BÁO CÁO THỰC HÀNH

Môn học: Công nghệ mạng khả lập trình

Buổi báo cáo: Lab 04

Tên chủ đề: Lập trình hiện thực và test thử nghiệm một Network Monitor trên mạng SDN/OpenFlow

GVHD: Phan Xuân Thiện

Ngày thực hiện: 20/11/2025

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	
Link source code	https://github.com/KhanhLe04/nt541-lab/tree/master/LAB%204
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

1. Tạo mạng SDN/OpenFlow với Topology tuỳ ý.

- Khởi tạo Ryu Controller

```
khanh@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
```

- Chạy file **q1_topology.py** để khởi tạo topology. Lúc này, thấy đã khởi tạo và kết nối thành công với controller

```
khanh@ubuntu:~/nt541-lab/LAB 4$ sudo python3 q1_topology.py
*** Creating network
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h1, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h2, s1) (10.00Mbit 1ms dela
y) (10.00Mbit 1ms delay) (h3, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h4, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms d
elay) (h5, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h6, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h7, s2) (10.
00Mbit 1ms delay) (10.00Mbit 1ms delay) (h8, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h9, s3) (10.00Mbit 1ms delay) (
10.00Mbit 1ms delay) (h10, s3) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h11, s3) (10.00Mbit 1ms delay) (10.00Mbit 1ms del
ay) (h12, s3) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h13, s4) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h14, s4) (10
.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h15, s4) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h16, s4) (20.00Mbit 2ms delay
) (20.00Mbit 2ms delay) (s1, s2) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (s2, s3) (20.00Mbit 2ms delay) (20.00Mbit 2ms de
lay) (s3, s4) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (s4, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Starting network\n*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...(10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay)
(20.00Mbit 2ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms
delay) (20.00Mbit 2ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mb
it 2ms delay) (20.00Mbit 2ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (
20.00Mbit 2ms delay) (20.00Mbit 2ms delay)
*** Network started; enter CLI to test (pingall, iperf, ...)\n*** Starting CLI:
mininet>
```

```
packet in 0000000000000002 00:00:00:00:00:0b 33:33:00:00:00:02 6
packet in 0000000000000002 00:00:00:00:00:0a 33:33:00:00:00:02 6
packet in 0000000000000002 00:00:00:00:00:0b 33:33:00:00:00:02 6
packet in 0000000000000002 00:00:00:00:00:0a 33:33:00:00:00:02 6
packet in 0000000000000002 ba:f9:d5:94:df:1c 33:33:00:00:00:fb 6
packet in 0000000000000002 ba:00:27:a8:00:aa 33:33:00:00:00:02 6
packet in 0000000000000002 00:00:00:00:00:09 33:33:00:00:00:16 6
packet in 0000000000000002 00:00:00:00:00:05 33:33:00:00:00:16 6
packet in 0000000000000002 66:0d:c2:d6:23:cb 33:33:00:00:00:fb 6
packet in 0000000000000004 9a:8a:fa:57:25:76 33:33:00:00:00:02 5
packet in 0000000000000004 5a:f1:ff:f7:0f:ab 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:01 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:0d 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:02 33:33:00:00:00:02 5
packet in 0000000000000004 5a:f1:ff:f7:0f:ab 33:33:00:00:00:fb 5
packet in 0000000000000004 00:00:00:00:00:03 33:33:00:00:00:02 5
packet in 0000000000000004 9a:8a:fa:57:25:76 33:33:00:00:00:fb 5
packet in 0000000000000004 0a:48:a3:7a:e7:88 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:0d 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:0f 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:03 33:33:00:00:00:02 6
packet in 0000000000000001 00:00:00:00:00:08 33:33:00:00:00:02 6
packet in 0000000000000001 00:00:00:00:00:01 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:01 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:10 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:04 33:33:00:00:00:02 5
packet in 0000000000000001 ba:00:27:a8:00:aa 33:33:00:00:00:fb 5
packet in 0000000000000001 c2:e4:5c:2c:64:21 33:33:00:00:00:fb 5
packet in 0000000000000001 00:00:00:00:00:07 33:33:00:00:00:02 5
packet in 0000000000000001 ba:00:27:a8:00:aa 33:33:00:00:00:fb 5
packet in 0000000000000001 00:00:00:00:00:0f 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:0d 33:33:00:00:00:02 5
packet in 0000000000000001 66:0d:c2:d6:23:cb 33:33:00:00:00:fb 5
packet in 0000000000000001 00:00:00:00:00:05 33:33:00:00:00:02 5
```

- Kiểm tra kết nối với pingall

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h2 -> h1 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h3 -> h1 h2 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h4 -> h1 h2 h3 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h5 -> h1 h2 h3 h4 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h6 -> h1 h2 h3 h4 h5 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
h7 -> h1 h2 h3 h4 h5 h6 h8 h9 h10 h11 h12 h13 h14 h15 h16
h8 -> h1 h2 h3 h4 h5 h6 h7 h9 h10 h11 h12 h13 h14 h15 h16
h9 -> h1 h2 h3 h4 h5 h6 h7 h8 h10 h11 h12 h13 h14 h15 h16
h10 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h11 h12 h13 h14 h15 h16
h11 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h12 h13 h14 h15 h16
h12 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h13 h14 h15 h16
h13 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h14 h15 h16
h14 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h15 h16
h15 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h16
h16 -> h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15
*** Results: 0% dropped (240/240 received)
mininet>
```

- Kiểm tra hiệu năng với iperf

```
mininet> h1 iperf -s &
mininet> h12 iperf -c h1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.12 port 47242 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.4 sec  14.1 MBytes  11.4 Mbits/sec
mininet>
```

2. Viết chương trình để hiện thực một network monitor trên mạng SDN/OpenFlow

- Chạy file q2_monitor.py để khởi tạo Ryu Controller với Monitor

```
khanh@ubuntu:~/nt541-lab/LAB 4$ ryu-manager q2_monitor.py
loading app q2_monitor.py
loading app ryu.controller.ofp_handler
instantiating app q2_monitor.py of SimpleMonitor13
instantiating app ryu.controller.ofp_handler of OFPHandler
```

- Giải thích source code q2_monitor.py
  - Switch_features_handler: khi nhận SwitchFeatures, cài flow miss (priority 0) match rỗng, action gửi về controller (OFPP_CONTROLLER) để nhận PacketIn.
  - add_flow: dựng và gửi FlowMod với priority/match/actions, tận dụng buffer_id nếu có.

```python
@set_ev_cls(ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER)
def switch_features_handler(self, ev):
    datapath = ev.msg.datapath
    ofproto = datapath.ofproto
    parser = datapath.ofproto_parser
    match = parser.OFPMatch()
    actions = [parser.OFPActionOutput(ofproto.OFPP_CONTROLLER,
                                      ofproto.OFPCML_NO_BUFFER)]
    self.add_flow(datapath, 0, match, actions)

def add_flow(self, datapath, priority, match, actions, buffer_id=None):
    ofproto = datapath.ofproto
    parser = datapath.ofproto_parser
    inst = [parser.OFPInstructionActions(ofproto.OFPIT_APPLY_ACTIONS,
                                         actions)]
    if buffer_id:
        mod = parser.OFPFlowMod(datapath=datapath, buffer_id=buffer_id,
                                priority=priority, match=match,
                                instructions=inst)
    else:
        mod = parser.OFPFlowMod(datapath=datapath, priority=priority,
                                match=match, instructions=inst)
    datapath.send_msg(mod)
```

- Packet_in_handler: Học ánh xạ src -> in_port vào mac_to_port[dpid]. Nếu biết đích (dst), xuất đúng cổng, nếu chưa biết thì flood. Khi biết đích, cài thêm flow match với in_port, eth_dst, eth_src để giảm PacketIn. Ngoài ra, gửi PacketOut (có/không kèm data tùy thuộc vào buffer_id).

```python
@set_ev_cls(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER)
def _packet_in_handler(self, ev):
    if ev.msg.msg_len < ev.msg.total_len:
        self.logger.debug("packet truncated: only %s of %s bytes",
                          ev.msg.msg_len, ev.msg.total_len)
    msg = ev.msg
    datapath = msg.datapath
    ofproto = datapath.ofproto
    parser = datapath.ofproto_parser
    in_port = msg.match['in_port']
    pkt = packet.Packet(msg.data)
    eth = pkt.get_protocols(ethernet.ethernet)[0]
    if eth.ethertype == ether_types.ETH_TYPE_LLDP:
        return
    dst = eth.dst
    src = eth.src
    dpid = datapath.id
    self.mac_to_port.setdefault(dpid, {})
    self.mac_to_port[dpid][src] = in_port
    if dst in self.mac_to_port[dpid]:
        out_port = self.mac_to_port[dpid][dst]
    else:
        out_port = ofproto.OFPP_FLOOD
    actions = [parser.OFPActionOutput(out_port)]
    if out_port != ofproto.OFPP_FLOOD:
        match = parser.OFPMatch(in_port=in_port, eth_dst=dst, eth_src=src)
        if msg.buffer_id != ofproto.OFP_NO_BUFFER:
            self.add_flow(datapath, 1, match, actions, msg.buffer_id)
            return
        else:
            self.add_flow(datapath, 1, match, actions)
    data = None
    if msg.buffer_id == ofproto.OFP_NO_BUFFER:
        data = msg.data
    out = parser.OFPPacketOut(datapath=datapath, buffer_id=msg.buffer_id,
                              in_port=in_port, actions=actions, data=data)
    datapath.send_msg(out)
```

- Switch_state_handler: Thêm hoặc bớt datapath vào self.datapaths khi vào MAIN_DISPATCHER hoặc DEAD_DISPATCHER, giúp log connect/disconnect
- monitor_loop: Vòng lặp, gọi request_stats(dp) cho từng datapath rồi sleep statistics_interval
- request_stats: Gửi OFPFlowStatsRequest và OFPPortStatsRequest tới switch

```python
@set_ev_cls(ofp_event.EventOFPStateChange, [MAIN_DISPATCHER, DEAD_DISPATCHER])
def _switch_state_handler(self, ev):
    dp = ev.datapath
    if ev.state == MAIN_DISPATCHER:
        self.datapaths[dp.id] = dp
        self.logger.info("Switch connected: %s", dp.id)
    elif ev.state == DEAD_DISPATCHER:
        if dp.id in self.datapaths:
            del self.datapaths[dp.id]
            self.logger.info("Switch disconnected: %s", dp.id)

def _monitor_loop(self):
    while True:
        for dp in self.datapaths.values():
            self._request_stats(dp)
        hub.sleep(self.statistics_interval)

def _request_stats(self, datapath):
    parser = datapath.ofproto_parser
    ofp = datapath.ofproto
    datapath.send_msg(parser.OFPFlowStatsRequest(datapath))
    datapath.send_msg(parser.OFPPortStatsRequest(datapath, 0, ofp.OFPP_ANY))
```

- flow_stats_reply_handler và _port_stats_reply_handler: Giúp định dạng log thống kê.

```python
@set_ev_cls(ofp_event.EventOFPFlowStatsReply, MAIN_DISPATCHER)
def _flow_stats_reply_handler(self, ev):
    dpid = ev.msg.datapath.id
    flows = [s for s in ev.msg.body if s.priority != 0]
    lines = []
    lines.append("=" * 66)
    lines.append(f"Flow Statistics for Switch: {dpid:016x}")
    lines.append("-" * 66)
    lines.append(f"{'in-port':<10}{'eth-dst':<22}{'packets':>10}{'bytes':>12}{'duration (s)':>14}")
    lines.append("-" * 66)
    for stat in sorted(flows, key=lambda s: (s.match.get("in_port", 0), s.match.get("eth_dst", ""))):
        in_port = stat.match.get("in_port", "-")
        eth_dst = stat.match.get("eth_dst", "-")
        duration = stat.duration_sec + stat.duration_nsec / 1e9
        lines.append(
            f"{in_port!s:<10}{eth_dst:<22}{stat.packet_count:>10}{stat.byte_count:>12}{duration:>14.2f}"
        )
    lines.append("-" * 66)
    self.logger.info("\n".join(lines))

@set_ev_cls(ofp_event.EventOFPPortStatsReply, MAIN_DISPATCHER)
def _port_stats_reply_handler(self, ev):
    dpid = ev.msg.datapath.id
    lines = []
    lines.append("=" * 66)
    lines.append(f"Port Statistics for Switch: {dpid:016x}")
    lines.append("-" * 66)
    lines.append(f"{'Port':<8}|{'Rx-Pkts':>10}|{'Rx-Bytes':>12}|{'Tx-Pkts':>10}|{'Tx-Bytes':>12}|{'Errors':>8}")
    lines.append("-" * 66)
    for stat in sorted(ev.msg.body, key=lambda s: s.port_no):
        port_label = "LOCAL" if stat.port_no == ofproto_v1_3.OFPP_LOCAL else str(stat.port_no)
        errors = stat.rx_errors + stat.tx_errors
        lines.append(
            f"{port_label:<8}|{stat.rx_packets:>10}|{stat.rx_bytes:>12}|"
            f"{stat.tx_packets:>10}|{stat.tx_bytes:>12}|{errors:>8}"
        )
    lines.append("-" * 66)
    self.logger.info("\n".join(lines))
```

- Chạy topology Mininet với file q1_topology.py như ở câu 1.

```
khanh@ubuntu:~/nt541-lab/LAB 4$ sudo python3 q1_topology.py
*** Creating network
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h1, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h2, s1) (10.00Mbit 1ms delay) (10.00Mbit 1m
s delay) (h3, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h4, s1) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h5, s2) (10.00Mbit 1ms de
lay) (10.00Mbit 1ms delay) (h6, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h7, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h8, s2)
 (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h9, s3) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h10, s3) (10.00Mbit 1ms delay) (10.00Mbit
1ms delay) (h11, s3) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h12, s3) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h13, s4) (10.00Mbit 1
ms delay) (10.00Mbit 1ms delay) (h14, s4) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h15, s4) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (
h16, s4) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (s1, s2) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (s2, s3) (20.00Mbit 2ms delay) (20.
00Mbit 2ms delay) (s3, s4)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Starting network\n*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...(10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay) (10.00Mbit 1ms d
elay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (10.00Mbit 1ms delay) (10
.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1
ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay)
*** Network started; enter CLI to test (pingall, iperf, ...)\n*** Starting CLI:
mininet>
```

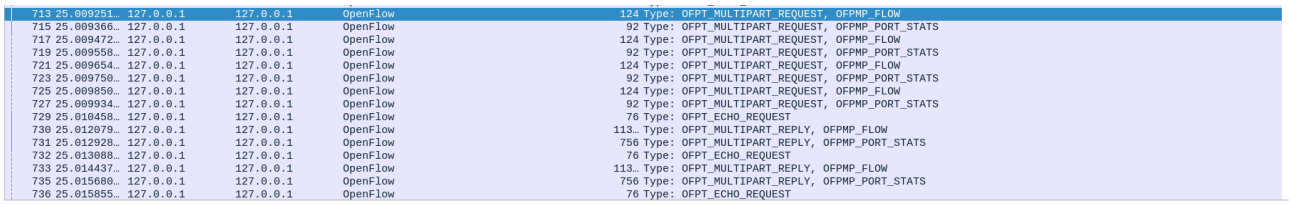
- Kiểm tra log của Ryu Controller, lúc này nó sẽ gửi các thông số flow mỗi 10 giây.

```
khanh@ubuntu:~/nt541-lab/LAB 4$ ryu-manager q2_monitor.py
loading app q2_monitor.py
loading app ryu.controller.ofp_handler
instantiating app q2_monitor.py of SimpleSwitchMonitor
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch connected: 1
Switch connected: 4
Switch connected: 3
Switch connected: 2
================================================================
Flow Statistics for Switch: 0000000000000001
----------------------------------------------------------------
in-port   eth-dst                 packets       bytes  duration (s)
----------------------------------------------------------------
----------------------------------------------------------------
================================================================
Port Statistics for Switch: 0000000000000001
----------------------------------------------------------------
Port     |   Rx-Pkts|    Rx-Bytes|   Tx-Pkts|    Tx-Bytes|  Errors
----------------------------------------------------------------
1        |        14|        1212|        44|        4923|       0
2        |        14|        1212|        43|        4853|       0
3        |        14|        1212|        42|        4763|       0
4        |        14|        1212|        43|        4853|       0
5        |        33|        3624|        16|        2102|       0
LOCAL    |         0|           0|         0|           0|       0
----------------------------------------------------------------
================================================================
Flow Statistics for Switch: 0000000000000004
----------------------------------------------------------------
in-port   eth-dst                 packets       bytes  duration (s)
----------------------------------------------------------------
----------------------------------------------------------------
================================================================
Port Statistics for Switch: 0000000000000004
```

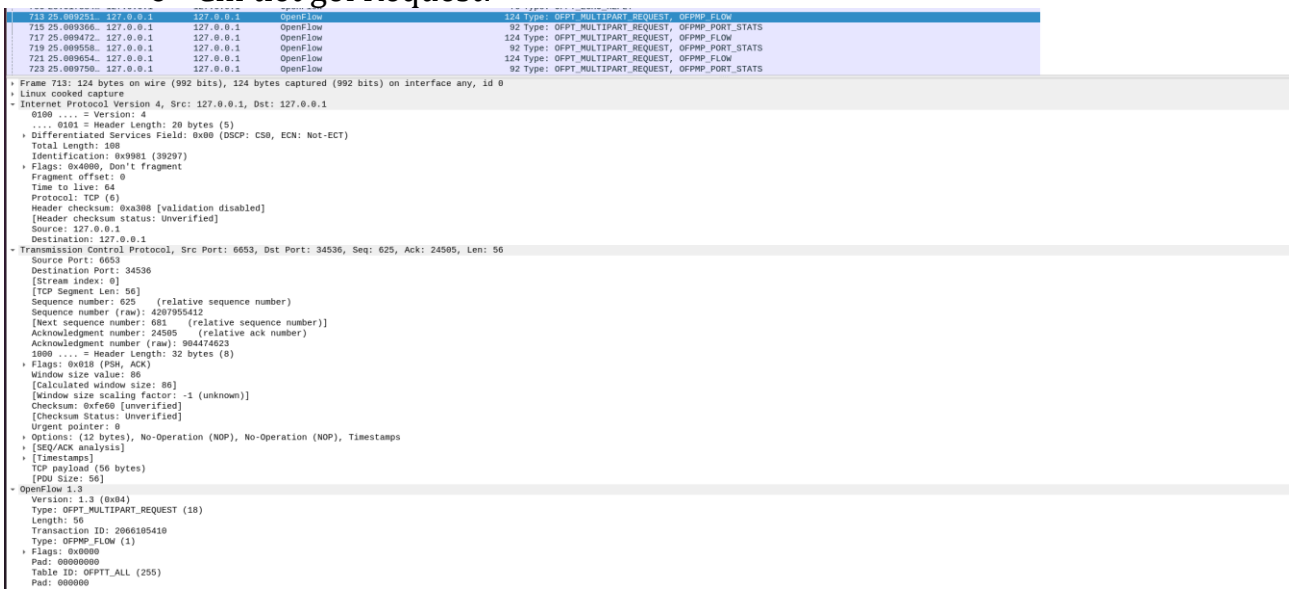
- Thực hiện pingall, thấy log trên Controller như sau. Lúc này các flow entry đã được thêm vào và controller hiển thị thông tin.

```
----------------------------------------------------------------
================================================================
Port Statistics for Switch: 0000000000000002
----------------------------------------------------------------
Port     |   Rx-Pkts|    Rx-Bytes|   Tx-Pkts|    Tx-Bytes|  Errors
----------------------------------------------------------------
1        |        78|        5692|       311|       22159|       0
2        |        78|        5692|       311|       22179|       0
3        |        78|        5692|       310|       22089|       0
4        |        78|        5712|       309|       21964|       0
5        |       241|       18091|       377|       27329|       0
6        |       373|       27514|       356|       25992|       0
LOCAL    |         0|           0|         0|           0|       0
----------------------------------------------------------------
================================================================
Flow Statistics for Switch: 0000000000000001
----------------------------------------------------------------
in-port   eth-dst                packets       bytes  duration (s)
----------------------------------------------------------------
1         00:00:00:00:00:02              2         140         71.90
1         00:00:00:00:00:03              2         140         71.86
1         00:00:00:00:00:04              2         140         71.83
1         00:00:00:00:00:05              2         140         71.79
1         00:00:00:00:00:06              2         140         71.74
1         00:00:00:00:00:07              2         140         71.69
1         00:00:00:00:00:08              2         140         71.64
1         00:00:00:00:00:09              2         140         71.58
1         00:00:00:00:00:0a              2         140         71.51
1         00:00:00:00:00:0b              2         140         71.44
1         00:00:00:00:00:0c              2         140         71.36
1         00:00:00:00:00:0d              2         140         71.28
1         00:00:00:00:00:0e              2         140         71.19
1         00:00:00:00:00:0f              2         140         71.10
```

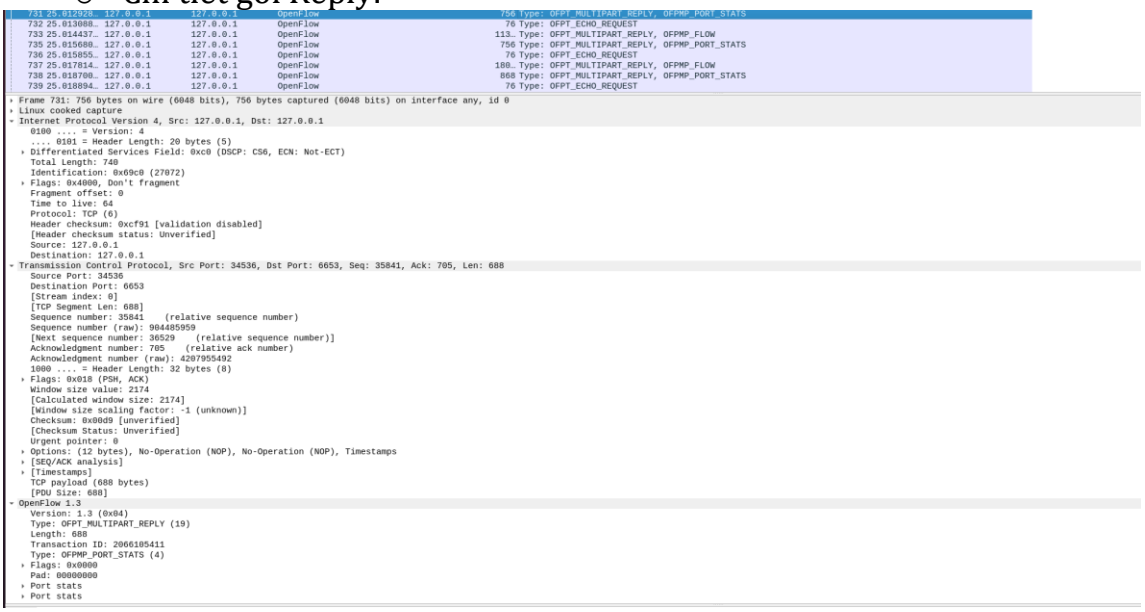
- Mở Wireshark và bắt các gói tin trao đổi giữa switch và controller (statistics request và reply):
    - Quan sát thấy rằng các gói OFPT_MULTIPART_REQUEST được conntroller gửi đến các swith và các switch cùng gửi các gói OFPT_MULTIPART_REPLY để phản hồi 2 request về flow và port.



- Chi tiết gói Request:



- Chi tiết gói Reply:



3. Tiến hành test chương trình với lưu lượng network traffic lớn.

- Dùng h12 làm iperf server. H4 iperf tới H12 với gói tin udp, bandwidth 10Mbit/s, thời gian là 30 giây.

```
mininet> h12 iperf -s -u -p 5001 &
mininet> h4 iperf -c h12 -u -b 10M -t 30 -p 5001
------------------------------------------------------------
Client connecting to 10.0.0.12, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1121.52 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.4 port 56137 connected with 10.0.0.12 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-30.0 sec  37.5 MBytes  10.5 Mbits/sec
[  3] Sent 26749 datagrams
[  3] Server Report:
[  3]  0.0-31.2 sec  35.8 MBytes  9.62 Mbits/sec   0.582 ms 1216/26749 (4.5%)
[  3] 0.0000-31.2259 sec  214 datagrams received out-of-order
mininet>
```

- Quan sát log của Controller. Thấy các statistic của các switch có network traffic tăng lên rất cao.

```
================================================================
Port Statistics for Switch: 0000000000000002
----------------------------------------------------------------
Port     |   Rx-Pkts|    Rx-Bytes|   Tx-Pkts|    Tx-Bytes|  Errors
----------------------------------------------------------------
1        |        79|        5762|       332|       23794|       0
2        |        79|        5762|       333|       23884|       0
3        |        79|        5762|       332|       23794|       0
4        |        79|        5782|       332|       23794|       0
5        |     25892|    38791414|       402|       37868|       0
6        |       392|       37578|     26014|    38799860|       0
LOCAL    |         0|           0|         0|           0|       0
----------------------------------------------------------------
```

```
================================================================
Port Statistics for Switch: 0000000000000003
----------------------------------------------------------------
Port     |   Rx-Pkts|    Rx-Bytes|   Tx-Pkts|    Tx-Bytes|  Errors
----------------------------------------------------------------
1        |        78|        5692|       332|       23774|       0
2        |        78|        5692|       333|       23864|       0
3        |        78|        5692|       332|       23774|       0
4        |        85|       14806|     25976|    38796552|       0
5        |     26014|    38799860|       392|       37578|       0
6        |       256|       19316|       386|       27872|       0
LOCAL    |         0|           0|         0|           0|       0
----------------------------------------------------------------
================================================================
```

```
================================================================
Port Statistics for Switch: 0000000000000001
----------------------------------------------------------------
Port     |   Rx-Pkts|    Rx-Bytes|   Tx-Pkts|    Tx-Bytes|  Errors
----------------------------------------------------------------
1        |        80|        5832|       335|       24211|       0
2        |        79|        5762|       336|       24281|       0
3        |        79|        5762|       333|       23996|       0
4        |     25724|    38778610|       342|       33270|       0
5        |       402|       37868|     25892|    38791414|       0
LOCAL    |         0|           0|         0|           0|       0
----------------------------------------------------------------
```

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