



SIT325: Advanced Network Security

T2_2022, Deakin University, VIC

Activity: Module 1, Task 1.1P, Pass Task

Released on 11th July, due date of submission is 25th July 23:59hrs.

Note: Queries should be forwarded to your respective tutor or unit chair via email only. Give at least 2 business days for us to reply or give feedback.

Aim of the Module: In this module, using network emulator tool (mininet) we design a network using real world topologies (AARNet topology and DFN topology). And measure the performance evaluation of the network. Later, we will extend this work to 5G test bed.

So, to accomplish this overall aim we have decomposed this big aim into many subtasks. Task 1.1P is one of the tasks in which we are installing required necessary softwares required in this unit.

Task

Step 1: Before we start installing the SDN applications we need to prepare our desktop/system/laptop or any LAB computer with the followings:

- Download and Install virtualization Software
- Download and Install Ubuntu Linux on a VM
- Test connectivity to the VM from your desktop or LAN

Download and Install virtualization Software

The recommended virtualization platform to run the various SDN applications is VirtualBox which can be installed on Windows / MacOS / Linux desktops. It can be downloaded from the following link: <https://www.virtualbox.org/wiki/Downloads>

Download and Install Ubuntu Linux on a VM

Go to <https://ubuntu.com/download/desktop> your computer's web browser. You can download the Ubuntu disk image (also known as an ISO file) here.

Further instructions on installing Ubuntu in VirtualBox can be found in the below link:

<https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox>

Test connectivity to the VM from your desktop or LAN

Using any tool, show that your host machine (Desktop) can connect to Virtual Machine (Ubuntu).

Step 2: Download and install "mininet" a network emulation tool. Create a small minimal network topology (one controller, 2 switches, and few hosts to each switch)
<http://mininet.org/>

For documentation, please refer <https://github.com/mininet/mininet/wiki/Documentation>

To submit:

1. Submit a word file, show your source code + screen shorts to show us the successful setup of the default topology, i.e., the minimal topology, which includes one OpenFlow kernel switch connected to two hosts, plus the OpenFlow reference controller. Example as below.

```
keshav@keshav-VirtualBox:~/mininet$ sudo mn
[sudo] password for keshav:
*** 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>
```

2. Submit a word file, show your source code + screen shorts to show us the successful setup of the Custom Topology.

Custom Topologies Mininet supports a simple Python API to create custom network topologies. You can create your custom topology by writing a few lines of Python code. Customize the topology as shown in the figure below.

