

Lab-Report

Report No: 01

Course code: ICT-3208

Course title: Computer Network Lab

Date of Performance: 23 / 10 /2020

Date of Submission: 26 / 10 / 2020

Submitted by

Name: Golam Kibria Tuhin

ID: IT-18015

3th year 2nd semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Experiment No: 01

Experiment Name : Basic mininet commands

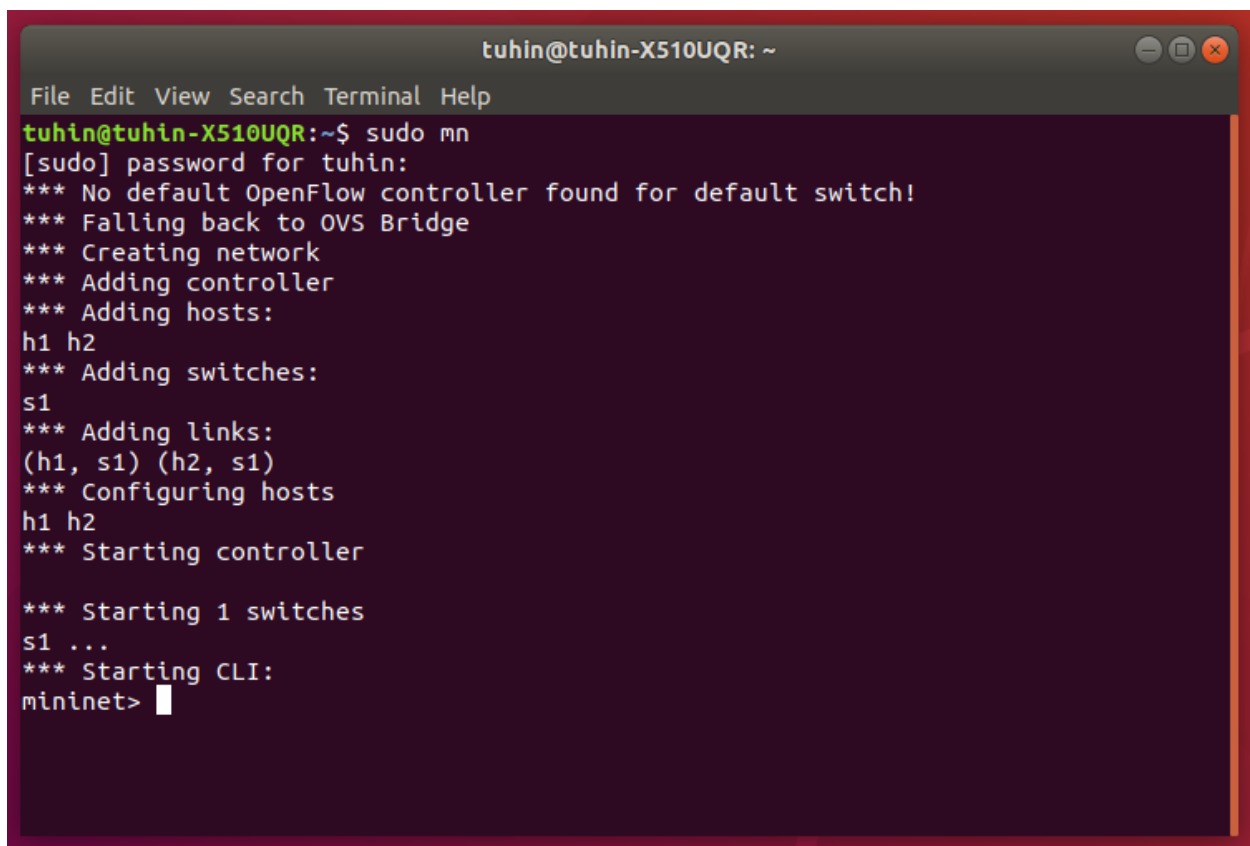
Create Virtual Network :

We will be using CLI (`sudo mn` command) to manage our virtual network. The default topology includes two hosts (h1,h2), OpenFlow Switch(s1) and OpenFlow controller(c0).

Interact with Hosts and Switches :

Start a minimal topology and enter the CLI :

\$ sudo mn



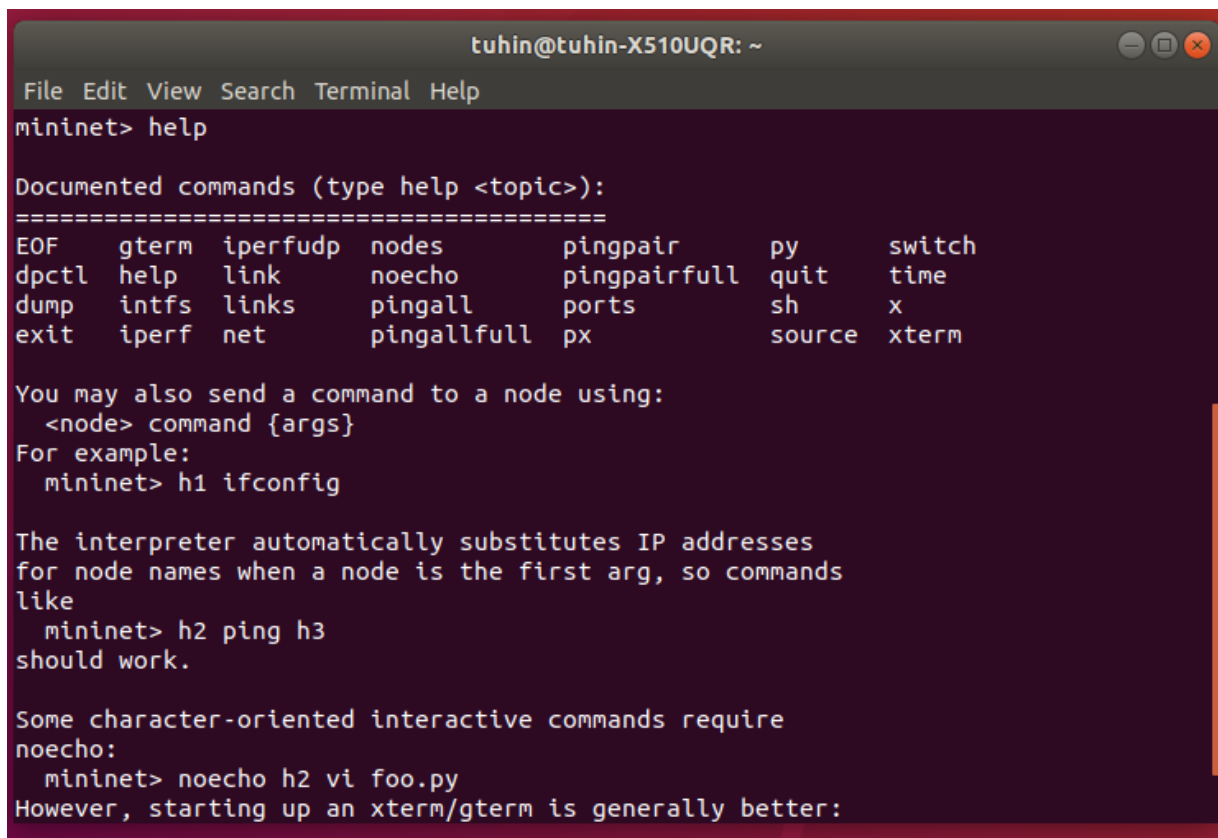
```
tuhin@tuhin-X510UQR: ~
File Edit View Search Terminal Help
tuhin@tuhin-X510UQR:~$ sudo mn
[sudo] password for tuhin:
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

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

mininet >

Again Display Mininet CLI commands:

mininet> help



```
tuhin@tuhin-X510UQR: ~
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     intfz  links     pingall    ports       sh      x
exit     iperf  net       pingallfull px          source  xterm

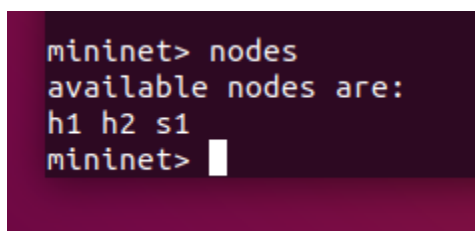
You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

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



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

Display links:

mininet> net

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

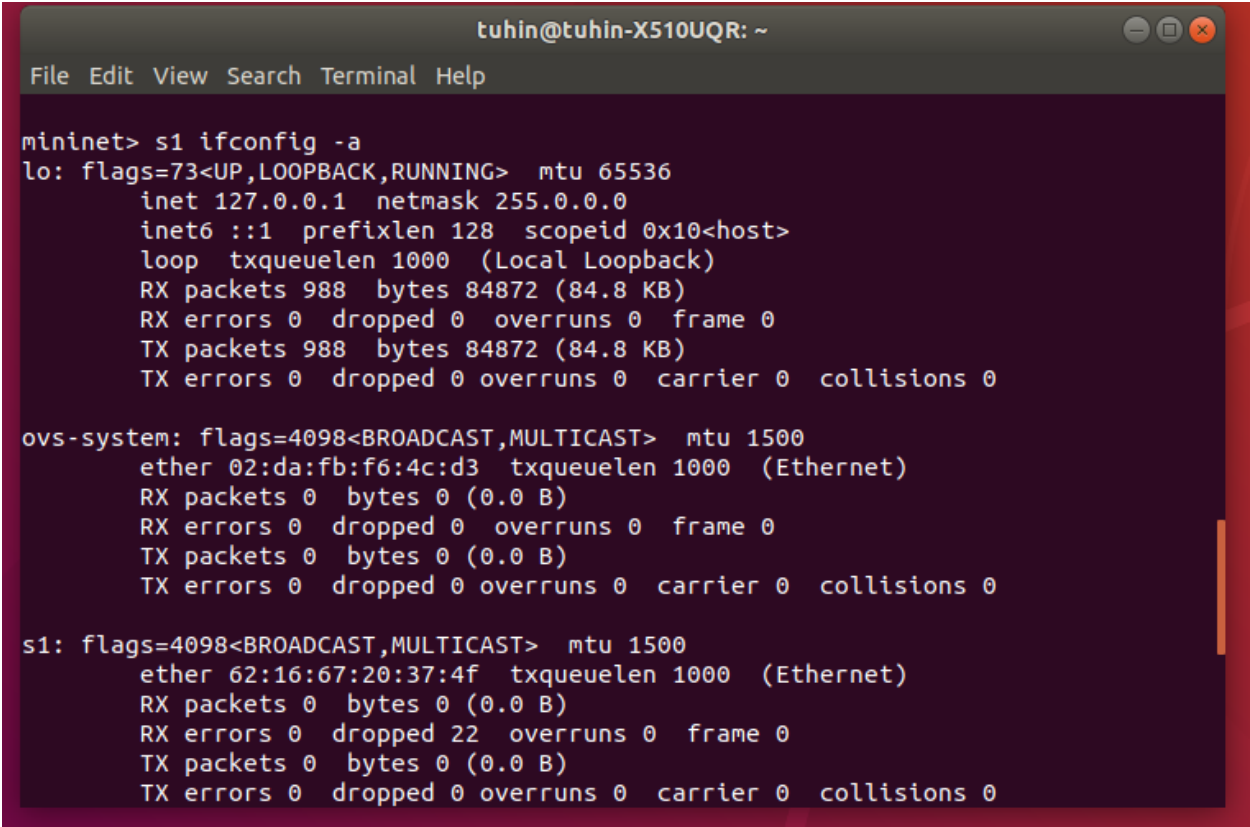
Dump information about all nodes:

mininet> dump

```
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=5181>
<Host h2: h2-eth0:10.0.0.2 pid=5183>
<OVSBridge s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=5188>
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



```
tuhin@tuhin-X510UQR: ~
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 988 bytes 84872 (84.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 988 bytes 84872 (84.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 02:da:fb:f6:4c:d3 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 62:16:67:20:37:4f txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 22 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

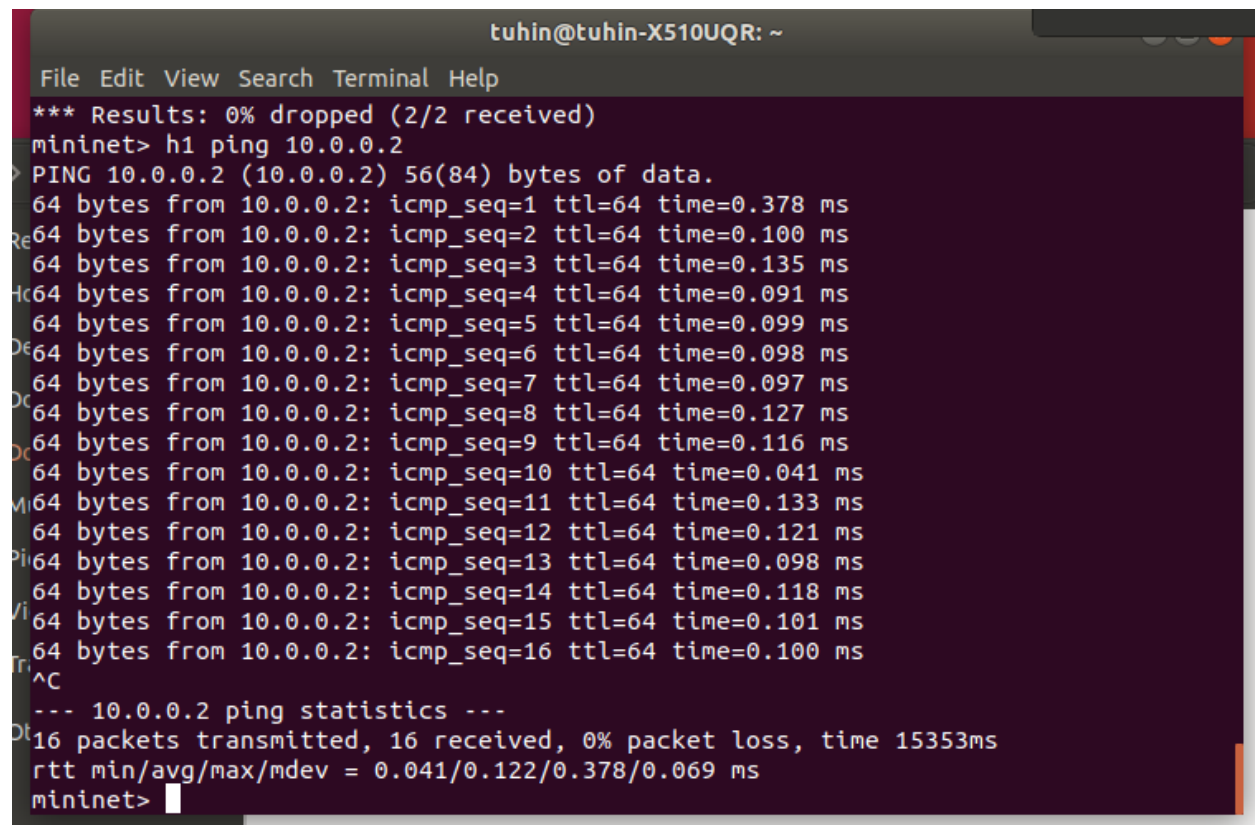
This command executes the `ifconfig` Linux command on host h1. The command shows host h1's interfaces. The display indicates that host h1 has an interface `h1-eth0` configured with IP address `10.0.0.1`, and another interface `lo` configured with IP address `127.0.0.1`.

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



```
tuhin@tuhin-X510UQR: ~  
File Edit View Search Terminal Help  
*** Results: 0% dropped (2/2 received)  
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=0.378 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.100 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.135 ms  
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.091 ms  
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.099 ms  
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.098 ms  
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.097 ms  
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.127 ms  
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.116 ms  
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.041 ms  
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.133 ms  
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.121 ms  
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.098 ms  
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=0.118 ms  
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=0.101 ms  
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=0.100 ms  
^C  
--- 10.0.0.2 ping statistics ---  
16 packets transmitted, 16 received, 0% packet loss, time 15353ms  
rtt min/avg/max/mdev = 0.041/0.122/0.378/0.069 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

```
mininet> exit
*** Stopping 0 controllers

*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 697.464 seconds
tuhin@tuhin-X510UQR:~$
```

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

\$ sudo mn -c

```
tuhin@tuhin-X510UQR: ~
File Edit View Search Terminal Help
tuhin@tuhin-X510UQR:~$ sudo mn -c
[sudo] password for tuhin:
*** 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-openflowd
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
rm -f ~/.ssh/mn/*
*** Cleanup complete.
tuhin@tuhin-X510UQR:~$
```

Discussion :

Mininet is a network emulator which creates realistic virtual network . From this lab how to install mininet successfully . And I have also learn the basic command and procedure of mininet from this lab.