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：方毓楚、郑浩、陈伟等（排名不分先后）

学院	地理与海洋科学学院	专业	地理信息科学
学号	191830173	姓名	徐嵩
Email	song.xv@outlook.com	开始/完成日期	20220305

1. 实验名称

Lab 1: Switchyard & Mininet

2. 实验目的

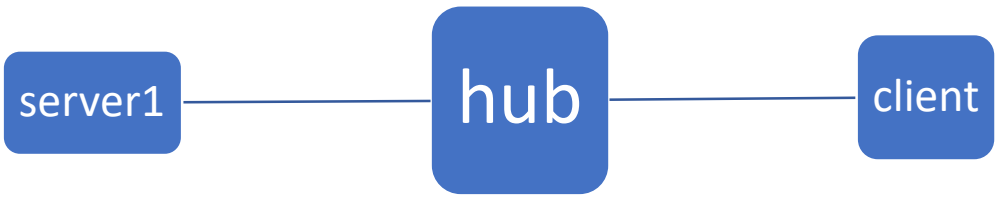
熟悉 switchyard 和 mininet 等工具的使用，构造一个集线器。

3. 实验内容

1. Modify the Mininet topology
2. Modify the logic of a device
3. Modify the test scenario of a device
4. Run your device in Mininet
5. Capture using Wireshark

4. 实验结果

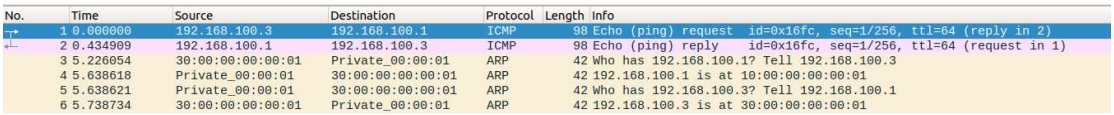
1. 删除 server2 端后，拓扑结构如图：



2. 修改 hub 代码后的运行日志：

```
10:02:01 2022/03/05     INFO Saving iptables state and installing switchyard ru
es
10:02:01 2022/03/05     INFO Using network devices: hub-eth1 hub-eth0
10:02:12 2022/03/05     INFO Flooding packet Ethernet 30:00:00:00:00:01->ff:ff:
f:ff:ff:ff ARP | Arp 30:00:00:00:00:01:192.168.100.3 00:00:00:00:00:00:192.168.
00.1 to hub-eth1
10:02:12 2022/03/05     INFO in: 1 out: 1
10:02:13 2022/03/05     INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:
0:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 30:00:00:00:00:01:192.168.
00.3 to hub-eth0
10:02:13 2022/03/05     INFO in: 2 out: 2
10:02:13 2022/03/05     INFO Flooding packet Ethernet 30:00:00:00:00:01->10:00:
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoRequest 5702
 (56 data bytes) to hub-eth1
10:02:13 2022/03/05     INFO in: 3 out: 3
10:02:13 2022/03/05     INFO Flooding packet Ethernet 10:00:00:00:00:01->30:00:
0:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoReply 5702 1
56 data bytes) to hub-eth0
10:02:13 2022/03/05     INFO in: 4 out: 4
```

3. 创建了一个需要被广播的 testcase
   使用命令 source ./syenv/bin/activate 进入 syenv python 虚拟环境
   使用命令 swyard -t myhub_testscenario.py myhub.py 测试 myhub.py 实现
4. 运行 Mininet：
   使用命令 sudo python start_mininet.py 进入 mininet
   使用命令 hub xterm 打开 hub 的终端
   在 xterm 中使用命令 source ./sysenv/bin/activate 启动 syenv python 虚拟环境
   在 xterm 中使用命令 swyard myhub.py 运行修改后的 hub 程序
   使用命令 hub wireshark & 监视 hub 的流量
   使用命令 client ping -c 1 server1 使端设备 client ping 端设备 server1
5. 用 wireshark 抓包
   在特权模式下将 pcap 或 pcang 文件保存
   使用命令 sudo chown $USER:$USER <filename> 切换文件所有者
   抓包内容：
   Client 首先通过 ARP 协议寻找并记录 192.168.100.1 的 MAC 地址
   然后发送 ICMP 报文进行 ping 测试



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.3	192.168.100.1	ICMP	98	Echo (ping) request id=0x16fc, seq=1/256, ttl=64 (reply in 2)
2	0.434909	192.168.100.1	192.168.100.3	ICMP	98	Echo (ping) reply id=0x16fc, seq=1/256, ttl=64 (request in 1)
3	5.226054	30:00:00:00:00:01	Private_00:00:01	ARP	42	Who has 192.168.100.1? Tell 192.168.100.3
4	5.638618	Private_00:00:01	30:00:00:00:00:01	ARP	42	192.168.100.1 is at 10:00:00:00:00:01
5	5.638621	Private_00:00:01	30:00:00:00:00:01	ARP	42	Who has 192.168.100.3? Tell 192.168.100.1
6	5.738734	30:00:00:00:00:01	Private_00:00:01	ARP	42	192.168.100.3 is at 30:00:00:00:00:01

```
▸ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
▸ Ethernet II, Src: 30:00:00:00:00:01 (30:00:00:00:00:01), Dst: Private_00:00:01 (10:00:00:00:00:01)
▸ Internet Protocol Version 4, Src: 192.168.100.3, Dst: 192.168.100.1
▸ Internet Control Message Protocol
```

5. 核心代码

```
1. 1. step 1: modify myhub.py
2.
3. ```
4. nodes = {
5.     "server1": {
6.         "mac": "10:00:00:00:00:{:02x}",
```

```
        "ip": "192.168.100.1/24"
    },
    "client": {
        "mac": "30:00:00:00:00:{:02x}",
        "ip": "192.168.100.3/24"
    },
    "hub": {
        "mac": "40:00:00:00:00:{:02x}",
    }
}
```

2. step 2: modify the logic of myhub.py

```
packets_in = 0
packets_out = 0

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

    log_info (f"In {net.name} received packet {packet} on {fromIface}")
    packets_in += 1
    eth = packet.get_header(Ethernet)
    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)
                packets_out += 1
                log_info (f"in: {packets_in} out: {packets_out}")
```

````
3. step 3: add my testcase: a broadcast frame should be sent out all ports except ingress

```
testpkt = new_packet(
    "40:00:00:00:00:02",
    "ff:ff:ff:ff:ff:ff",
    "192.168.0.1",
    "255.255.255.255"
)
s.expect(
    PacketInputEvent("eth0", testpkt, display=Ethernet),
    ("my testcase: a broadcast frame should arrive on eth0")
)
s.expect(
    PacketOutputEvent("eth1", testpkt, "eth2", testpkt, display=Ethernet),
    ("my testcase: a broadcast frame should be sent out other ports")
)
```
````

6. 总结与感想

了解了集线器是工作在物理层的设备，工作方式是将收到的目标不是自身的所有包广播，对目标是自身的包不做处理。