

# Data Center Networking Technology

## Project 2

**2021.03.16**

Timothy William

[timothywilliam.cs06g@g2.nctu.edu.tw](mailto:timothywilliam.cs06g@g2.nctu.edu.tw)

# Outline

- Project Info
- Descriptions and Objectives
- Project Content
- Step-by-Step Instructions
- Demo
- Reference

# Project Info

## Goal:

- In this project, student will learn how to use Mininet and a SDN controller (Ryu) to emulate a simple network system

**Project assigned: 03/16/2021**

**Project deadline: 03/30/2021**

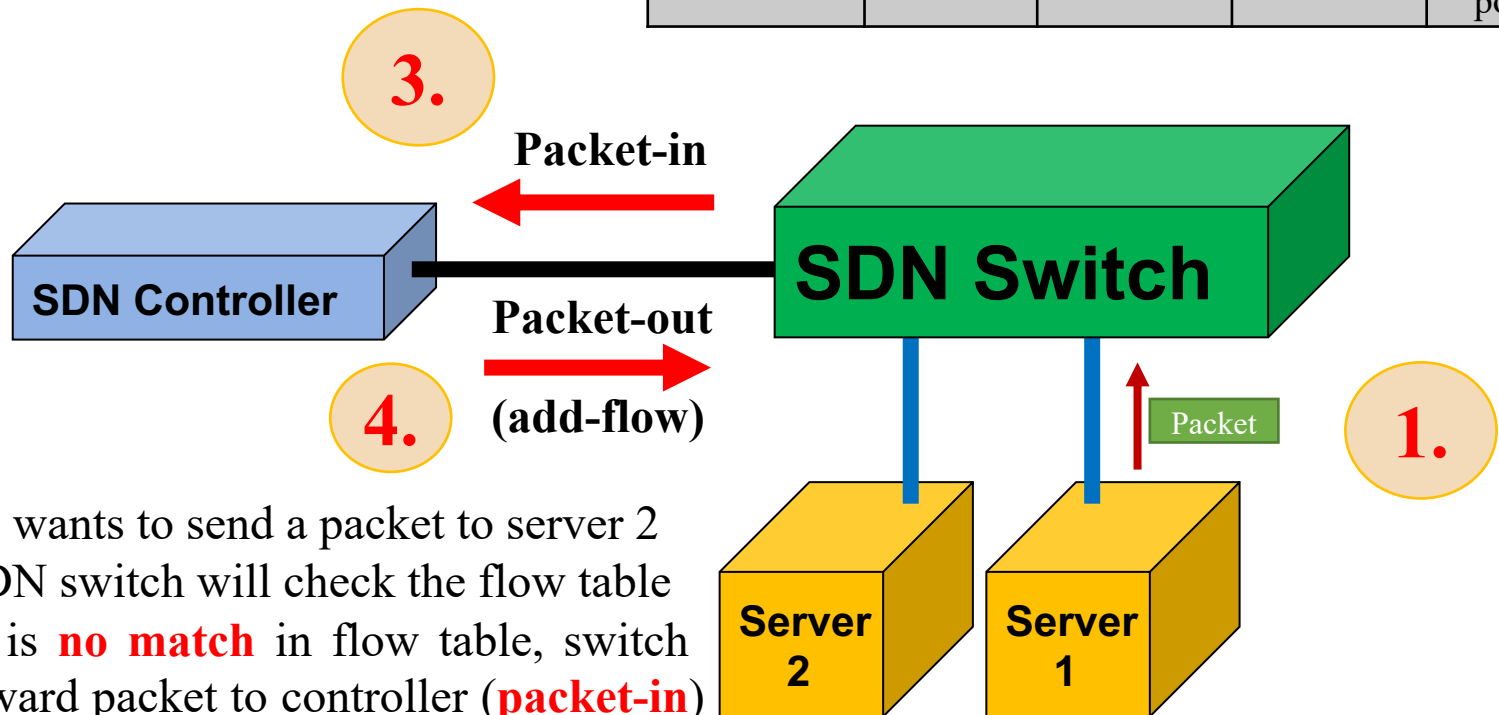
# Descriptions and Objectives (1/3)

2.

## About SDN Switch:

Flow table in switch

Ingress Port	Ethec Src	Ether Dst	...	Action
Port 1	192.168.14.3	*		Output port=2

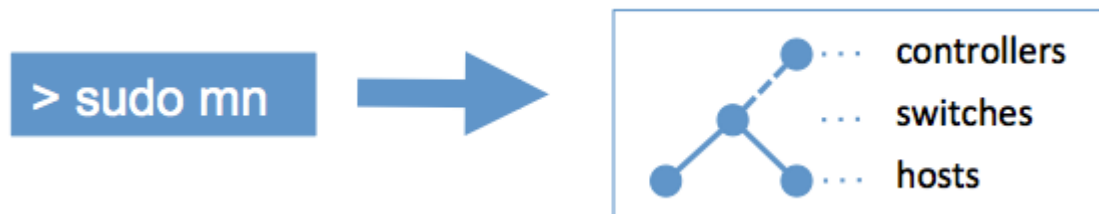


1. Server 1 wants to send a packet to server 2
2. First, SDN switch will check the flow table
3. If there is **no match** in flow table, switch will forward packet to controller (**packet-in**)
4. Controller will decide the **action** and send packet back to switch (**packet-out**), it can also add new flow entry in switch's flow table

# Descriptions and Objectives (2/3)

## About Mininet:

- Mininet creates a realistic **virtual network**, running real kernel, switch and application code
- It runs a collection of end-hosts, switches, routers, and links on a single Linux kernel
- The switches are OpenFlow-enabled



# Descriptions and Objectives (3/3)

## About SDN Controller Ryu:

- RYU supports the OpenFlow 1.0, 1.2, 1.3, 1.4 and 1.5
- RYU can work in conjunction with OpenStack for cloud computing
- Written in Python



# Project Content

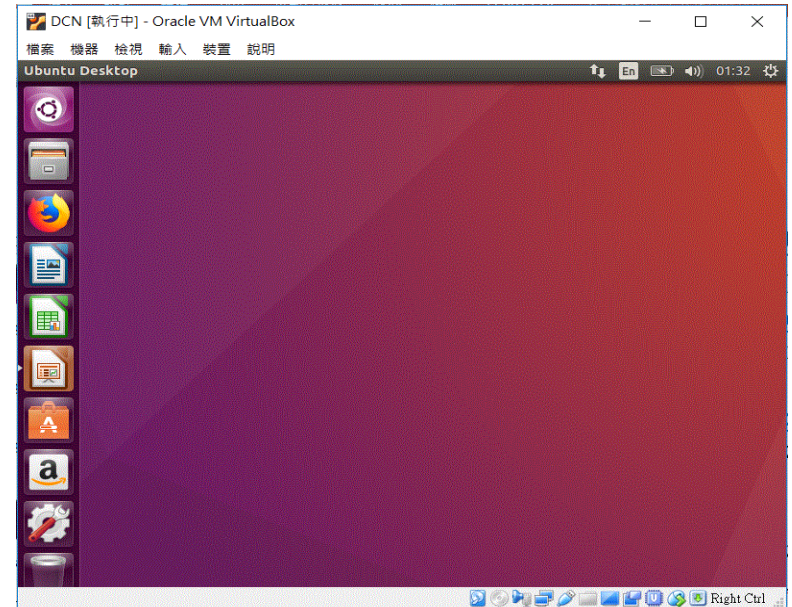
1. Create a VM which runs Ubuntu 16.04
2. Install Mininet
3. Install Ryu
4. Run Mininet and Ryu to emulate a simple SDN network system
5. Study SDN controller's sample code

# Step-by-Step Instructions (1/11)

## Step 1: Create a VM running Ubuntu 16.04

1. Go to <https://www.virtualbox.org/wiki/Downloads> and download VirtualBox
2. Go to <http://releases.ubuntu.com/16.04/> download the Ubuntu 16.04 desktop ISO file

3. Use Virtual Box to install Ubuntu ISO file



4. Following is the reference video  
<https://www.youtube.com/watch?v=RBU1xMP-SGc>



# Step-by-Step Instructions (2/11)

## Step 2: Install Mininet

Open a new terminal and enter the following commands

**1. Install git**

```
sudo apt-get install -y git
```

**2. Clone the mininet repository with git**

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

**3. Install mininet (takes 3-7 minutes)**

```
cd mininet/util
```

```
sudo ./install.sh -a
```

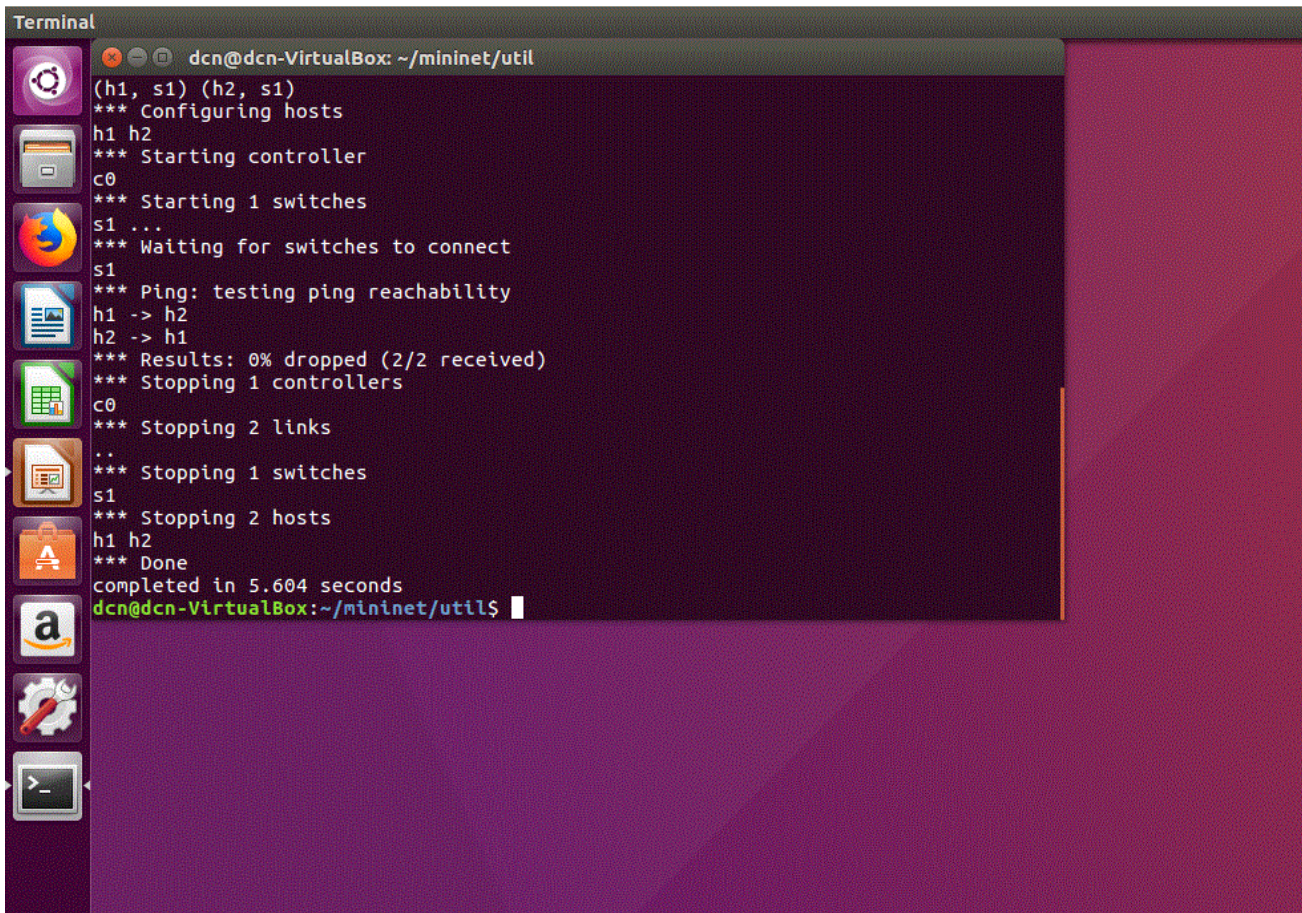
**4. Test the mininet installation and then close the terminal**

```
sudo mn --test pingall
```

# Step-by-Step Instructions (3/11)

## Step 2: Install Mininet

- You should see the following screen for a successful install

A terminal window titled "Terminal" with a dark background and light text. The window shows the output of a Mininet installation script. The output includes commands like (h1, s1) (h2, s1), status messages like "Configuring hosts", "Starting controller", "Starting 1 switches", "Waiting for switches to connect", "Ping: testing ping reachability", "Results: 0% dropped (2/2 received)", "Stopping 1 controllers", "Stopping 2 links", "Stopping 1 switches", "Stopping 2 hosts", and a final "Done" message. The terminal also shows the user's prompt "dcn@dcn-VirtualBox: ~/mininet/util\$" and the time taken to complete the installation, "completed in 5.604 seconds". The terminal window is part of a desktop environment with a sidebar on the left containing various application icons.

```
Terminal
dcn@dcn-VirtualBox: ~/mininet/util
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.604 seconds
dcn@dcn-VirtualBox:~/mininet/util$
```

# Step-by-Step Instructions (4/11)

## Step 3: Install Ryu

Open a new terminal and enter the following commands

### 1. Install required packages (5-10 minutes)

```
sudo apt-get install -y python-pip python-dev build-essential
sudo pip install --upgrade pip
sudo pip install --upgrade six
sudo apt-get install -y python-eventlet python-routes
sudo apt-get install -y python-webob python-paramiko
sudo pip install --upgrade setuptools
```

### 2. Download and install ryu

```
git clone git://github.com/osrg/ryu.git
cd ryu
sudo pip install .
```

### 3. Test (the result is on next page)

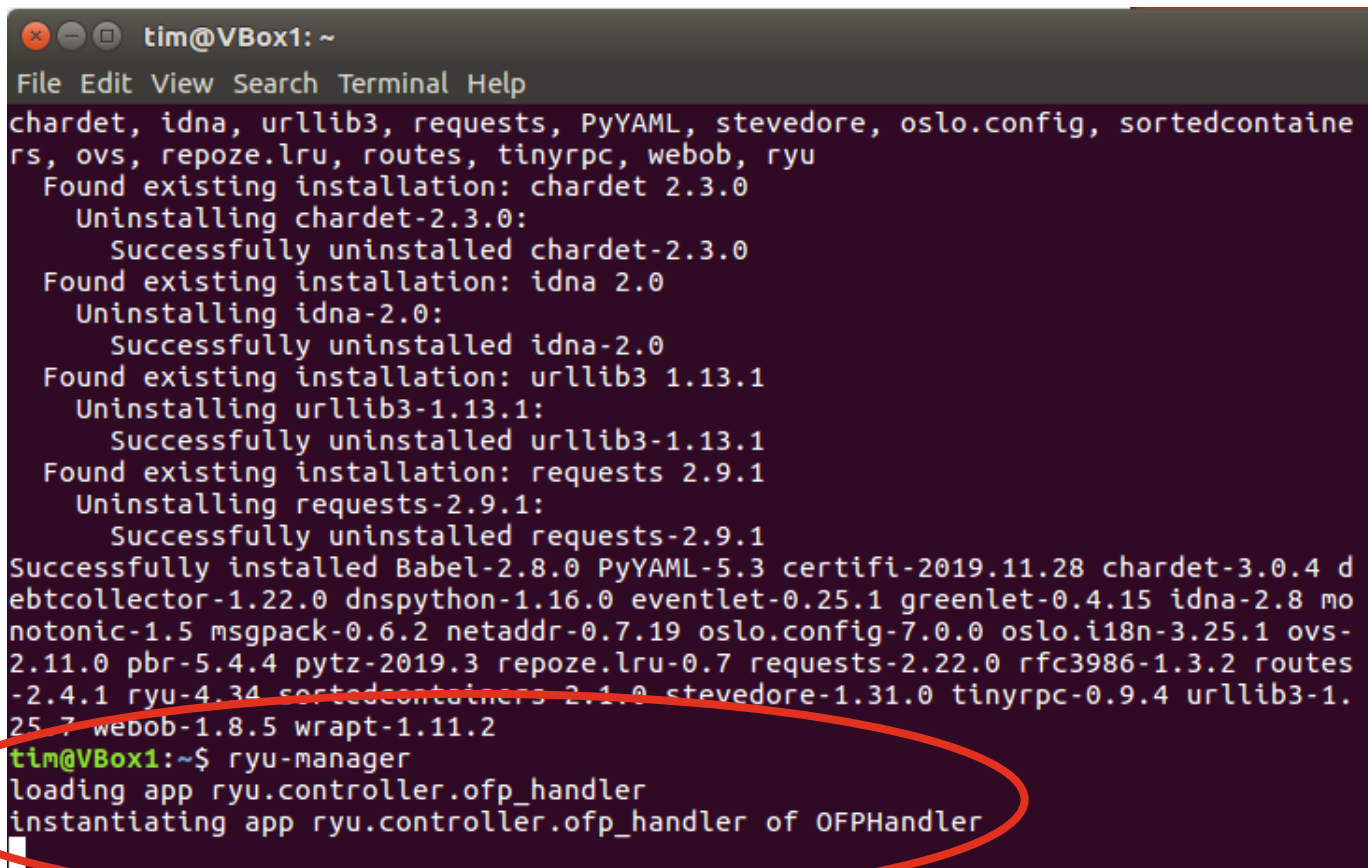
**ryu-manager**

### 4. Press “Ctrl+C” to leave ryu-manger and then close the terminal

# Step-by-Step Instructions (5/11)

## Step 3: Install Ryu

- If you can see this output, then Ryu is installed
- Quit **ryu-manager** by pressing “Ctrl+C” and close the terminal afterwards



```
tim@VBox1: ~  
File Edit View Search Terminal Help  
chardet, idna, urllib3, requests, PyYAML, stevedore, oslo.config, sortedcontainers,  
rs, ovs, repoze.lru, routes, tinyrpc, webob, ryu  
Found existing installation: chardet 2.3.0  
Uninstalling chardet-2.3.0:  
Successfully uninstalled chardet-2.3.0  
Found existing installation: idna 2.0  
Uninstalling idna-2.0:  
Successfully uninstalled idna-2.0  
Found existing installation: urllib3 1.13.1  
Uninstalling urllib3-1.13.1:  
Successfully uninstalled urllib3-1.13.1  
Found existing installation: requests 2.9.1  
Uninstalling requests-2.9.1:  
Successfully uninstalled requests-2.9.1  
Successfully installed Babel-2.8.0 PyYAML-5.3 certifi-2019.11.28 chardet-3.0.4 d  
ebcollector-1.22.0 dnspython-1.16.0 eventlet-0.25.1 greenlet-0.4.15 idna-2.8 mo  
notonic-1.5 msgpack-0.6.2 netaddr-0.7.19 oslo.config-7.0.0 oslo.i18n-3.25.1 ovs-  
2.11.0 pbr-5.4.4 pytz-2019.3 repoze.lru-0.7 requests-2.22.0 rfc3986-1.3.2 routes  
-2.4.1 ryu-4.34 sortedcontainers-2.1.0 stevedore-1.31.0 tinyrpc-0.9.4 urllib3-1.  
25.7 webob-1.8.5 wrapt-1.11.2  
tim@VBox1:~$ ryu-manager  
loading app ryu.controller.ofp_handler  
instantiating app ryu.controller.ofp_handler of OFPHandler
```

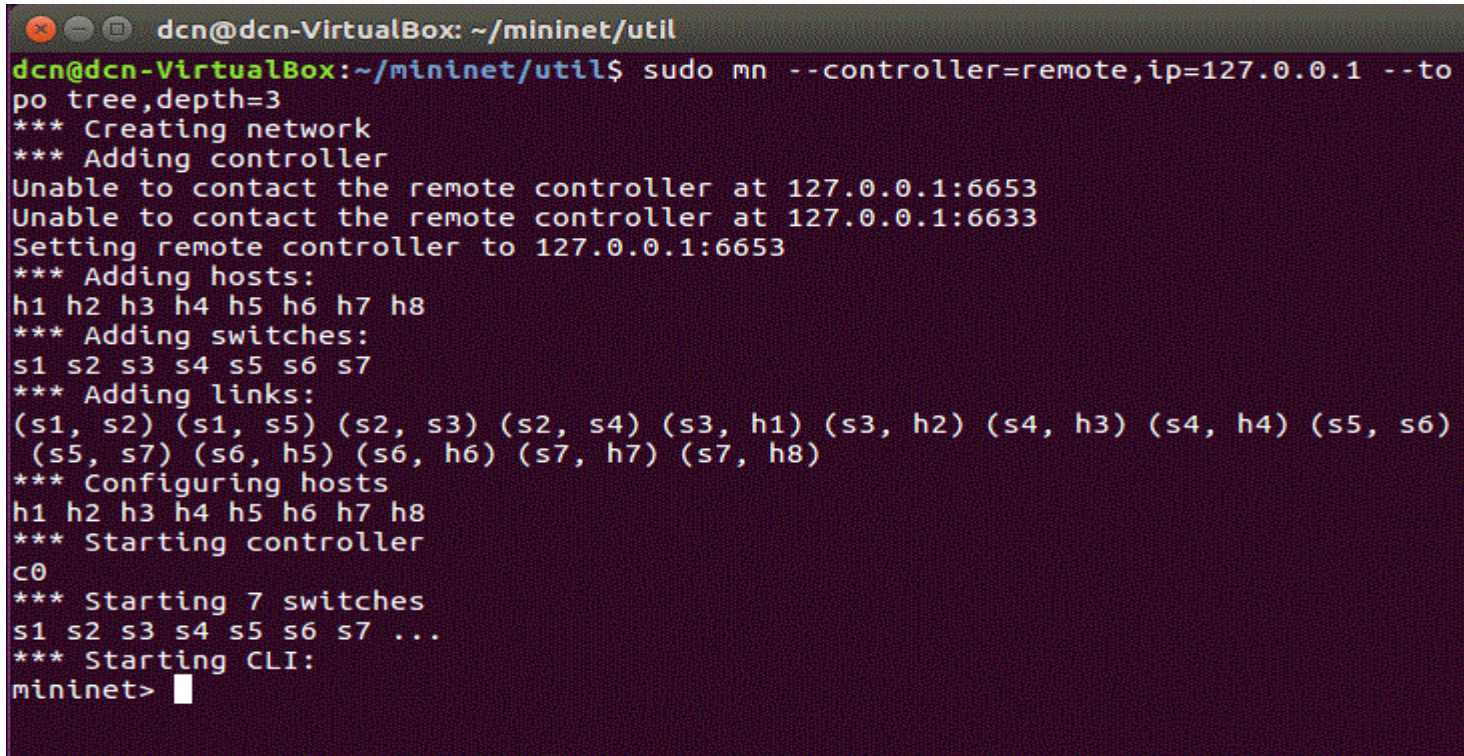


# Step-by-Step Instructions (6/11)

## Step 4: Run Mininet and Ryu to emulate a simple SDN network system

1. **Open a new terminal** and run mininet

```
sudo mn --controller=remote,ip=127.0.0.1 --topo tree,depth=3
```

A screenshot of a terminal window titled 'dcn@dcn-VirtualBox: ~/mininet/util'. The terminal shows the execution of the command 'sudo mn --controller=remote,ip=127.0.0.1 --topo tree,depth=3'. The output of the command is displayed in a light green font on a dark background. The output includes status messages for creating the network, adding a controller, adding hosts (h1-h8), adding switches (s1-s7), adding links, configuring hosts, starting the controller (c0), starting switches, and starting the CLI. The prompt 'mininet>' is shown at the bottom.

```
dcn@dcn-VirtualBox: ~/mininet/util
dcn@dcn-VirtualBox:~/mininet/util$ sudo mn --controller=remote,ip=127.0.0.1 --topo tree,depth=3
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7
*** Adding links:
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6)
(s5, s7) (s6, h5) (s6, h6) (s7, h7) (s7, h8)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
c0
*** Starting 7 switches
s1 s2 s3 s4 s5 s6 s7 ...
*** Starting CLI:
mininet> █
```

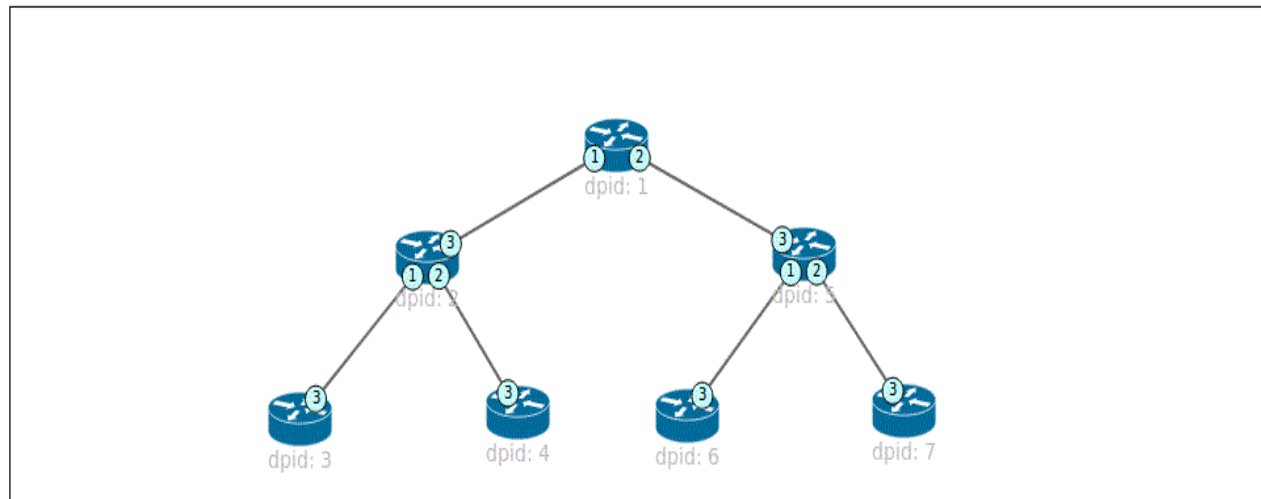
# Step-by-Step Instructions (7/11)

## Step 4: Run Mininet and Ryu to emulate a simple SDN network system

```
sudo mn --controller=remote,ip=127.0.0.1 --topo tree,depth=3
```

- The command above will create the tree topology below

### Ryu Topology Viewer



# Step-by-Step Instructions (8/11)

## Step 4: Run Mininet and Ryu to emulate a simple SDN network system

### 2. **Open another new terminal** and go to Ryu installation folder

- Get the Ryu installation directory:

```
pip show ryu
```

- Go to the specified directory, e.g.:

```
cd /usr/local/lib/python2.7/dist-packages
```

```
root@localhost:~# pip show ryu
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade to Python 3 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/with-new-python/#moving-forward
Name: ryu
Version: 4.34
Summary: Component-based Software-defined Networking Framework
Home-page: https://ryu-sdn.org
Author: Ryu project team
Author-email: ryu-devel@lists.sourceforge.net
License: Apache License 2.0
Location: /usr/local/lib/python2.7/dist-packages
Requires: webob, ovs, tynyrpc, routes, six, netaddr, msgpack, eventlet, oslo.config
Required-by:
root@localhost:~#
```

# Step-by-Step Instructions (9/11)

## Step 4: Run Mininet and Ryu to emulate a simple SDN network system

### 3. Check for Ryu installation folder existence

- You can see the Ryu folder (using '**ls**' command)

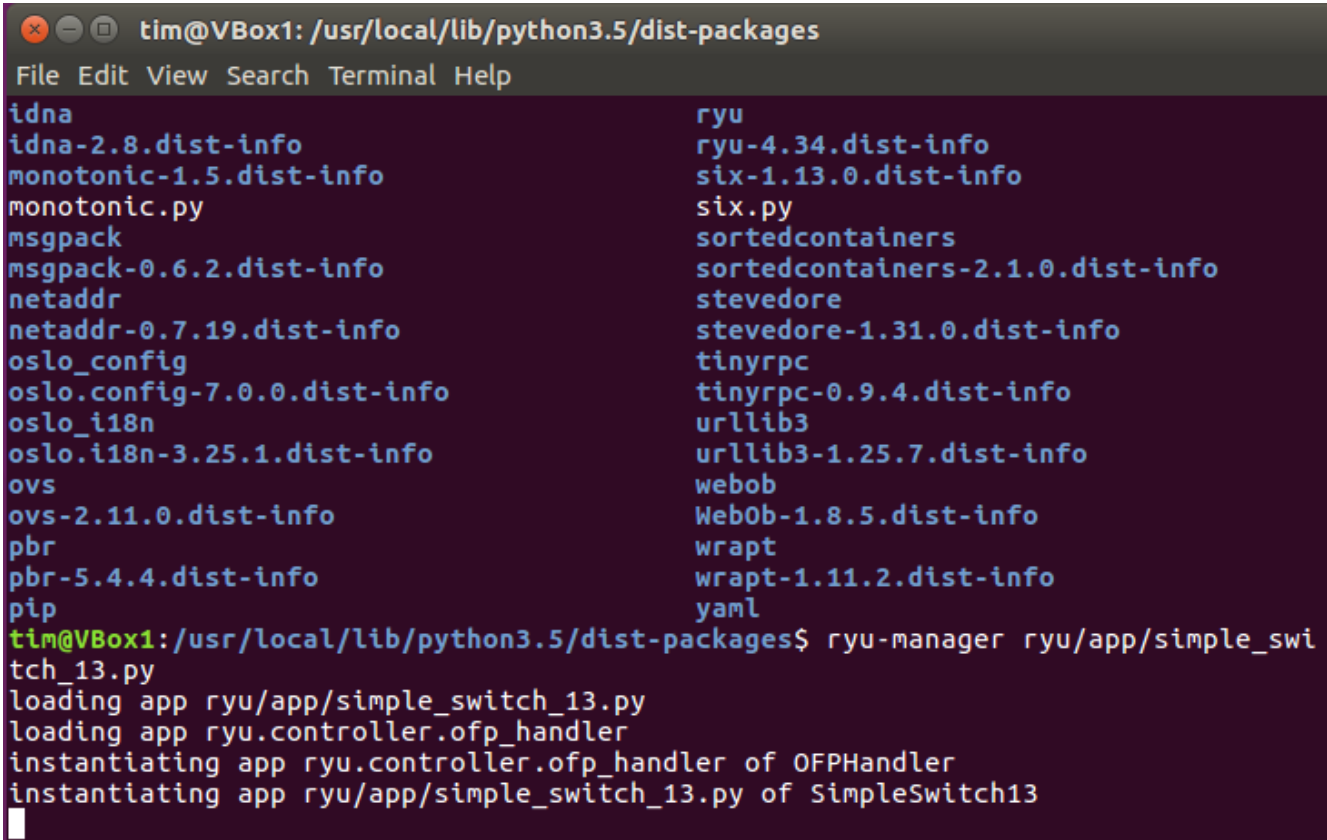
```
tim@VBox1:/usr/local/lib/python3.5/dist-packages$ ls
babel                                pip-19.3.1.dist-info
Babel-2.8.0.dist-info                __pycache__
certifi                              pytz
certifi-2019.11.28.dist-info         pytz-2019.3.dist-info
chardet                             PyYAML-5.3.dist-info
chardet-3.0.4.dist-info              repoze
debtcollector                        repoze.lru-0.7.dist-info
debtcollector-1.22.0.dist-info        repoze.lru-0.7-py3.6-nspkg.pth
dns                                  requests
dnspython-1.16.0.dist-info            requests-2.22.0.dist-info
eventlet                             rfc3986
eventlet-0.25.1.dist-info             rfc3986-1.3.2.dist-info
greenlet-0.4.15.dist-info             routes
greenlet.cpython-35m-x86_64-linux-gnu.so Routes-2.4.1.dist-info
idna                                  ryu
idna-2.8.dist-info                   ryu-4.34.dist-info
monotonic-1.5.dist-info               six-1.13.0.dist-info
monotonic.py                         six.py
```



# Step-by-Step Instructions (10/11)

**Step 4: Run Mininet and Ryu to emulate a simple SDN network system**

**4. Run controller sample code : `simple_switch_13.py`**  
**`ryu-manager ryu/app/simple_switch_13.py`**



```
tim@VBox1: /usr/local/lib/python3.5/dist-packages
File Edit View Search Terminal Help
idna
idna-2.8.dist-info
monotonic-1.5.dist-info
monotonic.py
msgpack
msgpack-0.6.2.dist-info
netaddr
netaddr-0.7.19.dist-info
oslo_config
oslo.config-7.0.0.dist-info
oslo_i18n
oslo.i18n-3.25.1.dist-info
ovs
ovs-2.11.0.dist-info
pbr
pbr-5.4.4.dist-info
pip
ryu
ryu-4.34.dist-info
six-1.13.0.dist-info
six.py
sortedcontainers
sortedcontainers-2.1.0.dist-info
stevedore
stevedore-1.31.0.dist-info
tinyrpc
tinyrpc-0.9.4.dist-info
urllib3
urllib3-1.25.7.dist-info
webob
WebOb-1.8.5.dist-info
wrapt
wrapt-1.11.2.dist-info
yaml
tim@VBox1:/usr/local/lib/python3.5/dist-packages$ ryu-manager ryu/app/simple_switch_13.py
loading app ryu/app/simple_switch_13.py
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ryu/app/simple_switch_13.py of SimpleSwitch13
```

# Step-by-Step Instructions (11/11)

## Step 4: Run Mininet and Ryu to emulate a simple SDN network system

### 5. In the mininet terminal use ‘**pingall**’ command to test

- Every host should successfully ping to each other
- You should also see ‘packet in’ command in the ryu-manager

```
tim@VBox1: ~  
File Edit View Search Terminal Help  
*** Adding switches:  
s1 s2 s3 s4 s5 s6 s7  
*** Adding links:  
(s1, s2) (s1, s5) (s2, s3) (s2, s4) (s3, h1) (s3, h2) (s4, h3) (s4, h4) (s5, s6)  
(s5, s7) (s6, h5) (s6, h6) (s7, h7) (s7, h8)  
*** Configuring hosts  
h1 h2 h3 h4 h5 h6 h7 h8  
*** Starting controller  
c0  
*** Starting 7 switches  
s1 s2 s3 s4 s5 s6 s7 ...  
*** Starting CLI:  
mininet> pingall  
*** Ping: testing ping reachability  
h1 -> h2 h3 h4 h5 h6 h7 h8  
h2 -> h1 h3 h4 h5 h6 h7 h8  
h3 -> h1 h2 h4 h5 h6 h7 h8  
h4 -> h1 h2 h3 h5 h6 h7 h8  
h5 -> h1 h2 h3 h4 h6 h7 h8  
h6 -> h1 h2 h3 h4 h5 h7 h8  
h7 -> h1 h2 h3 h4 h5 h6 h8  
h8 -> h1 h2 h3 h4 h5 h6 h7  
*** Results: 0% dropped (56/56 received)  
mininet> ☐
```

```
tim@VBox1: /usr/local/lib/python3.5/dist-packages  
File Edit View Search Terminal Help  
packet in 5 d6:f0:8f:aa:a4:17 33:33:00:00:00:fb 3  
packet in 6 d6:f0:8f:aa:a4:17 33:33:00:00:00:fb 3  
packet in 7 d6:f0:8f:aa:a4:17 33:33:00:00:00:fb 3  
packet in 2 36:e4:c7:4c:09:a3 33:33:00:00:00:02 2  
packet in 3 36:e4:c7:4c:09:a3 33:33:00:00:00:02 3  
packet in 1 36:e4:c7:4c:09:a3 33:33:00:00:00:02 1  
packet in 5 36:e4:c7:4c:09:a3 33:33:00:00:00:02 3  
packet in 6 d6:cb:06:ea:b8:8b 33:33:00:00:00:02 2  
packet in 7 36:e4:c7:4c:09:a3 33:33:00:00:00:02 3  
packet in 6 36:e4:c7:4c:09:a3 33:33:00:00:00:02 3  
packet in 4 2e:87:eb:cc:be:33 33:33:00:00:00:02 2  
packet in 5 d6:cb:06:ea:b8:8b 33:33:00:00:00:02 1  
packet in 2 2e:87:eb:cc:be:33 33:33:00:00:00:02 2  
packet in 1 d6:cb:06:ea:b8:8b 33:33:00:00:00:02 2  
packet in 7 d6:cb:06:ea:b8:8b 33:33:00:00:00:02 3  
packet in 3 2e:87:eb:cc:be:33 33:33:00:00:00:02 3
```

# DEMO

- We will have DEMO on **04/01 Thursday**
  - Please go to [this link](#) and choose the time that you prefer
  - Demo location will be written there too
- TA will ask questions about controller sample code (**simple\_switch\_13.py**)
  - This is to make sure you understand the controller program
  - You will need to learn and review the logic of the code
- **03/23 Tues** at ED302 13:20 – 14:10 we have help session
  - Students who have problems or questions can attend
- Email TA, if you have any questions

## Reference:

- Mininet: <http://mininet.org/>
- Ryu: <http://osrg.github.io/ryu/>
- Ryu book <http://osrg.github.io/ryu-book/en/Ryubook.pdf>

(There is detailed explanations for sample code in chap. 2)