Video chạy code SDN CK
Đầu tiên kích hoạt môi trường ảo:

```
source ~/ryu39-env/bin/activate
```

1 BASELINE topology 3 host

1.1 Mục tiêu (đưa vào mục 3.1 báo cáo)

Baseline dùng **Ryu simple_switch_13 (Learning Switch)** làm đối chứng: mạng chỉ L2 switching để host liên lạc bình thường, **không có kiểm tra ARP cache, không phát hiện spoofing, không rate-limit ARP**.

Kỳ vọng: bình thường ổn định; khi bị spoofing/flooding thì **không phát hiện, không ngăn chặn** → ARP poisoning hoặc tăng tải ARP/Packet-In.

1.2 Chạy baseline

Terminal 1 (controller):

```
ryu-manager ryu.app.simple_switch_13
```

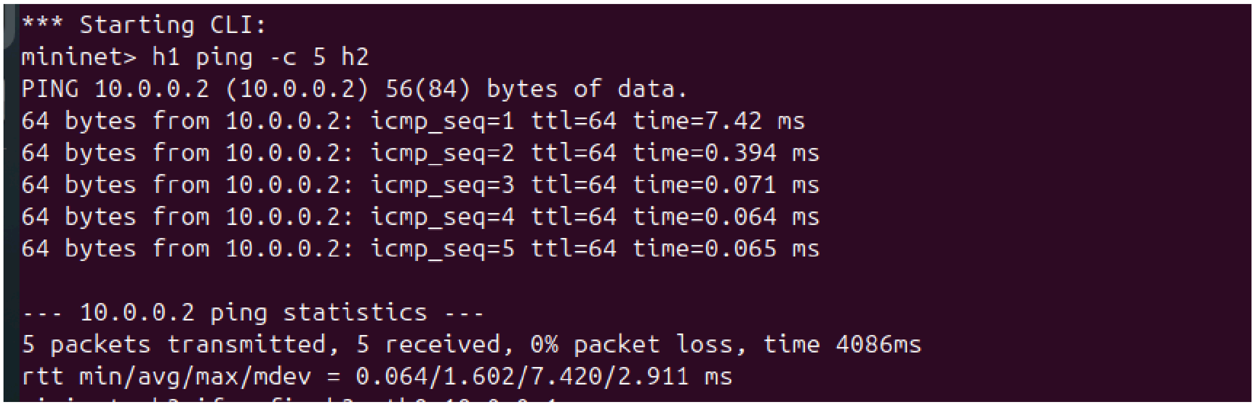
Terminal 2 (Mininet topo single,3):

```
sudo /usr/bin/mn --topo single,3 \
  --controller=remote,ip=127.0.0.1 \
  --mac --switch=ovs,protocols=OpenFlow13
```

1.3 Kịch bản A – Normal traffic

Trong Mininet:

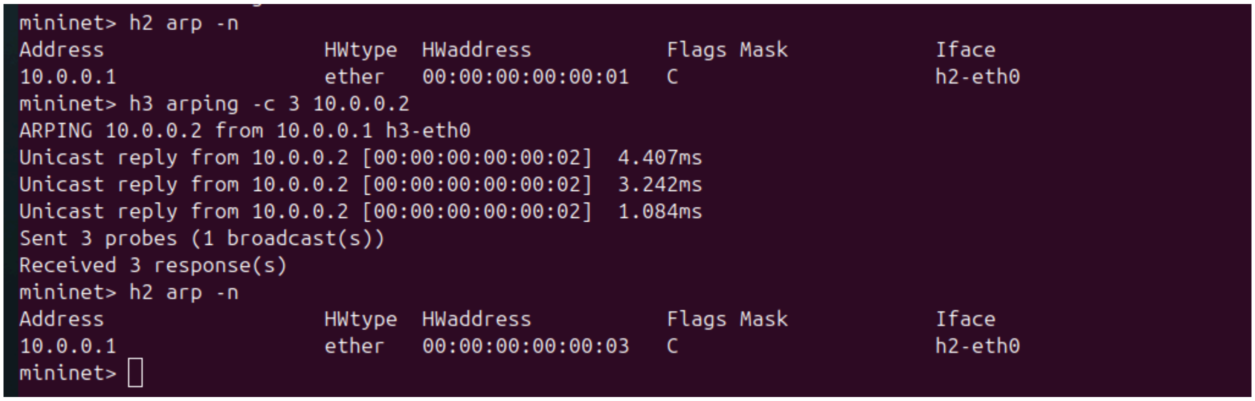
```
h1 ping -c 5 h2
```



```
*** Starting CLI:
mininet> h1 ping -c 5 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=7.42 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.394 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.071 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.064 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.065 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4086ms
rtt min/avg/max/mdev = 0.064/1.602/7.420/2.911 ms
```

1.4 Kịch bản B – ARP Spoofing/Poisoning trong baseline

```
h3 ifconfig h3-eth0 10.0.0.1
h2 arp -n          # xem trước: 10.0.0.1 -> MAC của h1 (...:01)
h3 arping -c 3 10.0.0.2
h2 arp -n          # xem sau: 10.0.0.1 -> MAC của h3 (...:03)
```

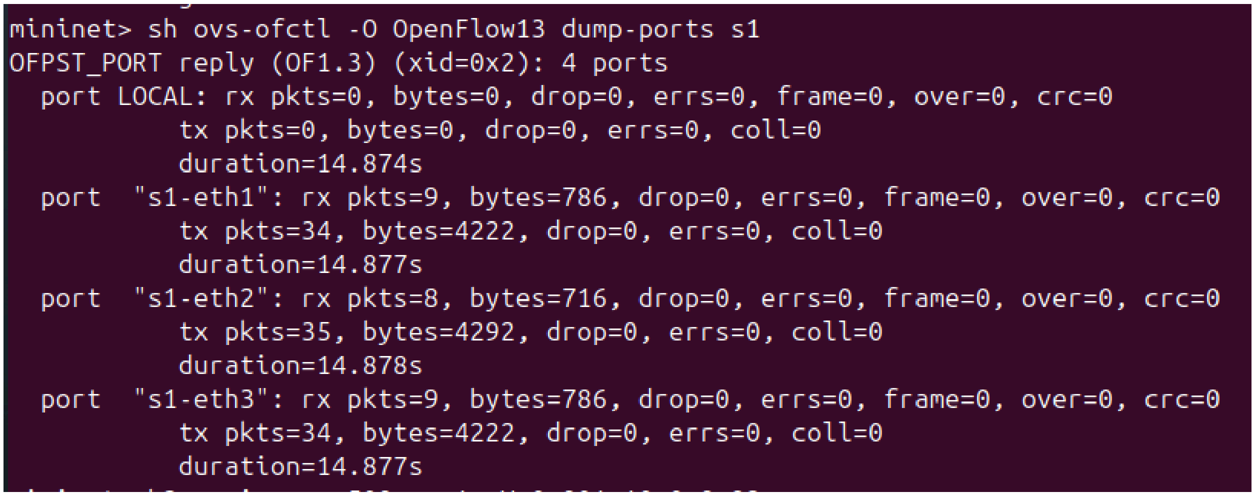


```
mininet> h2 arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.1                 ether   00:00:00:00:00:01   C                     h2-eth0
mininet> h3 arping -c 3 10.0.0.2
ARPING 10.0.0.2 from 10.0.0.1 h3-eth0
Unicast reply from 10.0.0.2 [00:00:00:00:00:02]  4.407ms
Unicast reply from 10.0.0.2 [00:00:00:00:00:02]  3.242ms
Unicast reply from 10.0.0.2 [00:00:00:00:00:02]  1.084ms
Sent 3 probes (1 broadcast(s))
Received 3 response(s)
mininet> h2 arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.1                 ether   00:00:00:00:00:03   C                     h2-eth0
mininet> 
```

1.5 Kịch bản C – ARP Flooding trong baseline + chứng cứ dump-ports

Trước flood:

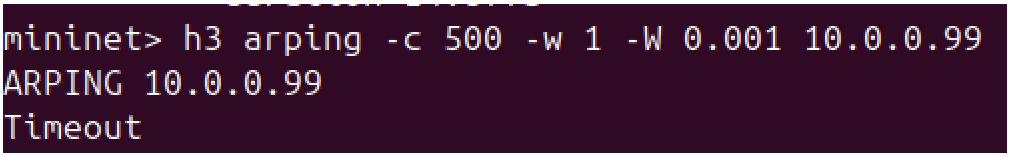
```
sh ovs-ofctl -O OpenFlow13 dump-ports s1
```



```
mininet> sh ovs-ofctl -O OpenFlow13 dump-ports s1
OFPST_PORT reply (OF1.3) (xid=0x2): 4 ports
  port LOCAL: rx pkts=0, bytes=0, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=0, bytes=0, drop=0, errs=0, coll=0
           duration=14.874s
  port  "s1-eth1": rx pkts=9, bytes=786, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=34, bytes=4222, drop=0, errs=0, coll=0
           duration=14.877s
  port  "s1-eth2": rx pkts=8, bytes=716, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=35, bytes=4292, drop=0, errs=0, coll=0
           duration=14.878s
  port  "s1-eth3": rx pkts=9, bytes=786, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=34, bytes=4222, drop=0, errs=0, coll=0
           duration=14.877s
```

Flood:

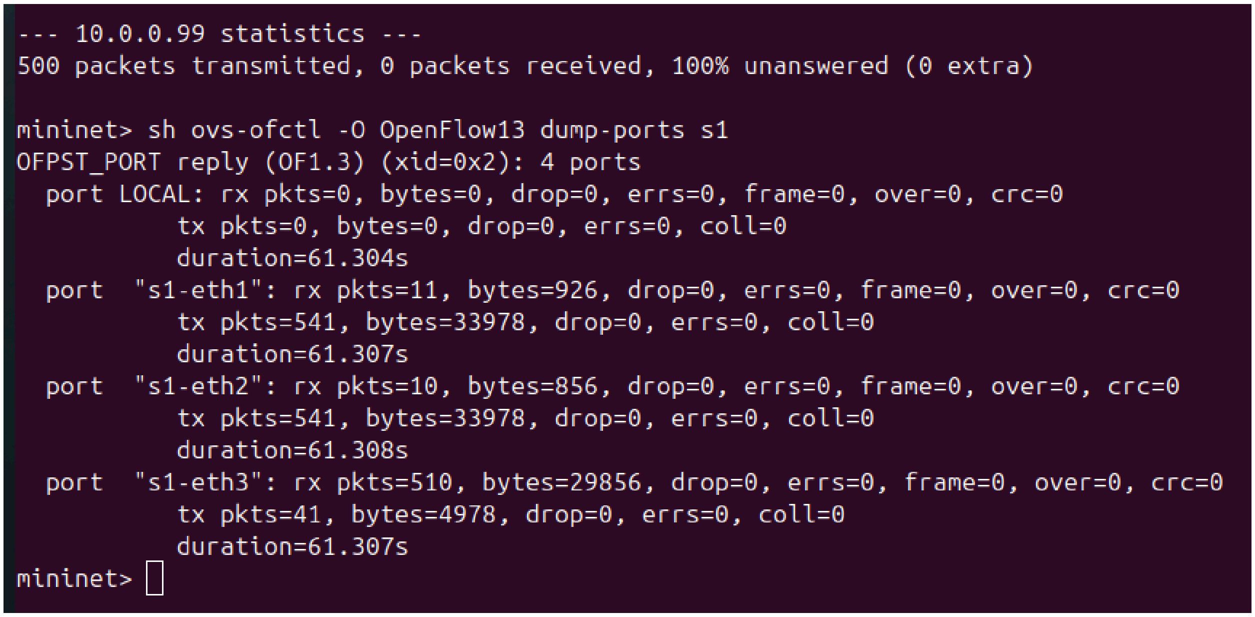
h3 arping -c 500 -w 1 -W 0.001 10.0.0.99



```
mininet> h3 arping -c 500 -w 1 -W 0.001 10.0.0.99
ARPING 10.0.0.99
Timeout
```

Sau flood:

sh ovs-ofctl -O OpenFlow13 dump-ports s1



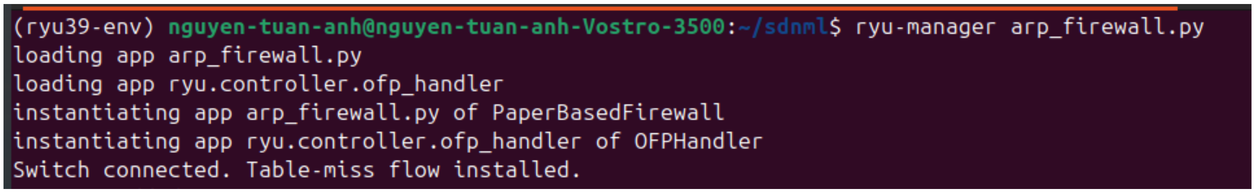
```
--- 10.0.0.99 statistics ---
500 packets transmitted, 0 packets received, 100% unanswered (0 extra)

mininet> sh ovs-ofctl -O OpenFlow13 dump-ports s1
OFPST_PORT reply (OF1.3) (xid=0x2): 4 ports
  port LOCAL: rx pkts=0, bytes=0, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=0, bytes=0, drop=0, errs=0, coll=0
           duration=61.304s
  port  "s1-eth1": rx pkts=11, bytes=926, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=541, bytes=33978, drop=0, errs=0, coll=0
           duration=61.307s
  port  "s1-eth2": rx pkts=10, bytes=856, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=541, bytes=33978, drop=0, errs=0, coll=0
           duration=61.308s
  port  "s1-eth3": rx pkts=510, bytes=29856, drop=0, errs=0, frame=0, over=0, crc=0
           tx pkts=41, bytes=4978, drop=0, errs=0, coll=0
           duration=61.307s
mininet> 
```

2 FIREWALL ARP

2.1 Chạy firewall ARP

Terminal 1:

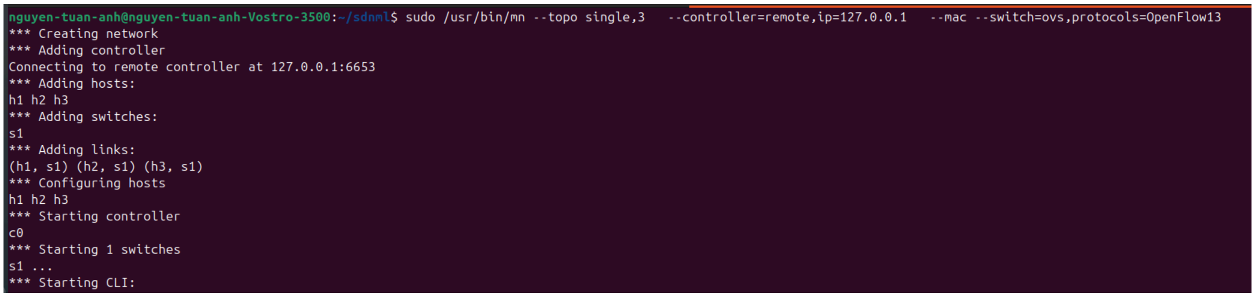
```
ryu-manager arp_firewall.py
```



```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnml$ ryu-manager arp_firewall.py
loading app arp_firewall.py
loading app ryu.controller.ofp_handler
instantiating app arp_firewall.py of PaperBasedFirewall
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch connected. Table-miss flow installed.
```

Terminal 2:

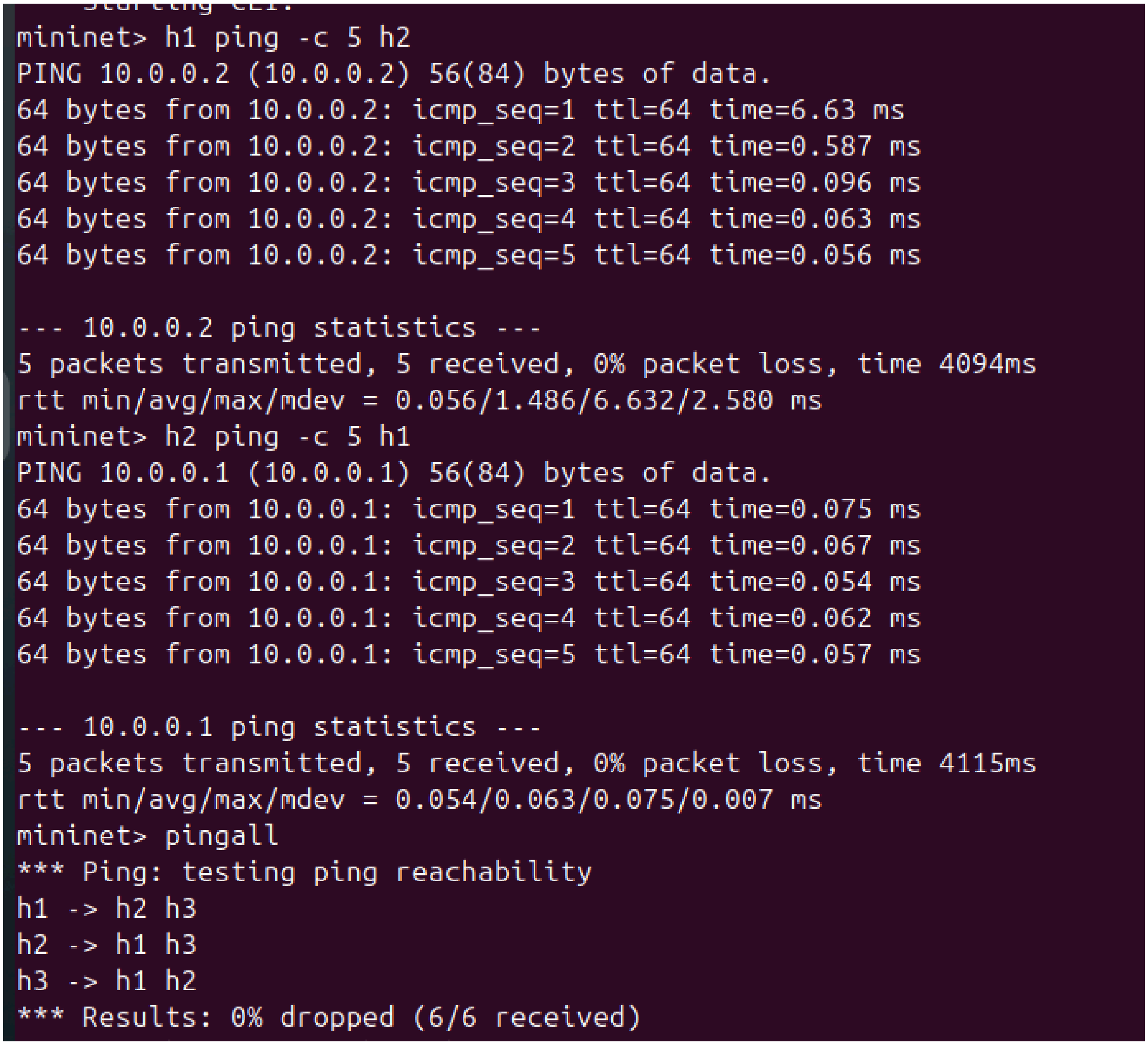
```
sudo /usr/bin/mn --topo single,3 \
  --controller=remote,ip=127.0.0.1 \
  --mac --switch=ovs,protocols=OpenFlow13
```



```
nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnml$ sudo /usr/bin/mn --topo single,3   --controller=remote,ip=127.0.0.1   --mac --switch=ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
Connecting to remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
```

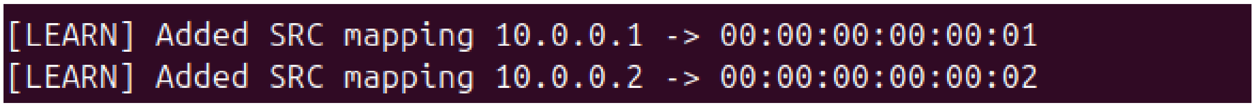
2.2 Normal traffic (để firewall “learn” ARP cache)

```
h1 ping -c 5 h2
h2 ping -c 5 h1
pingall
```



```
mininet> h1 ping -c 5 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=6.63 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.587 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.056 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4094ms
rtt min/avg/max/mdev = 0.056/1.486/6.632/2.580 ms
mininet> h2 ping -c 5 h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.075 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.067 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.062 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.057 ms

--- 10.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4115ms
rtt min/avg/max/mdev = 0.054/0.063/0.075/0.007 ms
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

Firewall học ARP cache từ traffic bình thường:

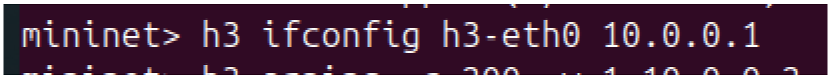


```
[LEARN] Added SRC mapping 10.0.0.1 -> 00:00:00:00:00:01
[LEARN] Added SRC mapping 10.0.0.2 -> 00:00:00:00:00:02
```

2.3 Kịch bản ARP Spoofing Detection

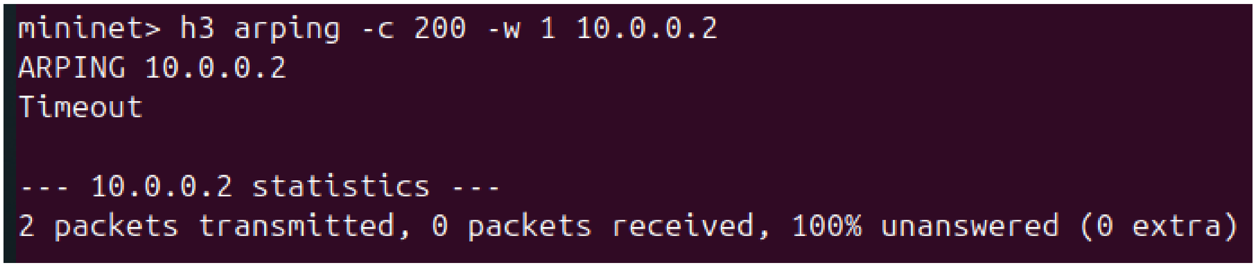
Attacker giả IP của h1:

h3 ifconfig h3-eth0 10.0.0.1

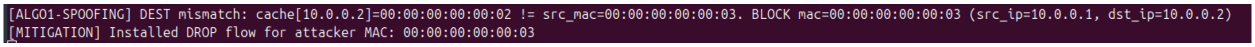


```
mininet> h3 ifconfig h3-eth0 10.0.0.1
```

Gửi ARP Request tới h2:
h3 arping -c 200 -w 1 10.0.0.2



```
mininet> h3 arping -c 200 -w 1 10.0.0.2
ARPING 10.0.0.2
Timeout

--- 10.0.0.2 statistics ---
2 packets transmitted, 0 packets received, 100% unanswered (0 extra)
```

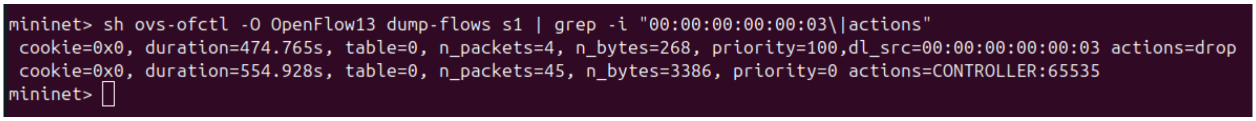


```
[ALGO1-SPOOFING] DEST mismatch: cache[10.0.0.2]=00:00:00:00:00:02 != src_mac=00:00:00:00:00:03. BLOCK mac=00:00:00:00:00:03 (src_ip=10.0.0.1, dst_ip=10.0.0.2)
[MITIGATION] Installed DROP flow for attacker MAC: 00:00:00:00:00:03
```

h3 dùng MAC thật ...:03 nhưng giả IP 10.0.0.1. Firewall phát hiện sai lệch so với ARP cache đã học và kích hoạt cơ chế chặn.

Lệnh kiểm tra flow trên OVS (s1)

sh ovs-ofctl -O OpenFlow13 dump-flows s1 | grep -i "00:00:00:00:00:03\|actions"



```
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1 | grep -i "00:00:00:00:00:03\|actions"
 cookie=0x0, duration=474.765s, table=0, n_packets=4, n_bytes=268, priority=100,dl_src=00:00:00:00:00:03 actions=drop
 cookie=0x0, duration=554.928s, table=0, n_packets=45, n_bytes=3386, priority=0 actions=CONTROLLER:65535
mininet> 
```

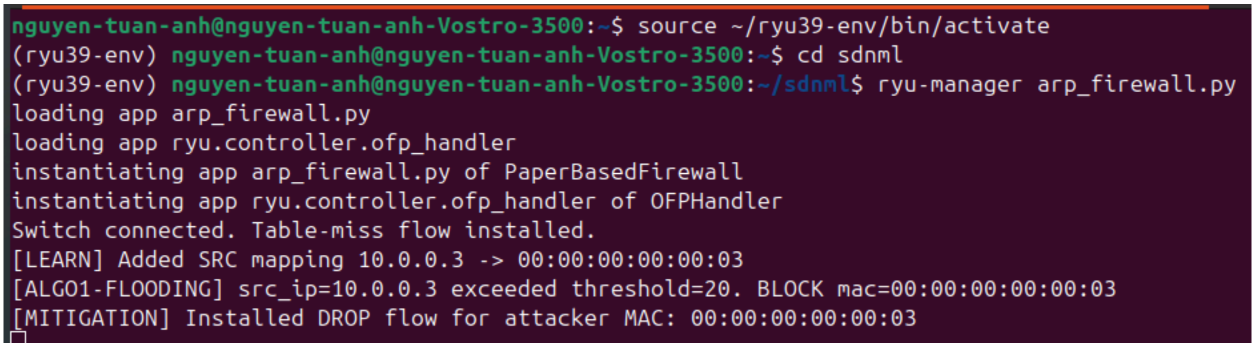
Kết quả cho thấy switch đã cài một flow có độ ưu tiên cao priority=100 khớp theo địa chỉ MAC nguồn dl_src=00:00:00:00:00:03 và thực hiện actions=drop, tức là mọi gói từ attacker sẽ bị loại bỏ ngay tại switch. Dòng priority=0 actions=CONTROLLER:65535 là rule table-miss mặc định, dùng để chuyển các gói không khớp rule cụ thể lên controller xử lý.

h3 ifconfig h3-eth0 10.0.0.3

2.4 Kịch bản ARP Flooding

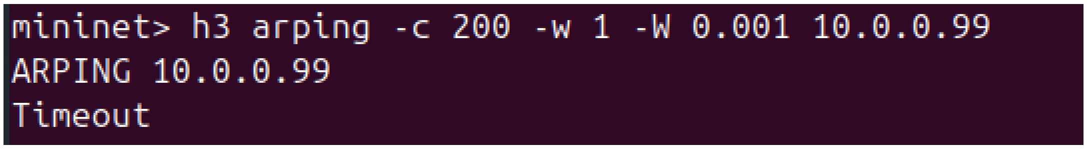
ARP_THRESHOLD = 20 req/s, cửa sổ 1 giây.

h3 arping -c 200 -w 1 -W 0.001 10.0.0.99
ryu-manager arp_firewall.py



```
nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~$ source ~/ryu39-env/bin/activate
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~$ cd sdnml
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnml$ ryu-manager arp_firewall.py
loading app arp_firewall.py
loading app ryu.controller.ofp_handler
instantiating app arp_firewall.py of PaperBasedFirewall
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch connected. Table-miss flow installed.
[LEARN] Added SRC mapping 10.0.0.3 -> 00:00:00:00:00:03
[ALGO1-FLOODING] src_ip=10.0.0.3 exceeded threshold=20. BLOCK mac=00:00:00:00:00:03
[MITIGATION] Installed DROP flow for attacker MAC: 00:00:00:00:00:03
```

```
sudo /usr/bin/mn --topo single,3 \
  --controller=remote,ip=127.0.0.1 \
  --mac --switch=ovs,protocols=OpenFlow13
```



```
mininet> h3 arping -c 200 -w 1 -W 0.001 10.0.0.99
ARPING 10.0.0.99
Timeout
```

Show minh chứng tốc độ cụ thể theardhold bao nhiêu:

- Bật tcpdump để lưu ARP vào pcap

h3 bash -lc 'tcpdump -i h3-eth0 -n -U -w /tmp/h3_arp.pcap arp > /tmp/h3_tcpdump.log 2>&1 & echo $!'

- Kiểm tra tcpdump đang chạy và file đang được ghi

h3 bash -lc 'pgrep -a tcpdump; ls -l /tmp/h3_arp.pcap /tmp/h3_tcpdump.log; tail -n 2 /tmp/h3_tcpdump.log'

- Thực hiện ARP flooding bằng arping

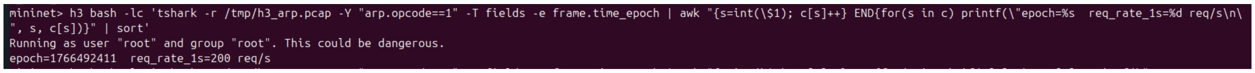
h3 arping -c 200 -w 1 -W 0.001 10.0.0.99

- Kiểm tra kích thước pcap

h3 ls -lh /tmp/h3_arp.pcap

- Tính req/s theo từng cửa sổ 1 giây từ pcap

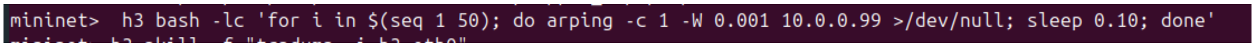
h3 bash -lc 'tshark -r /tmp/h3_arp.pcap -Y "arp.opcode==1" -T fields -e frame.time_epoch | awk "{s=int(\$1); c[s]++} END{for(s in c) printf(\"epoch=%s req_rate_1s=%d req/s\n\", s, c[s])}" | sort'



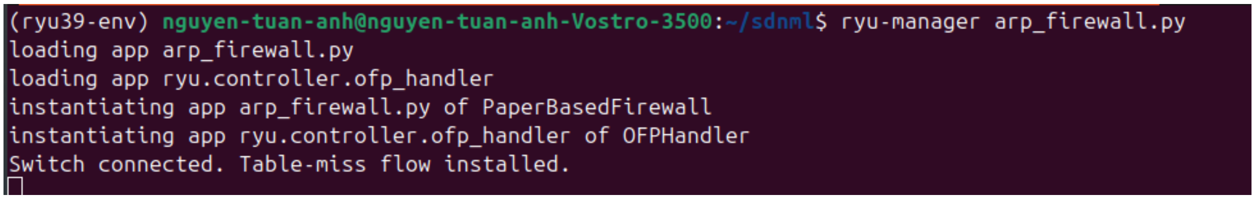
```
mininet> h3 bash -lc 'tshark -r /tmp/h3_arp.pcap -Y "arp.opcode==1" -T fields -e frame.time_epoch | awk "{s=int(\$1); c[s]++} END{for(s in c) printf(\"epoch=%s  req_rate_1s=%d req/s\n\
", s, c[s])}" | sort'
Running as user "root" and group "root". This could be dangerous.
epoch=1766492411  req_rate_1s=200 req/s
```

Kịch bản né ngưỡng :

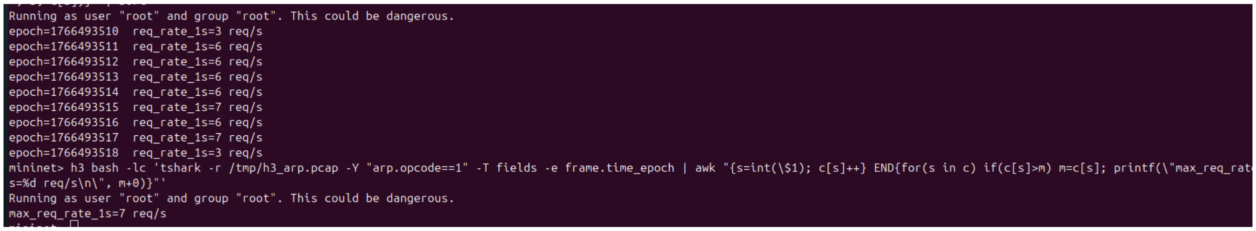
Câu lệnh được thực thi trong host **h3** và tạo lưu lượng ARP có kiểm soát bằng cách gửi **50 ARP Request** tới địa chỉ 10.0.0.99. Mỗi vòng lặp chỉ gửi **1 gói** (-c 1) rồi tạm dừng **0.1 giây** (sleep 0.10) trước khi gửi tiếp, nhằm mô phỏng kịch bản **low-rate ARP**



```
mininet>  h3 bash -lc 'for i in $(seq 1 50); do arping -c 1 -W 0.001 10.0.0.99 >/dev/null; sleep 0.10; done'
```

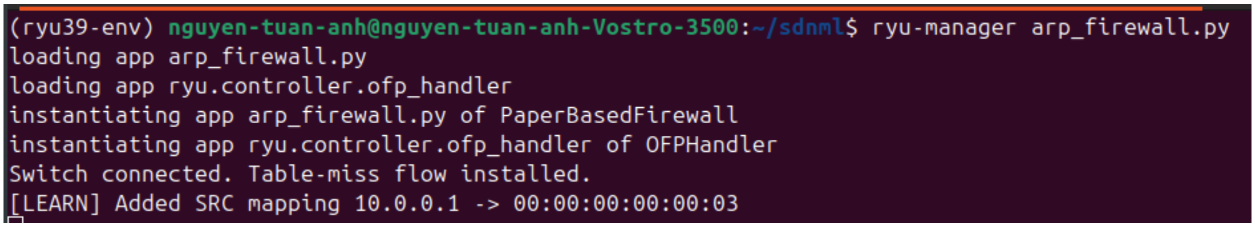


```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnml$ ryu-manager arp_firewall.py
loading app arp_firewall.py
loading app ryu.controller.ofp_handler
instantiating app arp_firewall.py of PaperBasedFirewall
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch connected. Table-miss flow installed.
```

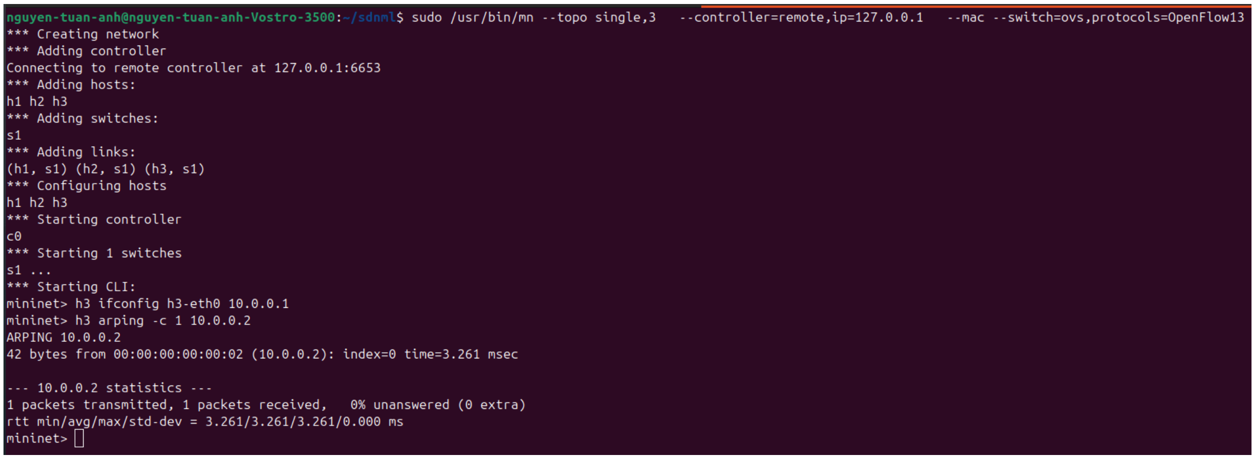


```
Running as user "root" and group "root". This could be dangerous.
epoch=1766493510  req_rate_1s=3 req/s
epoch=1766493511  req_rate_1s=6 req/s
epoch=1766493512  req_rate_1s=6 req/s
epoch=1766493513  req_rate_1s=6 req/s
epoch=1766493514  req_rate_1s=6 req/s
epoch=1766493515  req_rate_1s=7 req/s
epoch=1766493516  req_rate_1s=6 req/s
epoch=1766493517  req_rate_1s=7 req/s
epoch=1766493518  req_rate_1s=3 req/s
mininet> h3 bash -lc 'tshark -r /tmp/h3_arp.pcap -Y "arp.opcode==1" -T fields -e frame.time_epoch | awk "{s=int(\$1); c[s]++} END{for(s in c) if(c[s]>m) m=c[s]; printf(\"max_req_rat
s=%d req/s\n\", m+0)}"'
Running as user "root" and group "root". This could be dangerous.
max_req_rate_1s=7 req/s
```

Kịch bản: Tấn công mạo danh khi Cache rỗng:



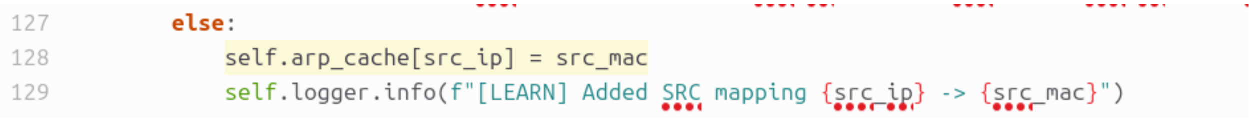
```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnml$ ryu-manager arp_firewall.py
loading app arp_firewall.py
loading app ryu.controller.ofp_handler
instantiating app arp_firewall.py of PaperBasedFirewall
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch connected. Table-miss flow installed.
[LEARN] Added SRC mapping 10.0.0.1 -> 00:00:00:00:00:03
```



```
nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnml$ sudo /usr/bin/mn --topo single,3   --controller=remote,ip=127.0.0.1   --mac --switch=ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
Connecting to remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h3 ifconfig h3-eth0 10.0.0.1
mininet> h3 arping -c 1 10.0.0.2
ARPING 10.0.0.2
42 bytes from 00:00:00:00:00:02 (10.0.0.2): index=0 time=3.261 msec

--- 10.0.0.2 statistics ---
1 packets transmitted, 1 packets received,   0% unanswered (0 extra)
rtt min/avg/max/std-dev = 3.261/3.261/3.261/0.000 ms
mininet>
```

Ở trạng thái khởi động ban đầu, **ARP cache của controller rỗng**. Khi attacker **h3 đổi IP thành 10.0.0.1** và gửi ARP Request (arping), controller **chưa có ánh xạ IP–MAC trước đó** nên **học lần đầu** ánh xạ
10.0.0.1 → MAC của h3 ([LEARN] Added SRC mapping).

Minh chứng trong code làm :



```
        else:
            self.arp_cache[src_ip] = src_mac
            self.logger.info(f"[LEARN] Added SRC mapping {src_ip} -> {src_mac}")
```

Do **thuật toán Algorithm 1 áp dụng cơ chế "learn-first" khi cache rỗng**, không có mâu thuẫn IP–MAC để so sánh, nên **tấn công mạo danh không bị phát hiện và không bị chặn** ở bước này.

3 THU DỮ LIỆU cho ML — collector.py

3.1 Thu benign

Theo đúng file hướng dẫn của bạn: dùng label.txt, set DATA_LABEL, DATA_OUT, chạy collector và log ra file.

```
echo benign > label.txt
DATA_LABEL=benign DATA_OUT=data/benign_arp_traffic.csv \
ryu-manager collector.py --ofp-tcp-listen-port 6653 2>&1 | tee logs/collector_benign.log

sudo /usr/bin/mn --topo single,4 \
  --controller=remote,ip=127.0.0.1 \
  --mac --switch=ovs,protocols=OpenFlow13
```

-Tạo traffic benign

```
Pingall
h1 ping -c 10 h2
h2 ping -c 10 h3
h1 ping -c 10 h3
```

3.2 Thu attack

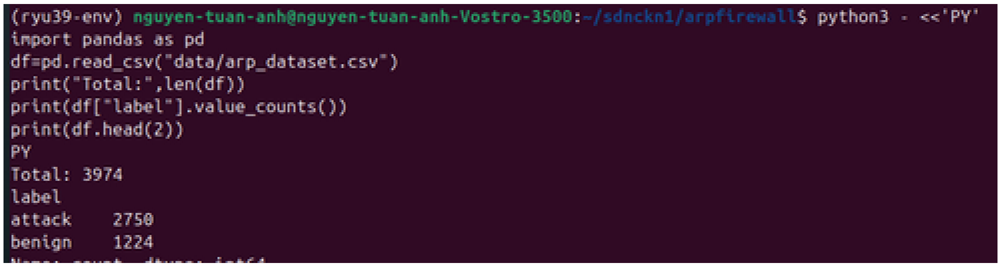
```
echo attack > label.txt
DATA_LABEL=attack DATA_OUT=data/attack_arp_traffic.csv \
ryu-manager collector.py --ofp-tcp-listen-port 6653 2>&1 | tee logs/collector_attack.log
```

Tạo dữ liệu attack:

```
h3 ifconfig h3-eth0 10.0.0.1
h3 apring -c 10 10.0.0.2
h3 apring -c 10 10.0.0.1
h3 ifconfig h3-eth0 10.0.0.3
h3 ifconfig h3-eth0 10.0.0.4
h3 apring -c 10 10.0.0.1
h3 apring -c 50 -W 0.20 10.0.0.1
h3 apring -c 50 -W 0.10 10.0.0.1
h3 apring -c 80 -W 0.04 10.0.0.1
h3 apring -c 200 -W 0.02 10.0.0.1
h3 apring -c 50 -W 0.1 10.0.0.1
```

4 HUẤN LUYỆN

Dữ liệu huấn luyện và phân bố nhãn



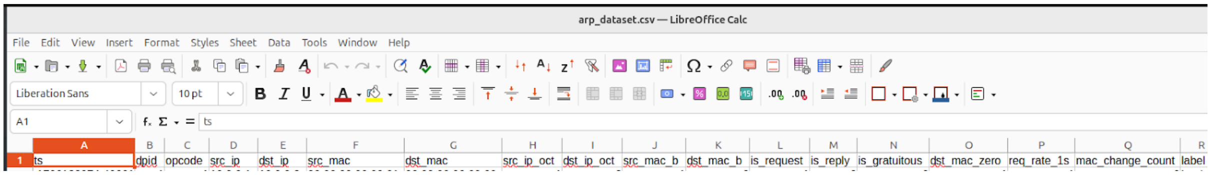
```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnckn1/arpfirewall$ python3 - <<'PY'
import pandas as pd
df=pd.read_csv("data/arp_dataset.csv")
print("Total:",len(df))
print(df["label"].value_counts())
print(df.head(2))
PY
Total: 3974
label
attack     2750
benign     1224
```

Thống kê số mẫu và phân bố nhãn của tập dữ liệu ARP”.

Tập đặc trưng sử dụng cho mô hình

Mô hình DNN sử dụng **11 đặc trưng** đúng theo cấu hình trong bước đánh giá:

opcode, src_ip_oct, dst_ip_oct, src_mac_b, dst_mac_b, is_request, is_reply, is_gratuitous, dst_mac_zero, req_rate_1s, mac_change_count.

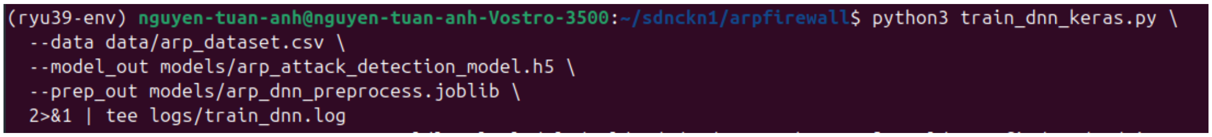

arp_dataset.csv — LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation Sans
10 pt
A1
ts

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	ts	dpid	opcode	src_ip	dst_ip	src_mac	dst_mac	src_ip_oct	dst_ip_oct	src_mac_b	dst_mac_b	is_request	is_reply	is_gratuitous	dst_mac_zero	req_rate_1s	mac_change_count	label

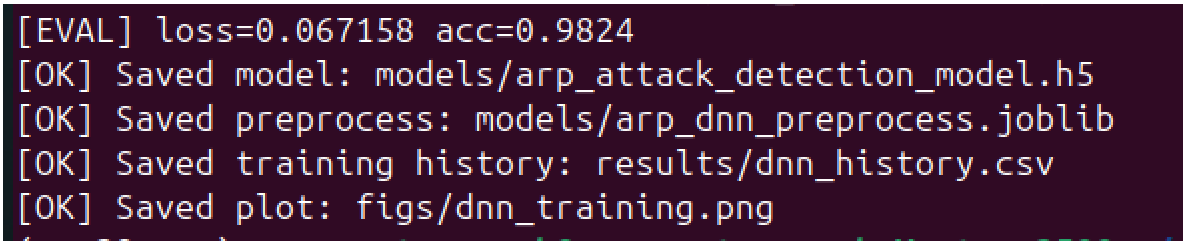
Quy trình huấn luyện mô hình DNN

Mô hình được huấn luyện bằng script train_dnn_keras.py với lệnh:

```
python3 train_dnn_keras.py \
  --data data/arp_dataset.csv \
  --model_out models/arp_attack_detection_model.h5 \
  --prep_out models/arp_dnn_preprocess.joblib \
  2>&1 | tee logs/train_dnn.log
```

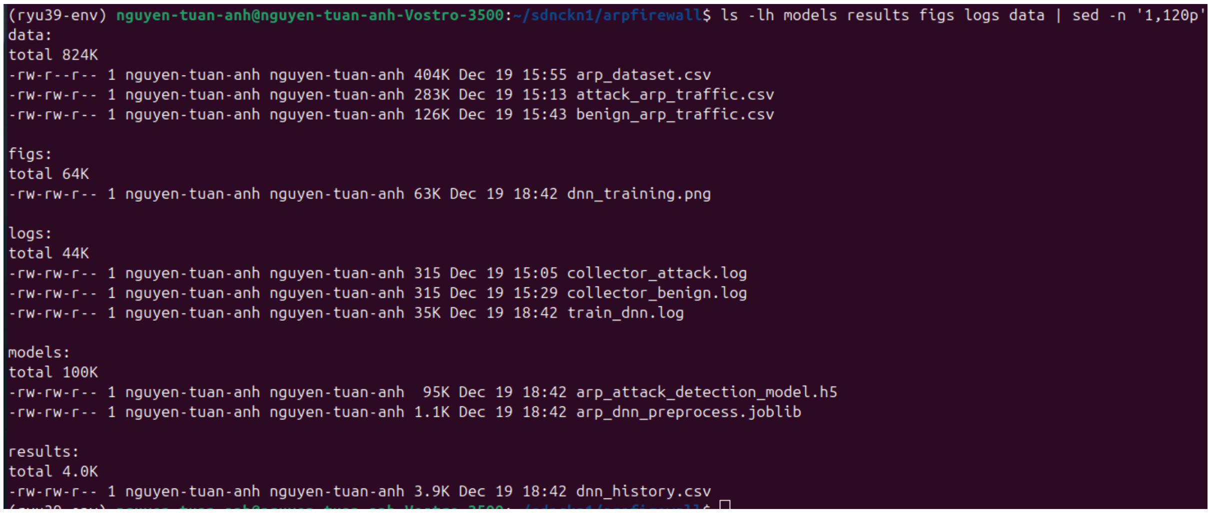


```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnckn1/arpfirewall$ python3 train_dnn_keras.py \
  --data data/arp_dataset.csv \
  --model_out models/arp_attack_detection_model.h5 \
  --prep_out models/arp_dnn_preprocess.joblib \
  2>&1 | tee logs/train_dnn.log
```



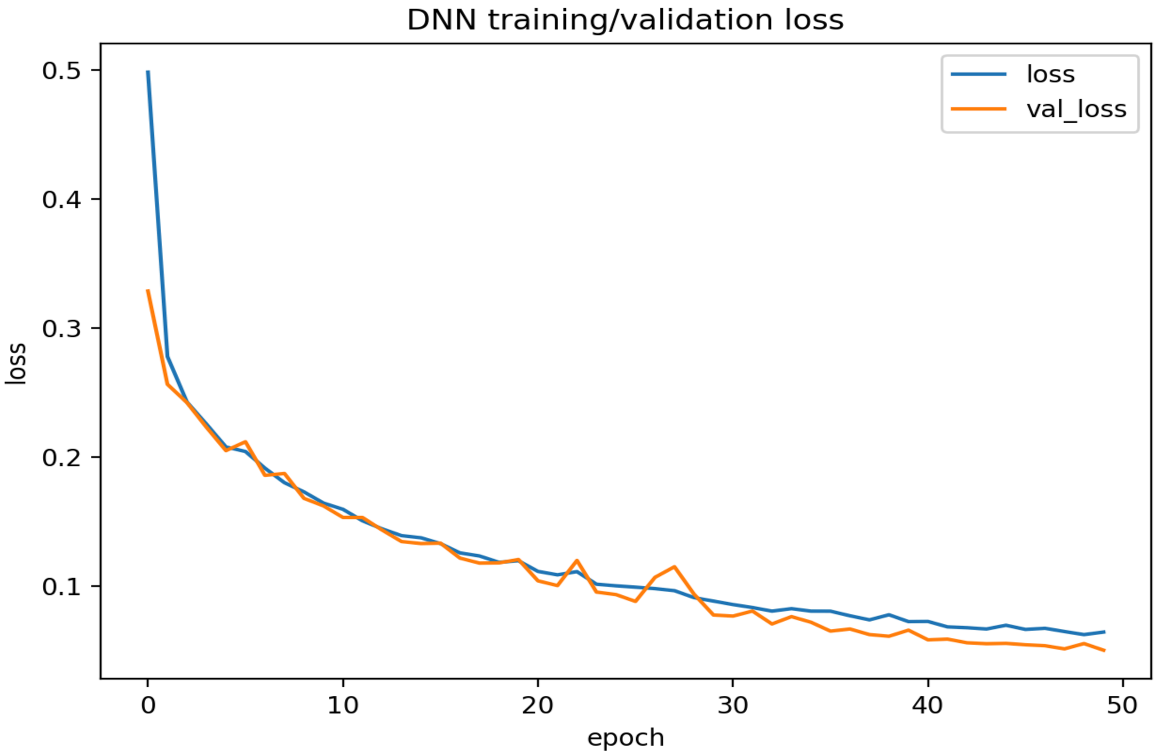
```
[EVAL] loss=0.067158 acc=0.9824
[OK] Saved model: models/arp_attack_detection_model.h5
[OK] Saved preprocess: models/arp_dnn_preprocess.joblib
[OK] Saved training history: results/dnn_history.csv
[OK] Saved plot: figs/dnn_training.png
```

Kết thúc train và chỉ số [EVAL]



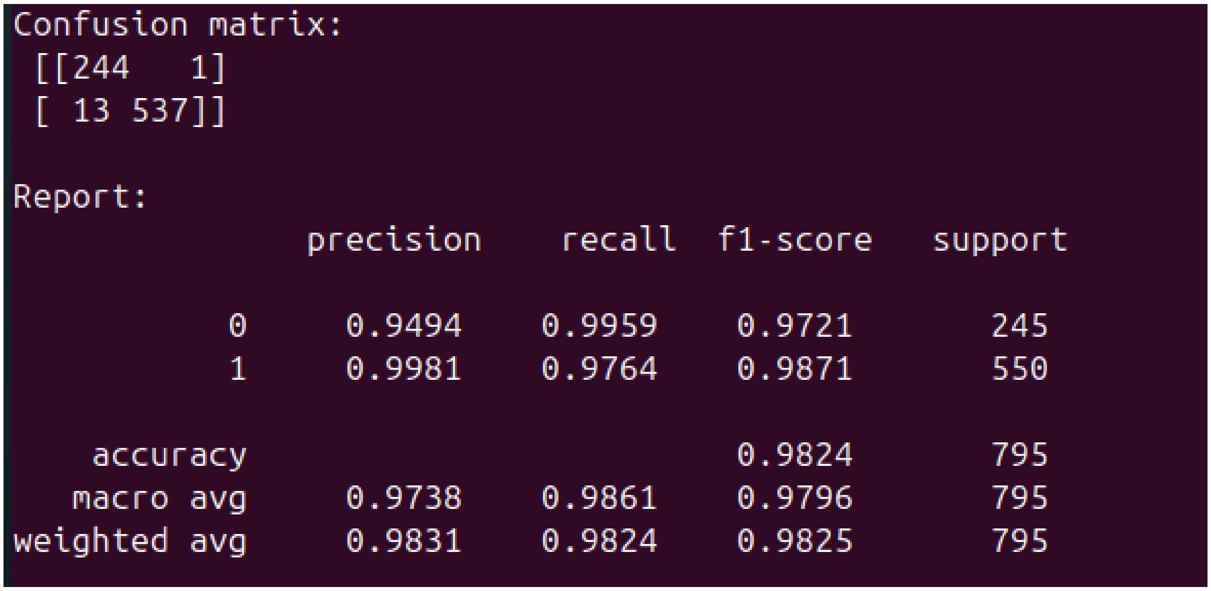
```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnckn1/arpfirewall$ ls -lh models results figs logs data | sed -n '1,120p'
data:
total 824K
-rw-r--r-- 1 nguyen-tuan-anh nguyen-tuan-anh 404K Dec 19 15:55 arp_dataset.csv
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 283K Dec 19 15:13 attack_arp_traffic.csv
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 126K Dec 19 15:43 benign_arp_traffic.csv

figs:
total 64K
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 63K Dec 19 18:42 dnn_training.png

logs:
total 44K
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 315 Dec 19 15:05 collector_attack.log
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 315 Dec 19 15:29 collector_benign.log
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 35K Dec 19 18:42 train_dnn.log

models:
total 100K
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh  95K Dec 19 18:42 arp_attack_detection_model.h5
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 1.1K Dec 19 18:42 arp_dnn_preprocess.joblib

results:
total 4.0K
-rw-rw-r-- 1 nguyen-tuan-anh nguyen-tuan-anh 3.9K Dec 19 18:42 dnn_history.csv
```

Artefact sinh ra sau huấn luyện:



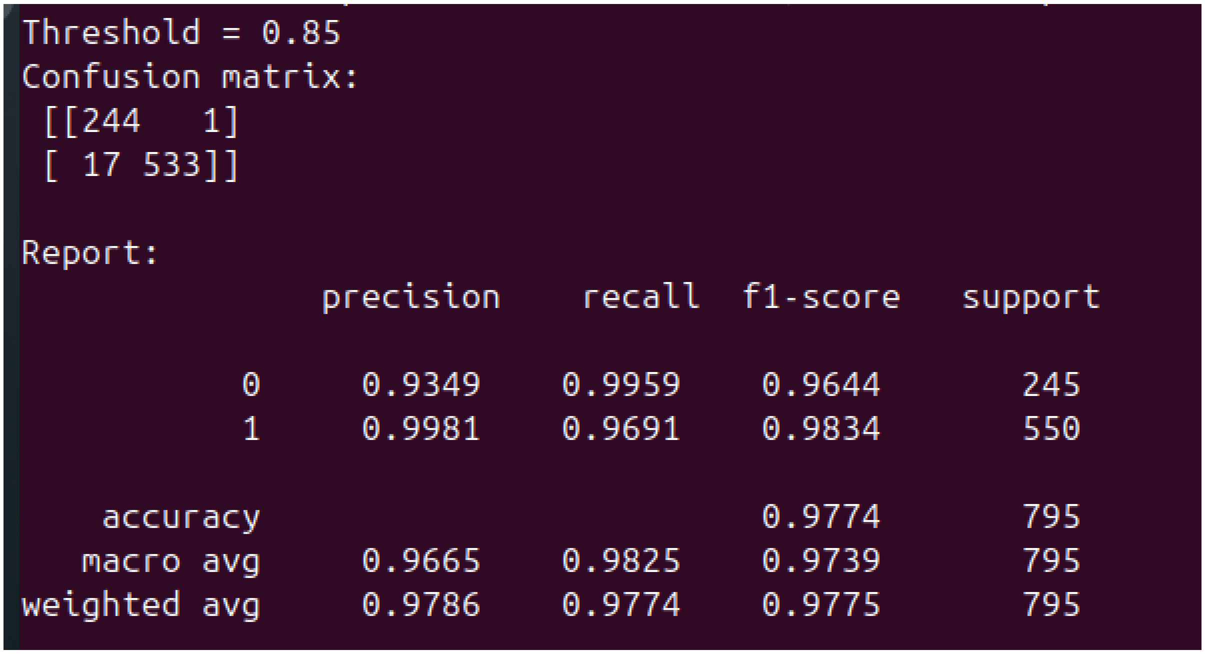
Đường cong loss/accuracy theo epoch của mô hình DNN

Ngưỡng 0.5

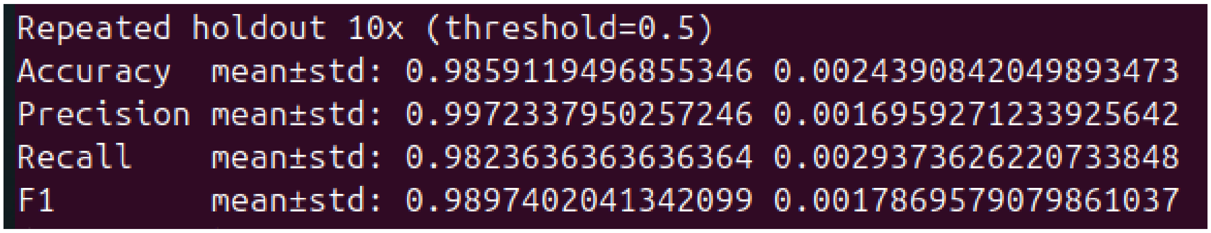


```
Confusion matrix:
 [[244   1]
 [ 13 537]]

Report:
               precision    recall  f1-score   support

           0     0.9494    0.9959    0.9721       245
           1     0.9981    0.9764    0.9871       550

    accuracy                         0.9824       795
   macro avg     0.9738    0.9861    0.9796       795
weighted avg     0.9831    0.9824    0.9825       795
```

threshold = 0.85

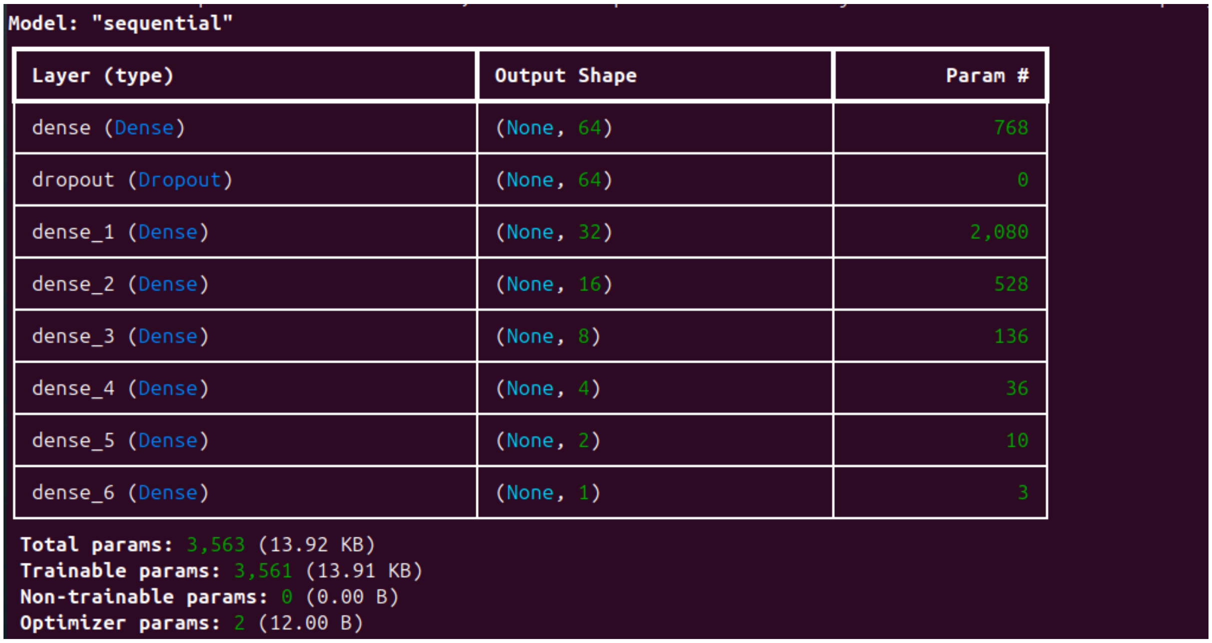


```
Threshold = 0.85
Confusion matrix:
 [[244   1]
 [ 17 533]]

Report:
               precision    recall  f1-score   support

           0      0.9349    0.9959    0.9644       245
           1      0.9981    0.9691    0.9834       550

    accuracy                          0.9774       795
   macro avg      0.9665    0.9825    0.9739       795
weighted avg      0.9786    0.9774    0.9775       795
```



```
Repeated holdout 10x (threshold=0.5)
Accuracy  mean±std: 0.9859119496855346 0.0024390842049893473
Precision mean±std: 0.9972337950257246 0.0016959271233925642
Recall    mean±std: 0.9823636363636364 0.0029373626220733848
F1        mean±std: 0.9897402041342099 0.0017869579079861037
```

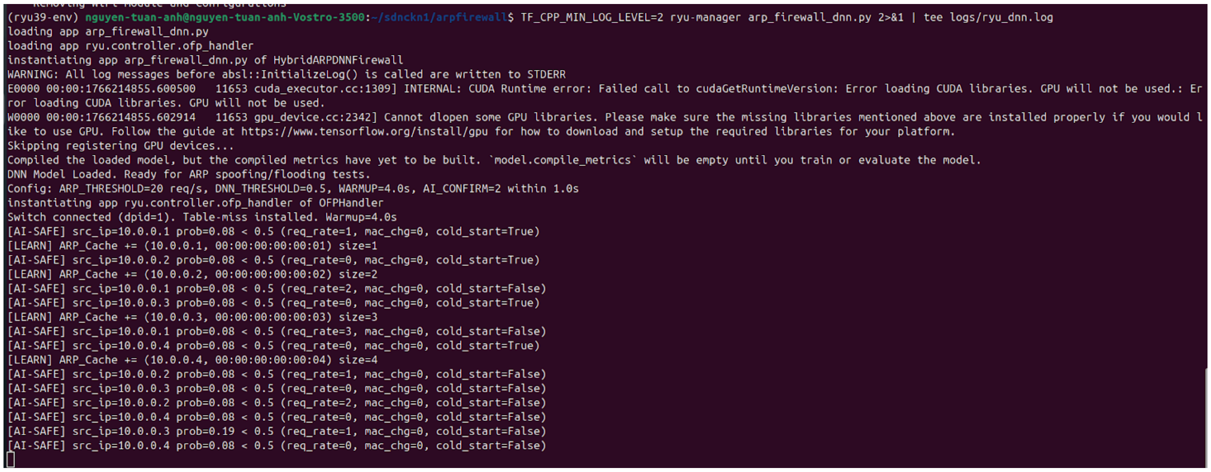
```
python3 - <<'PY'
from tensorflow.keras.models import load_model
m = load_model("models/arp_attack_detection_model.h5")
m.summary()
PY
```



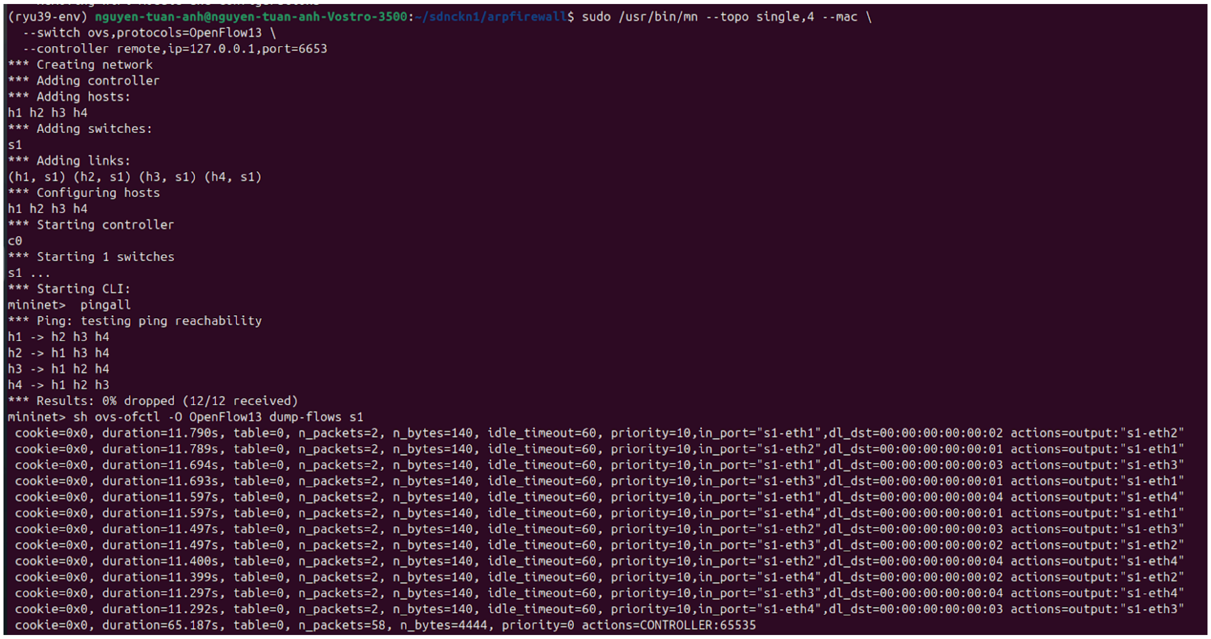
```
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                          ┃ Output Shape                ┃         Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
│ dense (Dense)                         │ (None, 64)                  │             768 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dropout (Dropout)                     │ (None, 64)                  │               0 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_1 (Dense)                       │ (None, 32)                  │           2,080 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_2 (Dense)                       │ (None, 16)                  │             528 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_3 (Dense)                       │ (None, 8)                   │             136 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_4 (Dense)                       │ (None, 4)                   │              36 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_5 (Dense)                       │ (None, 2)                   │              10 │
├───────────────────────────────────────┼─────────────────────────────┼─────────────────┤
│ dense_6 (Dense)                       │ (None, 1)                   │               3 │
└───────────────────────────────────────┴─────────────────────────────┴─────────────────┘
 Total params: 3,563 (13.92 KB)
 Trainable params: 3,561 (13.91 KB)
 Non-trainable params: 0 (0.00 B)
 Optimizer params: 2 (12.00 B)
```

```
python3 - <<'PY'
from tensorflow.keras.models import load_model
from tensorflow.keras.utils import plot_model

m = load_model("models/arp_attack_detection_model.h5")
plot_model(m, to_file="figs/dnn_arch.png", show_shapes=True,
show_layer_names=True)
print("[OK] Saved figs/dnn_arch.png")
PY
```

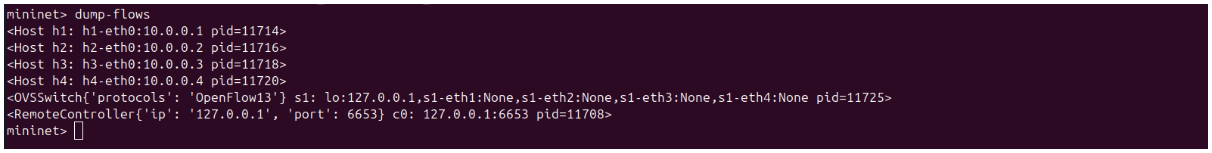
để ra hình ảnh



```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnckn1/arpfirewall$ TF_CPP_MIN_LOG_LEVEL=2 ryu-manager arp_firewall_dnn.py 2>&1 | tee logs/ryu_dnn.log
loading app arp_firewall_dnn.py
loading app ryu.controller.ofp_handler
instantiating app arp_firewall_dnn.py of HybridARPDNNFirewall
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1766214855.600500   11653 cuda_executor.cc:1309] INTERNAL: CUDA Runtime error: Failed call to cudaGetRuntimeVersion: Error loading CUDA libraries. GPU will not be used.: Er
ror loading CUDA libraries. GPU will not be used.
W0000 00:00:1766214855.602914   11653 gpu_device.cc:2342] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
DNN Model Loaded. Ready for ARP spoofing/flooding tests.
Config: ARP_THRESHOLD=20 req/s, DNN_THRESHOLD=0.5, WARMUP=4.0s, AI_CONFIRM=2 within 1.0s
instantiating app ryu.controller.ofp_handler of OFPHandler
Switch connected (dpid=1). Table-miss installed. Warmup=4.0s
[AI-SAFE] src_ip=10.0.0.1 prob=0.08 < 0.5 (req_rate=1, mac_chg=0, cold_start=True)
[LEARN] ARP_Cache += (10.0.0.1, 00:00:00:00:00:01) size=1
[AI-SAFE] src_ip=10.0.0.2 prob=0.08 < 0.5 (req_rate=0, mac_chg=0, cold_start=True)
[LEARN] ARP_Cache += (10.0.0.2, 00:00:00:00:00:02) size=2
[AI-SAFE] src_ip=10.0.0.1 prob=0.08 < 0.5 (req_rate=2, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.3 prob=0.08 < 0.5 (req_rate=0, mac_chg=0, cold_start=True)
[LEARN] ARP_Cache += (10.0.0.3, 00:00:00:00:00:03) size=3
[AI-SAFE] src_ip=10.0.0.1 prob=0.08 < 0.5 (req_rate=3, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.4 prob=0.08 < 0.5 (req_rate=0, mac_chg=0, cold_start=True)
[LEARN] ARP_Cache += (10.0.0.4, 00:00:00:00:00:04) size=4
[AI-SAFE] src_ip=10.0.0.2 prob=0.08 < 0.5 (req_rate=1, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.3 prob=0.08 < 0.5 (req_rate=0, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.2 prob=0.08 < 0.5 (req_rate=2, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.4 prob=0.08 < 0.5 (req_rate=0, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.3 prob=0.19 < 0.5 (req_rate=1, mac_chg=0, cold_start=False)
[AI-SAFE] src_ip=10.0.0.4 prob=0.08 < 0.5 (req_rate=0, mac_chg=0, cold_start=False)
```



```
(ryu39-env) nguyen-tuan-anh@nguyen-tuan-anh-Vostro-3500:~/sdnckn1/arpfirewall$ sudo /usr/bin/mn --topo single,4 --mac \
  --switch ovs,protocols=OpenFlow13 \
  --controller remote,ip=127.0.0.1,port=6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>  pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> sh ovs-ofctl -O OpenFlow13 dump-flows s1
 cookie=0x0, duration=11.790s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth1",dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=11.789s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth2",dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=11.694s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth1",dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=11.693s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth3",dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=11.597s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth1",dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=11.597s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth4",dl_dst=00:00:00:00:00:01 actions=output:"s1-eth1"
 cookie=0x0, duration=11.497s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth2",dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=11.497s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth3",dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=11.400s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth2",dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=11.399s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth4",dl_dst=00:00:00:00:00:02 actions=output:"s1-eth2"
 cookie=0x0, duration=11.297s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth3",dl_dst=00:00:00:00:00:04 actions=output:"s1-eth4"
 cookie=0x0, duration=11.292s, table=0, n_packets=2, n_bytes=140, idle_timeout=60, priority=10,in_port="s1-eth4",dl_dst=00:00:00:00:00:03 actions=output:"s1-eth3"
 cookie=0x0, duration=65.187s, table=0, n_packets=58, n_bytes=4444, priority=0 actions=CONTROLLER:65535
```



```
mininet> dump-flows
<Host h1: h1-eth0:10.0.0.1 pid=11714>
<Host h2: h2-eth0:10.0.0.2 pid=11716>
<Host h3: h3-eth0:10.0.0.3 pid=11718>
<Host h4: h4-eth0:10.0.0.4 pid=11720>
<OVSSwitch{'protocols': 'OpenFlow13'} s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None pid=11725>
<RemoteController{'ip': '127.0.0.1', 'port': 6653} c0: 127.0.0.1:6653 pid=11708>
mininet> 
```