



Sir Visvesvaraya Institute of Technology, Nashik

Department of Computer Engineering

LAB MANUAL

Laboratory Practice VI

(410256)

Semester-VIII

Companion Courses:

Software Defined Networking (410252

C)

INDEX

Name of the Subject/Course: **410256 - Laboratory Practice VI**

Class: **BE**

Sr. No.	Assignment Title
410252(C) : Software Defined Networks	
1	Prepare setup for Mininet network emulation environment with the help of Virtualbox and Mininet. Demonstrate the basic commands in Mininet and emulate different custom network topology(Simple, Linear, and Tree).View flow tables.
2	After studying open source POX and Floodlight controller, Install controller and run custom topology using remote controller like POX and floodlight controller. Recognize inserted flows by controllers
3	Prepare setup for Mininet network emulation environment with the help of Virtualbox and Mininet. Demonstrate the basic commands in Mininet and emulate different custom network topology(Simple, Linear, and Tree).View flow tables.
4	After studying open source POX and Floodlight controller, Install controller and run custom topology using remote controller like POX and floodlight controller. Recognize inserted flows by controllers
5	Create a SDN environment on Mininet and configure a switch to provide a firewall functionality using POX controller. Ref: https://github.com/mininet/openflow-tutorial/wiki/Create- Firewall

Assignment No.1

Aim: Prepare setup for Mininet network emulation environment with the help of Virtualbox and Mininet. Demonstrate the basic commands in Mininet and emulate different custom network topology (Simple, Linear, and Tree).View flow tables.

Objective:

1. Learn installation on Mininet emulator.
2. Learn basic commands in Mininet
3. Learn how to create different topologies in Mininet

Theory:

Mininet is a virtual testbed enabling the development and testing of network tools and protocols. With a single command, Mininet can create a realistic virtual network on any type of machine (Virtual Machine (VM), cloud-hosted, or native). Therefore, it provides an inexpensive solution and streamlined development running in line with production networks¹. Mininet offers the following features:

- Fast prototyping for new networking protocols.
- Simplified testing for complex topologies without the need of buying expensive hardware.
- Realistic execution as it runs real code on the Unix and Linux kernels.
- Open source environment backed by a large community contributing extensive documentation.

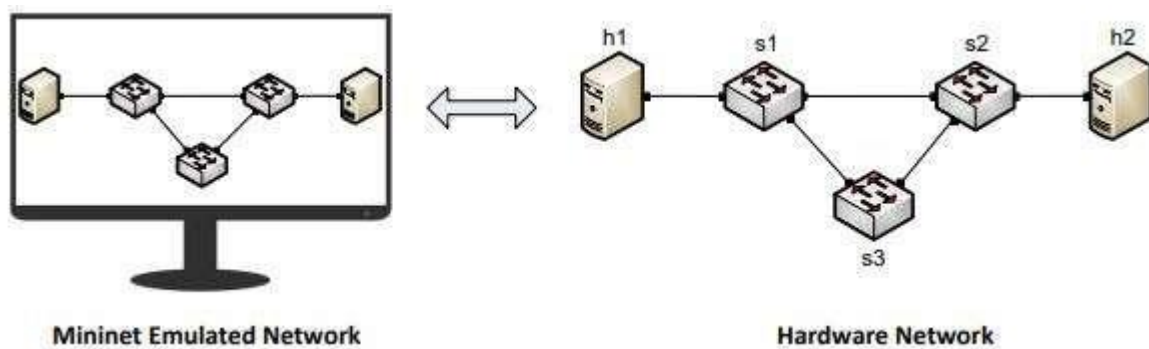


Figure 1. Hardware network vs. Mininet emulated network.

Mininet is useful for development, teaching, and research as it is easy to customize and interact with it through the CLI or the GUI. Mininet was originally designed to experiment with OpenFlow² and Software-Defined Networking (SDN)³. This lab, however, only focuses on emulating a simple network environment without SDN-based devices. Mininet's logical nodes

can be connected into networks. These nodes are sometimes called containers, or more accurately, network namespaces. Containers consume sufficiently fewer resources than networks of over a thousand nodes have created, running on a single laptop. A Mininet container is a process (or group of processes) that no longer has access to all the host system's native network interfaces. Containers are then assigned virtual Ethernet interfaces, which are connected to other containers through a virtual switch⁴.

. Mininet connects a host and a switch using a virtual Ethernet (veth) link. The veth link is analogous to a wire connecting two virtual interfaces, as illustrated below.

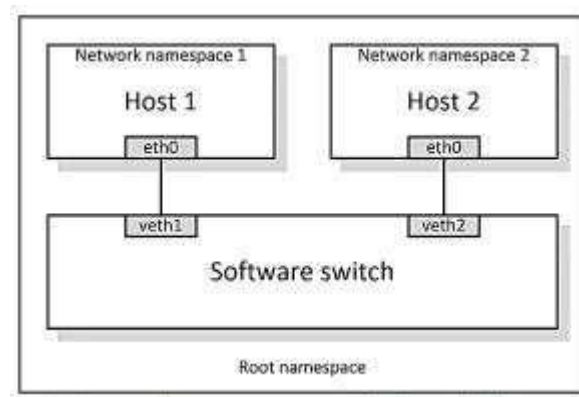


Figure 2. Network namespaces and virtual Ethernet links.

Each container is an independent network namespace, a lightweight virtualization feature that provides individual processes with separate network interfaces, routing tables, and Address Resolution Protocol (ARP) tables. Mininet provides network emulation opposed to simulation, allowing all network software at any layer to be simply run as is; i.e. nodes run the native network software of the physical machine. On the other hand, in a simulated environment applications and protocol implementations need to be ported to run within the simulator before they can be used.

Invoke Mininet using the default topology

Step 1. Launch a Linux terminal by holding the Ctrl+Alt+T keys or by clicking on the Linux terminal icon.

Step 2. To start a minimal topology, enter the command shown below. When prompted for a password, type password and hit enter. Note that the password will not be visible as you type it.

\$sudo mn

```
sdn@admin: ~  
File Actions Edit View Help  
sdn@admin: ~  
sdn@admin:~$ sudo mn  
[sudo] password for sdn:  
*** 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:  
containernet>
```

The above command starts Mininet with a minimal topology, which consists of a switch connected to two hosts as shown below.

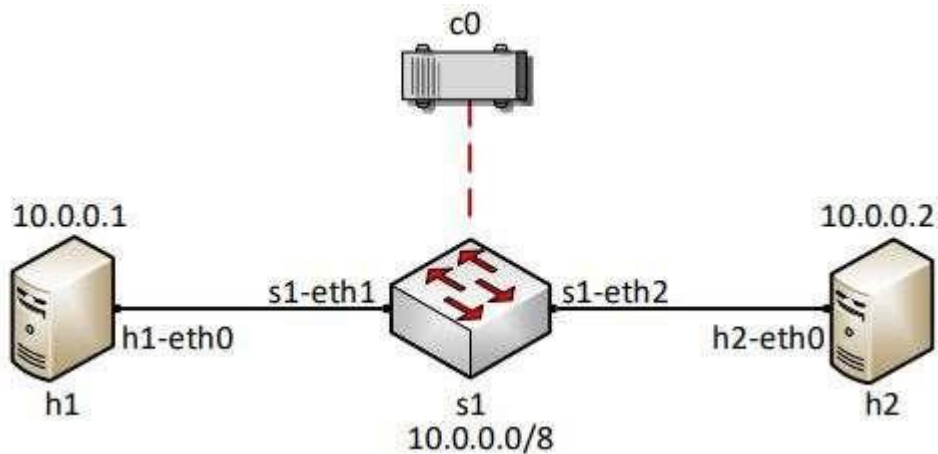


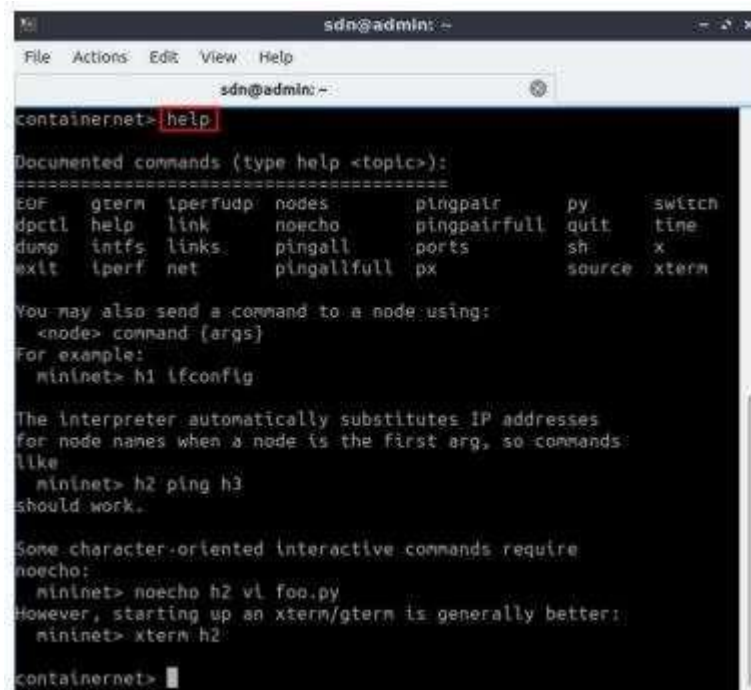
Figure 4. Mininet's default minimal topology

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

mininet>

Step 3. To display the list of Mininet CLI commands and examples on their usage, type the following command:

help



```
sdn@admin: ~  
File Actions Edit View Help  
sdn@admin: ~  
containernet> help  
  
Documented commands (type help <topic>):  
=====  
EOF  gterm  lperfudp  nodes    pingpair  py    switch  
dctl  help   link      noecho   pingpairfull  quit  time  
dump  intfs  links     pingall  ports     sh    x  
exit  lperf  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 vl foo.py  
However, starting up an xterm/gterm is generally better:  
  mininet> xterm h2  
  
containernet>
```

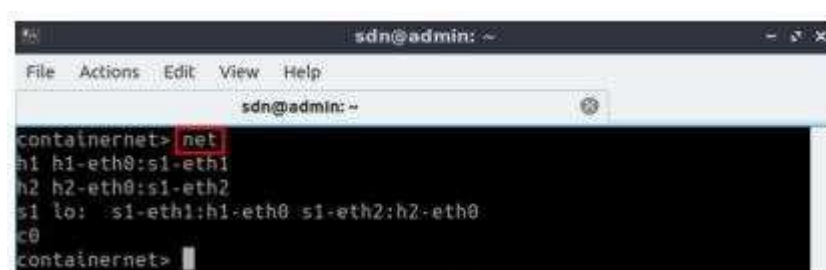
Step 4. To display the available nodes, type the following command:

nodes



```
sdn@admin: ~  
File Actions Edit View Help  
sdn@admin: ~  
containernet> nodes  
available nodes are:  
c0 h1 h2 s1  
containernet>
```

Step 5. It is useful sometimes to display the links between the devices in Mininet to understand the topology. Issue the command shown below to see the available links.



```
sdn@admin: ~  
File Actions Edit View Help  
sdn@admin: ~  
containernet> net  
h1 h1-eth0:s1-eth1  
h2 h2-eth0:s1-eth2  
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0  
c0  
containernet>
```

Test connectivity

h1 ping h2

Mininet>dump: This command shows the dump information about all nodes available in the current Mininet network.

Creation of Topologies in mininet;

Linear topology

In mininet we have various topologies like minimal, single, reversed, linear, tree topology etc.

- **Minimal/ Simple:** It is the most basic topology with two hosts and one switch. To run minimal topology

we simply run the following command in the terminal window i.e.

Sudo mn --topo minimal

- **Single Topology:** It is the simple topology with one switch and N hosts. To run this topology we

run following command in terminal window i.e.

Sudo mn –topo single,3

- **Reversed Topology:** It is similar to the single connection but order of connection between hosts and switch is reversed. To run reversed topology we use the command in terminal window i.e.

Sudo mn –topo reversed,3

Conclusion: We learn how to install Mininet and basic commands, and how to create topology.

FAQ:-

1] What is difference between Emulator and simulator?

2] What is difference between Software defined network and traditional network?

3] What is the use of SDN or Application of SDN

4] What are the challenges in Traditional network