实验报告

1. 实验名称

Lab1: Switchyard & Mininet

2. 实验目的

学习 Mininet、Switchyard 等

3. 实验内容

Step1 修改 mininet 拓扑，删除 server2。
删除 server2 对应的代码即可

```
nodes = {
    "server1": {
        "mac": "10:00:00:00:00:{:02x}",
        "ip": "192.168.100.1/24"
    },
    "server2": {
        "mac": "20:00:00:00:00:{:02x}",
        "ip": "192.168.100.2/24"
    },
    "client": {
        "mac": "30:00:00:00:00:{:02x}",
        "ip": "192.168.100.3/24"
    },
    "hub": {
        "mac": "40:00:00:00:00:{:02x}",
    }
}
```

Step2 统计进出 hub 的 packets 数量。
分别在接收和发送 packet 的位置处添加统计 cnt 的代码

```
def main(net: switchyard.llnetbase.LLNetBase):
    my_interfaces = net.interfaces()
    mymacs = [intf.ethaddr for intf in my_interfaces]
    incnt = 0
    outcnt = 0
    while True:
        try:
            _, fromIface, packet = net.recv_packet()
        except NoPackets:
            continue
        except Shutdown:
            break

        log_debug (f"In {net.name} received packet {packet} on {fromIface}")
        eth = packet.get_header(Ethernet)
        incnt += 1
        if eth is None:
            log_info("Received a non-Ethernet packet?!")
            return
        if eth.dst in mymacs:
            log_info("Received a packet intended for me")
        else:
            for intf in my_interfaces:
                if fromIface!= intf.name:
                    log_info (f"Flooding packet {packet} to {intf.name}")
                    net.send_packet(intf, packet)
                    outcnt += 1
        log_info (f"in:{incnt} out:{outcnt}")
    net.shutdown()
```

Step3 创建新测试用例。
新用例：构造一个从某主机（from 90:00:00:00:00:00:09）广播(to ff:ff:ff:ff:ff:ff)发送的消息，从 eth1 送到 hub，再从 eth0 和 eth2 转发出去

```
#test case 4
mypkt = new_packet(
    "90:00:00:00:00:09",
    "ff:ff:ff:ff:ff:ff",
    "172.16.42.9",
    "255.255.255.255"
)
s.expect(
    PacketInputEvent("eth1", testpkt, display=Ethernet),
    ("An Ethernet frame with a broadcast destination address "
     "should arrive on eth1")
)
s.expect(
    PacketOutputEvent("eth0", testpkt, "eth2", testpkt, display=Ethernet),
    ("The Ethernet frame with a broadcast destination address should be "
     "forwarded out ports eth0 and eth2")
)
```

Step4
在 mininet 中运行 switchyard
开启 hub，

```
*** Starting CLI:
mininet> xterm hub
mininet>
```

```
root@njucs-VirtualBox:~/workspace/lab-1-xkxkzzZ# cd ..
root@njucs-VirtualBox:~/workspace# source ./syenv/bin/activate
(syenv) root@njucs-VirtualBox:~/workspace# cd lab-1-xkxkzzZ/
(syenv) root@njucs-VirtualBox:~/workspace/lab-1-xkxkzzZ# swyard myhub.py
14:22:16 2024/03/16     INFO Saving iptables state and installing switchyard rul
es
14:22:16 2024/03/16     INFO Using network devices: hub-eth1 hub-eth0
```

然后 pingall

Step5
创造流量数据，用 wireshark 捕获数据包并保存
开启 hub，用 wireshark 监听 client，用 server1 来 ping client

```
*** Starting CLI:
mininet> xterm hub
mininet> client wireshark &
mininet> server1 ping -c 1 client
PING 192.168.100.3 (192.168.100.3) 56(84) bytes of data.
64 bytes from 192.168.100.3: icmp_seq=1 ttl=64 time=1081 ms

--- 192.168.100.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1081.296/1081.296/1081.296/0.000 ms
mininet>
```

4. 实验结果

Step1 成功删除 server2

```
*** Starting CLI:
mininet> nodes
available nodes are:
client hub server1
mininet> net
client client-eth0:hub-eth0
hub hub-eth0:client-eth0 hub-eth1:server1-eth0
server1 server1-eth0:hub-eth1
```

Step2　运行并 Log 结果如下，成功统计 in 和 out 数量

```
h2
12:23:20 2024/03/16     INFO in:1 out:2
12:23:20 2024/03/16     INFO Flooding packet Ethernet 20:00:00:00:00:01->30:00:00:00:00:0
2 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth1
12:23:20 2024/03/16     INFO Flooding packet Ethernet 20:00:00:00:00:01->30:00:00:00:00:0
2 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth2
12:23:20 2024/03/16     INFO in:2 out:4
12:23:20 2024/03/16     INFO Flooding packet Ethernet 30:00:00:00:00:02->20:00:00:00:00:0
1 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 data bytes) to eth0
12:23:20 2024/03/16     INFO Flooding packet Ethernet 30:00:00:00:00:02->20:00:00:00:00:0
1 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 data bytes) to eth2
12:23:20 2024/03/16     INFO in:3 out:6
12:23:20 2024/03/16     INFO Received a packet intended for me
12:23:20 2024/03/16     INFO in:4 out:6
```

Step3 成功通过创建的测试用例

```
    20:00:00:00:00:01 should arrive on eth1
6   Ethernet frame destined to 20:00:00:00:00:01 should be
    flooded outeth0 and eth2
7   An Ethernet frame should arrive on eth2 with destination
    address the same as eth2's MAC address
8   The hub should not do anything in response to a frame
    arriving with a destination address referring to the hub
    itself.
9   An Ethernet frame with a broadcast destination address
    should arrive on eth1
10  The Ethernet frame with a broadcast destination address
    should be forwarded out ports eth0 and eth2


All tests passed!
```

Step4

hub 可以正常工作，client 和 server1 可以互相 ping 到

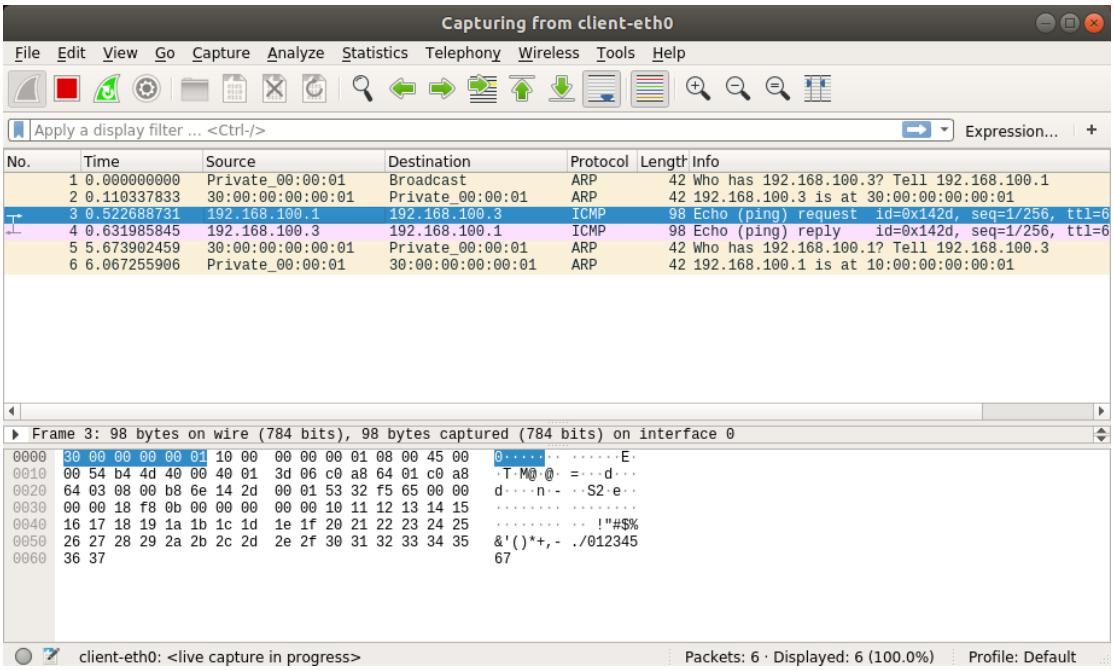
```
net.ipv6.conf.default.disable_ipv6 = 1
*** Starting controller

*** Starting 0 switches

*** Starting CLI:
mininet> xterm hub
mininet> pingall
*** Ping: testing ping reachability
client -> X server1
hub -> X X
server1 -> client X
*** Results: 66% dropped (2/6 received)
mininet>
```

"Node: hub"

```
13:26:45 2024/03/16     INFO in:4 out:4
13:26:45 2024/03/16     INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:0
0:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoRequest 4237 1
 (56 data bytes) to hub-eth0
13:26:45 2024/03/16     INFO in:5 out:5
13:26:46 2024/03/16     INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:0
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoReply 4237 1 (
56 data bytes) to hub-eth1
13:26:46 2024/03/16     INFO in:6 out:6
13:26:50 2024/03/16     INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:0
0:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 00:00:00:00:00:00:192.168.1
00.3 to hub-eth0
13:26:50 2024/03/16     INFO in:7 out:7
```

step5

server1（192.168.100.1），client（192.168.100.03）



在此数据包中，server1 先确定 client 位置，然后发送 ping 请求，然后收到 client 的 ping 回复

5. 核心代码

start_mininet.py

myhub.py

testcases/myhub_testscenario.py

6. 总结与感想

了解了 Mininet、Switchyard、wireshark 等的用法

内容可以自由变通，根据手册编写