**Lab 2 – CCGC 5002 - Openflow Test Bed Setup – due by Week 3, Saturday 11:30pm**

Compile solutionsinto a single word/pdf file and upload in the Lab 2 folder under Assignments tab of Blackboard.

## 1. Objective

Setup the SDN Test environment to practice Openflow usecases with RYU SDN Controller.

**Tools to be installed:**

| **OS** | **Ubuntu 19+ Desktop** |
| --- | --- |
| **Test Bed** | Mininet |
| **Controller** | RYU |
| **Switch** | Openvswitch |
| **Packet Capture** | Wireshark |
| **Traffic Generator** | IPerf |

**Skip the installations and update steps if using vCloud**

## 2. Installation

**Requirements:**

OS: Ubuntu 19.x

CPU: 2 Cores +

RAM: 8GB +

HDD: 25GB+

If you are using Windows or other OS, you can install Ubuntu 19 as a Virtual Machine(VM) using Virtual Box .

You can download the ISO installer from the below link,

<https://www.ubuntu.com/download/desktop>

How to install ubuntu using Virtualbox

https://www.youtube.com/watch?v=x5MhydijWmc

As first step, please run this command.

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install git gcc python-dev libffi-dev libssl-dev libxml2-dev libxslt1-dev zlib1g-dev python-pip

Once we installed the above packages, python 2.7 and relevant dependents are installed properly.

Note: Sudo will give you secure access to run the commands, you may need to type the vm password ‘humber’.

If you are using vCloud for this lab, then DO NOT install or upgrade anything. They already come preinstalled with the image in the library.

**To check the python version:**

*python –version*

Example:

tanvir@tanvir-vm:~$ python --version

Python 2.7.15rc1

tanvir@tanvir-vm:~$

At the time of writing, Ubuntu official repository has the following tools versions

Tool Name       Version

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Openvswitch :     2.9.2

Wireshark   :     2.6.6

IPERF       :     2.0.10

Mininet :     2.2.2

RYU will be installed using PIP.

Tool Name     Version

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

RYU       :     4.31

1. **Openvswitch Installation**

*sudo apt-get install openvswitch-switch*

To verify :

ovs-vsctl -version

Example:

tanvir@tanvir-vm:~$ ovs-vsctl --version

ovs-vsctl (Open vSwitch) 2.13.0

DB Schema 8.2.0

tanvir@tanvir-vm:~$

1. **Wireshark Installation**

*sudo apt-get install wireshark*

To verify :

wireshark –version

tanvir@tanvir-vm:~$ wireshark --version

Wireshark 2.6.6 (Git v2.6.6 packaged as 2.6.6-1~ubuntu18.04.0)

1. **Traffic Test Tools installation**

IPERF:

*sudo apt-get install iperf*

CURL:

*sudo apt-get install curl*

HPING3:

*sudo apt-get install hping3*

Apache Bench:

*sudo apt-get install apache2-utils*

To verify :

*iperf –version*

*Log:*

tanvir@tanvir-vm:~$ iperf -v

iperf version 2.0.13 (21 Jan 2019) pthreads

tanvir@tanvir-vm:~$

1. **RYU installation**

*sudo pip install ryu*

To verify :

*ryu-manager –version*

Example:

tanvir@tanvir-vm:~$ ryu-manager --version

ryu-manager 4.34

tanvir@tanvir-vm:~$

Note: Make sure you do with "sudo pip install ryu" . ryu package will install ryu-manager binary in /usr/local/bin folder. Hence installation requires sudo access.

1. **Mininet Installation**

*sudo apt-get install mininet*

To verify :

*mn –version*

Log:

tanvir@tanvir-vm:~$mn --version

2.2.2

tanvir@tanvir-vm:~$

## 3. **Quick Testing**

Open 4 Terminals:

1. In Terminal1,

sudo wireshark

And start the capture for "loopback" or "any" interface.

1. In Terminal2,

ryu-manager ryu.app.simple\_switch\_13

1. In Terminal3,

sudo mn --controller=remote,ip=127.0.0.1 --mac --switch=ovsk,protocols=OpenFlow13 --topo=single,4

you will get the mininet prompt. In mininet prompt, type pingall command

*pingall*

Logs:

tanvir@tanvir-vm:~$ sudo mn --controller=remote,ip=127.0.0.1 --mac --switch=ovsk,protocols=OpenFlow13 --topo=single,4

\*\*\* Creating network

\*\*\* Adding controller

Connecting to remote controller at 127.0.0.1:6653

\*\*\* Adding hosts:

h1 h2 h3 h4

\*\*\* Adding switches:

s1

\*\*\* Adding links:

(h1, s1) (h2, s1) (h3, s1) (h4, s1)

\*\*\* Configuring hosts

h1 h2 h3 h4

\*\*\* Starting controller

c0

\*\*\* Starting 1 switches

s1 ...

\*\*\* Starting CLI:

mininet> pingall

\*\*\* Ping: testing ping reachability

h1 -> h2 h3 h4

h2 -> h1 h3 h4

h3 -> h1 h2 h4

h4 -> h1 h2 h3

\*\*\* Results: 0% dropped (12/12 received)

mininet>

\*\*\*\*\*\*\*\*Take clear screenshot of mininet for the above.

1. In Terminal 4,

*sudo ovs-vsctl show*

*sudo ovs-ofctl -O OpenFlow13 dump-flows s1*

Logs:

tanvir@tanvir-vm:~$ sudo ovs-vsctl show

[sudo] password for tanvir:

a315e8b4-dd3f-42f6-b84a-e967e02660a4

  Bridge "s1"

      Controller "tcp:127.0.0.1:6653"

          is\_connected: true

      Controller "ptcp:6654"

      fail\_mode: secure

      Port "s1-eth4"

          Interface "s1-eth4"

      Port "s1"

          Interface "s1"

              type: internal

      Port "s1-eth1"

          Interface "s1-eth1"

      Port "s1-eth3"

          Interface "s1-eth3"

      Port "s1-eth2"

          Interface "s1-eth2"

  ovs\_version: "2.9.2"

tanvir@tanvir-vm:~$ sudo ovs-ofctl -O OpenFlow13 dump-flows s1

cookie=0x0, duration=115.430s, table=0, n\_packets=3, n\_bytes=238, priority=1,in\_port="s1-eth2",dl\_src=00:00:00:00:00:02,dl\_dst=00:00:00:00:00:01 actions=output:"s1-eth1"

cookie=0x0, duration=115.421s, table=0, n\_packets=2, n\_bytes=140, priority=1,in\_port="s1-eth1",dl\_src=00:00:00:00:00:01,dl\_dst=00:00:00:00:00:02 actions=output:"s1-eth2"

cookie=0x0, duration=115.410s, table=0, n\_packets=3, n\_bytes=238, priority=1,in\_port="s1-eth3",dl\_src=00:00:00:00:00:03,dl\_dst=00:00:00:00:00:01 actions=output:"s1-eth1"

cookie=0x0, duration=115.404s, table=0, n\_packets=2, n\_bytes=140, priority=1,in\_port="s1-eth1",dl\_src=00:00:00:00:00:01,dl\_dst=00:00:00:00:00:03 actions=output:"s1-eth3"

cookie=0x0, duration=115.391s, table=0, n\_packets=3, n\_bytes=238, priority=1,in\_port="s1-eth4",dl\_src=00:00:00:00:00:04,dl\_dst=00:00:00:00:00:01 actions=output:"s1-eth1"

cookie=0x0, duration=115.380s, table=0, n\_packets=2, n\_bytes=140, priority=1,in\_port="s1-eth1",dl\_src=00:00:00:00:00:01,dl\_dst=00:00:00:00:00:04 actions=output:"s1-eth4"

cookie=0x0, duration=115.370s, table=0, n\_packets=3, n\_bytes=238, priority=1,in\_port="s1-eth3",dl\_src=00:00:00:00:00:03,dl\_dst=00:00:00:00:00:02 actions=output:"s1-eth2"

cookie=0x0, duration=115.368s, table=0, n\_packets=2, n\_bytes=140, priority=1,in\_port="s1-eth2",dl\_src=00:00:00:00:00:02,dl\_dst=00:00:00:00:00:03 actions=output:"s1-eth3"

cookie=0x0, duration=115.361s, table=0, n\_packets=3, n\_bytes=238, priority=1,in\_port="s1-eth4",dl\_src=00:00:00:00:00:04,dl\_dst=00:00:00:00:00:02 actions=output:"s1-eth2"

cookie=0x0, duration=115.359s, table=0, n\_packets=2, n\_bytes=140, priority=1,in\_port="s1-eth2",dl\_src=00:00:00:00:00:02,dl\_dst=00:00:00:00:00:04 actions=output:"s1-eth4"

cookie=0x0, duration=115.346s, table=0, n\_packets=3, n\_bytes=238, priority=1,in\_port="s1-eth4",dl\_src=00:00:00:00:00:04,dl\_dst=00:00:00:00:00:03 actions=output:"s1-eth3"

cookie=0x0, duration=115.344s, table=0, n\_packets=2, n\_bytes=140, priority=1,in\_port="s1-eth3",dl\_src=00:00:00:00:00:03,dl\_dst=00:00:00:00:00:04 actions=output:"s1-eth4"

cookie=0x0, duration=164.572s, table=0, n\_packets=58, n\_bytes=4276, priority=0 actions=CONTROLLER:65535

tanvir@tanvir-vm:~$

1. check the openflow messages in wireshark

Stop the Wireshark capture,

In the filter type "openflow" to see the OPENFLOW Messages.

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Take screenshot of wireshark screen.

**Quick check on subnet knowledge: you have 2 VMs in vCloud as you created from template library. How can you identify from the ip that they are located in 2 different subnets?**

## References

<https://www.ubuntu.com/download/desktop>

<https://osrg.github.io/ryu/>

<http://ryu.readthedocs.io/en/latest/getting_started.html>

<http://mininet.org/>

<https://iperf.fr/>

<https://www.wireshark.org>

<https://www.openvswitch.org/>

Others:

sudo apt-get install openssh-server net-tools

Note: the notes are written using python version 2 and pip