南京大学本科生实验报告

课程名称：**计算机网络**　　　　　　任课教师：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	**221220143**	姓名	**陈雨檬**
Email	**2843438797@qq.com**	开始/完成日期	**2024.3.6/2024.3.14**

1. 实验名称

Lab1: Switchyard & Mininet

2. 实验目的

熟悉操作环境，进行一些简单的例子熟悉软件与指令操作。

3. 实验内容

Step1：Modify the Mininet topology

How: 构建 nodes 时删除 server2 相关的节点代码即可。

Step2：Modify the logic of a device

How: 在 hub 接收到 packet 时为变量 numin 增加 1；如果 packet 不是传递给 hub 而是传递给 hub 的接口设备为变量 numout 增加 1，再打印出来即可。

Step3：Modify the test scenario of a device

How: 添加自定义一个样例，属于 testcase1 情况，修改部分参数供测试。

Step4：Run your device in Mininet

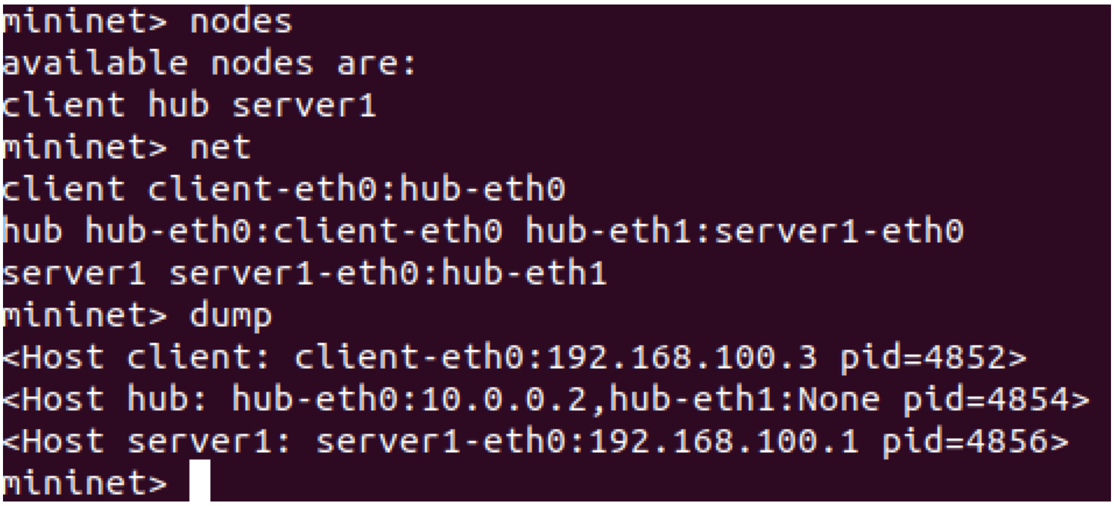
How: 类似于 Switchyard 节中内容，通过 sudo python start_mininet.py 运行当前修改过的自定义拓补，再通过 xterm hub 进入 hub 主机，在 hub 主机中通过 swyard myhub.py 运行当前修改过的 hub 程序，再通过 pingall 测试新的拓补网络。

Step5：Capture using Wireshark

How: 运行构建的拓补后，通过 client wireshark &指令进入客户主机端，再通过 client ping -c1 server1 发出流量包，即可在 wireshark 中捕获流量。

4. 实验结果

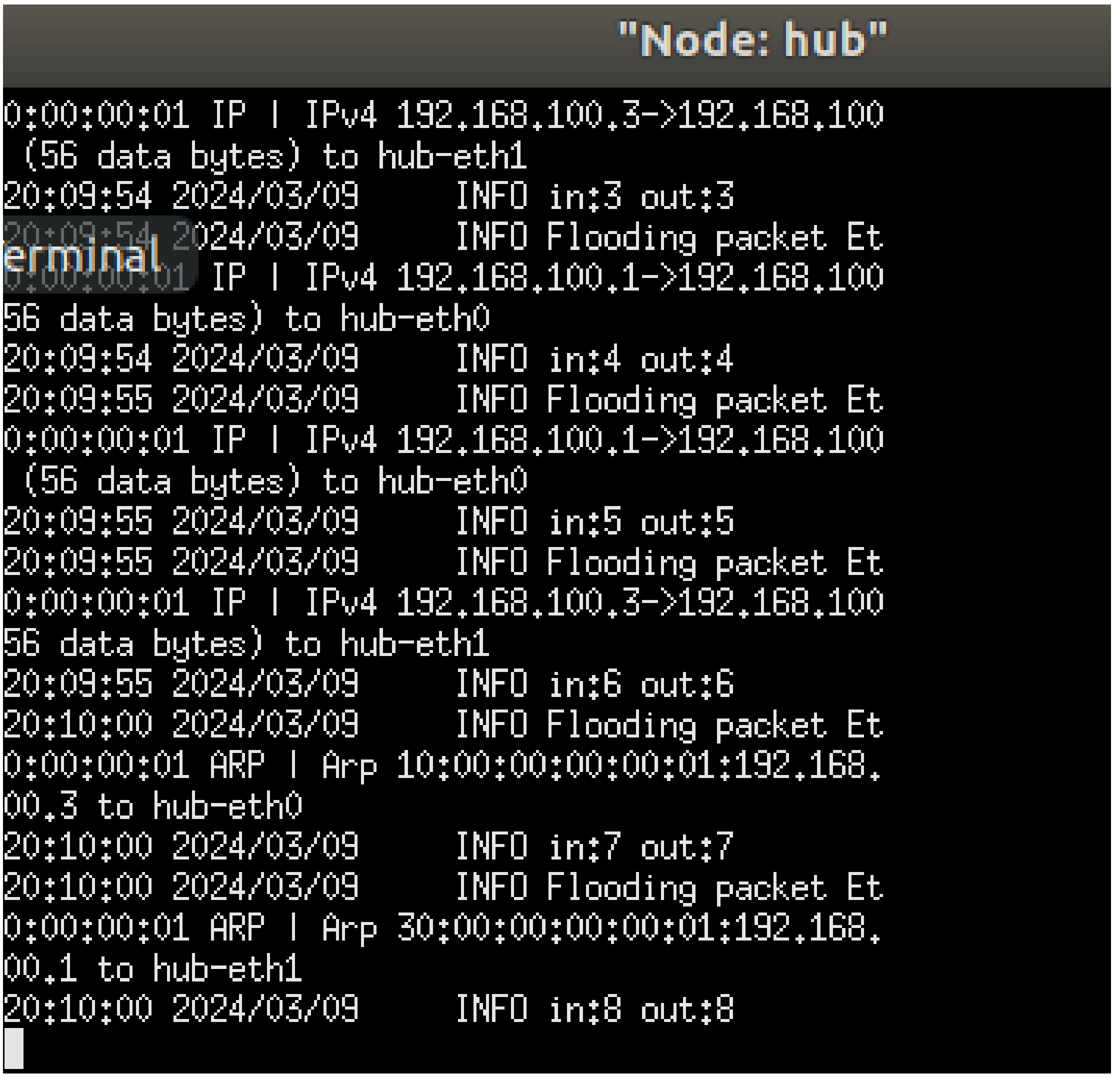
Step1：删除 server2 节点后的网络信息如下



```
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
mininet> dump
<Host client: client-eth0:192.168.100.3 pid=4852>
<Host hub: hub-eth0:10.0.0.2,hub-eth1:None pid=4854>
<Host server1: server1-eth0:192.168.100.1 pid=4856>
mininet> 
```

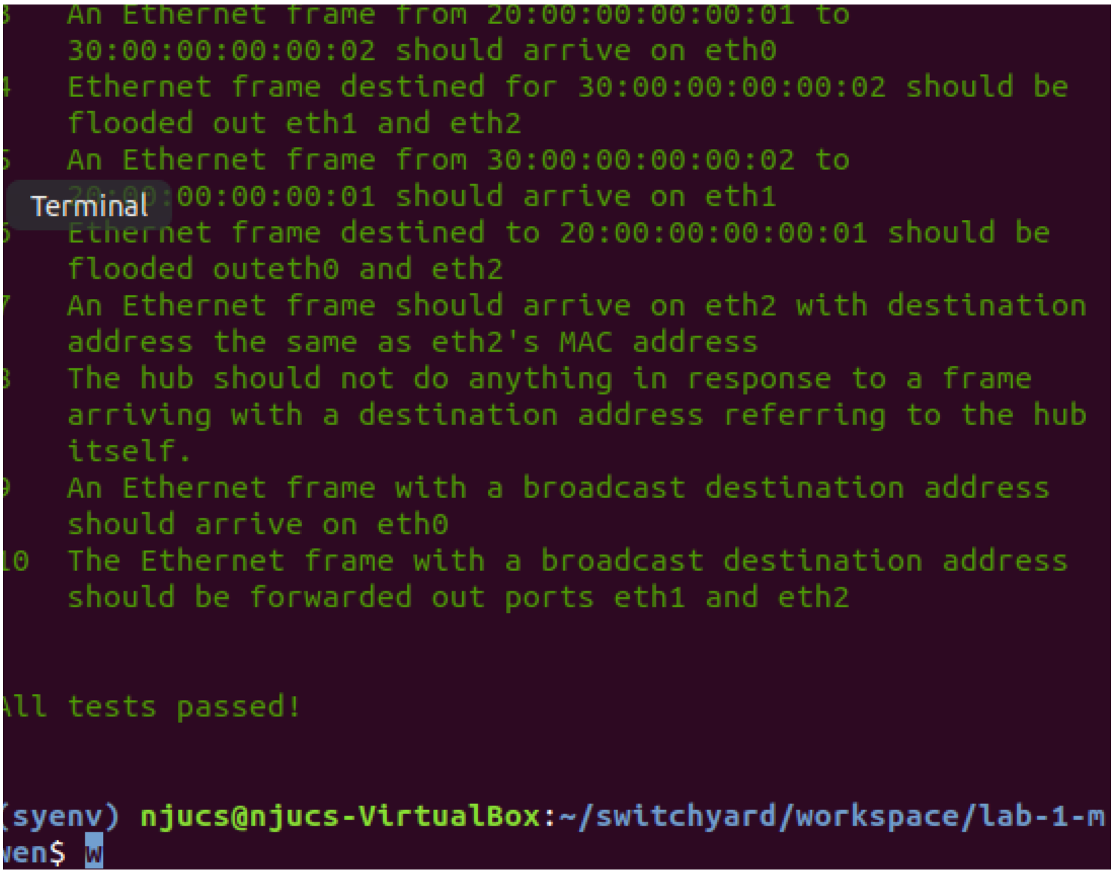
Step2：在 step1 删除 server2 的基础上进行，hub 运行代码

myhub.py, mininet 进行 pingall，打印日志结果如下



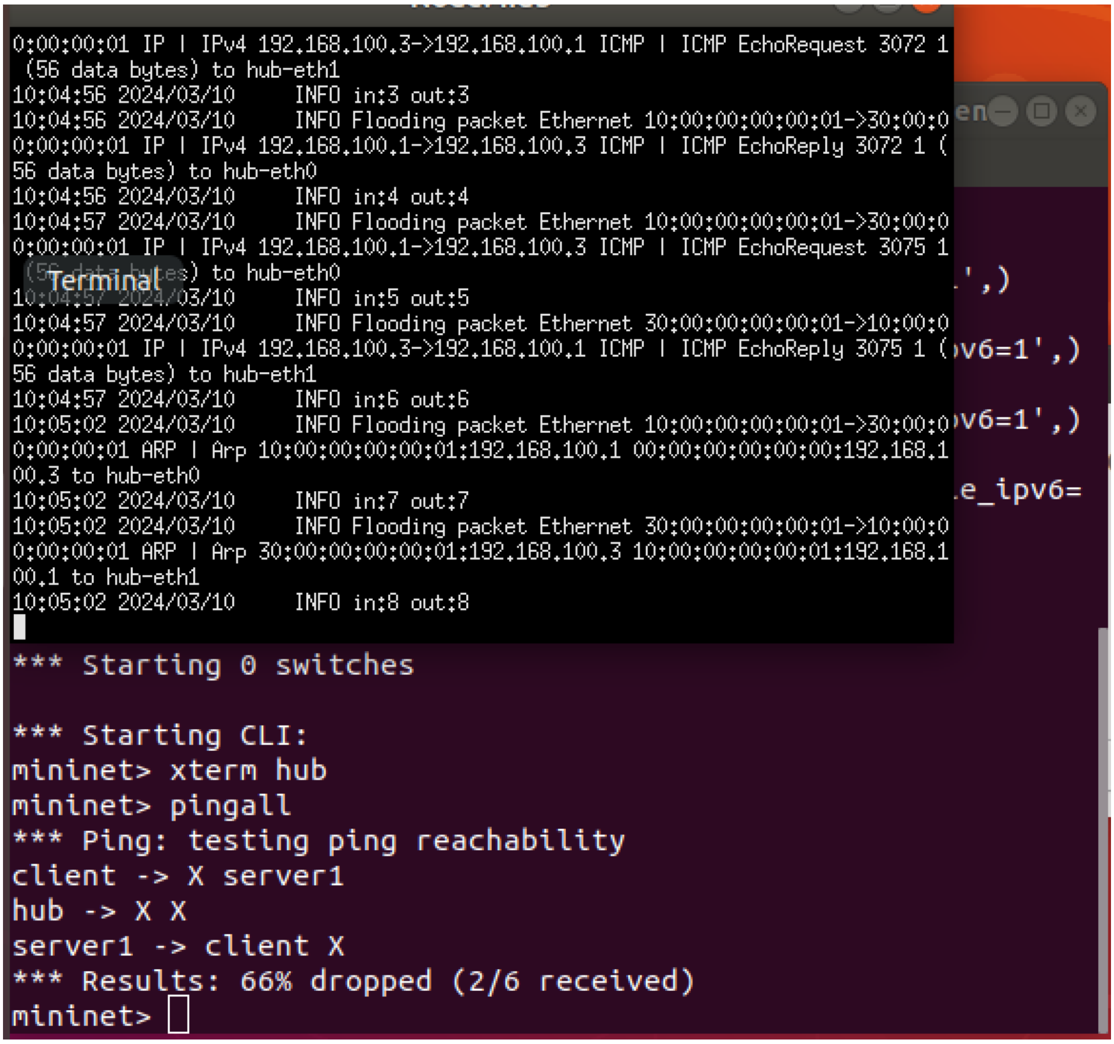
```
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100
 (56 data bytes) to hub-eth1
20:09:54 2024/03/09     INFO in:3 out:3
20:09:54 2024/03/09     INFO Flooding packet Et
0:00:00:01 IP | IPv4 192.168.100.1->192.168.100
56 data bytes) to hub-eth0
20:09:54 2024/03/09     INFO in:4 out:4
20:09:55 2024/03/09     INFO Flooding packet Et
0:00:00:01 IP | IPv4 192.168.100.1->192.168.100
 (56 data bytes) to hub-eth0
20:09:55 2024/03/09     INFO in:5 out:5
20:09:55 2024/03/09     INFO Flooding packet Et
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100
56 data bytes) to hub-eth1
20:09:55 2024/03/09     INFO in:6 out:6
20:10:00 2024/03/09     INFO Flooding packet Et
0:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.
00.3 to hub-eth0
20:10:00 2024/03/09     INFO in:7 out:7
20:10:00 2024/03/09     INFO Flooding packet Et
0:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.
00.1 to hub-eth1
20:10:00 2024/03/09     INFO in:8 out:8
```

Step3：添加自定义样例后观察 test 是否 pass，结果如下



```
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    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 eth0
10   The Ethernet frame with a broadcast destination address
     should be forwarded out ports eth1 and eth2


All tests passed!


(syenv) njucs@njucs-VirtualBox:~/switchyard/workspace/lab-1-m
wen$ w
```

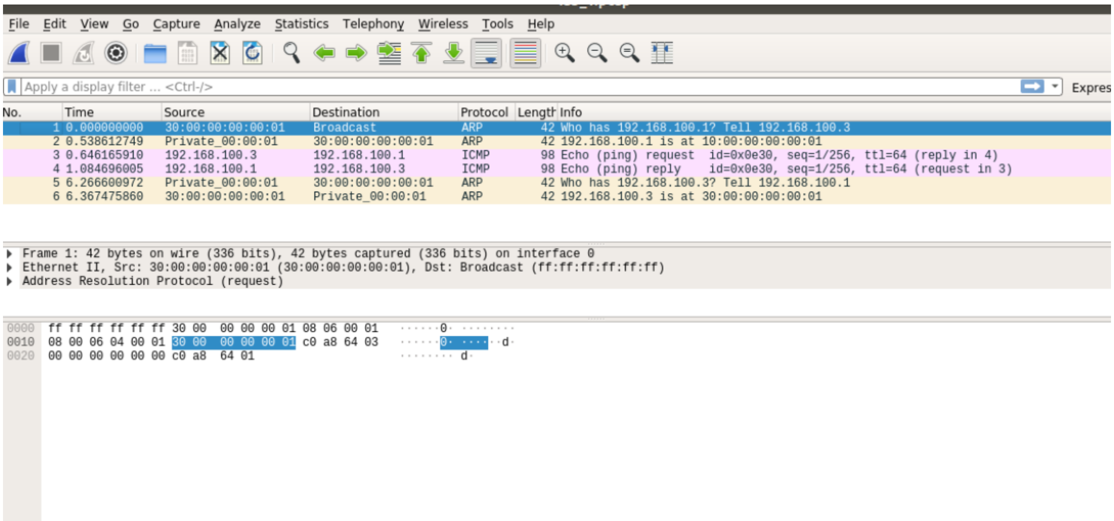
Step4: 运行时的结果如下



```
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoRequest 3072 1
 (56 data bytes) to hub-eth1
10:04:56 2024/03/10     INFO in:3 out:3
10:04:56 2024/03/10     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 EchoReply 3072 1 (
56 data bytes) to hub-eth0
10:04:56 2024/03/10     INFO in:4 out:4
10:04:57 2024/03/10     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 3075 1
 (56 data bytes) to hub-eth0
10:04:57 2024/03/10     INFO in:5 out:5
10:04:57 2024/03/10     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 3075 1 (
56 data bytes) to hub-eth1
10:04:57 2024/03/10     INFO in:6 out:6
10:05:02 2024/03/10     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 00:00:00:00:00:00:192.168.1
00.3 to hub-eth0
10:05:02 2024/03/10     INFO in:7 out:7
10:05:02 2024/03/10     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-eth1
10:05:02 2024/03/10     INFO in:8 out:8
```

```
*** 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>
```

Step5：运行时的结果与分析如下



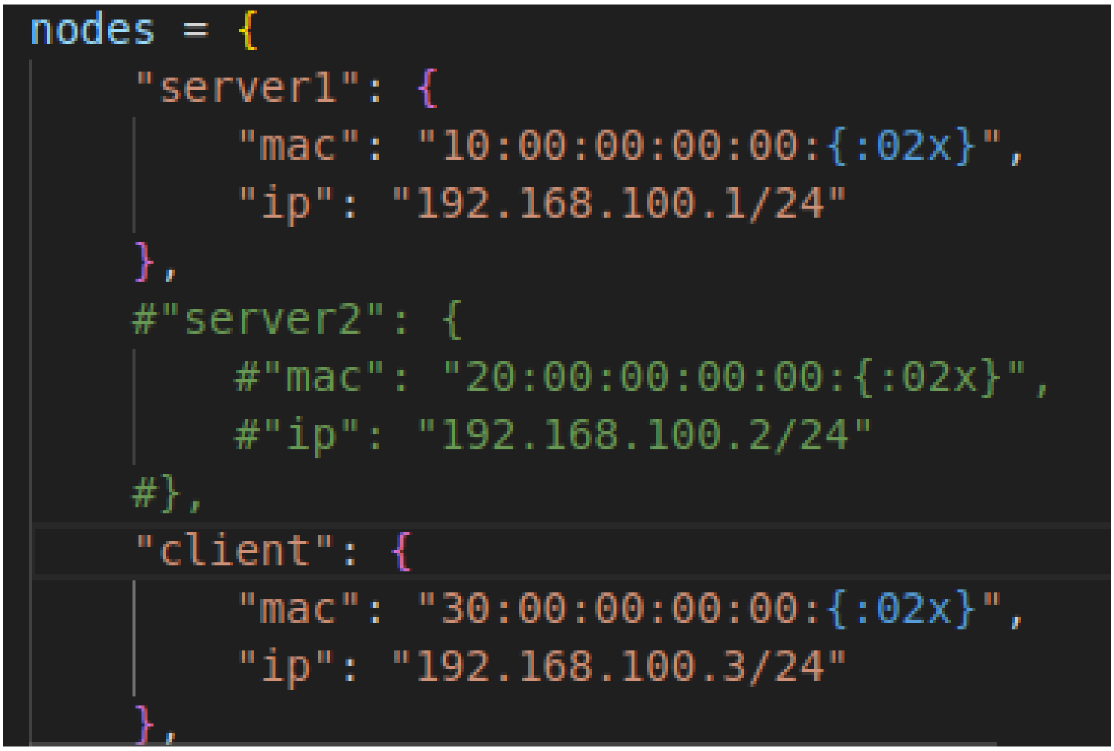
```
No.   Time          Source              Destination         Protocol Length Info
    1 0.000000000   30:00:00:00:00:01   Broadcast           ARP       42 Who has 192.168.100.1? Tell 192.168.100.3
    2 0.538612749   Private_00:00:01    30:00:00:00:00:01   ARP       42 192.168.100.1 is at 10:00:00:00:00:01
    3 0.646165910   192.168.100.3       192.168.100.1       ICMP      98 Echo (ping) request  id=0x0e30, seq=1/256, ttl=64 (reply in 4)
    4 1.084696005   192.168.100.1       192.168.100.3       ICMP      98 Echo (ping) reply    id=0x0e30, seq=1/256, ttl=64 (request in 3)
    5 6.266600972   Private_00:00:01    30:00:00:00:00:01   ARP       42 Who has 192.168.100.3? Tell 192.168.100.1
    6 6.367475860   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: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: 30:00:00:00:00:01 (30:00:00:00:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 30 00  00 00 00 01 08 06 00 01   ······0· ········
0010  08 00 06 04 00 01 30 00  00 00 00 01 c0 a8 64 03   ······0· ······d·
0020  00 00 00 00 00 00 c0 a8  64 01                     ········ d·
```

分析：client ping -c1 server1 发出流量包后，首先发出广播信号询

问 server1 的 MAC 地址，得到 server1 的 MAC 地址后，再发出 Request，得到 Reply，Reply 时也要先发送一个广播信号询问 client 的 MAC 地址，得到后再发出一个 Reply。

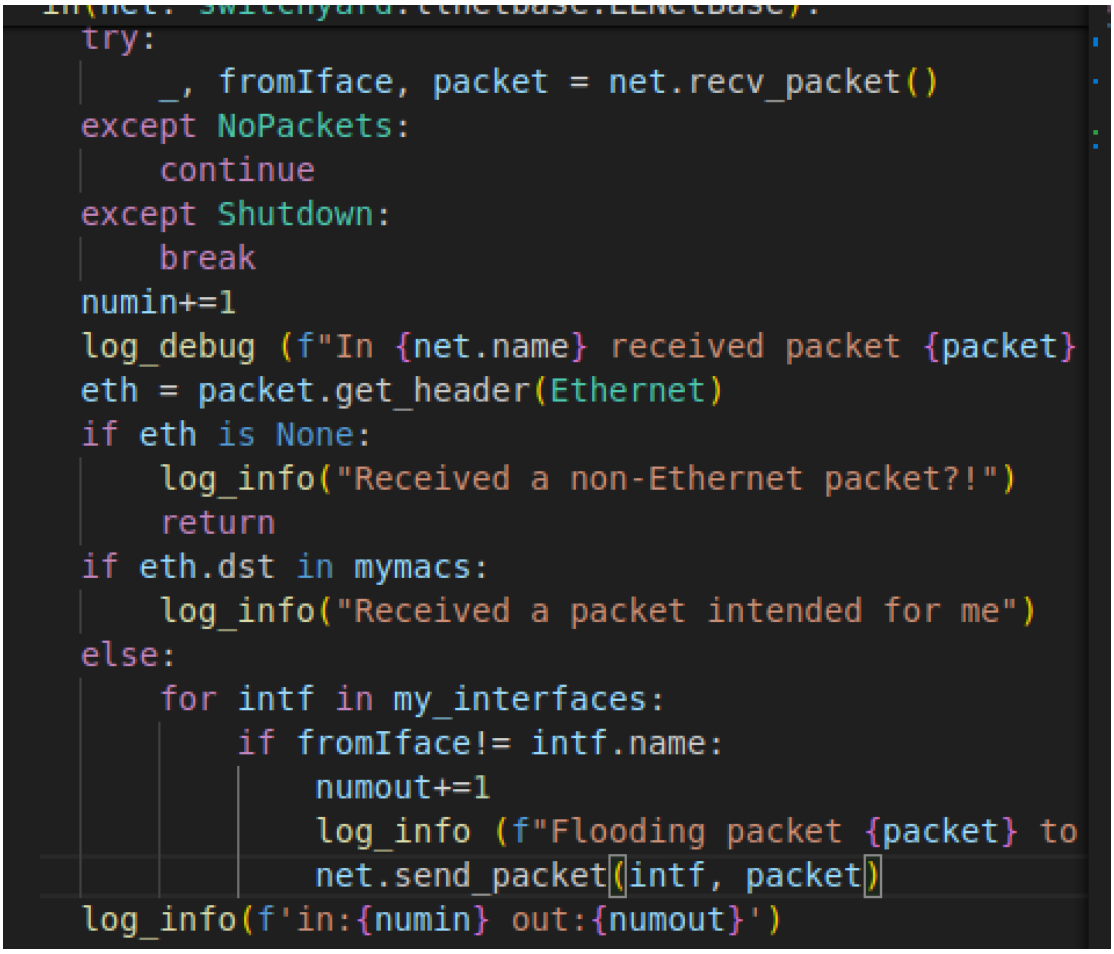
5. 核心代码

Step1 核心代码



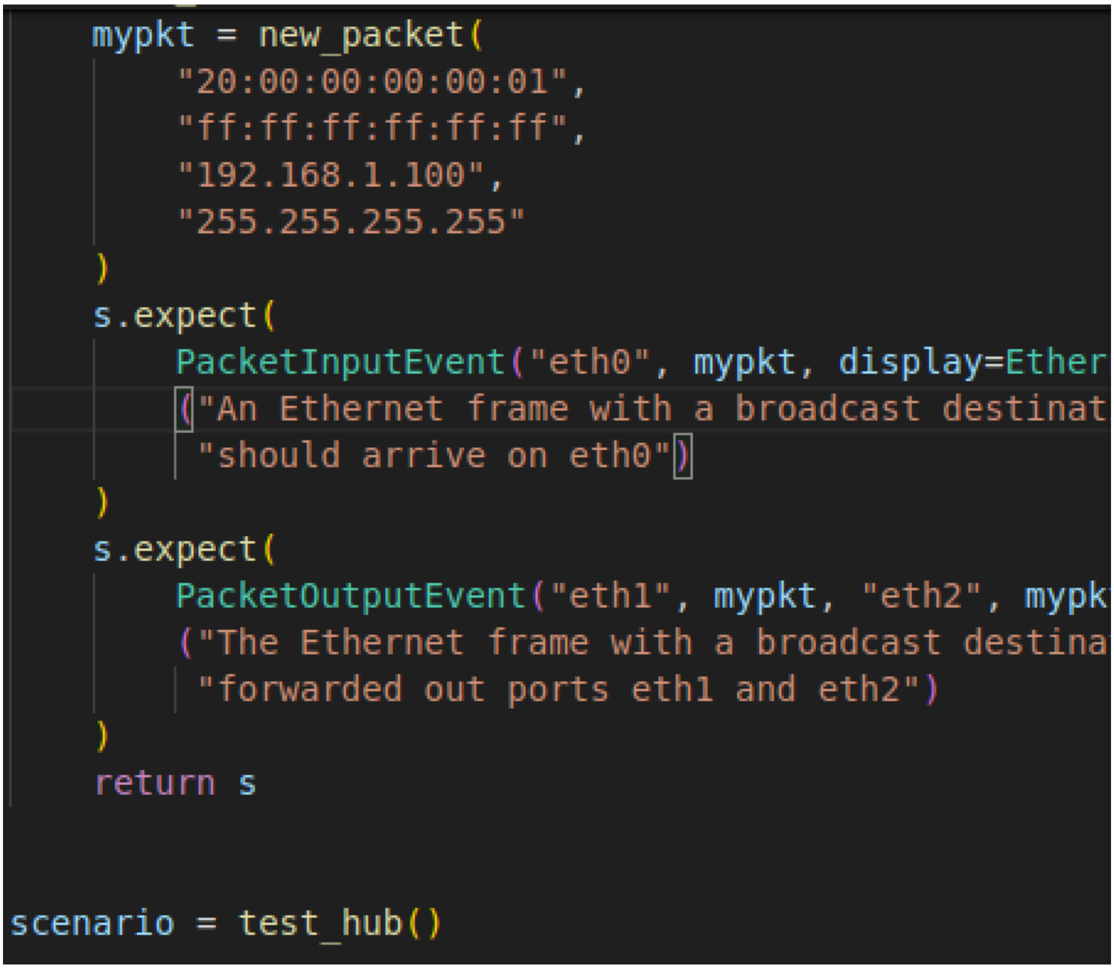
```
nodes = {
    "server1": {
        "mac": "10:00:00:00:00:{:02x}",
        "ip": "192.168.100.1/24"
    },
    #"server2": {
        #"mac": "20:00:00:00:00:{:02x}",
        #"ip": "192.168.100.2/24"
    #},
    "client": {
        "mac": "30:00:00:00:00:{:02x}",
        "ip": "192.168.100.3/24"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

Step2 核心代码



```
  in(net: switchyard.llnetbase.LLNetBase):
    try:
        _, fromIface, packet = net.recv_packet()
    except NoPackets:
        continue
    except Shutdown:
        break
    numin+=1
    log_debug (f"In {net.name} received packet {packet}
    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:
                numout+=1
                log_info (f"Flooding packet {packet} to
                net.send_packet(intf, packet)
    log_info(f'in:{numin} out:{numout}')
```

Step3 添加的样例代码如下



```
    mypkt = new_packet(
        "20:00:00:00:00:01",
        "ff:ff:ff:ff:ff:ff",
        "192.168.1.100",
        "255.255.255.255"
    )
    s.expect(
        PacketInputEvent("eth0", mypkt, display=Ether
        ("An Ethernet frame with a broadcast destinat
         "should arrive on eth0")
    )
    s.expect(
        PacketOutputEvent("eth1", mypkt, "eth2", mypk
        ("The Ethernet frame with a broadcast destina
         "forwarded out ports eth1 and eth2")
    )
    return s


scenario = test_hub()
```

Step4 代码修改过程与 step1、step2，step4 一致，无核心代码。

Step5 Step5 无核心代码

6. 总结与感想

使用虚拟机和新的软件总会遇到各种各样的问题（×）和看不明白的数据，更多时候需要靠自己去解决，学习的过程其实是解决问题的过程。