

# 计算机网络 lab\_1 实验报告

181840135 梁俊凯

我想主要描述一下实验要求的步骤内容，我对本次实验的理解，以及我还没有明白的地方。

Task1: 需要下载三个工具，mininet, wireshark, switchyard 并按照实验要求测试并熟悉三个工具的使用。在 wireshark 的安装中，无论我怎样更换源都会提示依赖错，后将 ubuntu19.04 更新至 19.10 顺利解决。在实验过程中我对各个包的理解：

1>mininet: 通过 start\_mininet.py 进行启动，可以通过修改自己的 start\_mininet.py 来产生不同的网络结构，运行时需要 root 权限，我的理解是它会产生一个实验性的教学网络来模拟真正网络的运作，但我还不清楚其中的网络是否与本机的适配器相关联，还是仅仅通过代码产生的一个虚拟网络？接下来我准备通过阅读手册来理解这一问题

2>wireshark: 抓包软件，需要运行在 usermod 可以让我们近距离观察 mininet 中的包是什么样子，只需在 mininet 对某一节点执行指令即可。我认为 wireshark 并不是为 mininet 设计的，而是可以为真机的适配器抓包，但它是怎么为 mininet 服务的我并不理解，还是代码结构的问题。

3>switchyard 我理解是在 mininet 框架下的应用视角，比如其中的 start\_mininet.py 就是 mininet 的启动代码，其中的 hubtest 文件也是根据 mininet 做一些测试。

Task2: 分为 5 个 step，我会逐一描述我的工作

1>改变 mininet 的结构，我删除掉了 start\_mininet.py 中建立 server2 的代码，如下注释部分

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

for node in ['server1','client']:
    # all links are 10Mb/s, 100 millisecond prop delay
    self.addLink(node, 'hub', bw=10, delay='100ms')
```

当运行时执行 nodes 命令显示节点，我看到只有 client, hub, server1 三个节点，符合预期。但此时执行 pingall 时会显示 100% dropped，这个问题我决定留到以后。

2>myhub.py 添加信息，我定义了两个变量 in\_packet, out\_packet 分别表示入包数量和出包数量，初始化为 0，当执行流通过 try...except 函数后证明 resv\_packet 未抛出异常，接包成功，此时 in\_packet 加 1，当执行流调用 send\_packet 后证明发包，out\_packet+1，每次 while 循环后都会打印 in, out 的数值，运行结果如下：可以看到 in, out 输出

```
0:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoRequest 11999 1 (56 data bytes) to hub-eth0
10:03:42 2020/09/29      INFO in:3 out:3
10:03:43 2020/09/29      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 11999 1 (56 data bytes) to hub-eth1
10:03:43 2020/09/29      INFO in:4 out:4
10:03:43 2020/09/29      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 12003 1 (56 data bytes) to hub-eth1
10:03:43 2020/09/29      INFO in:5 out:5
10:03:43 2020/09/29      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 12003 1 (56 data bytes) to hub-eth0
10:03:43 2020/09/29      INFO in:6 out:6
10:03:48 2020/09/29      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-eth1
10:03:48 2020/09/29      INFO in:7 out:7
10:03:48 2020/09/29      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-eth0
10:03:48 2020/09/29      INFO in:8 out:8
```

此时 mininet 终端提示 66% dropped

3>修改 hubtest.py，我选择应用 mk\_pkt 来写入自己的测试，我尝试了不同于 30:00:00:00:00:02 和 20:00:00:00:00:01 的源 mac 地址，发现运行时都 fail 了，但是 hubtest.py 中并没有这两个地址的初始化，猜测可能与 mininet 的初始代码有关，于是我就使用的一个目的广播地址，改变了 test case 1 中传输的参数（eth0, eth1 等）并得到了一个可以通过的 test。运行结果如下，其中 9 和 10 是我的测试的输出

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

All tests passed!
```

4>在 mininet 中使用 switchyard 代码。此部在 step2 展示 in, out 时已经运行过, 具体操作是在启动 mininet 后使用 xterm hub/server/clinet 然后在出现的 xterm 终端中执行 myhub.py 代码, 具体步骤如图: 包括进入 python 环境和运行代码

```
root@ljk-ThinkPad-13:/home/ljk/Network_project/lab_1# source /home/ljk/switchyard/syenv/bin/activate
(syenv) root@ljk-ThinkPad-13:/home/ljk/Network_project/lab_1# ./myhub.py
bash: ./myhub.py: Permission denied
(syenv) root@ljk-ThinkPad-13:/home/ljk/Network_project/lab_1# swyard myhub.py
10:22:50 2020/09/29      INFO Saving iptables state and installing switchyard rules
10:22:50 2020/09/29      INFO Using network devices: hub-eth0 hub-eth1
```

5>抓包, 我在启动 start\_mininet.py 后打开 wireshark 进行抓包, 然后执行 pingall 指令, 观察包的变化

1	0.000000000	Private_00:00:01	Broadcast	ARP	42 Who has 192.168.100.3? Tell 192.168.100.1
2	1.008101605	Private_00:00:01	Broadcast	ARP	42 Who has 192.168.100.3? Tell 192.168.100.1
3	2.031798482	Private_00:00:01	Broadcast	ARP	42 Who has 192.168.100.3? Tell 192.168.100.1

可以看到包的内容是广播, 询问谁有对应的 IP 并希望得到回答, 本次 pingall drop 率为 100%, 故看到没有人回答, 我还没有关于 IP 的知识, 故暂未理解为何没有得到回答这个行为。

最后提交时, 因为我将 lab\_1 文件夹放在了 assignment1 的外面, 故 git push origin master 不成功, 后将文件夹整体移动至 assignment1 内并重新 git commit, 由于之前都在 assignment1 外部做实验 git 记录都在外部的 Network\_project 目录下的 .git 中保存, 故可能有一些中途 commit 的消失。(可以查询)

本次实验报告结束, 感谢阅读!