.2/8 to host h1 and host h2 respectively. To test connectivity between them, you can use the command ping.

The command shown below.

mininet> h1 ping 10.0.0.2

```
File Edit View Search Terminal Help
mininet> h1 ping 10.0.0.2
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=1.35 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.090 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.091 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.089 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.090 ms
^C
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9186ms
rtt min/avg/max/mdev = 0.086/0.216/1.351/0.378 ms
mininet>
```

This command tests the connectivity between host h1 and host h2. To stop the test, press Ctrl+c .

Stop the emulation by typing the following command:

mininet> exit

```
rtt min/avg/max/mdev = 0.086/0.216/1.351/0.378 ms
mininet> exit
*** Stopping 0 controllers

*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 309.318 seconds
Tethye@tethye:~$
```

Mininet crashes for some reason, clean it up by the following command:

\$ sudomn -c

```
File Edit View Search Terminal Help
Tethye@tethye:~$ 
Tethye@tethye:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openfl
owd ovs-controller udpbwtest mnexec ivs 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
```

Discussion :

Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking. A network emulator which creates a network of virtual hosts, switches, controllers, and links is known as Mininet.