

# SDN Lab1

TA: 陳俊廷

Lab: ED817

Email: [andy1995030978@gmail.com](mailto:andy1995030978@gmail.com)

# Outline

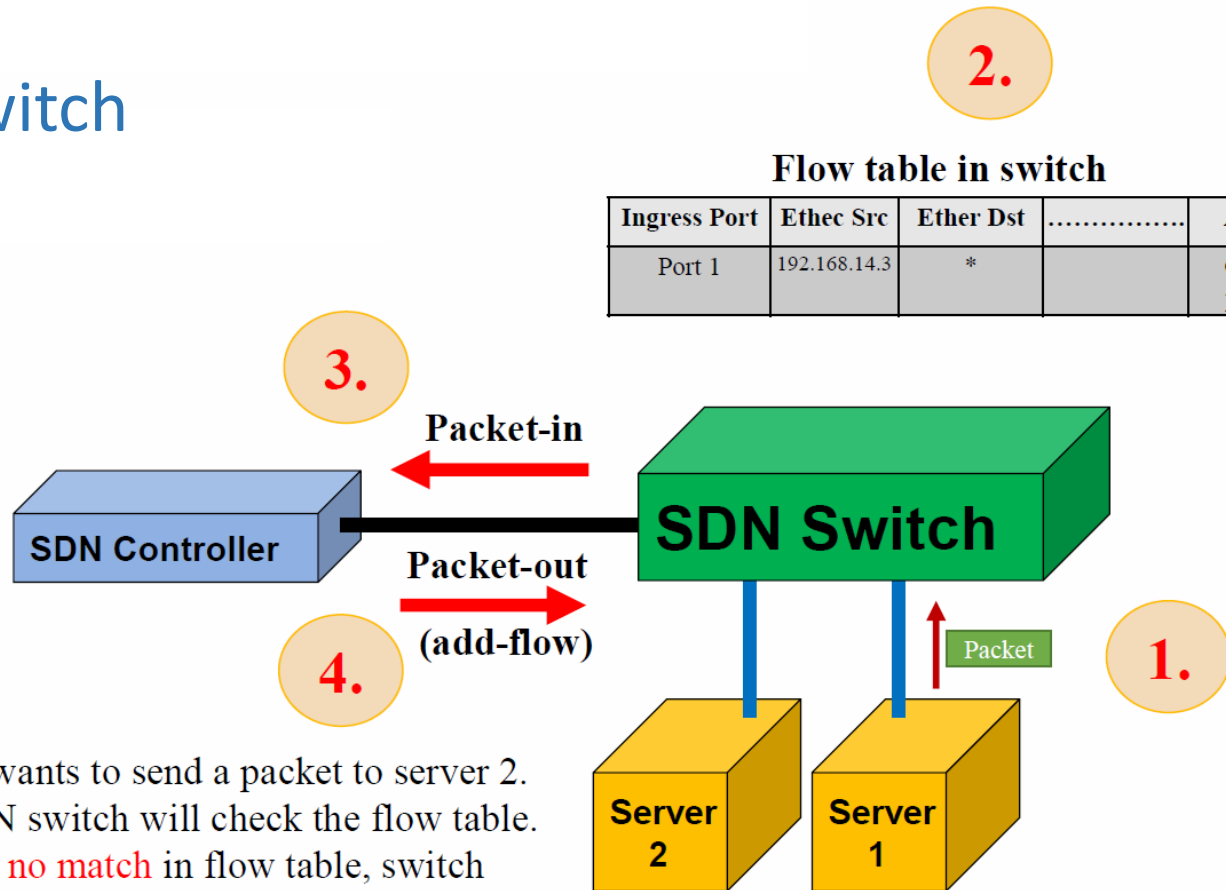
- Goals
- Objectives
- Lab Content
- Instructions
- Reference

# Goals

- Learn how to use Virtual Machine and be familiar with Ubuntu Linux
- Learn basic ideas of Software Defined Network (SDN) and SDN switches
- Learn how to use Mininet to create a simple network topology
- Learn how to construct SDN controller Ryu and use it to control flows on the created topology

# Objectives

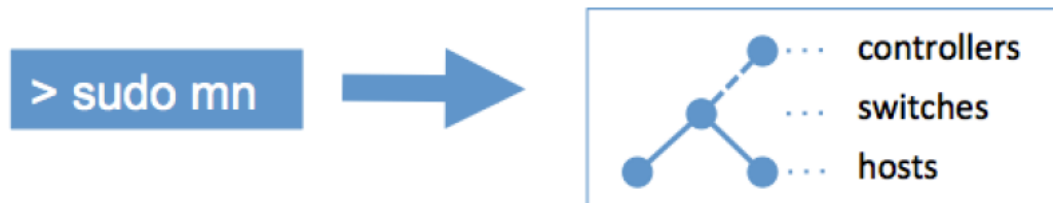
- SDN switch



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**)

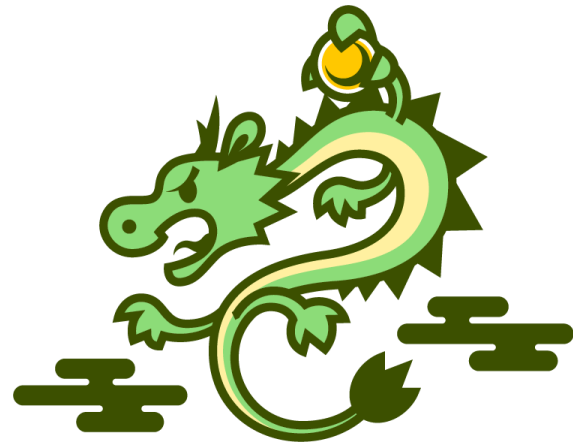
# Objectives

- Mininet
  - It 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 created switches are OpenFlow-enabled



# Objectives

- SDN controller RYU
  - RYU supports the OpenFlow1.0, 1.2, 1.3 and 1.4
  - RYU can work in conjunction with OpenStack for cloud computing
  - Written in Python



# Lab Content

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

# Instructions

- Step 1: Create a VM which runs Ubuntu
  - Download 32-bits VMware player 6.0.7 or the latest version of Virtual Box
  - Download the latest version of Ubuntu (32-bits)
  - Install them step-by-step
  - For created VM at ED713
    - user name: sdn
    - password: mininet



# Instructions

- Step 2: Install Mininet
  - Install git  
`sudo apt-get install git`
  - Download mininet by git  
`git clone git://github.com/mininet/mininet`
  - Install mininet (it will take 5~10 minutes)  
`cd mininet/util`  
`./install.sh -a`
  - Test  
`sudo mn --test pingall`

# Instructions

- Step 2: Install Mininet

```
sdn@ubuntu:~/mininet/util$ sudo mn --test pingall
*** 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 ...
*** 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 5.614 seconds
sdn@ubuntu:~/mininet/util$
```

# Instructions

- Step 3: Install Ryu

- Install the following packages

- ```
sudo apt-get install python-pip python-dev build-essential
```

- ```
sudo pip install --upgrade pip
```

- ```
sudo easy_install pip
```

- ```
sudo pip install --upgrade six
```

- ```
sudo apt-get install python-eventlet python-routes
```

- ```
sudo apt-get install python-webob python-paramiko
```

- Download and install Ryu

- ```
sudo pip install ryu
```

- Test

- ```
ryu-manager
```

- Press “Ctrl+C” to leave ryu-manger

# Instructions

- Step 4: Run Mininet and Ryu to emulate a simple SDN network system
  - Run mininet and create a tree topology with depth=3

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

```
sdn@ubuntu:~/mininet/util$ ryu-manager
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
^Csdn@ubuntu:~/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> 
```

# Instructions

- Step 4: Run Mininet and Ryu to emulate a simple SDN network system
  - Open a new terminal and go to following folder  
`cd /usr/local/lib/python2.7/dist-packages`  
`ls`

```
sdn@ubuntu:~$ cd /usr/local/lib/python2.7/dist-packages
sdn@ubuntu:/usr/local/lib/python2.7/dist-packages$ ls
babel                               msgpack                             pbr
Babel-2.3.4.dist-info              msgpack_python-0.4.8.dist-info      pbr-1.10.0.dist-info
debtcollector                      netaddr                             pip
debtcollector-1.8.0.dist-info      netaddr-0.7.18.dist-info            pip-8.1.2.dist-info
easy-install.pth                   oslo_config                          pytz
funcsigs                           oslo.config-3.17.0.dist-info        pytz-2016.6.1.dist-info
funcsigs-1.0.2.dist-info            oslo_i18n                          /rfc3986
mininet-2.3.0d1-py2.7.egg          oslo.i18n-3.9.0.dist-info          /rfc3986-0.4.1.dist-info
sdn@ubuntu:/usr/local/lib/python2.7/dist-packages$
```

The terminal output shows the contents of the directory `/usr/local/lib/python2.7/dist-packages`. The package `ryu` is circled in red in the original image.

# Instructions

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

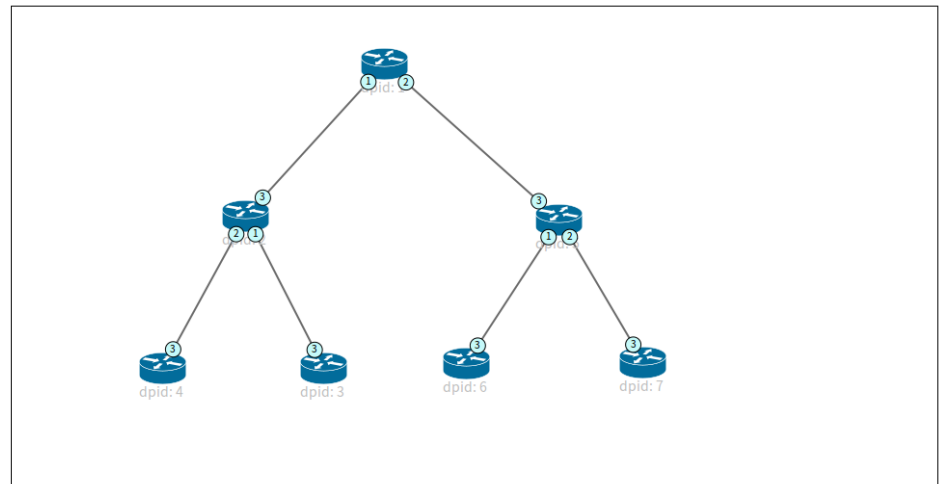
- Run the Ryu topology viewer

`PYTHONPATH=. ryu run --observe-links ryu/app/gui_topology/gui_topology.py`

- Open the web browser

`http://127.0.0.1:8080`

**Ryu Topology Viewer**



# Instructions

- Step 4: Run Mininet and Ryu to emulate a simple SDN network system
  - Press “Ctrl+C” to stop the topology viewer
  - Enter following command to make switches supports OF 1.3 (in this topology we have 7 switches s1~s7)

```
sudo ovs-vsctl set bridge s1 protocols=OpenFlow13
sudo ovs-vsctl set bridge s2 protocols=OpenFlow13
sudo ovs-vsctl set bridge s3 protocols=OpenFlow13
sudo ovs-vsctl set bridge s4 protocols=OpenFlow13
sudo ovs-vsctl set bridge s5 protocols=OpenFlow13
sudo ovs-vsctl set bridge s6 protocols=OpenFlow13
sudo ovs-vsctl set bridge s7 protocols=OpenFlow13
```

# Instructions

- Step 4: Run Mininet and Ryu to emulate a simple SDN network system
  - Run controller sample code: `simple_switch_13.py`  
`ryu-manager ryu/app/simple_switch_13.py`
  - Go back to terminal with Mininet  
`pingall`

```
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> █
```



# Instructions

- Step 5: Study SDN controller's sample code
  - Open `simple_switch_13.py`
  - Study the sample code with reference, `Ryubook.pdf` (Chap.1)
  - TA will check Step 1 to Step 4 and ask some simple questions about the sample code

# Reference

- VMware player 6.0.7:  
[https://my.vmware.com/web/vmware/free#desktop\\_end\\_user\\_computing/vmware\\_player/6\\_0|PLAYER-607](https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0|PLAYER-607)
- Ubuntu: <https://www.ubuntu-tw.org/modules/tinyd0/>
- Mininet: <http://mininet.org/>
- Ryu: <http://osrg.github.io/ryu/>
- Ryu book: <http://osrg.github.io/ryu-book/en/Ryubook.pdf>