



课程实验报告

实验名称 Switchyard & Mininet

课程名称 计算机网络

院 系 计算机科学与技术系

学 号 191220129

姓 名 邢尚禹

邮 箱 191220129@smail.nju.edu.cn

实验日期 2021 年 3 月

目录

1	实验目的	2
2	实验内容	2
3	实验过程及结果	2
3.1	Modify the Mininet topology	2
3.2	Modify the logic of a device	3
3.3	Modify the test scenario of a device	5
3.4	Run your device in Mininet	7
3.5	Capture using Wireshark	7
4	总结与感想	7

1 实验目的

- 学习 linux, python, git, vscode 等工具的基本使用；
- 学习使用 mininet 搭建模拟网络；
- 学习使用 wireshark 抓包；
- 学习使用 switchyard 实现硬件逻辑。

2 实验内容

1. 安装并配置 linux, python, git, vscode 等工具；
2. 了解 mininet, wireshark 和 switchyard 的使用方法；
3. 对指定的代码进行修改，以完成特定的功能。

3 实验过程及结果

task 1,2,3 此处略过，本节仅简述 task 4 的实验过程和结果。

3.1 Modify the Mininet topology

选择第一个实现：删除 server2，只需将 start_mininet.py 的第 30 至 33 行的 server2 注释即可。

```
1 nodes = {  
2     "server1": {  
3         "mac": "10:00:00:00:00:{:02x}",  
4         "ip": "192.168.100.1/24"  
5     },  
6     # "server2": {  
7     #     "mac": "20:00:00:00:00:{:02x}",  
8     #     "ip": "192.168.100.2/24"  
9     # },  
10    "client": {
```

```

11         "mac": "30:00:00:00:00:{:02x}",
12         "ip": "192.168.100.3/24"
13     },
14     "hub": {
15         "mac": "40:00:00:00:00:{:02x}",
16     }
17 }

```

修改后的结构为: server1 — hub — client

3.2 Modify the logic of a device

要实现接受和发送的包的数量统计,只需要添加全局变量 `in_count` 和 `out_count`,接受和发送时分别 +1 即可。在循环的最后,通过 `log_info` 函数格式化字符串输出。

```

1 while True:
2     try:
3         _, fromIface, packet = net.recv_packet()
4     except NoPackets:
5         continue
6     except Shutdown:
7         break
8     in_count += 1
9     log_debug (f"In {net.name} received packet {packet} on {
        fromIface}")
10    eth = packet.get_header(Ethernet)
11    if eth is None:
12        log_info("Received a non-Ethernet packet?!")
13        return
14    if eth.dst in mymacs:
15        log_info("Received a packet intended for me")
16    else:
17        for intf in my_interfaces:
18            if fromIface != intf.name:
19                log_info (f"Flooding packet {packet} to {intf.
                    name}")
20                net.send_packet(intf, packet)

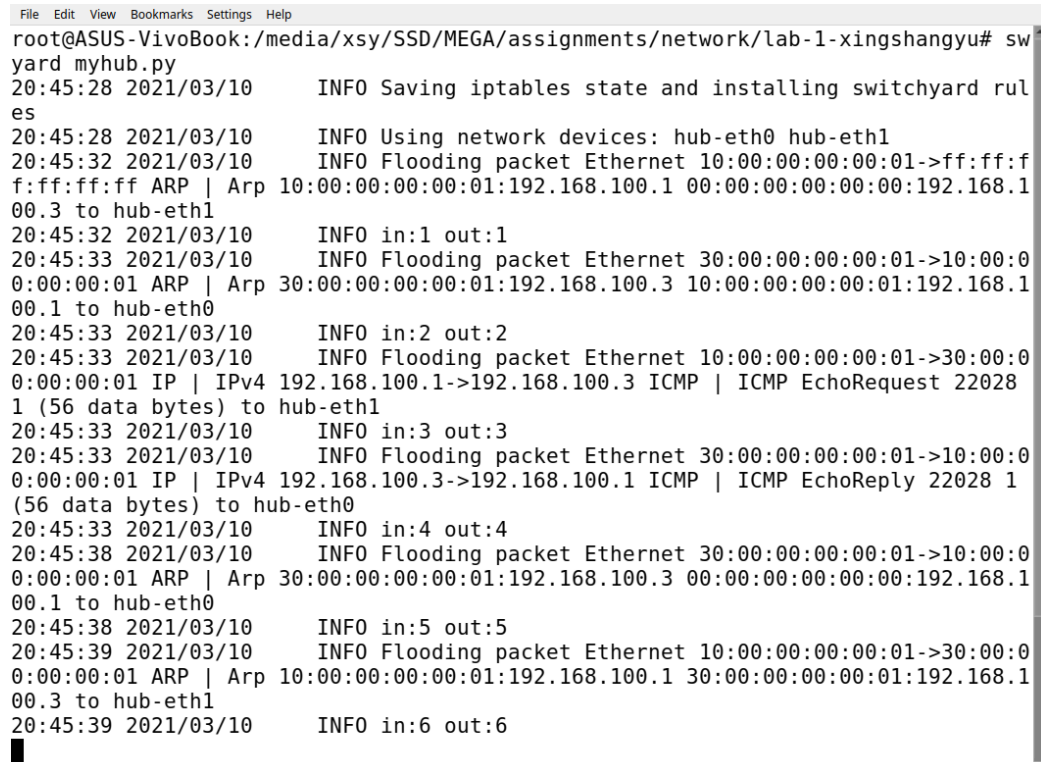
```

```

21         out_count += 1
22     log_info("in:%d out:%d" % (in_count, out_count))

```

在 mininet CLI 中执行 `server1 ping 192.168.100.3 -c 1` 命令，在 switchyard 中得到如下 log：



```

File Edit View Bookmarks Settings Help
root@ASUS-VivoBook:/media/xsy/SSD/MEGA/assignments/network/lab-1-xingshangyu# sw
yard myhub.py
20:45:28 2021/03/10      INFO Saving iptables state and installing switchyard rul
es
20:45:28 2021/03/10      INFO Using network devices: hub-eth0 hub-eth1
20:45:32 2021/03/10      INFO Flooding packet Ethernet 10:00:00:00:00:01->ff:ff:f
f:ff:ff:ff 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-eth1
20:45:32 2021/03/10      INFO in:1 out:1
20:45:33 2021/03/10      INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:0
0:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.100.3 10:00:00:00:00:01:192.168.1
00.1 to hub-eth0
20:45:33 2021/03/10      INFO in:2 out:2
20:45:33 2021/03/10      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 22028
1 (56 data bytes) to hub-eth1
20:45:33 2021/03/10      INFO in:3 out:3
20:45:33 2021/03/10      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 22028 1
(56 data bytes) to hub-eth0
20:45:33 2021/03/10      INFO in:4 out:4
20:45:38 2021/03/10      INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:0
0:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.100.3 00:00:00:00:00:00:192.168.1
00.1 to hub-eth0
20:45:38 2021/03/10      INFO in:5 out:5
20:45:39 2021/03/10      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 30:00:00:00:00:01:192.168.1
00.3 to hub-eth1
20:45:39 2021/03/10      INFO in:6 out:6

```

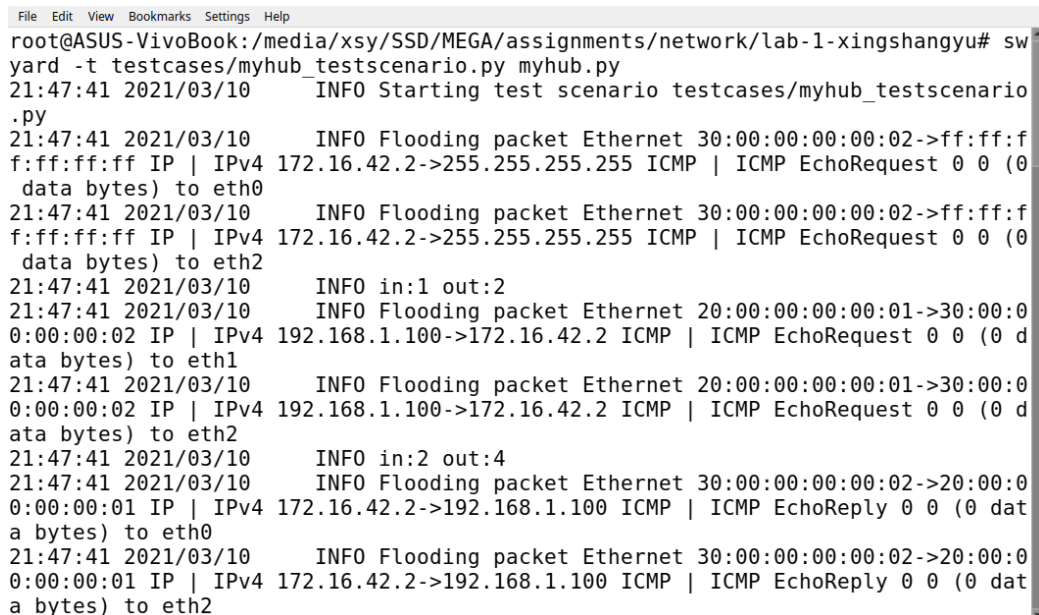
图 1: hublog

3.3 Modify the test scenario of a device

通过调用 `new_packet` 函数，添加一个测试：client 向 hub 发送一个 frame。根据 hub 的逻辑，不会出现任何包的转发。在最后一个 `expect` 语句前添加如下代码：

```
1 # my testcase: a frame from client to hub should result in
   nothing happening
2 testpkt = new_packet(
3     "30:00:00:00:00:02",
4     "10:00:00:00:00:01",
5     "192.168.1.100",
6     "172.16.42.2"
7 )
8 s.expect(
9     PacketInputEvent("eth0", testpkt, display=Ethernet),
10    ("A frame from client to hub should result in nothing
    happening.")
11 )
```

测试结果：



```
File Edit View Bookmarks Settings Help
root@ASUS-VivoBook:/media/xsy/SSD/MEGA/assignments/network/lab-1-xingshangyu# sw
yard -t testcases/myhub_testscenario.py myhub.py
21:47:41 2021/03/10      INFO Starting test scenario testcases/myhub_testscenario
.py
21:47:41 2021/03/10      INFO Flooding packet Ethernet 30:00:00:00:00:02->ff:ff:f
f:ff:ff:ff IP | IPv4 172.16.42.2->255.255.255.255 ICMP | ICMP EchoRequest 0 0 (0
data bytes) to eth0
21:47:41 2021/03/10      INFO Flooding packet Ethernet 30:00:00:00:00:02->ff:ff:f
f:ff:ff:ff IP | IPv4 172.16.42.2->255.255.255.255 ICMP | ICMP EchoRequest 0 0 (0
data bytes) to eth2
21:47:41 2021/03/10      INFO in:1 out:2
21:47:41 2021/03/10      INFO Flooding packet Ethernet 20:00:00:00:00:01->30:00:0
0:00:00:02 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 d
ata bytes) to eth1
21:47:41 2021/03/10      INFO Flooding packet Ethernet 20:00:00:00:00:01->30:00:0
0:00:00:02 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 d
ata bytes) to eth2
21:47:41 2021/03/10      INFO in:2 out:4
21:47:41 2021/03/10      INFO Flooding packet Ethernet 30:00:00:00:00:02->20:00:0
0:00:00:01 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 dat
a bytes) to eth0
21:47:41 2021/03/10      INFO Flooding packet Ethernet 30:00:00:00:00:02->20:00:0
0:00:00:01 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 dat
a bytes) to eth2
```

图 2: hubtest_1

```
File Edit View Bookmarks Settings Help
21:47:41 2021/03/10 INFO in:3 out:6
21:47:41 2021/03/10 INFO Received a packet intended for me
21:47:41 2021/03/10 INFO in:4 out:6
21:47:41 2021/03/10 INFO Received a packet intended for me
21:47:41 2021/03/10 INFO in:5 out:6

Results for test scenario hub tests: 9 passed, 0 failed, 0 pending

Passed:
1 An Ethernet frame with a broadcast destination address
  should arrive on eth1
2 The Ethernet frame with a broadcast destination address
  should be forwarded out ports eth0 and eth2
3 An Ethernet frame from 20:00:00:00:00:01 to
  30:00:00:00:00:02 should arrive on eth0
4 Ethernet frame destined for 30:00:00:00:00:02 should be
  flooded out eth1 and eth2
5 An Ethernet frame from 30:00:00:00:00:02 to
  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
```

图 3: hubtest_2

```
File Edit View Bookmarks Settings Help
  should arrive on eth1
2 The Ethernet frame with a broadcast destination address
  should be forwarded out ports eth0 and eth2
3 An Ethernet frame from 20:00:00:00:00:01 to
  30:00:00:00:00:02 should arrive on eth0
4 Ethernet frame destined for 30:00:00:00:00:02 should be
  flooded out eth1 and eth2
5 An Ethernet frame from 30:00:00:00:00:02 to
  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 A frame from client to hub should result in nothing
  happening.
9 The hub should not do anything in response to a frame
  arriving with a destination address referring to the hub
  itself.

All tests passed!

root@ASUS-VivoBook:/media/xsy/SSD/MEGA/assignments/network/lab-1-xingshangyu#
```

图 4: hubtest_3

3.4 Run your device in Mininet

运行步骤:

1. 运行 mininet: `sudo python3 start_mininet.py`
2. 运行 hub switchyard: `hub xterm &`, 在 xterm 中运行 switchyard: `swyard myhub.py`
3. 在 mininet 中生成一些 traffic, 这里让 server1 ping client: `server1 ping 192.168.100.3 -c 1`

3.5 Capture using Wireshark

运行上述前 2 步后, 运行 wireshark: `client wireshark`。然后让 client ping server 得到 6 个包, 记录见文件 `capture.pcapng`。捕获得的包为:

1. client 询问 server 的 mac address;
2. 回应 server 的 mac address;
3. client 向 server 发送 ping request;
4. server 询问 client 的 mac address;
5. 回应 server 的 mac address;
6. server 向 client 发送 ping reply。

4 总结与感想

- 做实验前要先了解相关工具的使用, 可以事半功倍;
- 英文阅读能力很重要。