BÁO CÁO THỰC HÀNH

Môn học: Công nghệ mạng khả lập trình

Buổi báo cáo: Lab 03

Tên chủ đề: Lập trình mạng SDN/OpenFlow với Topology tuỳ ý trong Mininet

GVHD: Phan Xuân Thiện

Ngày thực hiện: 30/10/2025

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	
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

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

- **Cài đặt Mininet:**
  - Tải mã nguồn mininet từ GitHub

```
khanhle@khanhle:~$ git clone https://github.com/mininet/mininet
Cloning into 'mininet'...
remote: Enumerating objects: 10388, done.
remote: Counting objects: 100% (131/131), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 10388 (delta 104), reused 71 (delta 71), pack-reused 10257 (from 3)
Receiving objects: 100% (10388/10388), 3.36 MiB | 3.12 MiB/s, done.
Resolving deltas: 100% (6906/6906), done.
khanhle@khanhle:~$
```

  - Chạy script cài đặt Mininet. Lúc này, kiểm tra thấy rằng đã cài đặt thành công

```
khanhle@khanhle:~$ mininet/util/install.sh -a
Detected Linux distribution: Ubuntu 22.04 jammy amd64
sys.version_info(major=3, minor=10, micro=12, releaselevel='final', serial=0)
Detected Python (python3) version 3
Installing all packages except for -eix (doxypy, ivs, nox-classic)...
Install Mininet-compatible kernel if necessary
[sudo] password for khanhle:
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2,776 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,065 kB]
Get:10 https://pkg.jenkins.io/debian-stable binary/ Packages [30.0 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [473 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [19.0 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [4,832 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [403 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [898 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [640 B]
Get:17 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,242 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [307 kB]

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
----------------------------------------------------------------------
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/khanhle/oflops/example_modules/snmp_cpu'
make[2]: Leaving directory '/home/khanhle/oflops/example_modules/snmp_cpu'
make[2]: Entering directory '/home/khanhle/oflops/example_modules'
make[3]: Entering directory '/home/khanhle/oflops/example_modules'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/khanhle/oflops/example_modules'
make[2]: Leaving directory '/home/khanhle/oflops/example_modules'
make[1]: Leaving directory '/home/khanhle/oflops/example_modules'
Making install in cbench
make[1]: Entering directory '/home/khanhle/oflops/cbench'
make[2]: Entering directory '/home/khanhle/oflops/cbench'
 /usr/bin/mkdir -p '/usr/local/bin'
  /bin/bash ../libtool   --mode=install /usr/bin/install -c cbench '/usr/local/bin'
libtool: install: /usr/bin/install -c cbench /usr/local/bin/cbench
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/khanhle/oflops/cbench'
make[1]: Leaving directory '/home/khanhle/oflops/cbench'
Making install in doc
make[1]: Entering directory '/home/khanhle/oflops/doc'
make[1]: Nothing to be done for 'install'.
make[1]: Leaving directory '/home/khanhle/oflops/doc'
Enjoy Mininet!
```

```
khanhle@khanhle:~$ sudo mn
*** 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 ...
*** Starting CLI:
mininet>
```

- **Viết chương trình tạo mạng SDN/OpenFlow với topology như hình 1**
  - Tạo file topo.py với nội dung như sau

```
khanh@ubuntu:~/nt541/lab3$ cat topo.py
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink
from mininet.util import dumpNodeConnections

class CustomTopo(Topo):

    def build(self):

        # Add switches
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')

        # Add hosts and connect them to switches
        # For S1
        h1 = self.addHost('h1', ip='10.0.0.1/24')
        h2 = self.addHost('h2', ip='10.0.0.2/24')
        h3 = self.addHost('h3', ip='10.0.0.3/24')
        h4 = self.addHost('h4', ip='10.0.0.4/24')
        self.addLink(h1, s1, bw=10, delay='1ms')
        self.addLink(h2, s1, bw=10, delay='1ms')
        self.addLink(h3, s1, bw=10, delay='1ms')
        self.addLink(h4, s1, bw=10, delay='1ms')

        # For S2
        h5 = self.addHost('h5', ip='10.0.0.5/24')
        h6 = self.addHost('h6', ip='10.0.0.6/24')
        h7 = self.addHost('h7', ip='10.0.0.7/24')
        h8 = self.addHost('h8', ip='10.0.0.8/24')
        self.addLink(h5, s2, bw=10, delay='1ms')
        self.addLink(h6, s2, bw=10, delay='1ms')
        self.addLink(h7, s2, bw=10, delay='1ms')
        self.addLink(h8, s2, bw=10, delay='1ms')

        # For S3
        h9 = self.addHost('h9', ip='10.0.0.9/24')
        h10 = self.addHost('h10', ip='10.0.0.10/24')
        h11 = self.addHost('h11', ip='10.0.0.11/24')
        h12 = self.addHost('h12', ip='10.0.0.12/24')
        self.addLink(h9, s3, bw=10, delay='1ms')
        self.addLink(h10, s3, bw=10, delay='1ms')
        self.addLink(h11, s3, bw=10, delay='1ms')
        self.addLink(h12, s3, bw=10, delay='1ms')
        h13 = self.addHost('h13', ip='10.0.0.13/24')
        h14 = self.addHost('h14', ip='10.0.0.14/24')
        h15 = self.addHost('h15', ip='10.0.0.15/24')
        h16 = self.addHost('h16', ip='10.0.0.16/24')
        self.addLink(h13, s4, bw=10, delay='1ms')

        self.addLink(h14, s4, bw=10, delay='1ms')
        self.addLink(h15, s4, bw=10, delay='1ms')
        self.addLink(h16, s4, bw=10, delay='1ms')
        self.addLink(s1, s2, bw=20, delay='2ms')
        self.addLink(s2, s3, bw=20, delay='2ms')
        self.addLink(s3, s4, bw=20, delay='2ms')

def setup_network():
    # Create network
    net = Mininet(topo=CustomTopo(),
    controller=RemoteController('c0', ip='127.0.0.1', port=6653),
    link=TCLink,
    autoSetMacs=True)
    net.start()
    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel('info')
    setup_network()
khanh@ubuntu:~/nt541/lab3$
```

- Khởi tạo 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
```

- Khởi tạo Topology, thấy rằng đã kết nối tới Controller thành công.

```
khanh@ubuntu:~/nt541/lab3$ sudo python3 topo.py
*** Creating network
*** Adding controller
*** 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) (10.0
0Mbit 1ms 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 delay) (10.00Mbit 1ms delay) (h6, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h7, s2) (10.00Mbit 1ms delay) (10.00
Mbit 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 1ms delay) (10.00Mbit 1ms delay) (h14, s4) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h15, s
4) (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 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) (10.00Mb
it 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (10.00Mbi
t 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)
*** Starting CLI:
mininet>


packet in 0000000000000003 00:00:00:00:00:01 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 0000000000000003 00:00:00:00:00:0c 33:33:00:00:00:02 4
packet in 0000000000000002 00:00:00:00:00:0c 33:33:00:00:00:02 6
packet in 0000000000000004 00:00:00:00:00:0c 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:0c 33:33:00:00:00:02 5
packet in 0000000000000003 00:00:00:00:00:0b 33:33:00:00:00:02 3
packet in 0000000000000002 00:00:00:00:00:0b 33:33:00:00:00:02 6
packet in 0000000000000004 00:00:00:00:00:0b 33:33:00:00:00:02 5
packet in 0000000000000001 00:00:00:00:00:0b 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:0e 33:33:00:00:00:02 2
packet in 0000000000000004 00:00:00:00:00:0d 33:33:00:00:00:02 1
packet in 0000000000000003 00:00:00:00:00:0e 33:33:00:00:00:02 6
packet in 0000000000000003 00:00:00:00:00:0d 33:33:00:00:00:02 6
packet in 0000000000000002 00:00:00:00:00:0e 33:33:00:00:00:02 6
packet in 0000000000000002 00:00:00:00:00:0d 33:33:00:00:00:02 6
packet in 0000000000000001 00:00:00:00:00:0e 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 46:b2:1e:61:40:29 33:33:00:00:00:fb 5
packet in 0000000000000003 56:d6:dc:bf:13:88 33:33:00:00:00:fb 5
packet in 0000000000000002 0a:e0:6b:37:97:f4 33:33:00:00:00:fb 5
packet in 0000000000000002 32:90:dd:09:28:d2 33:33:00:00:00:fb 6
packet in 0000000000000004 56:d6:dc:bf:13:88 33:33:00:00:00:fb 5
packet in 0000000000000001 32:90:dd:09:28:d2 33:33:00:00:00:fb 5
packet in 0000000000000003 0a:e0:6b:37:97:f4 33:33:00:00:00:fb 5
packet in 0000000000000004 0a:e0:6b:37:97:f4 33:33:00:00:00:fb 5
packet in 0000000000000004 86:6b:96:14:a8:c5 33:33:00:00:00:fb 5
packet in 0000000000000003 26:8c:f0:bf:d1:43 33:33:00:00:00:fb 6
packet in 0000000000000002 26:8c:f0:bf:d1:43 33:33:00:00:00:fb 6
packet in 0000000000000001 26:8c:f0:bf:d1:43 33:33:00:00:00:fb 5
packet in 0000000000000001 00:00:00:00:00:02 33:33:00:00:00:02 2
packet in 0000000000000002 00:00:00:00:00:02 33:33:00:00:00:02 5
packet in 0000000000000003 00:00:00:00:00:02 33:33:00:00:00:02 5
packet in 0000000000000004 00:00:00:00:00:02 33:33:00:00:00:02 5
```

2. Test mạng SDN/OpenFlow được tạo ra, gồm: test kết nối, test hiệu suất của liên kết giữa hai host bất kỳ trong mạng.

- Test kết nối giữa các host với ping all:

```
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)
```

- Test hiệu suất liên kết giữa 2 host h1 và h13

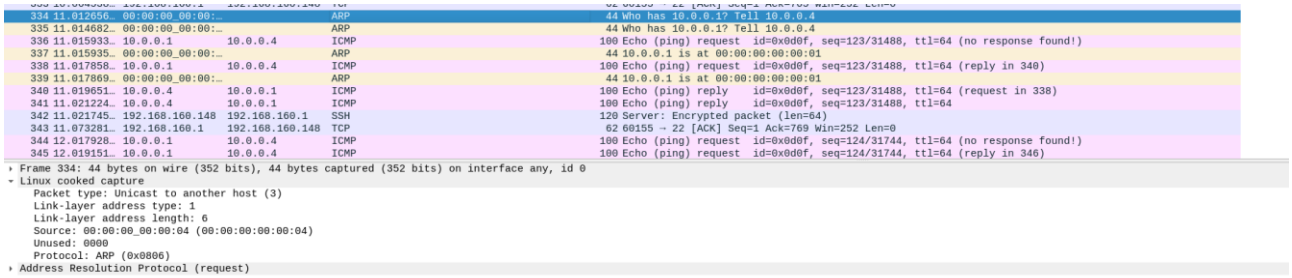
```
mininet> h1 iperf -s &
mininet> h13 iperf -c h1
------------------------------------------------------------
Client connecting to 10.0.0.1, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  3] local 10.0.0.13 port 55940 connected with 10.0.0.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.5 sec  14.1 MBytes  11.3 Mbits/sec
mininet>
```

3. **Mở Wireshark, tiến hành bắt các gói tin OpenFlow trao đổi giữa Controller và các Switch trong 2 trường hợp:**
   a. **Ping h1 tới h4**

- Tiến hành ping h1 tới h4:

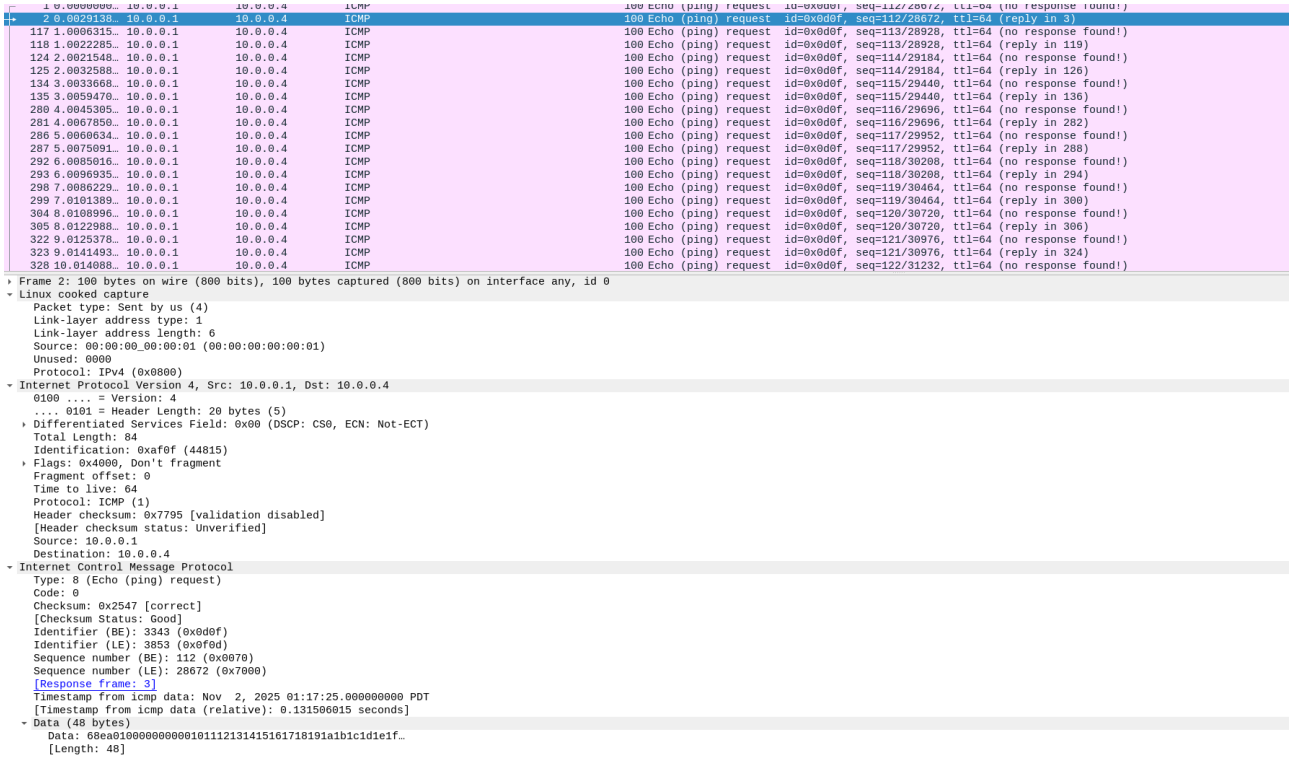
```
mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=18.0 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=5.71 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=6.65 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=5.94 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=5.95 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=7.60 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=6.84 ms
^C
--- 10.0.0.4 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 5.707/8.100/18.023/4.096 ms
mininet>
```

- Gói tin ARP:



  - Vì h1 và h4 nằm trong cùng 1 switch, vì vậy giao thức ARP sẽ dùng để 2 host này kết nối với nhau

- Gói tin ICMP:



- Giải thích: Gói tin ICMP được sinh ra khi h1 ping tới h4, thể hiện được sự kết nối thành công giữa 2 host này

- Gói tin OpenFlow Mod:

```
Out port: 0
Out group: 0
▾ Flags: 0x0000
    .... .... .... ...0 = Send flow removed: False
    .... .... .... ..0. = Check overlap: False
    .... .... .... .0.. = Reset counts: False
    .... .... .... 0... = Don't count packets: False
    .... .... ...0 .... = Don't count bytes: False
  Pad: 0000
▾ Match
    Type: OFPMT_OXM (1)
    Length: 32
  ▾ OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0000 000. = Field: OFPXMT_OFB_IN_PORT (0)
      .... ...0 = Has mask: False
      Length: 4
      Value: 1
  ▾ OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0000 011. = Field: OFPXMT_OFB_ETH_DST (3)
      .... ...0 = Has mask: False
      Length: 6
      Value: 00:00:00_00:00:04 (00:00:00:00:00:04)
  ▾ OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)
      .... ...0 = Has mask: False
      Length: 6
      Value: 00:00:00_00:00:01 (00:00:00:00:00:01)
▾ Instruction
    Type: OFPIT_APPLY_ACTIONS (4)
    Length: 24
    Pad: 00000000
  ▾ Action
      Type: OFPAT_OUTPUT (0)
      Length: 16
      Port: 4
      Max length: 65509
      Pad: 000000000000
```

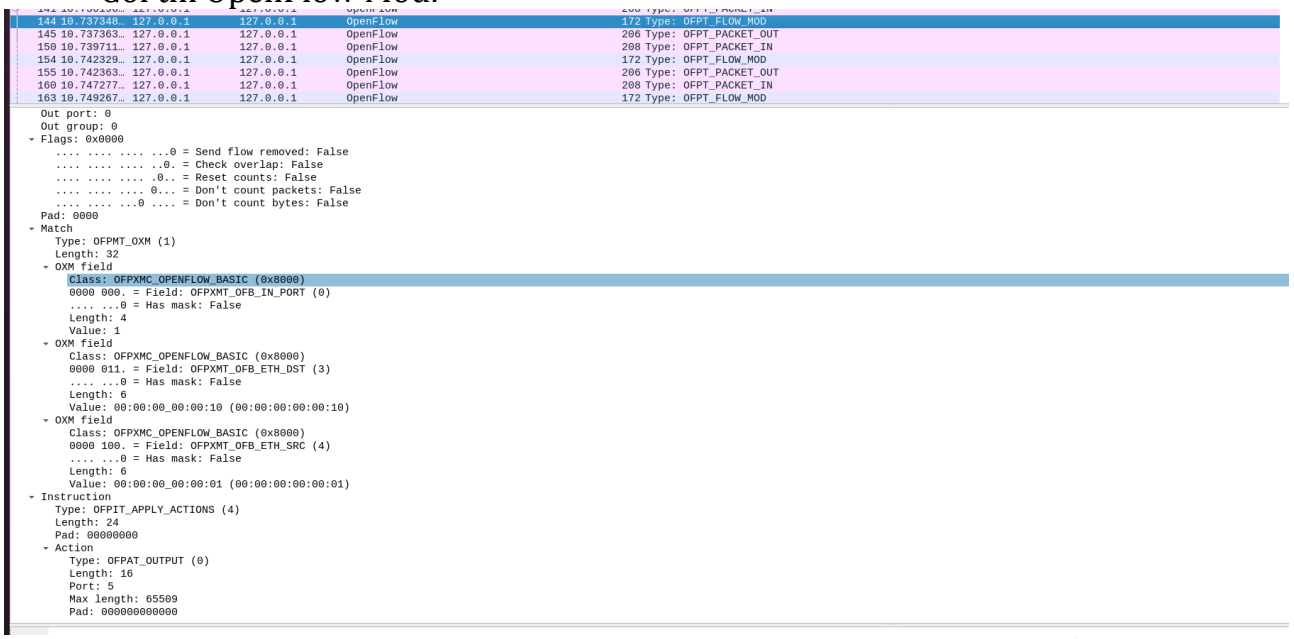
- Giải thích: Controller gửi xuống s1 2 gói tin để thêm hai flow entry vào flow table của s1:
  - IN_PORT=4
  - ETH_SRC = 00:00:00_00:00:01
  - ETH_DST: 00:00:00_00:00:04
  - OFPAT_OUTPUT: Chuyển tiếp gói tới port của s1

b. Ping h1 tới h16

- Tiến hành ping:

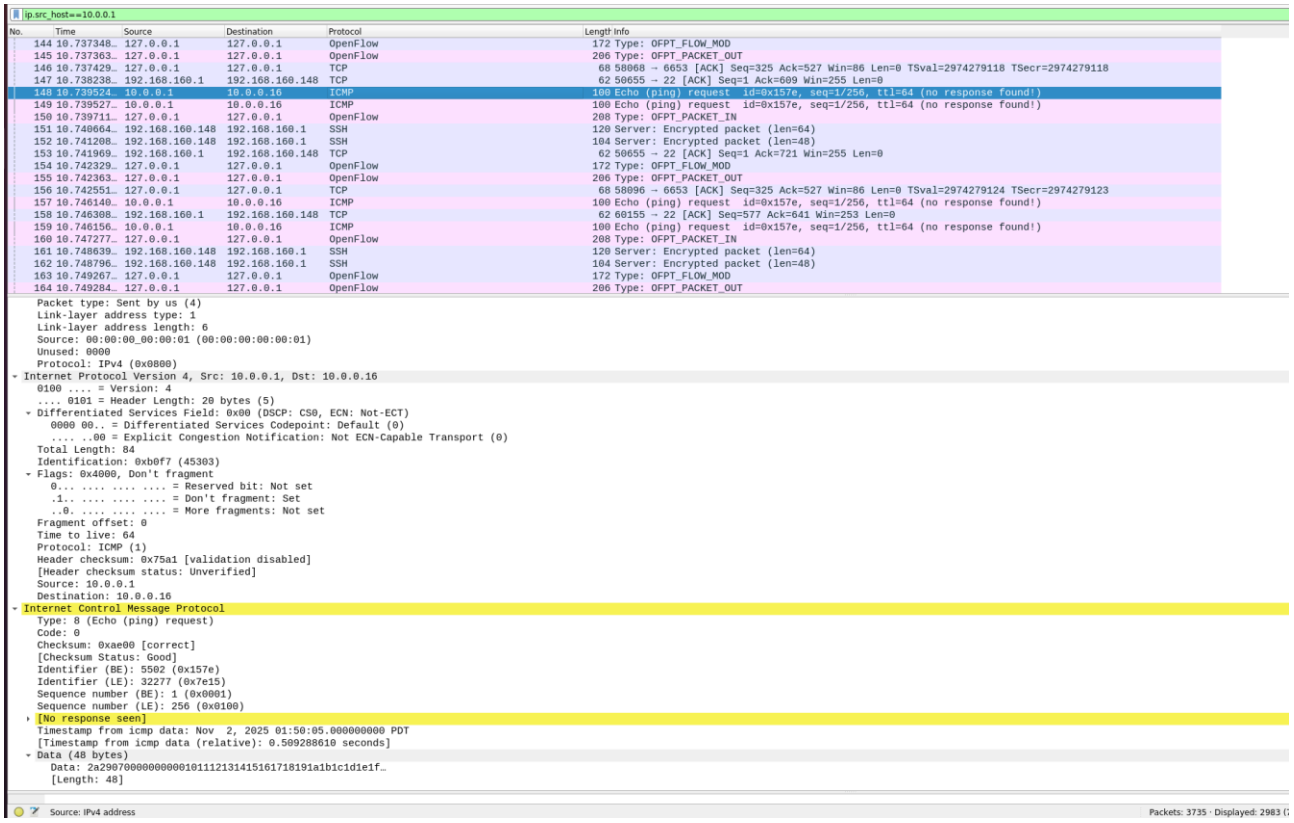
```
mininet> h1 ping h16
PING 10.0.0.16 (10.0.0.16) 56(84) bytes of data.
64 bytes from 10.0.0.16: icmp_seq=1 ttl=64 time=69.1 ms
64 bytes from 10.0.0.16: icmp_seq=2 ttl=64 time=27.2 ms
64 bytes from 10.0.0.16: icmp_seq=3 ttl=64 time=21.3 ms
64 bytes from 10.0.0.16: icmp_seq=4 ttl=64 time=21.9 ms
64 bytes from 10.0.0.16: icmp_seq=5 ttl=64 time=24.1 ms
^C
--- 10.0.0.16 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 21.274/32.715/69.071/18.298 ms
mininet>
```

- Gói tin OpenFlow Mod:



  - Giải thích: Vì h1 và h16 không kết nối vào chung 1 switch, dẫn tới việc controller phải thêm các flow entry ở cả 4 switch tạo thành flow từ s1 tới 4 (mỗi switch 2 flow entry cho gói tin từ h1 tới h16 và ngược lại)
    - Với gói tin ở trên, controller gửi xuống s3 trong gói tin OpenFlow Mod:
      - Gói tin thứ 1 có match field:
        - IN_PORT = 1
        - ETH_DST = 00:00:00_00:00:10
        - ETH_SRC = 00:00:00_00:00:01

- Gói tin ICMP:



  - Giải thích: Gói tin xuất hiện khi h1 ping tới h16. Việc ping thực hiện qua protocol ICMP.

4. Cài đặt OpenvSwitch và chạy thử mạng SDN/OpenFlow với OpenvSwitch (homework)

- Cài đặt OpenvSwitch

```
khanh@ubuntu:~/nt541/lab3$ sudo apt install openvswitch-switch
[sudo] password for khanh:
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
openvswitch-switch is already the newest version (2.13.8-0ubuntu1.4).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
khanh@ubuntu:~/nt541/lab3$
```

- Tạo file ovs_topo.py tương tự với topo.py ở câu 3. Nhưng thêm dòng switch=OVSSwitch ở RemoteController.

```
def setup_network():
    # Create network
    net = Mininet(topo=CustomTopo(),
    controller=RemoteController('c0', ip='127.0.0.1', port=6653),
    link=TCLink,
    switch=OVSSwitch,
    autoSetMacs=True)
    net.start()
    CLI(net)
    net.stop()
```

- 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 đoạn source code đã được tạo, thấy rằng đã tạo thành công topology

```
*** Creating network
*** Adding controller
*** 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) (10.0
0Mbit 1ms 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 delay) (10.00Mbit 1ms delay) (h6, s2) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h7, s2) (10.00Mbit 1ms delay) (10.00
Mbit 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 1ms delay) (10.00Mbit 1ms delay) (h14, s4) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (h15, s
4) (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 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) (10.00Mb
it 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (10.00Mbit 1ms delay) (20.00Mbit 2ms delay) (20.00Mbit 2ms delay) (10.00Mbi
t 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)
*** Starting CLI:
mininet>
```

- Kiểm tra với lệnh “sudo ovs-vsctl show” để kiểm tra thử topology đã được kết nối tới OpenvSwitch chưa. Thấy rằng các bridge s1 tới s4 đã kết nối thành công

```
khanh@ubuntu:~$ sudo ovs-vsctl show
[sudo] password for khanh:
1f00dcdb-fd72-4b46-a7ea-bd8a8dd6178f
    Bridge s3
        Controller "tcp:127.0.0.1:6653"
        fail_mode: secure
        Port s3
            Interface s3
                type: internal
        Port s3-eth3
            Interface s3-eth3
        Port s3-eth2
            Interface s3-eth2
        Port s3-eth5
            Interface s3-eth5
        Port s3-eth1
            Interface s3-eth1
        Port s3-eth4
            Interface s3-eth4
        Port s3-eth6
            Interface s3-eth6
    Bridge s4
        Controller "tcp:127.0.0.1:6653"
        fail_mode: secure
        Port s4-eth2
            Interface s4-eth2
        Port s4
            Interface s4
                type: internal
        Port s4-eth3
            Interface s4-eth3
        Port s4-eth1
            Interface s4-eth1
        Port s4-eth5
            Interface s4-eth5
        Port s4-eth4
            Interface s4-eth4
    Bridge s2
        Controller "tcp:127.0.0.1:6653"
        fail_mode: secure
        Port s2-eth4
            Interface s2-eth4
        Port s2-eth5
            Interface s2-eth5
        Port s2-eth1
            Interface s2-eth1
        Port s2
            Interface s2
                type: internal
        Port s2-eth6
```

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