

## 南京大学本科生实验报告

课程名称：计算机网络 任课教师：田臣/李文中 助教：

学院：计算机科学与技术系

专业（方向）：计算机科学与技术

学号：191840186

姓名：彭昕昂

Email: 2501087332@qq.com

开始/完成日期：2021/3/10~2021/3/24

### 1. 实验名称

Lab1:Switchyard&mininet

### 2. 实验目的

熟悉 switchyard 的框架，学习 wireshark 和 mininet 的基本用法

### 3. 实验内容

Step1 Modify the Mininet topology

要求在网络拓扑中删去 server2，只需将 start\_mininet.py 中与 server2 相关的代码注释掉

```

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 Modify the logic of a device

要求统计 packet 在 hub 中的进出情况，只需在 main 函数中增加两个变量 icnt (in count)，ocnt (out count) 加入循环并在需要的位置计数即可

```

def main(net: switchyard.llnetbase.LLNetBase):
    my_interfaces = net.interfaces()
    mymacs = [intf.ethaddr for intf in my_interfaces]

    icnt=0
    ocnt=0

    while True:
        try:
            _, fromIface, packet = net.recv_packet()
        except NoPackets:
            continue
        except Shutdown:
            return

        icnt+=1

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

```

### Step3 Modify the test scenario of a device

要求构建一个新的测试用例，只需仿照前文测试用例并修改部分参数即可

```
test = new_packet("20:00:00:00:00:01", "ff:ff:ff:ff:ff:ff", "192.168.1.100", '172.16.42.2')
s.expect(PacketInputEvent("eth1", test, display=Ethernet), "An Ethernet frame with a broadcast destination address should arrive on eth1")
s.expect(PacketOutputEvent("eth0", test, "eth2", test, display=Ethernet), "The Ethernet frame with a broadcast destination address should be forwarded out ports eth0 &
```

### Step4 Run your device in Mininet

过程:

`source ./syenv/bin/activate` 打开虚拟环境

`sudo python lab_1/start_mininet.py` 运行 mininet

`xterm hub` 启动 xterm，并运行虚拟环境

`swyard lab_1/myhub.py` 开始运行我们构建的网络拓扑

接着可通过 `pingall` 构造一些流量以验证自己构造的网络是否正确

### Step5 Capture using Wireshark

继 Step4, 在 mininet 中输入命令行 `client wireshark &` 即可在 client 上启动 wireshark, 选择对应网卡, 并通过 `server1 ping -c1 client`, `pingall` 等方法构造一些流量并在 wireshark 中抓包

## 4. 实验结果

## Step1

```
*** Starting CLI:
mininet> nodes
available nodes are:
client hub server1
mininet>
```

## Step2

```
root@njucs-VirtualBox:~/switchyard# source ./syenv/bin/activate
(syenv) root@njucs-VirtualBox:~/switchyard# swyard examples/myhub.py
19:19:54 2021/03/23 INFO Saving iptables state and installing switchyard rules
19:19:54 2021/03/23 INFO Using network devices: hub-eth0 hub-eth1
19:20:06 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:00:01->ff:ff:ff:ff:ff:ff ARP | Arp 30:00:00:00:00:01:192.168.100.3 00:00:00:00:00:00:192.168.100.1 to hub-eth1
19:20:06 2021/03/23 INFO in:1 out:1
19:20:06 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:00:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 30:00:00:00:00:01:192.168.100.3 to hub-eth0
19:20:06 2021/03/23 INFO in:2 out:2
19:20:06 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:00:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoRequest 3137 1 (56 data bytes) to hub-eth1
19:20:06 2021/03/23 INFO in:3 out:3
19:20:07 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoReply 3137 1 (56 data bytes) to hub-eth0
19:20:07 2021/03/23 INFO in:4 out:4
19:20:07 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoRequest 3140 1 (56 data bytes) to hub-eth0
19:20:07 2021/03/23 INFO in:5 out:5
19:20:07 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:00:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoReply 3140 1 (56 data bytes) to hub-eth1
19:20:07 2021/03/23 INFO in:6 out:6
19:20:12 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:00:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 00:00:00:00:00:00:192.168.100.3 to hub-eth0
19:20:12 2021/03/23 INFO in:7 out:7
19:20:12 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:00:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.100.3 10:00:00:00:00:01:192.168.100.1 to hub-eth1
19:20:12 2021/03/23 INFO in:8 out:8
```

## Step3

```

[sysenv] njucs@njucs-VirtualBox:~/switchyard$ swyard -t lab_1/testcases/myhub.testscenario.py lab_1/myhub.py
18:13:39 2021/03/23 INFO Starting test scenario lab_1/testcases/myhub.testscenario.py
18:13:39 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:02->ff:ff:ff:ff:ff:ff IP | IPv4 172.16.42.2->255.255.255.255 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth0
18:13:39 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:02->ff:ff:ff:ff:ff:ff IP | IPv4 172.16.42.2->255.255.255.255 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth2
18:13:39 2021/03/23 INFO In1 out1
18:13:39 2021/03/23 INFO Flooding packet Ethernet 20:00:00:00:01->30:00:00:00:00:02 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth1
18:13:39 2021/03/23 INFO Flooding packet Ethernet 20:00:00:00:01->30:00:00:00:00:02 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth2
18:13:39 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:02->20:00:00:00:00:01 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 data bytes) to eth0
18:13:39 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:02->20:00:00:00:00:01 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 data bytes) to eth2
18:13:39 2021/03/23 INFO In3 out3
18:13:39 2021/03/23 INFO Received a packet intended for me
18:13:39 2021/03/23 INFO In4 out3
18:13:40 2021/03/23 INFO Flooding packet Ethernet 20:00:00:00:01->ff:ff:ff:ff:ff:ff IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth0
18:13:40 2021/03/23 INFO Flooding packet Ethernet 20:00:00:00:01->ff:ff:ff:ff:ff:ff IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth2
18:13:40 2021/03/23 INFO In5 out4

Results for test scenario hub tests: 10 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 out eth0 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!

[sysenv] njucs@njucs-VirtualBox:~/switchyard$

```

## Step4

```

njucs@njucs-VirtualBox:~$ cd switchyard
njucs@njucs-VirtualBox:~/switchyard$ sudo python examples/start_mininet.py
[sudo] password for njucs:
*** Creating network
*** Adding hosts:
client hub server1
*** Adding switches:

*** Adding links:
(10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (client, hub) (10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (server1, hub)
*** Configuring hosts
client hub server1
client client-eth0 30:00:00:00:00:01
server1 server1-eth0 10:00:00:00:00:01
hub hub-eth0 40:00:00:00:00:01
hub hub-eth1 40:00:00:00:00:02
*** client : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** client : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
net.ipv6.conf.default.disable_ipv6 = 1
*** hub : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** hub : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
net.ipv6.conf.default.disable_ipv6 = 1
*** server1 : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** server1 : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
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 X
hub -> X X
server1 -> X X
*** Results: 100% dropped (0/6 received)
mininet> pingall
*** Ping: testing ping reachability
client -> X server1
hub -> X X
server1 -> client X
*** Results: 66% dropped (2/6 received)
mininet>

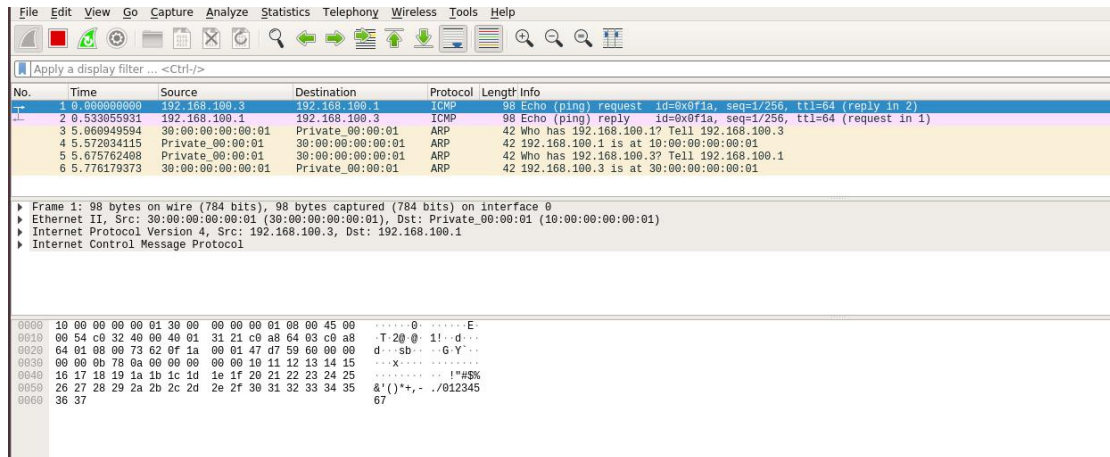
```

```

"Node: hub"
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoRequest 3804 1
(56 data bytes) to hub-eth1
19:54:28 2021/03/23 INFO In3 out:3
19:54:29 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoReply 3804 1 (
56 data bytes) to hub-eth0
19:54:29 2021/03/23 INFO In4 out:4
19:54:29 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoRequest 3807 1
(56 data bytes) to hub-eth0
19:54:29 2021/03/23 INFO In5 out:5
19:54:29 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoReply 3807 1 (
56 data bytes) to hub-eth1
19:54:29 2021/03/23 INFO In5 out:6
19:54:34 2021/03/23 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:00:01 ARP | Rsp 10:00:00:00:00:01:192.168.100.1 00:00:00:00:00:00:192.168.1
00.3 to hub-eth0
19:54:34 2021/03/23 INFO In7 out:7
19:54:34 2021/03/23 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:00:01 ARP | Rsp 30:00:00:00:00:01:192.168.100.3 10:00:00:00:00:01:192.168.1
00.1 to hub-eth1
19:54:34 2021/03/23 INFO In8 out:8

```

## Step5



## 5. 核心代码

见实验内容中每一步的代码截图

## 6. 总结与感想

本次实验了解了 switchyard 以及 mininet 和 wireshark 的基本用法，对课程内容有了初步了解，但是对代码执行原理以及 python 语言还不够熟练，希望能继续努力解决问题。