BÁO CÁO THỰC HÀNH

Môn học: Công nghệ mạng khả lập trình

Buổi báo cáo: Lab 02

GVHD: Phan Xuân Thiện

Ngày thực hiện: 16/10/2025

THÔNG TIN CHUNG:

Lớp: NT541.Q11.2

STT	Họ và tên	MSSV	Email
1	Lê Hữu Khánh	22520636	22520636@gm.uit.edu.vn

1. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	1 ngày
Link Video thực hiện *(nếu có)*	
Ý kiến *(nếu có)* + Khó khăn + Đề xuất …	
Điểm tự đánh giá	10

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

I. Yêu cầu 1

1. Cài đặt SDN/OpenFlow Controller (remote controller). Dùng Ryu controller (python)

- Clone Github Repo source code Ryu Controller

```
khanh@ubuntu:~$ git clone https://github.com/faucetsdn/ryu.git
fatal: destination path 'ryu' already exists and is not an empty directory.
khanh@ubuntu:~$ rm -rf ryu/
khanh@ubuntu:~$ git clone https://github.com/faucetsdn/ryu.git
Cloning into 'ryu' ...
remote: Enumerating objects: 26506, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
Receiving objects:  16% (4274/26506), 1.90 MiB | 25.00 KiB/s
```

- Cài đặt Ryu Controller

```
khanh@ubuntu:~$ cd ryu
khanh@ubuntu:~/ryu$ pip install .
Processing /home/khanh/ryu
Requirement already satisfied: eventlet==0.31.1 in /home/khanh/.local/lib/python3.8/site-packages
(from ryu==4.34) (0.31.1)
Requirement already satisfied: msgpack>=0.4.0 in /home/khanh/.local/lib/python3.8/site-packages (f
rom ryu==4.34) (1.1.1)
Requirement already satisfied: netaddr in /home/khanh/.local/lib/python3.8/site-packages (from ryu
==4.34) (1.3.0)
Requirement already satisfied: oslo.config>=2.5.0 in /home/khanh/.local/lib/python3.8/site-package
s (from ryu==4.34) (9.6.0)
Requirement already satisfied: ovs>=2.6.0 in /home/khanh/.local/lib/python3.8/site-packages (from
ryu==4.34) (3.6.0)
Requirement already satisfied: packaging==20.9 in /home/khanh/.local/lib/python3.8/site-packages (
from ryu==4.34) (20.9)
Requirement already satisfied: routes in /home/khanh/.local/lib/python3.8/site-packages (from ryu=
=4.34) (2.5.1)
Requirement already satisfied: six>=1.4.0 in /usr/lib/python3/dist-packages (from ryu==4.34) (1.14
.0)
Requirement already satisfied: tinyrpc==1.0.4 in /home/khanh/.local/lib/python3.8/site-packages (f
rom ryu==4.34) (1.0.4)
Requirement already satisfied: webob>=1.2 in /home/khanh/.local/lib/python3.8/site-packages (from
ryu==4.34) (1.8.9)
Requirement already satisfied: greenlet>=0.3 in /home/khanh/.local/lib/python3.8/site-packages (fr
om eventlet==0.31.1->ryu==4.34) (3.1.1)
Requirement already satisfied: dnspython<2.0.0,>=1.15.0 in /home/khanh/.local/lib/python3.8/site-p
ackages (from eventlet==0.31.1->ryu==4.34) (1.16.0)
Requirement already satisfied: requests>=2.18.0 in /usr/lib/python3/dist-packages (from oslo.confi
g>=2.5.0->ryu==4.34) (2.22.0)
Building wheels for collected packages: ryu
  Building wheel for ryu (setup.py) ... done
  Created wheel for ryu: filename=ryu-4.34-py3-none-any.whl size=2203583 sha256=a8bacb58f3b66cb92d
324bd410f30166ba1567e5bcafd549241f489c8636a649
  Stored in directory: /tmp/pip-ephem-wheel-cache-5qut6bja/wheels/f1/0a/b5/392102aa9e3e938501748f2
296ebb89229cd1f14070fc6a81c
Successfully built ryu
Installing collected packages: ryu
  Attempting uninstall: ryu
    Found existing installation: ryu 4.34
    Uninstalling ryu-4.34:
      Successfully uninstalled ryu-4.34
Successfully installed ryu-4.34
khanh@ubuntu:~/ryu$
```

2. Tạo mạng OpenFlow bằng Mininet, kết nối với Ryu Controller

- Viết source code định nghĩa topology. Lưu file dưới tên topo.py

```
khanh@ubuntu:~/nt541/lab2$ cat topo.py
from mininet.topo import Topo
class MyTopo(Topo):
    def build(self):
        # Add hosts and switches
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')

        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')

        # Add links
        self.addLink(h1, s1)
        self.addLink(h2, s2)
        self.addLink(h3, s3)
        self.addLink(h4, s4)

        self.addLink(s1, s2)
        self.addLink(s1, s3)
        self.addLink(s3, s4)
topos = {'mytopo': (lambda: MyTopo())}
khanh@ubuntu:~/nt541/lab2$
```

- Mở một terminal mới để chạy Ryu controller

```
khanhle@ubuntu:~$ ryu-manager ryu.app.simple_switch_13
loading app ryu.app.simple_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch_13 of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
```

- Ở terminal khác, tiến hành tạo mạng OpenFlow với Mininet, sử dụng lệnh:

"sudo mn --custom topo.py --topo mytopo --controller remote"

```
khanh@ubuntu:~/nt541/lab2$ sudo mn --custom topo.py --topo mytopo --controller remote
*** Creating network
*** Adding controller
Connecting to remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s1, s2) (s1, s3) (s3, s4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet>
```

- Kiểm tra lại terminal Ryu controller, thấy rằng đã kết nối thành công

```
packet in 4 06:c4:a0:a4:a7:97 33:33:00:00:00:02 2
packet in 2 06:c4:a0:a4:a7:97 33:33:00:00:00:02 2
packet in 1 6e:d0:8e:48:91:4d 33:33:00:00:00:02 3
packet in 2 6e:d0:8e:48:91:4d 33:33:00:00:00:02 2
packet in 1 de:c8:34:9b:06:6c 33:33:00:00:00:02 2
packet in 1 4e:d7:eb:99:09:d2 33:33:00:00:00:02 1
packet in 2 7a:98:79:5b:0e:07 33:33:00:00:00:02 2
packet in 3 de:c8:34:9b:06:6c 33:33:00:00:00:02 2
packet in 3 4e:d7:eb:99:09:d2 33:33:00:00:00:02 2
packet in 2 4e:d7:eb:99:09:d2 33:33:00:00:00:02 2
packet in 4 de:c8:34:9b:06:6c 33:33:00:00:00:02 2
packet in 4 4e:d7:eb:99:09:d2 33:33:00:00:00:02 2
packet in 4 4a:59:6e:4f:db:a6 33:33:00:00:00:02 2
packet in 3 56:3e:a4:70:67:9f 33:33:00:00:00:02 3
packet in 1 56:3e:a4:70:67:9f 33:33:00:00:00:02 3
packet in 2 56:3e:a4:70:67:9f 33:33:00:00:00:02 2
packet in 4 4a:59:6e:4f:db:a6 33:33:00:00:00:fb 2
packet in 3 56:3e:a4:70:67:9f 33:33:00:00:00:fb 3
packet in 1 56:3e:a4:70:67:9f 33:33:00:00:00:fb 3
packet in 2 56:3e:a4:70:67:9f 33:33:00:00:00:fb 2
packet in 1 de:c8:34:9b:06:6c 33:33:00:00:00:fb 2
packet in 2 7a:98:79:5b:0e:07 33:33:00:00:00:fb 2
packet in 3 de:c8:34:9b:06:6c 33:33:00:00:00:fb 2
packet in 4 de:c8:34:9b:06:6c 33:33:00:00:00:fb 2
packet in 3 7e:99:b1:48:0b:6e 33:33:00:00:00:fb 2
packet in 4 7e:99:b1:48:0b:6e 33:33:00:00:00:fb 2
packet in 1 6e:d0:8e:48:91:4d 33:33:00:00:00:fb 3
packet in 2 6e:d0:8e:48:91:4d 33:33:00:00:00:fb 2
```

- Kiểm tra các node hiện có

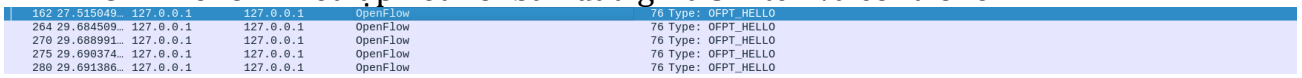
```
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 s1 s2 s3 s4
mininet>
```

- Test ping h1 với h2 và h4

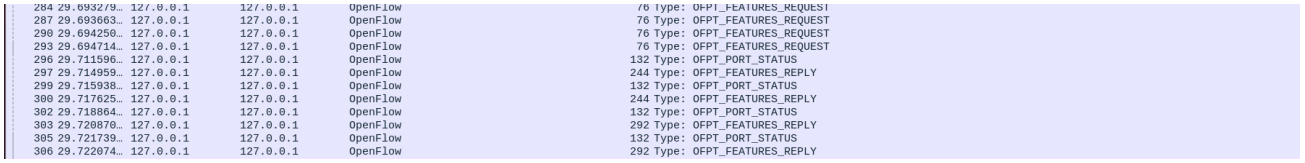
```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=10.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.357 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.068 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.095 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4066ms
rtt min/avg/max/mdev = 0.068/2.278/10.797/4.260 ms
mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=9.61 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.373 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.080 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.077 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.082 ms
^C
--- 10.0.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4064ms
rtt min/avg/max/mdev = 0.077/2.044/9.608/3.783 ms
mininet>
```

3. Dùng Wireshark tiến hành bắt các gói tin OpenFlow (v1.3)

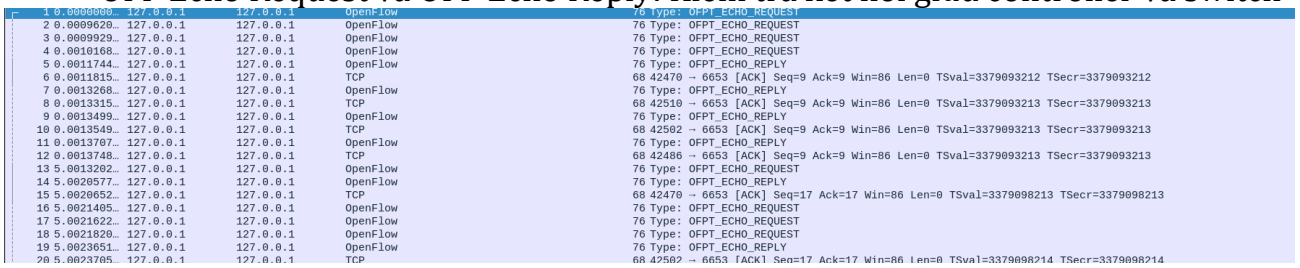
- OFP Hello: Thiết lập kết nối ban đầu giữa switch và controller



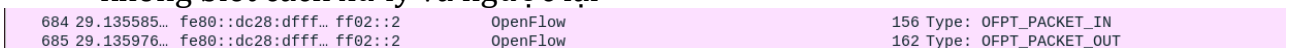
- OFP Features Request và OFP Features Reply: Cho phép controller lấy thông tin về switch



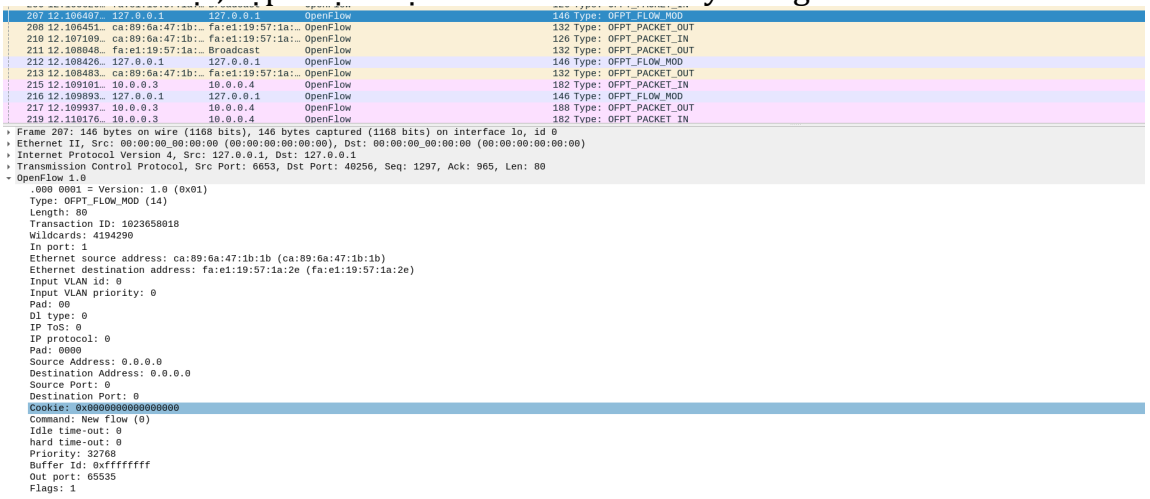
- OFP Echo Request và OFP Echo Reply: Kiểm tra kết nối giữa controller và switch



- OFP Packet In và OFP Packet Out: Gửi dữ liệu từ switch lên controller khi switch không biết cách xử lý và ngược lại

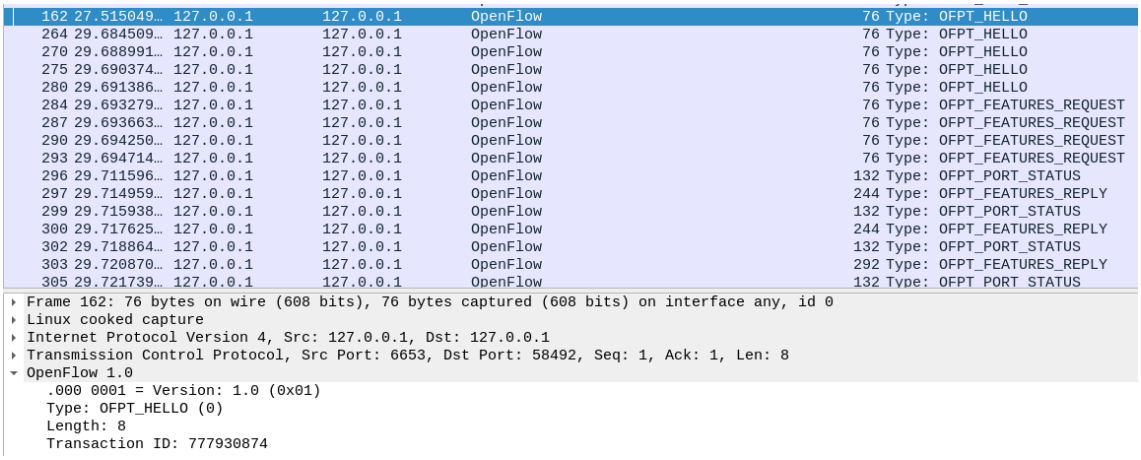


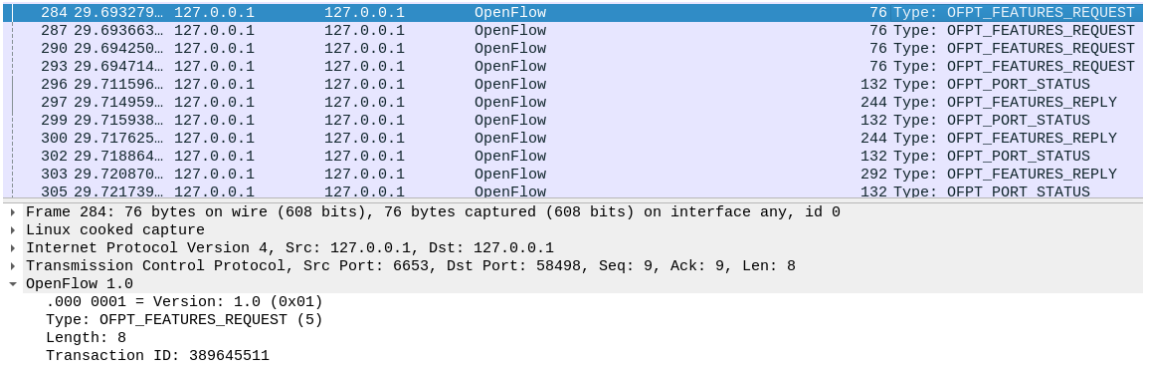
- OFP Flow Mod: Là thông điệp điều khiển được Controller gửi xuống Switch nhằm cài đặt, cập nhật hoặc xóa các Flow Entry trong Flow Table của switch

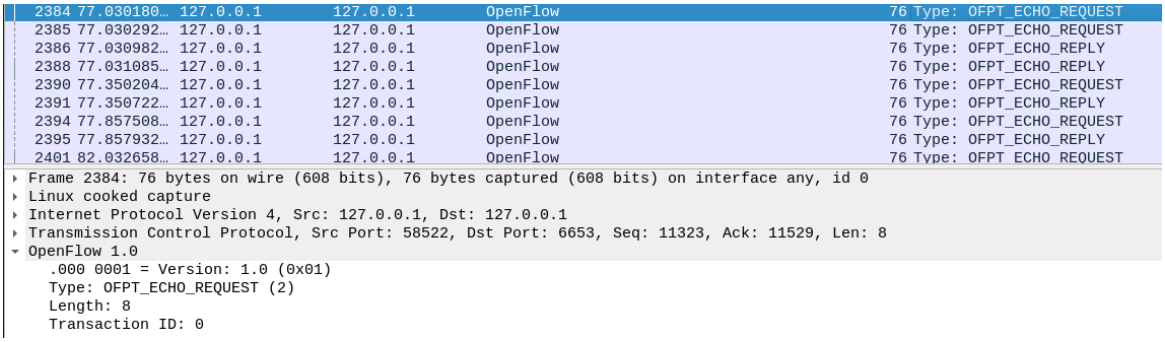


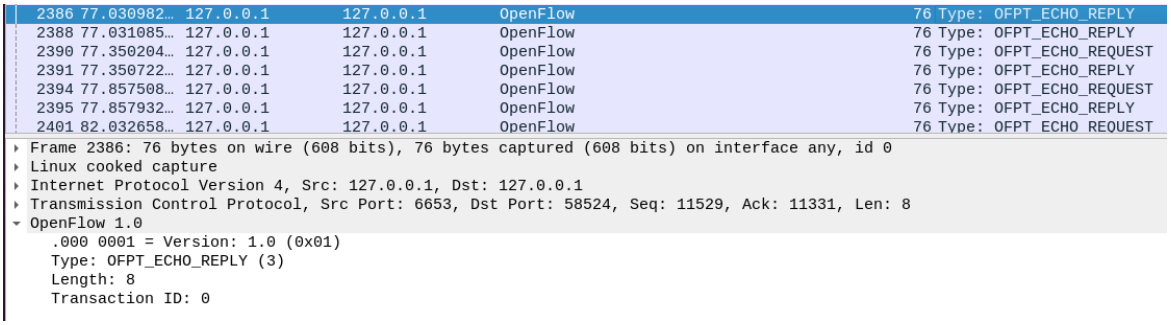
4. Liệt kê, mô tả các trường thông tin chính trong mỗi loại thông điệp ở bước 3

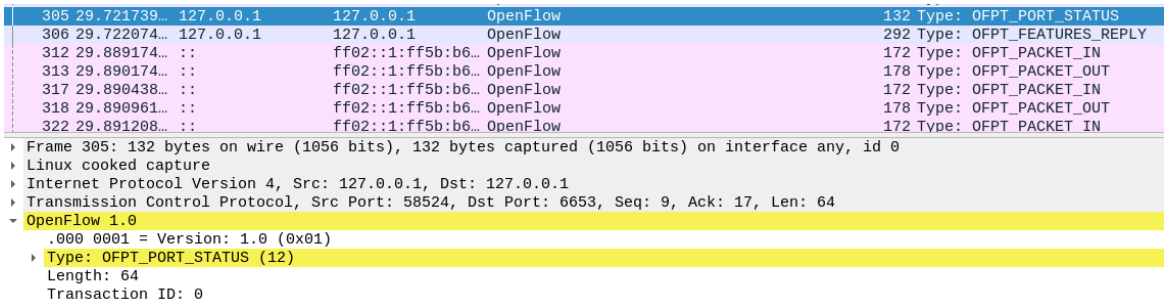
- OFP Hello, OFP Features Request, OFP Echo Request, OFP Echo Reply:
  - Version: Phiên bản của giao thức OpenFlow (1.0)
  - Type: Loại thông điệp
  - Length: Độ dài của thông điệp (8), riêng gói tin OFPT_PORT_STATUS có độ dài 64.
  - Transaction ID: Mã để theo dõi yêu cầu và phản hồi


162 27.515049… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_HELLO
264 29.684509… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_HELLO
270 29.688991… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_HELLO
275 29.690374… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_HELLO
280 29.691386… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_HELLO
284 29.693279… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
287 29.693663… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
290 29.694250… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
293 29.694714… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
296 29.711596… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
297 29.714959… 127.0.0.1 127.0.0.1 OpenFlow 244 Type: OFPT_FEATURES_REPLY
299 29.715938… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
300 29.717625… 127.0.0.1 127.0.0.1 OpenFlow 244 Type: OFPT_FEATURES_REPLY
302 29.718864… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
303 29.720870… 127.0.0.1 127.0.0.1 OpenFlow 292 Type: OFPT_FEATURES_REPLY
305 29.721739… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
Frame 162: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 6653, Dst Port: 58492, Seq: 1, Ack: 1, Len: 8
OpenFlow 1.0
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_HELLO (0)
Length: 8
Transaction ID: 777930874

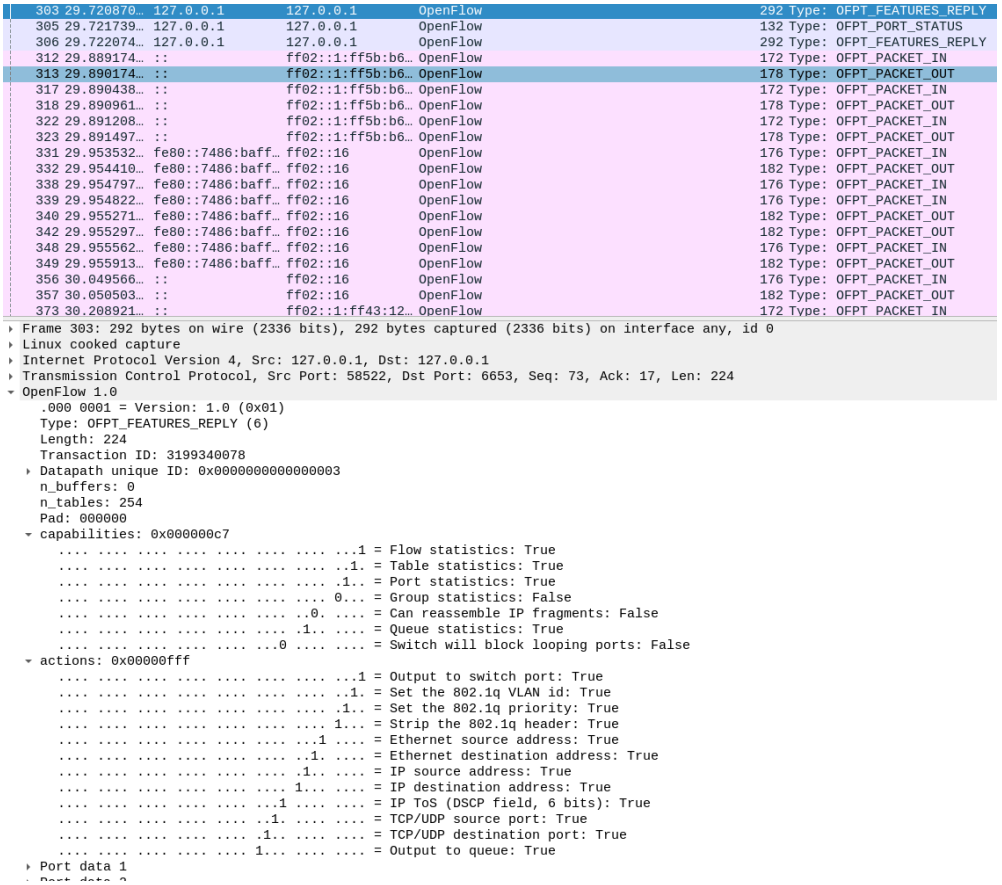

284 29.693279… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
287 29.693663… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
290 29.694250… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
293 29.694714… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_FEATURES_REQUEST
296 29.711596… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
297 29.714959… 127.0.0.1 127.0.0.1 OpenFlow 244 Type: OFPT_FEATURES_REPLY
299 29.715938… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
300 29.717625… 127.0.0.1 127.0.0.1 OpenFlow 244 Type: OFPT_FEATURES_REPLY
302 29.718864… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
303 29.720870… 127.0.0.1 127.0.0.1 OpenFlow 292 Type: OFPT_FEATURES_REPLY
305 29.721739… 127.0.0.1 127.0.0.1 OpenFlow 132 Type: OFPT_PORT_STATUS
Frame 284: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 6653, Dst Port: 58498, Seq: 9, Ack: 9, Len: 8
OpenFlow 1.0
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_FEATURES_REQUEST (5)
Length: 8
Transaction ID: 389645511


2384 77.030180… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
2385 77.030292… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
2386 77.030982… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2388 77.031085… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2390 77.350204… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
2391 77.350722… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2394 77.857508… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
2395 77.857932… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2401 82.032658… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
Frame 2384: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 58522, Dst Port: 6653, Seq: 11323, Ack: 11529, Len: 8
OpenFlow 1.0
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_ECHO_REQUEST (2)
Length: 8
Transaction ID: 0

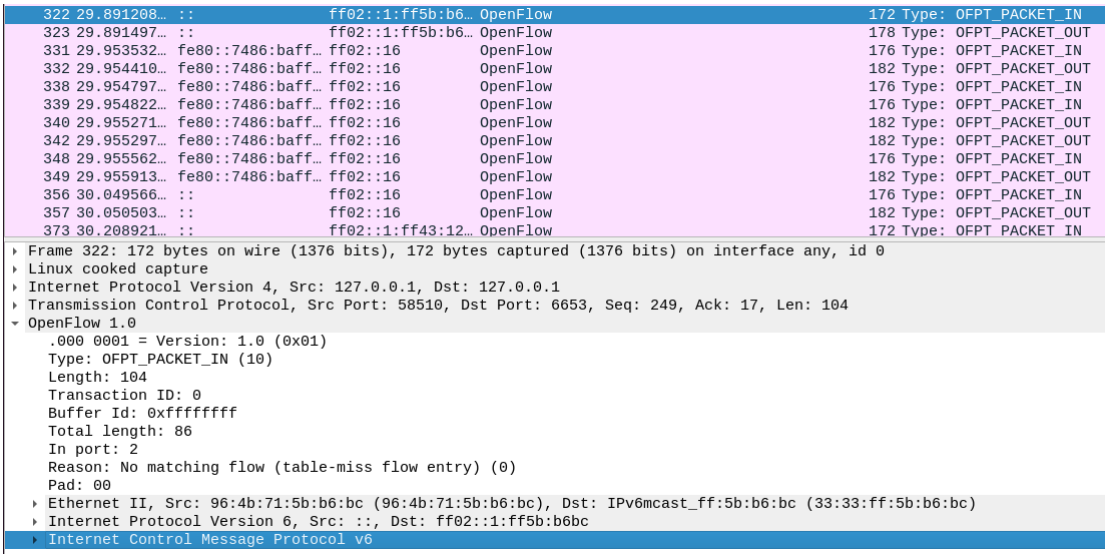

2386 77.030982… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2388 77.031085… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2390 77.350204… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
2391 77.350722… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2394 77.857508… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
2395 77.857932… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REPLY
2401 82.032658… 127.0.0.1 127.0.0.1 OpenFlow 76 Type: OFPT_ECHO_REQUEST
Frame 2386: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 6653, Dst Port: 58524, Seq: 11529, Ack: 11331, Len: 8
OpenFlow 1.0
.000 0001 = Version: 1.0 (0x01)
Type: OFPT_ECHO_REPLY (3)
Length: 8
Transaction ID: 0



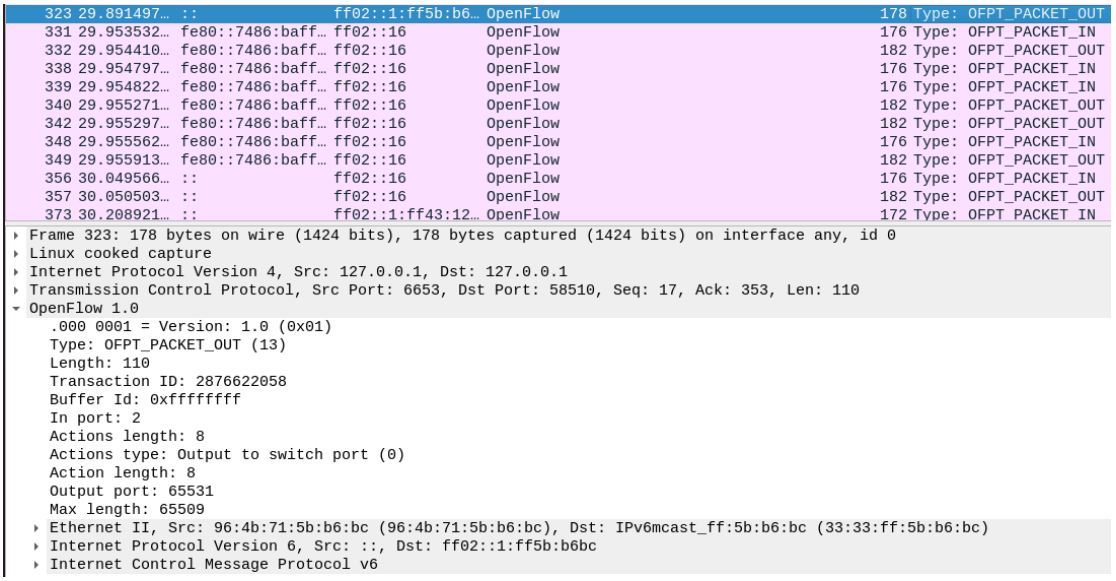
- OFP Features Reply: Có các trường thông tin về switch
  - N_tables: Số flow table mà switch hỗ trợ
  - Capabilities: Các khả năng mà switch hỗ trợ
  - Actions: Các hành động mà switch hỗ trợ
  - Port: Thông tin các port của switch



- OFP Packet In:
  - In port: Cổng mà switch nhận gói tin.
  - Reason: Lý do gói tin không được xử lý bởi flow rule trong flow table của switch.

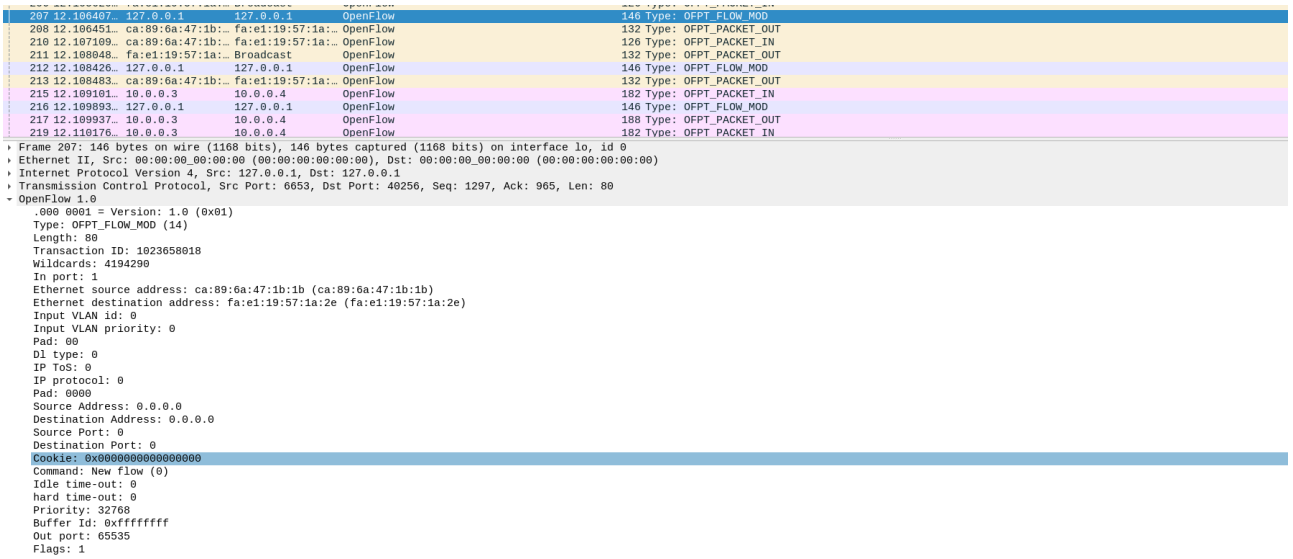


- OFP Packet Out:
  - Action type: Hành động thực hiện với gói tin
  - Output port: Cổng mà switch sẽ gửi gói tin đến
  - Max length: Độ dài tối đa



- OFP FlowMod:
  - Thông tin chung của gói tin: version, type, length, transaction ID
  - Các trường match fields bao gồm: wildcards, in_port, MAC nguồn, MAC đích, IP nguồn, IP đích, port nguồn, port đích …

- Các trường thiết lập flow bao gồm: cookie, command, idle time-out, hard time-out, priority…

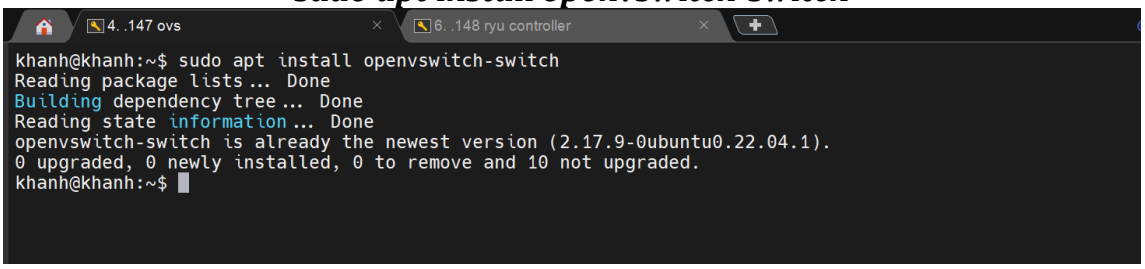


II. Yêu cầu 2

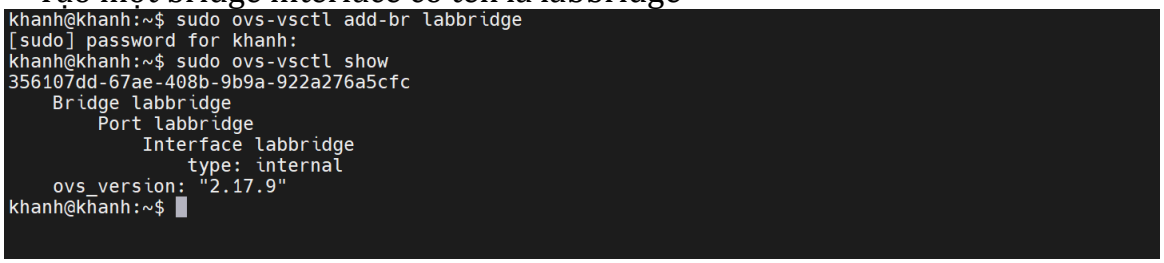
1. Tiến hành cài đặt một loại OpenFlow virtual Switch (vd: Open vSwitch)

- Ở trên máy sẽ làm ovs, cài đặt Open vSwitch với command:

"sudo apt install openvswitch-switch"



- Tạo một bridge interface có tên là labbridge



- Tạo các virtual port vport1 và vport2

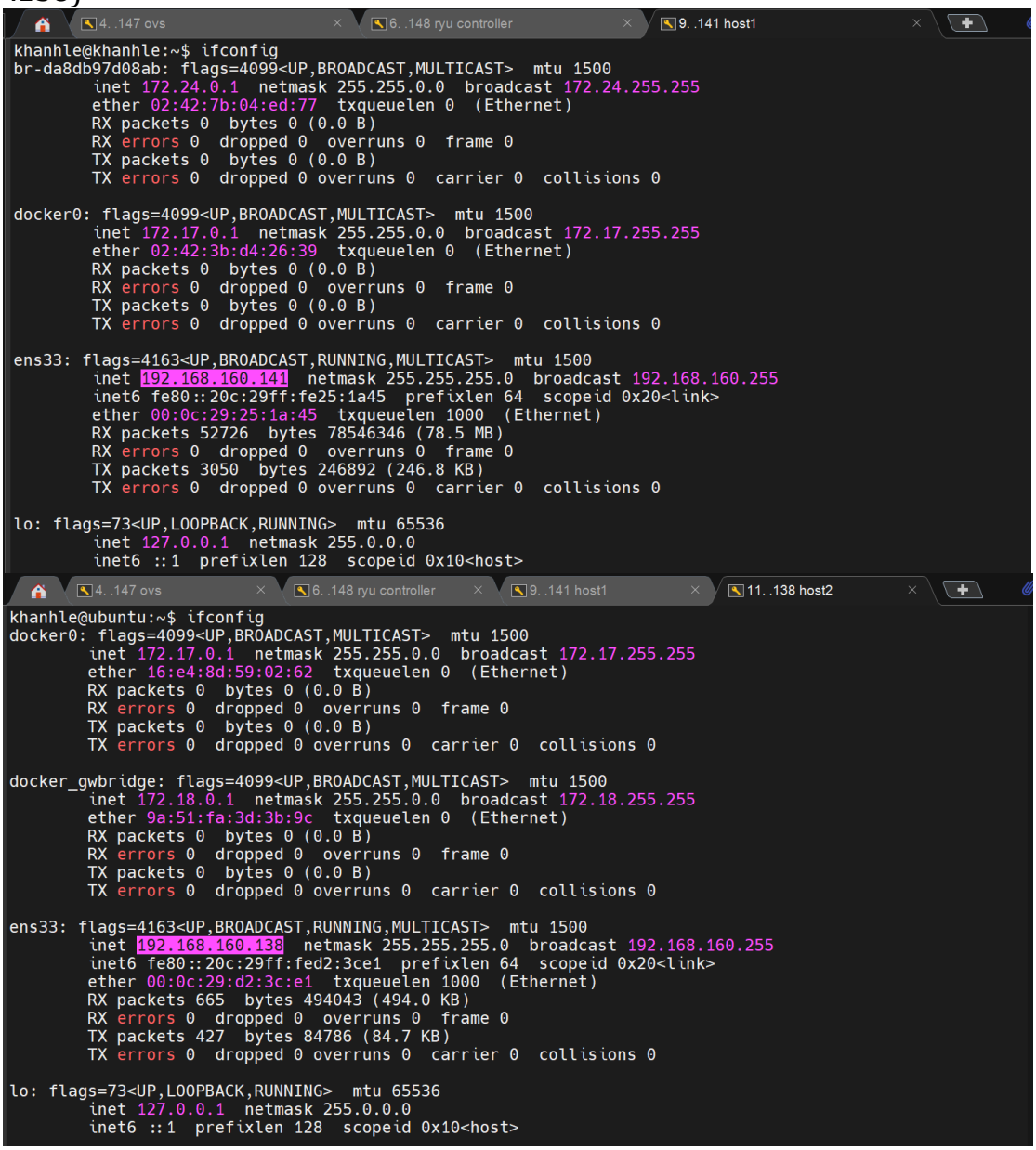
```
khanh@khanh:~$ sudo ip tuntap add mode tap vport1
khanh@khanh:~$ sudo ip tuntap add mode tap vport2
khanh@khanh:~$ sudo ifconfig vport1 up
khanh@khanh:~$ sudo ifconfig vport2 up
khanh@khanh:~$ ifconfig -a
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.160.147  netmask 255.255.255.0  broadcast 192.168.160.255
        inet6 fe80::56a7:a304:af49:3b3f  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:61:b2:24  txqueuelen 1000  (Ethernet)
        RX packets 218418  bytes 325350283 (325.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14649  bytes 1033587 (1.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

labbridge: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 62:99:59:ec:3f:43  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 272  bytes 28258 (28.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 272  bytes 28258 (28.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 46:56:5c:b4:26:67  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vport1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether aa:dd:c4:08:ef:c5  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vport2: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 4a:fe:99:e6:de:ae  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Tiếp theo, thêm 2 port trên vào interface bridge labbridge

```
khanh@khanh:~$ sudo ovs-vsctl add-port labbridge vport1
khanh@khanh:~$ sudo ovs-vsctl add-port labbridge vport2
khanh@khanh:~$ sudo ovs-vsctl show
356107dd-67ae-408b-9b9a-922a276a5cfc
    Bridge labbridge
        Port vport1
            Interface vport1
        Port labbridge
            Interface labbridge
                type: internal
        Port vport2
            Interface vport2
    ovs_version: "2.17.9"
khanh@khanh:~$
```

2. Kết nối switch đã cài đặt với một Ryu controller và một số Host (>= 2 host)

- Tiến hành tạo 2 máy ảo Ubuntu làm Host (một máy có IP .141, và một máy có IP .138)



- Trên máy ảo cài đặt Open vSwitch, kết nối Ryu controller tới với switch (192.168.160.148 là địa chỉ IP của máy Ryu controller)

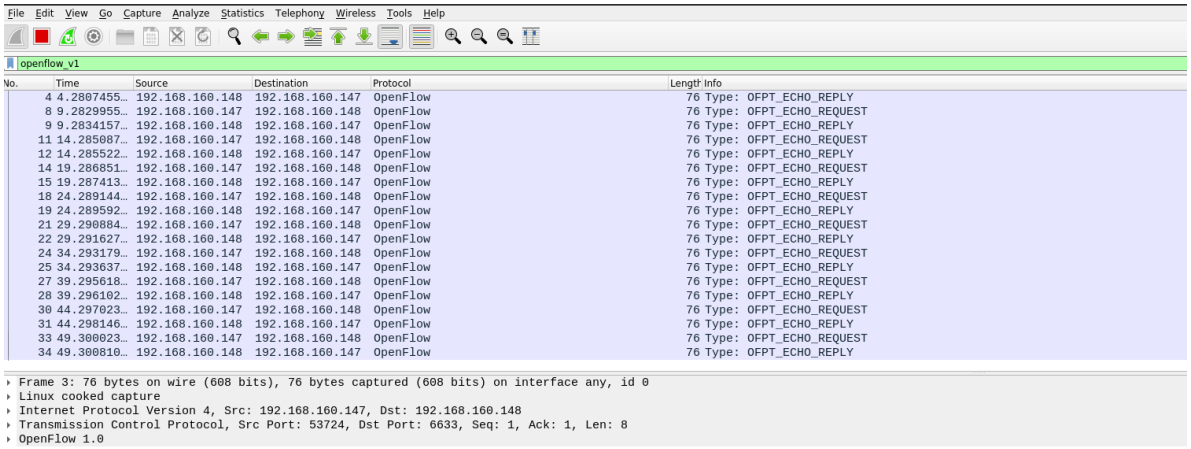
```
khanh@khanh:~$ sudo ovs-vsctl set-controller labbridge tcp:192.168.160.148:6633
khanh@khanh:~$ sudo ovs-vsctl show
356107dd-67ae-408b-9b9a-922a276a5cfc
    Bridge labbridge
        Controller "tcp:192.168.160.148:6633"
            is_connected: true
        Port vport1
            Interface vport1
        Port labbridge
            Interface labbridge
                type: internal
        Port vport2
            Interface vport2
    ovs_version: "2.17.9"
khanh@khanh:~$
```

- Kiểm tra kết nối ở máy Ryu controller, thấy rằng đã thành công

```
khanh@ubuntu:~$ ryu run ryu.app.simple_switch
loading app ryu.app.simple_switch
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch of SimpleSwitch
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 108410778173251 00:23:20:8d:87:74 ff:ff:ff:ff:ff:ff 65534
```

3. Cài Wireshark và tiến hành các bước bắt gói tin trong yêu cầu 1, 2

- Bắt gói tin bằng Wireshark, thấy rằng Switch và Controller đang giao tiếp với nhau.



4. Test performance của mạng đã tạo ra

- Cho máy host 1 với địa chỉ IP .141 làm iperf server

```
khanhle@khanhle:~$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.160.138, port 54216
[  5] local 192.168.160.141 port 5201 connected to 192.168.160.138 port 54218
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  97.5 MBytes   818 Mbits/sec
[  5]   1.00-2.00   sec   235 MBytes  1.97 Gbits/sec
[  5]   2.00-3.00   sec   273 MBytes  2.29 Gbits/sec
[  5]   3.00-4.00   sec   287 MBytes  2.41 Gbits/sec
[  5]   4.00-5.00   sec   285 MBytes  2.39 Gbits/sec
[  5]   5.00-6.00   sec   254 MBytes  2.13 Gbits/sec
[  5]   6.00-7.00   sec   281 MBytes  2.36 Gbits/sec
[  5]   7.00-8.00   sec   304 MBytes  2.55 Gbits/sec
[  5]   8.00-9.00   sec   290 MBytes  2.43 Gbits/sec
[  5]   9.00-10.00  sec   253 MBytes  2.12 Gbits/sec
[  5]  10.00-10.07  sec  20.5 MBytes  2.48 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.07  sec  2.52 GBytes  2.15 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

- Cho máy host 2 với địa chỉ .138 làm iperf client. Kiểm tra hiệu năng của mạng đã tạo ra

```
khanhle@ubuntu:~$ iperf3 -c 192.168.160.141
Connecting to host 192.168.160.141, port 5201
[  5] local 192.168.160.138 port 54218 connected to 192.168.160.141 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec   110 MBytes   923 Mbits/sec   81   1.37 MBytes
[  5]   1.00-2.00   sec   236 MBytes  1.98 Gbits/sec    0   1.51 MBytes
[  5]   2.00-3.00   sec   274 MBytes  2.30 Gbits/sec    0   1.63 MBytes
[  5]   3.00-4.00   sec   289 MBytes  2.42 Gbits/sec    0   1.73 MBytes
[  5]   4.00-5.00   sec   284 MBytes  2.38 Gbits/sec    0   1.84 MBytes
[  5]   5.00-6.00   sec   258 MBytes  2.16 Gbits/sec   11   1.43 MBytes
[  5]   6.00-7.00   sec   282 MBytes  2.37 Gbits/sec    0   1.57 MBytes
[  5]   7.00-8.00   sec   306 MBytes  2.57 Gbits/sec    0   1.70 MBytes
[  5]   8.00-9.00   sec   288 MBytes  2.41 Gbits/sec    0   1.82 MBytes
[  5]   9.00-10.00  sec   256 MBytes  2.15 Gbits/sec   19   1.42 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  2.52 GBytes  2.17 Gbits/sec  111             sender
[  5]   0.00-10.07  sec  2.52 GBytes  2.15 Gbits/sec                  receiver

iperf Done.
khanhle@ubuntu:~$
```

YÊU CẦU CHUNG

1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả *(nếu có)*; giải thích cho quan sát *(nếu có)*.
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

2) Báo cáo

- File .PDF hoặc .docx. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: LabX_MSSV1_MSSV2. (trong đó X là Thứ tự buổi Thực hành).

  Ví dụ: Lab01_21520001_21520002
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT