

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

学院：计算机科学与技术

学号：181860117

姓名：徐佳美

邮件：181860117@smail.nju.edu.cn

完成日期：2020.3.15

一. 实验名称

LAB1 Switchyard&Mininet

二. 实验目的

掌握实验环境和 vs code 等相关软件的使用；
学习含 hub 的网络传输问题。

三. 实验内容和核心代码

1.修改拓扑图：

这一步选择的是删除 server2 节点；只需在 start_mininet.py 中找到有关 server2 的信息并删除即可。如下：红色部分为删除内容

```
# server1
#      \
#      hub----client
#      /
# server2
#
nodeconfig = {'cpu':-1}
self.addHost('server1', **nodeconfig)
self.addHost('server2', **nodeconfig)
self.addHost('hub', **nodeconfig)
```

```
def setup_addressing(net):
    reset_macs(net, 'server1', '10:00:00:00:00:{:02x}')
    reset_macs(net, 'server2', '20:00:00:00:00:{:02x}')
    reset_macs(net, 'client', '30:00:00:00:00:{:02x}')
    reset_macs(net, 'hub', '40:00:00:00:00:{:02x}')
    set_ip(net, 'server1','hub','192.168.100.1/24')
    set_ip(net, 'server2','hub','192.168.100.2/24')
    set_ip(net, 'client','hub','192.168.100.3/24')
```

2.统计输入和输出包的数量:

定义两个变量 `in_pac` 和 `out_pac` 来计数;观察到在代码中有 `net.recv_packet()` 函数和 `net.send_packet()` 函数,显然是收到包和发送包的函数,在函数调用之后把相应的 `in_pac` 或者 `out_pac` 自增,再调用 `log_info` 打印信息即可。主要部分用红色标出:

```
in_pac=0
log_info("in_pac: {} ".format (in_pac))
out_pac=0
log_info("out_pac: {} ".format (out_pac))
while True:
    try:
        timestamp,dev,packet = net.recv_packet()
    except NoPackets:
        continue
    except Shutdown:
        return
    in_pac+=1
    log_info("in: {} ".format (in_pac))
```

```
        if dev != intf.name:
            log_info ("Flooding packet {} to
{}").format(packet, intf.name))
            net.send_packet(intf, packet)
            out_pac+=1
            log_info("out: {} ".format (out_pac))
```

3.修改测试场景:

先观察 `mk_pkt()` 函数,定义如下:

```
def mk_pkt(hwsrc, hwdst, ipsrc, ipdst, reply=False):
```

函数的名称为 `make package test`,可以确定这就是用来测试的函数;其参数内容为源地址的 `mac`,目的地的 `mac`,源地址的 `ip` 和目的地的 `ip`;观察已经给出的三个测试样例,可以确定节点的 `mac` 和 `ip` 的对应关系,如下表所示:

eth	mac	ip
eth1	30:00:00:00:00:02	172.16.42.2
eth2	20:00:00:00:00:01	192.168.1.100
eth0(hub)	10:00:00:00:00:03	

结合已知知识,给出的三个测试分别为①`eth1` 进行广播;②`eth1` 和 `eth2` 通信;③`eth2` 发信息给 `hub`;

选择增加测试样例,可以为 `eth2` 进行广播或者 `eth1` 发信息给 `hub`;仿照测

试 1 和 3，只要修改对应的 mac 和 ip 即可；
增加的测试如下：

```
# test case 4: a frame with dest address of one of the interfaces
should
    # result in nothing happening
    reqpkt = mk_pkt("30:00:00:00:00:01", "10:00:00:00:00:03",
'172.16.42.2', '172.16.42.2')
    s.expect(PacketInputEvent("eth1", reqpkt, display=Ethernet),
    "An Ethernet frame should arrive on eth2 with destination address
the same as eth1's MAC address")
    s.expect(PacketInputTimeoutEvent(1.0), "The hub should not do
anything in response to a frame arriving with a destination
address referring to the hub itself.")
    return s

    # test case 5: a frame with broadcast destination should get
sent out
    # all ports except ingress
    testpkt = mk_pkt("20:00:00:00:00:01", "ff:ff:ff:ff:ff:ff",
"192.168.1.100", "255.255.255.255")
    s.expect(PacketInputEvent("eth2", testpkt, display=Ethernet),
    "An Ethernet frame with a broadcast destination address should
arrive on eth2")
    s.expect(PacketOutputEvent("eth0", testpkt, "eth1", testpkt,
display=Ethernet), "The Ethernet frame with a broadcast
destination address should be forwarded out ports eth0 and eth1")
```

四. 实验结果

1. 增加新的测试场景后，测试通过；
如下图所示：

```
All tests passed!

(syem) njucs@jucs-VirtualBox:~/switchyard$ ./myhub.py
21:06:33 2020/03/15 INFO Starting test scenario lab_1/myhub.py
21:06:33 2020/03/15 INFO in: 0
21:06:33 2020/03/15 INFO out: 0
21:06:33 2020/03/15 INFO in: 1
21:06:33 2020/03/15 INFO Flooding packet Ethernet 30:00:00:00:00:02->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
21:06:33 2020/03/15 INFO out: 1
21:06:33 2020/03/15 INFO Flooding packet Ethernet 30:00:00:00:00:02->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
21:06:33 2020/03/15 INFO out: 2
21:06:33 2020/03/15 INFO in: 2
21:06:33 2020/03/15 INFO Flooding packet Ethernet 20:00:00:00:00:01->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
21:06:33 2020/03/15 INFO out: 3
21:06:33 2020/03/15 INFO Flooding packet Ethernet 20:00:00:00:00:01->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
21:06:33 2020/03/15 INFO out: 4
21:06:33 2020/03/15 INFO in: 3
21:06:33 2020/03/15 INFO Flooding packet Ethernet 30:00:00:00:00:02->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
21:06:33 2020/03/15 INFO out: 5
21:06:33 2020/03/15 INFO Flooding packet Ethernet 30:00:00:00:00:02->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
21:06:33 2020/03/15 INFO out: 6
21:06:33 2020/03/15 INFO in: 4
21:06:33 2020/03/15 INFO Received a packet intended for me

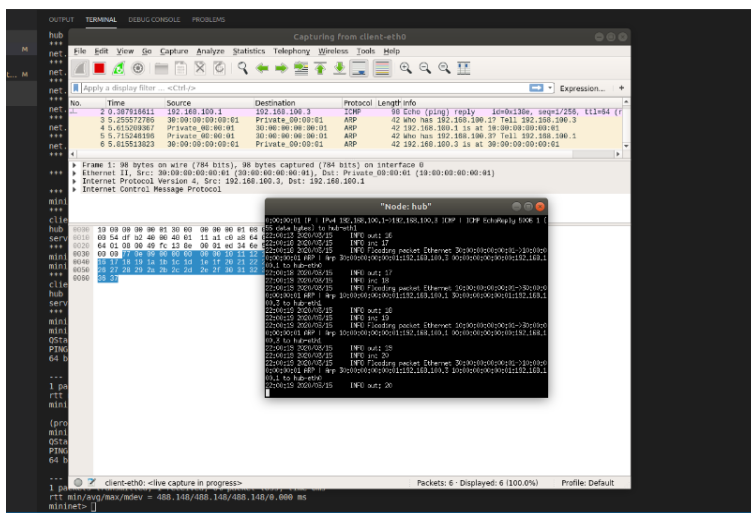
Results for test scenario hub tests: 0 passed, 0 failed, 0 pending

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.

All tests passed!

(syem) njucs@jucs-VirtualBox:~/switchyard$
```

2. 运行 mininet 和抓包，（pingall 时丢包率为 66%）



五. 感想与总结：

- ①认真看清实验要求，尤其是运行次序：某次测试丢包率为 100%，就是因为是在 ping 之前没有运行 myhub.py;
- ②实验之前先把课件看完，尤其是实验视频，不要直接上手实验。