



**MHRD**  
Ministry of Human  
Resource Development  
Government of India



## Software Based Networks:

### **SDN and Integration of Virtualization in Networks**

December 19 - 23, 2017

National Institute of Technology Karnataka (NITK), Surathkal, India

# Introduction to Mininet

## Objectives of the tutorial:

1. Mininet installation
2. Using Mininet
3. Creating a simple network topology using CLI and Python
4. Creating a simple network topology using Mininet GUI tool
5. Using POX controller in Mininet

## Prepared by:

Wireless Information Networking Group (WiNG),  
National Institute of Technology Karnataka, Surathkal,  
Mangalore, India.

Email address: wing [at] nitk [dot] ac [dot] in

## 1. Mininet installation

There are 4 ways to install Mininet:

1. Download Mininet VM and set up the environment
2. Native installation from source files
3. Installation through packages (e.g., apt-get install)
4. Upgrade an existing Mininet installation

In this tutorial, we follow the second method to install Mininet in Ubuntu system (preferably, Ubuntu 16.04). We assume you are installing Mininet on a newly installed Ubuntu OS. Steps to install Mininet by using other methods are described here: <http://mininet.org/download/>

Step 1: Open a new terminal:

**Ctrl + Alt + t**

(above command will open a new terminal)

Step 2: Install git:

**sudo apt-get install git**

(above command will install git version control in your machine)

Step 3: Go to Desktop via terminal:

**cd Desktop**

Step 4: Clone Mininet from github:

**git clone git://github.com/mininet/mininet**

(above command will download mininet source code from github)

Step 5: Go to the 'mininet' directory created during cloning and install Mininet:

**cd mininet**

**./util/install.sh -a**

(above command installs Mininet. Installation time depends on the Internet speed.)

Step 6: Verify whether Mininet is installed successfully:

**sudo mn --test pingall**

## 2. Using Mininet

After the successful installation, follow these steps to run default Mininet examples:

Step 1: Run following command to get access to Mininet terminal.

```
sudo mn
```

(above command will start mininet terminal)

A default topology will be loaded which has two hosts that are connected to a switch (OVSSwitch) and a default controller (ovs-controller). The terminal will start displaying something like this:

```
mininet>
```

Step 2: Use following commands inside the Mininet terminal to get information about topology:

1. To display nodes: **nodes**
2. To display links: **net**
3. To dump information about all nodes: **dump**

Use **help** to get more CLI commands.

Step 3: To see information related to any particular host, use the following command inside the Mininet terminal:

```
host-name ifconfig -a
```

Example: **h1 ifconfig -a**

Step 4: To ping from one node to another node:

```
host-name-1 ping host-name-2
```

Example: **h1 ping h2**

(above command will ping from h1 to h2)

Step 5: To ping all hosts from every host:

```
pingall
```

(above command will ping all hosts from every host)

## 3. Creating a simple topology using CLI and Python

### 3.1. Changing topology type and size

Mininet provides some inbuilt topologies like linear, single, minimal, reversed, torus and tree. To use them, open Mininet from your bash terminal by using the following command:

```
sudo mn --topo linear,4
```

(creates a topology of 4 nodes, each connected with a *separate switch*)

```
sudo mn --topo single,4
```

(creates a topology 4 nodes, each connected with a *single switch*)

### 3.2. Changing link parameters: use the following command

```
sudo mn --link tc,bw=10,delay=10ms
```

(above command will set bandwidth to 10Mbps and delay to 10ms for all the links in network)

### 3.3. Creating custom topology: steps to create a custom topology as shown below



Step 1: Go to 'Desktop/mininet' directory via terminal

```
cd Desktop/mininet
```

Step 2: Locate and examine custom/topo-2sw-2host.py

```
ls custom/topo-2sw-2host.py
```

(above command will displays one of the custom topology script)

Step 3: Run this command to load custom topology from 'Desktop/mininet' directory:

```
sudo mn --custom ./custom/topo-2sw-2host.py --topo mytopo
```

(above command will open the Mininet terminal with custom topology)

Step 4: Test your topology using **pingall** application from Mininet terminal:

```
pingall
```

(above command will execute ping in every node to ping every other node in the topology)

## 4. Creating a simple topology using Mininet GUI tool

Step 1: Go to 'Desktop/mininet' directory via terminal

```
cd Desktop/mininet
```

Step 2: Execute following command to open GUI tool

```
sudo python ./examples/miniedit.py
```

(above command will open MiniEdit GUI window)

A window will open in which you can drag and drop the networking components and set their attributes.

Step 3: Create topology instructed by instructor and save it as a Python program.

Click on **File** in the top menu bar and select **Export Level 2 Script** from the drop-down menu. Type in the filename and save the file.

Example: save it under **Desktop/mininet/custom** folder as **mycustom.py**

Step 4. Go to 'Desktop/mininet' directory via terminal and run following command to load custom topology.

```
sudo python ./custom/mycustom.py
```

(above command will load custom topology created using miniedit tool)

## 5. Using POX controller in Mininet

POX controller gets installed when Mininet is installed with option “-a”.

Step 1: Open a new terminal and go to ‘Desktop/pox’ directory from your bash terminal:

```
cd Desktop/pox
```

Step 2: Type the following command to start pox controller:

```
sudo ./pox.py forwarding.l2_learning info.packet_dump  
samples.pretty_log log.level --INFO
```

(above command will start POX controller on the localhost i.e., 127.0.0.1)

Step 3: Open another terminal and load any topology and configure your network to use remote controller by using below command:

```
sudo mn --topo linear,4 --controller remote,ip=127.0.0.1
```

(creates a linear topology and configures to use remote controller running at 127.0.0.1)

Step 4: Execute the following Mininet command:

```
dpctl dump-flows
```

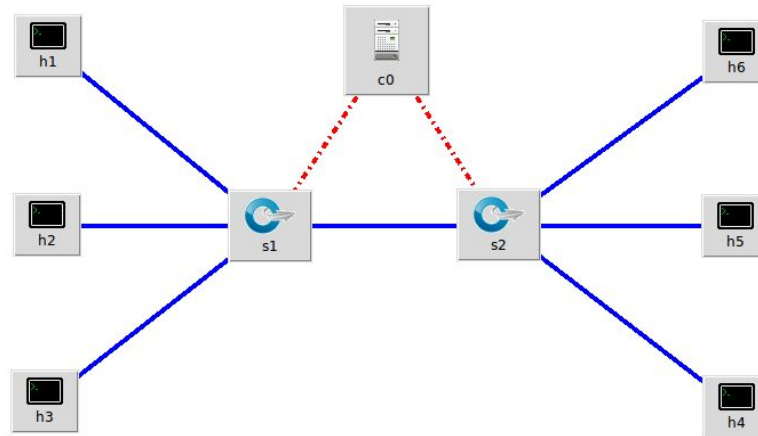
(above command displays contents of the flow tables on all switches)

Step 5: To check ARP tables on each host, execute the Mininet **arp** command.

Example: **h1 arp**

## Assignments

**Assignment 1:** Create a custom dumbbell topology as shown in the figure below.



Keep bandwidth and delay as 50 Mbps and 5 ms for all the links. Validate your topology using **dump**, **pingall** and **arp**.

**Assignment 2:** Configure the example described in assignment 1, but with an Opendaylight controller (you should install opendaylight first).

## Learn more!

Why mininet?: <http://openvswitch.org/support/ovscon2015/16/1305-lantz.pdf>

Mininet installation: <http://mininet.org/download/>

Mininet commands (topology, link variation, etc): <http://mininet.org/walkthrough/>

Miniedit: <http://www.brianlinkletter.com/how-to-use-miniedit-mininets-graphical-user-interface/>

Mininet with POX controller: <http://www.brianlinkletter.com/using-the-pox-sdn-controller/>