

Report

0816169 陳伯庭

Execution:

1. steps for finishing this lab and how to run my code:

First of all, I followed the content from the slide to setup the environment. Secondly, I modified “topo.py” by adding the constraints (bandwidth, delay, loss) in addLink function so as to build the topology via mininet which satisfy “topo.png”. Then I duplicated the sample code “SimpleController.py” two times, in each copy, I followed the forwarding rules in page 29 and 31, modified the function “switch_features_handler(self, ev)” . Last but not least, I measured simple code and my code. I ran topo.py in a terminal, and run “SimpleController.py” , “controller1” , “controller2” respectively in another terminal. Later, I enter “h1 ping h2” to make sure that ICMP and APR packets can reach the destination. Moreover, I create an

“out” directory in src to store the different results. After entering iperf command in mininet CLI, I was able to measure my bandwidth and store the result in ./out/result1 or 2 or 3. I also got the packets number in the terminal running the controller. Finally, I opened a new terminal and using the following command so as to output the forwarding rules. I would start over the above steps so as to measure different forwarding rules.

2. What is the meaning of the executing command?

ryu-manager: loads Ryu application and run it.

--observe-links: manifest the message between connection.

mn-c: clean up the mininet or RTNETLINK

mn --custom topo.py : choose py file by custom parameter.

--topo topo : according to the name of the last line of python.

--link tc : users can setup via connection.

--controller remote : setup controller, remote = controlled by outside controller.

3. Screenshot

SimpleController:

The screenshot displays a network simulation environment. On the left, there is a 'Quick connect...' panel with a tree view of the network topology. The tree shows a root node 'lab2-botingchen/src' with several sub-nodes: 'topo', 'out', 'pycache', 'topo.py', 'SimpleController.py', 'controller2.py', and 'controller1.py'. Below the tree is a 'Remote monitoring' button. The main area of the interface is a terminal window showing the output of the simulation. The terminal output includes ping statistics for 10.0.0.2, a client connecting to 10.0.0.1 on UDP port 5566, and a table of network performance metrics. The table shows a consistent transfer rate of 1.05 Mbits/sec and a bandwidth of 1.05 Mbits/sec. The terminal also shows a 'Server Report' with a total transfer of 1.17 Mbytes and a bandwidth of 981 Kbits/sec.

```
Quick connect...
/root/lab2-botingchen/src/
Name
  ...
  topo
  out
  __pycache__
  topo.py
  SimpleController.py
  controller2.py
  controller1.py
Remote monitoring

lab2-botingchen/src
local
.cache
.rnd
.profile
.mininet_history
.gitconfig
.bashrc
.bash_history
Remote monitoring

64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=10.0 ms
^C
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 10.079/10.083/10.089/0.004 ms
mininet> h1 iperf -s -u -i 1 -p 5566 > ./out/result1 &
mininet> h2 iperf -c 10.0.0.1 -u -i 1 -p 5566
-----
Client connecting to 10.0.0.1, UDP port 5566
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 10.0.0.2 port 42910 connected with 10.0.0.1 port 5566
[ ID] Interval      Transfer     Bandwidth
[ 3] 0.0- 1.0 sec   129 KBytes   1.06 Mbits/sec
[ 3] 1.0- 2.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 2.0- 3.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 3.0- 4.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 4.0- 5.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 5.0- 6.0 sec   129 KBytes   1.05 Mbits/sec
[ 3] 6.0- 7.0 sec   129 KBytes   1.06 Mbits/sec
[ 3] 7.0- 8.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 8.0- 9.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 9.0-10.0 sec   128 KBytes   1.05 Mbits/sec
[ 3] 0.0-10.0 sec   1.25 MBytes   1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec   1.17 MBytes   981 Kbits/sec   0.352 ms   57/ 893 (6.4%)
mininet>
```

The screenshot shows a terminal window with the command 'ovs-ofctl -O OpenFlow13 dump-f' executed. The output displays a list of OpenFlow flow entries. The entry for 'cookie=0x0, duration=100.458s, table=0, n_packets=0, n_bytes=0, priority=3,udp, in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.1,tp_dst=5566 actions=output:2' is highlighted with a red box. The entry for 'cookie=0x0, duration=100.458s, table=0, n_packets=837, n_bytes=1265544, priority=3,udp,in_port=2,nw_src=10.0.0.2,nw_dst=10.0.0.1,tp_dst=5566 actions=output:1' is also highlighted with a red box. The entry for 'cookie=0x0, duration=89.730s, table=0, n_packets=24, n_bytes=1792, priority=1,1 n_port=3,dl_dst=92:50:47:ff:b1:ea actions=output:1' is highlighted with a red box. The entry for 'cookie=0x0, duration=89.340s, table=0, n_packets=16, n_bytes=2814, priority=1,1 n_port=1,dl_dst=7a:aa:00:51:08:19 actions=output:3' is highlighted with a red box. The entry for 'cookie=0x0, duration=100.458s, table=0, n_packets=25825, n_bytes=1807498, priority=0 actions=CONTROLLER:65535' is highlighted with a red box.

```
H lab2-0816169 root ~ lab2-botingchen src ovs-ofctl -O OpenFlow13 dump-f
flows s2
OFPST FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=100.452s, table=0, n_packets=175, n_bytes=10500, priority=
65535,dl_dst=01:80:c2:00:00:0e,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=100.458s, table=0, n_packets=0, n_bytes=0, priority=3,udp,
in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.1,tp_dst=5566 actions=output:2
 cookie=0x0, duration=100.458s, table=0, n_packets=837, n_bytes=1265544, priorit
y=3,udp,in_port=2,nw_src=10.0.0.2,nw_dst=10.0.0.1,tp_dst=5566 actions=output:1
 cookie=0x0, duration=89.730s, table=0, n_packets=24, n_bytes=1792, priority=1,1
n_port=3,dl_dst=92:50:47:ff:b1:ea actions=output:1
 cookie=0x0, duration=89.340s, table=0, n_packets=16, n_bytes=2814, priority=1,1
n_port=1,dl_dst=7a:aa:00:51:08:19 actions=output:3
 cookie=0x0, duration=100.458s, table=0, n_packets=25825, n_bytes=1807498, prior
ity=0 actions=CONTROLLER:65535
```

Controller1:

```

Quick connect...
/root/lab2-botingchen/src/
Name
..
topo
out
_pycache_
topo.py
SimpleController.py
controller2.py
controller1.py
Remote monitoring

64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=20.0 ms
^C
--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 20.045/20.132/20.851/0.255 ms
mininet> h1 iperf -s -u -i 1 -p 5566 > ./out/result2 &
mininet> h2 iperf -c 10.0.0.1 -u -i 1 -p 5566
-----
[ 3] local 10.0.0.2 port 56604 connected with 10.0.0.1 port 5566
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec   129 KBytes    1.06 Mbits/sec
[ 3] 1.0- 2.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 2.0- 3.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 3.0- 4.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 4.0- 5.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 5.0- 6.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 6.0- 7.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 7.0- 8.0 sec   129 KBytes    1.06 Mbits/sec
[ 3] 8.0- 9.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 9.0-10.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 0.0-10.0 sec   1.25 MBytes    1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec   1.15 MBytes    959 Kbits/sec    1.356 ms    76/ 893 (8.5%)
mininet>

```

```

5 switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 449 packets
switch 2: count 818 packets
switch 2: count 818 packets
switch 2: count 818 packets
switch 2: count 818 packets

```

```

ity=0 actions=CONTROLLER:65535
H lab2 0816169 root ~ | lab2-botingchen | src ovs-ofctl -O OpenFlow13 dump-f
lows s2
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=173.147s, table=0, n_packets=281, n_bytes=16860, priority=
65535,dl_dst=01:80:c2:00:00:0e,dl_type=0x88cc actions=CONTROLLER:65535
 cookie=0x0, duration=173.153s, table=0, n_packets=0, n_bytes=0, priority=3,udp,
in_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.2,tp_dst=5566 actions=output:2
 cookie=0x0, duration=173.153s, table=0, n_packets=818, n_bytes=1236816, priorit
y=3,udp,in_port=3,nw_src=10.0.0.2,nw_dst=10.0.0.1,tp_dst=5566 actions=output:1
 cookie=0x0, duration=167.799s, table=0, n_packets=35755, n_bytes=1502830, prior
ity=1,in_port=2,dl_dst=0a:ae:da:c5:36:f3 actions=output:1
 cookie=0x0, duration=167.791s, table=0, n_packets=25, n_bytes=3696, priority=1,
in_port=1,dl_dst=1e:41:9a:0b:38:b0 actions=output:2
 cookie=0x0, duration=173.153s, table=0, n_packets=38222, n_bytes=1605380, prior
ity=0 actions=CONTROLLER:65535

```

Controller2:

```

Quick connect...
/root/lab2-botingchen/src/
Name
...
topo
out
__pycache__
topo.py
SimpleController.py
controller2.py
controller1.py
Sftp
Remote monitoring

64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=20.0 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7018ms
rtt min/avg/max/mdev = 20.050/27.410/78.893/19.459 ms
mininet> h1 iperf -s -u -i 1 -p 5566 > ./out/result3 &
mininet> h2 iperf -c 10.0.0.1 -u -i 1 -p 5566
-----
Client connecting to 10.0.0.1, UDP port 5566
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] Local 10.0.0.2 port 36643 connected with 10.0.0.1 port 5566
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec   129 KBytes    1.06 Mbits/sec
[ 3] 1.0- 2.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 2.0- 3.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 3.0- 4.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 4.0- 5.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 5.0- 6.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 6.0- 7.0 sec   129 KBytes    1.06 Mbits/sec
[ 3] 7.0- 8.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 8.0- 9.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 9.0-10.0 sec   128 KBytes    1.05 Mbits/sec
[ 3] 0.0-10.0 sec   1.25 MBytes    1.05 Mbits/sec
[ 3] Sent 893 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec   1.20 MBytes    1.01 Mbits/sec    0.896 ms    34/ 893 (3.8%)
mininet>

```

```

5 switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 0 packets
switch 2: count 624 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets
switch 2: count 860 packets

```

```

H lab2_0816169 root ~ | lab2-botingchen | src ovs-ofctl -O OpenFlow13 dump-f
lows s2
OFPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=89.220s, table=0, n_packets=151, n_bytes=9060, priority=65
535,d1,dst=01:80:c2:00:00:0e,d1_type=0x88cc,actions=CONTROLLER:65535
 cookie=0x0, duration=89.227s, table=0, n_packets=0, n_bytes=0, priority=3,udp,i
n_port=1,nw_src=10.0.0.1,nw_dst=10.0.0.1,tp_dst=5566,actions=output:2
 cookie=0x0, duration=89.227s, table=0, n_packets=860, n_bytes=1300320, priority
=3,udp,in_port=3,nw_src=10.0.0.2,nw_dst=10.0.0.1,tp_dst=5566,actions=output:1
 cookie=0x0, duration=84.624s, table=0, n_packets=9882, n_bytes=415492, priority
=1,in_port=2,d1,dst=ca:ae:dc:97:68:e7,actions=output:1
 cookie=0x0, duration=84.618s, table=0, n_packets=11, n_bytes=2324, priority=1,i
n_port=1,d1,dst=fe:dd:0a:d0:3f:cf,actions=output:2
 cookie=0x0, duration=89.227s, table=0, n_packets=18920, n_bytes=794696, priorit
y=0,actions=CONTROLLER:65535

```

Discription:

1. packet in 和 packet out 是來自於 controller 的指令。如果 openflow 交換器收到 packet in，要對於接收到的封包進行轉

送到 Controller 的動作。相反的收到 packet out 的話，要對於接收到來自 Controller 的封包轉送到指定的連接埠。

2. 一個 packet 在一個 flow table 中沒有發現能夠匹配的 flow entry.
3. 要撰寫一支 Ryu 應用程式，你只需要將你的應用程式類別繼承自 RyuApp 即可
4. OpenFlow 交換器以及 Flow table 的操作都是透過 Datapath 類別的實體來進行。 在一般的情況下，會由事件傳遞給事件管理的訊息中取得
5. 使用 UDP 以及設定協定為 IPv4
6. controller2.py，因為 loss 比較少