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 runnung 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.

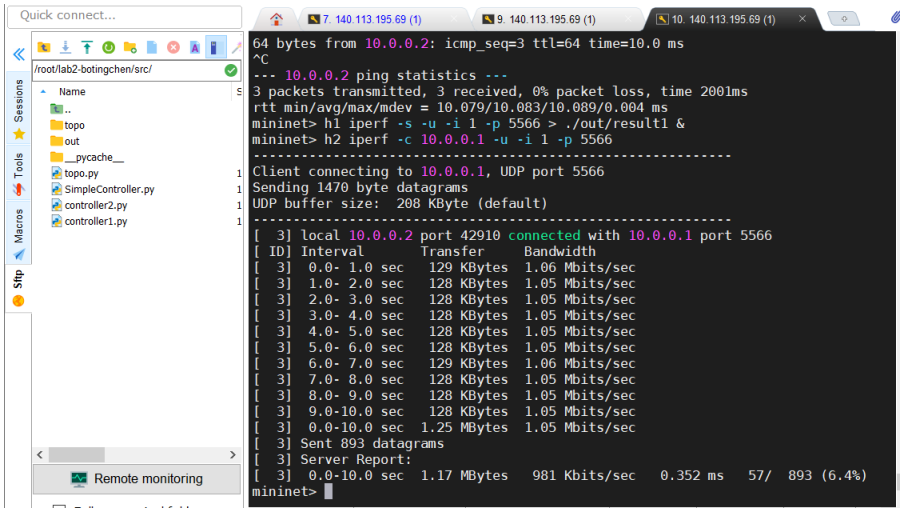
--topo topo : according to the name of the last line of python.

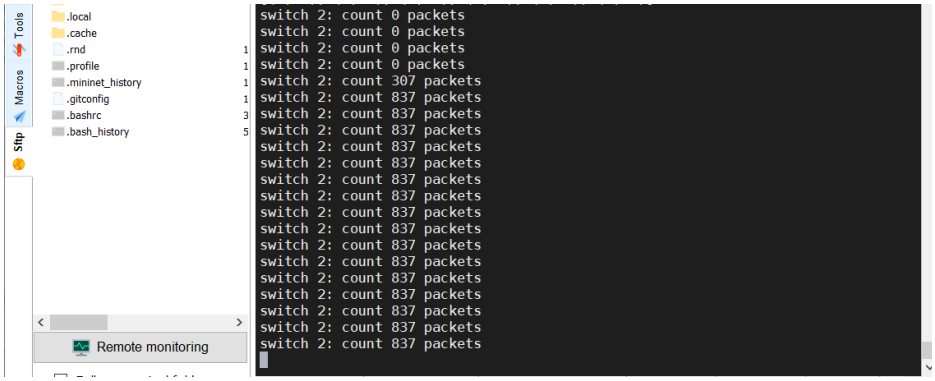
--link tc : users can setup via connection.

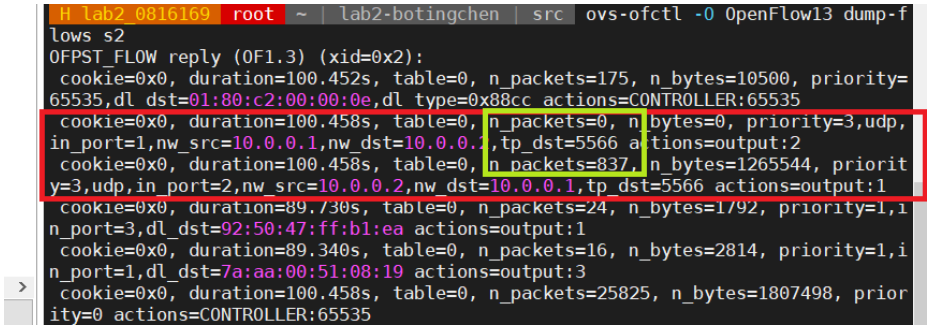
--controller remote : setup controller, remote = controlled by outside controller.

3. Screenshot

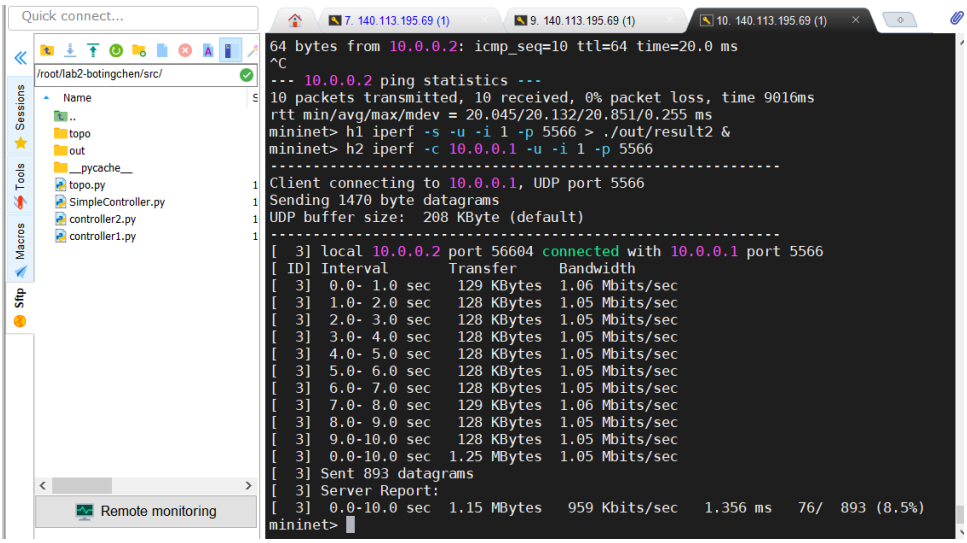
SimpleController:

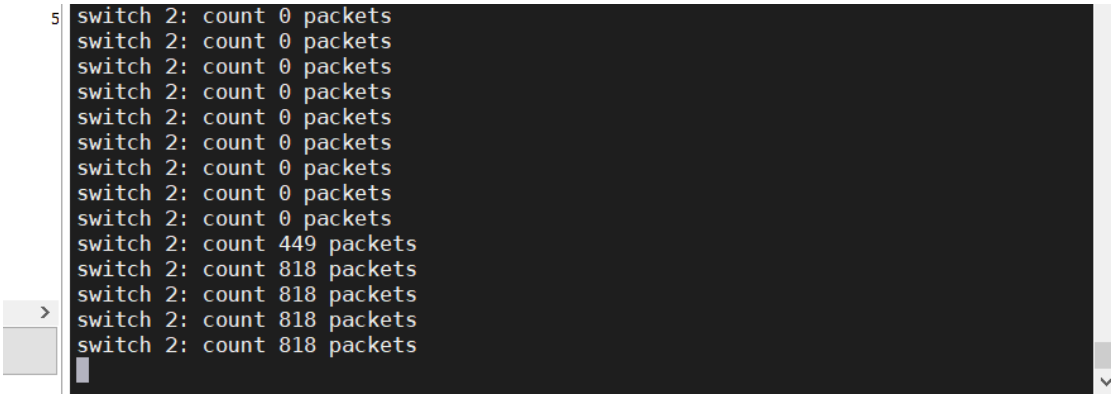


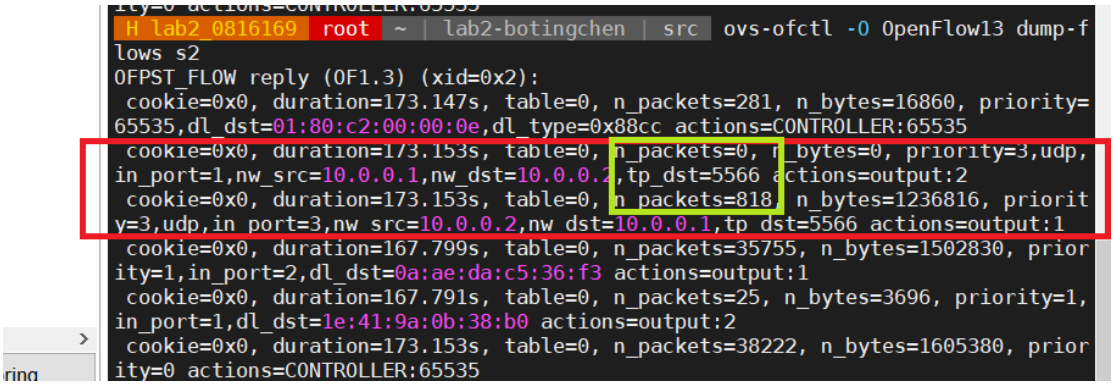




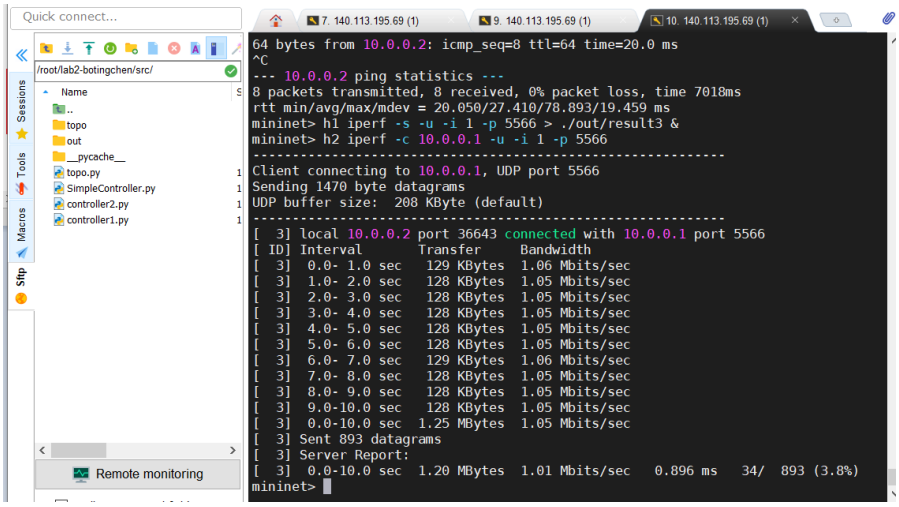
Controller1:

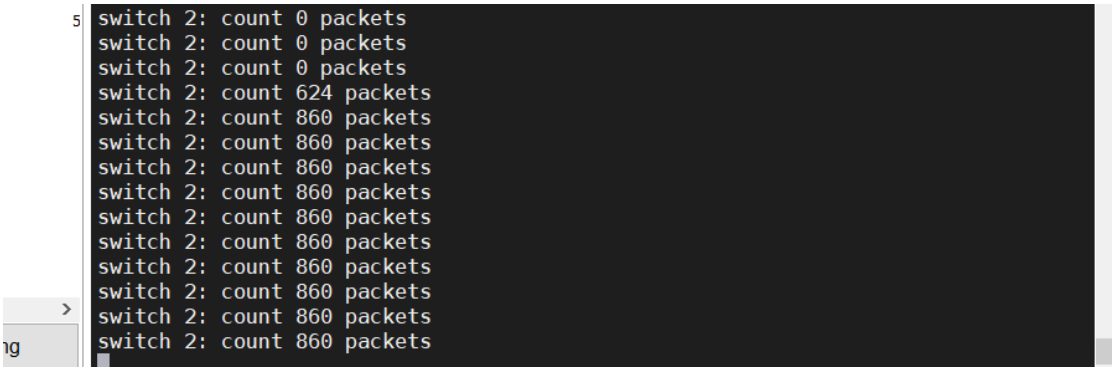


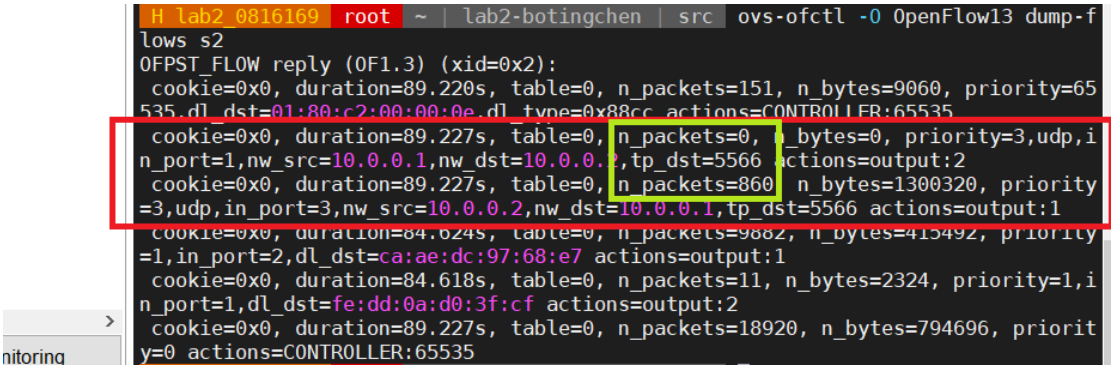




Controller2:







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