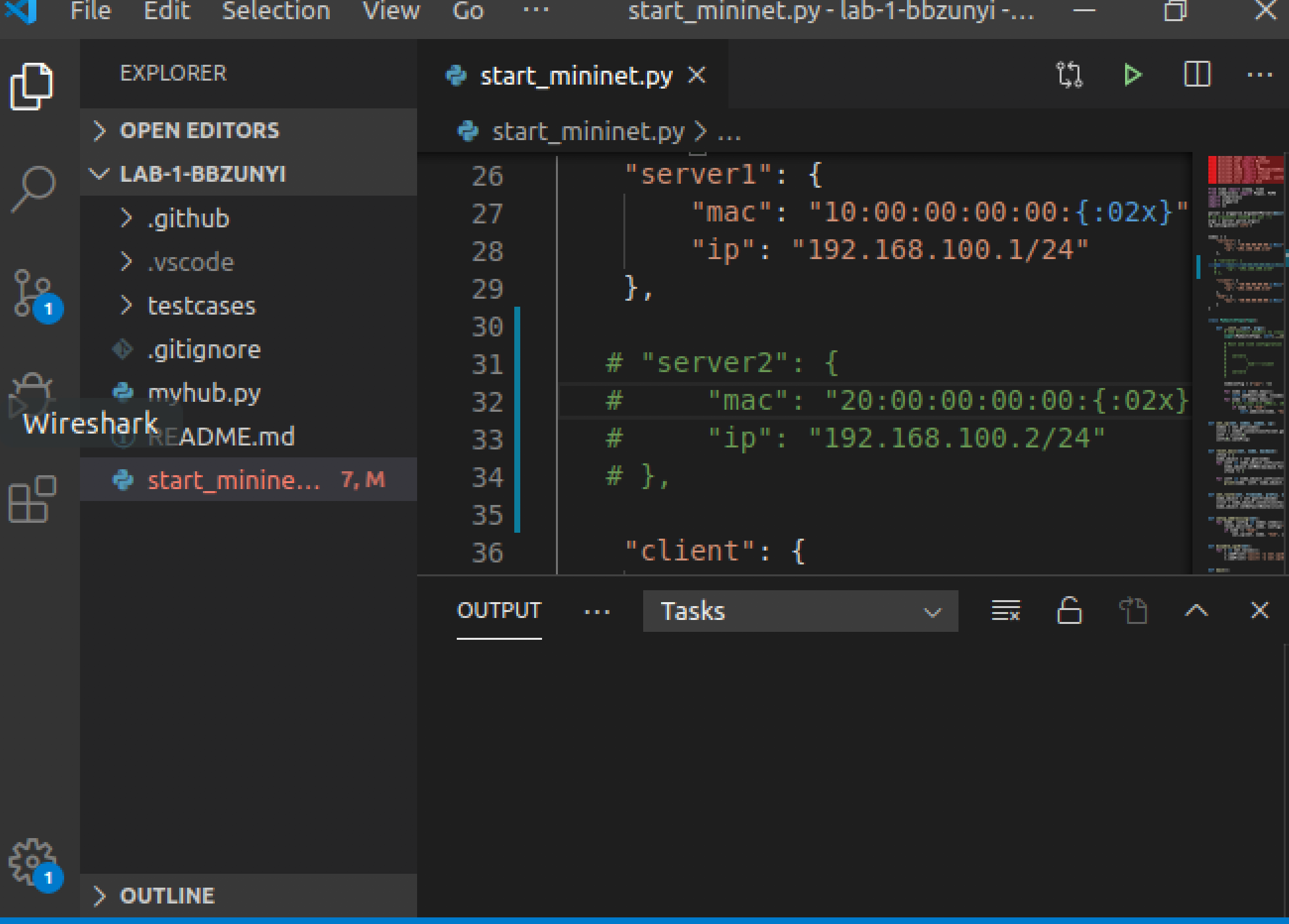
# 计网lab1 Switchyard&Mininet实验报告

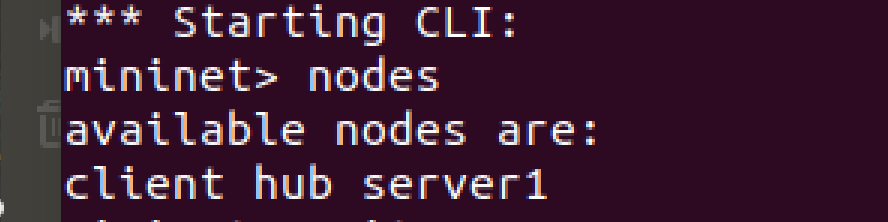
张洋彬 191220169

## Step1：修改mininet的拓扑结构

在这个环节，我选择了在拓扑图中删除server2，在start\_mininet.py中，将本段代码注释掉，即可完成删除server2的操作



删除结果如下所示：

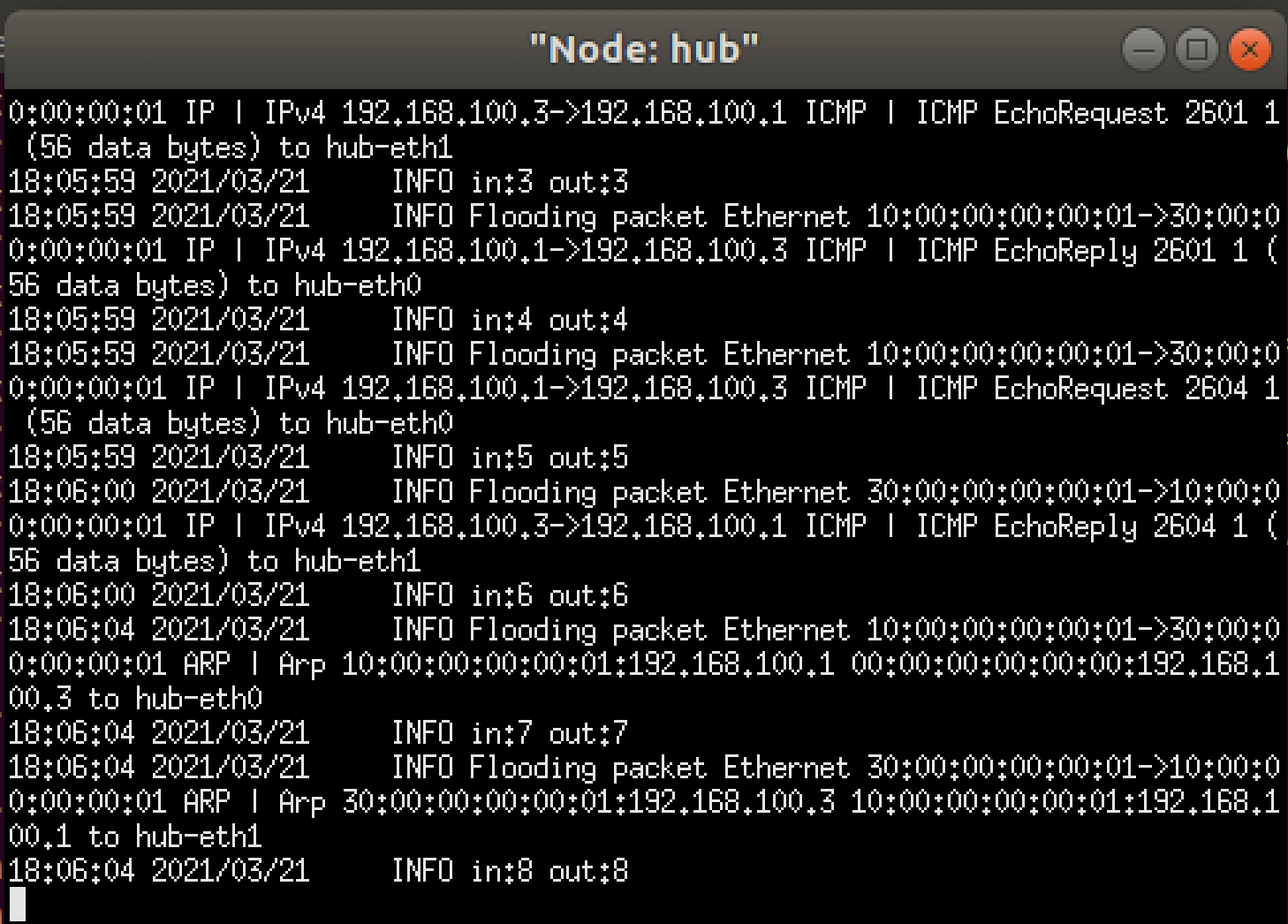


## Step2：修改设备的逻辑



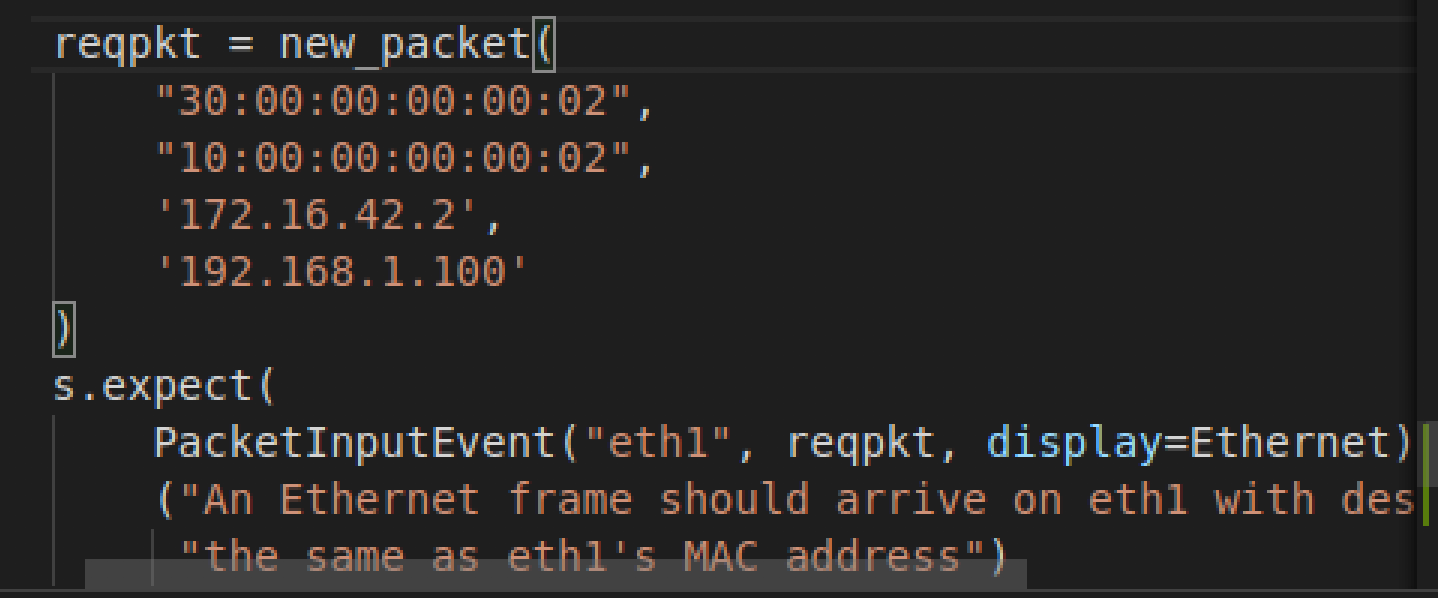


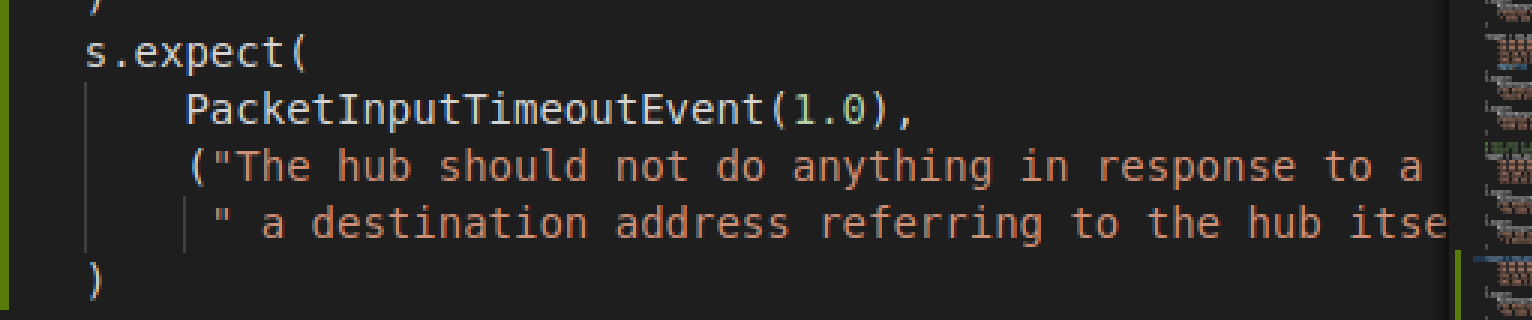
1. 定义in\_count&out\_count并赋予初值;
2. 当有包传输的时候，in\_count+1;
3. 当packet被发送走的时候，out\_count+1;
4. 每次循环都打印出in：in\_count out:out\_count(结果如下图所示)



## Step3 修改测试文件

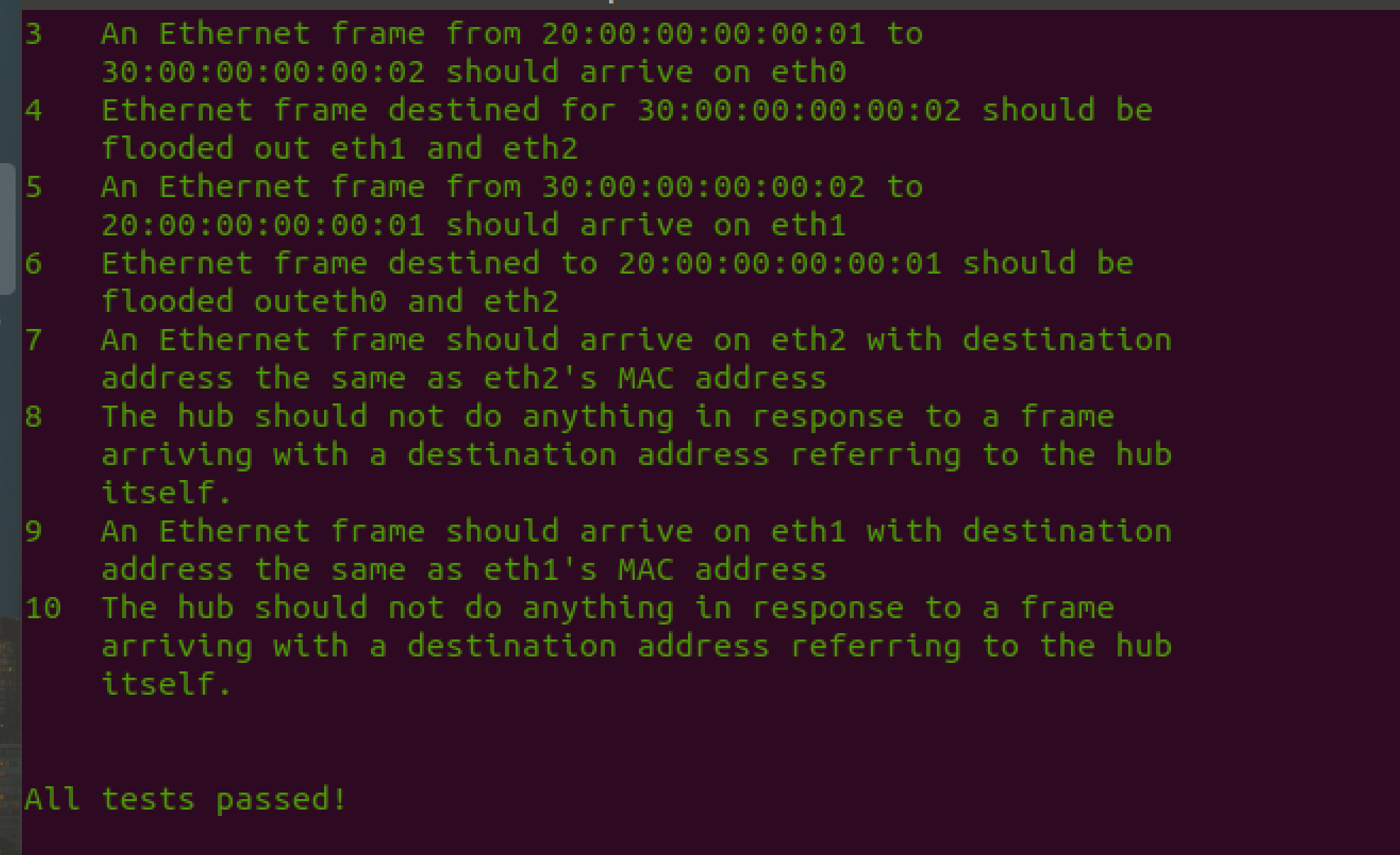
1、选择第一个任务：利用new\_packet创造一个新的测试





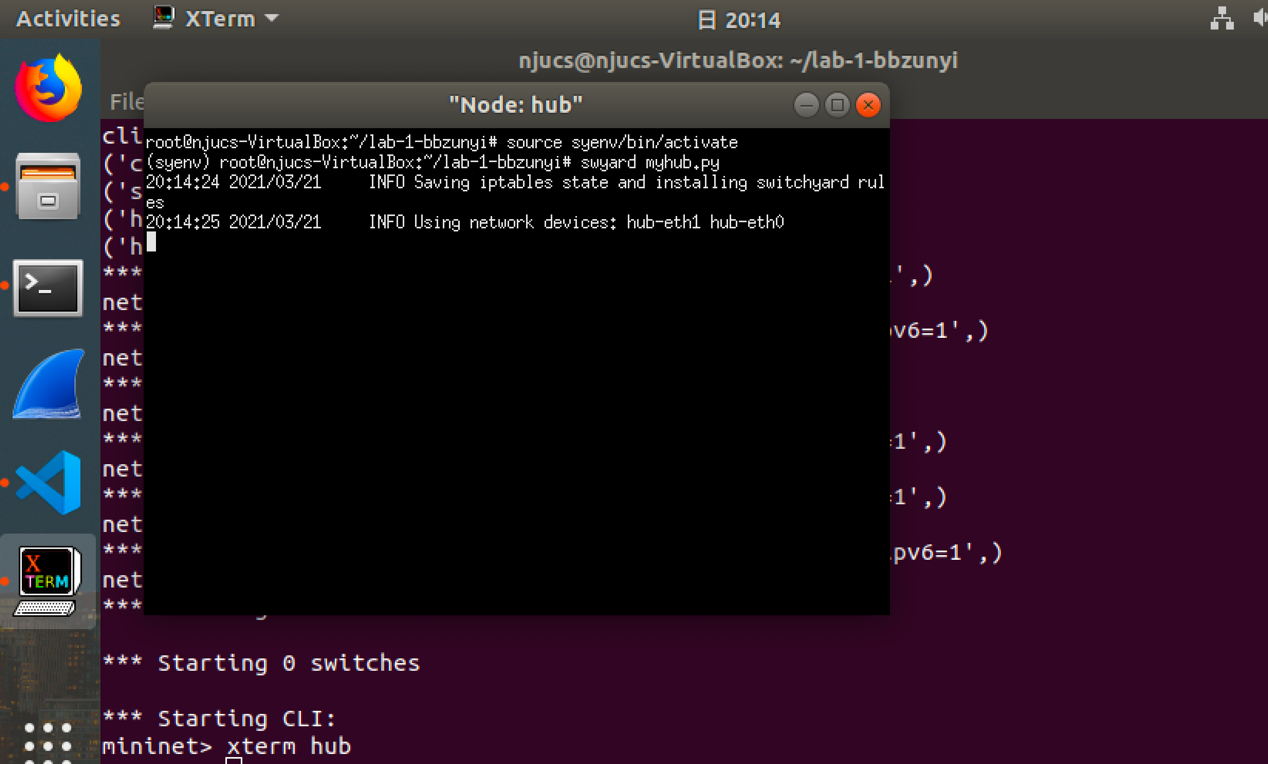
这里所做的操作是创建一个eth1的包，目的地址是eth1的Mac地址，所以并不会经过hub结点，所以说hub不会做任何事。

通过测试后，输出第九条和第十条信息。

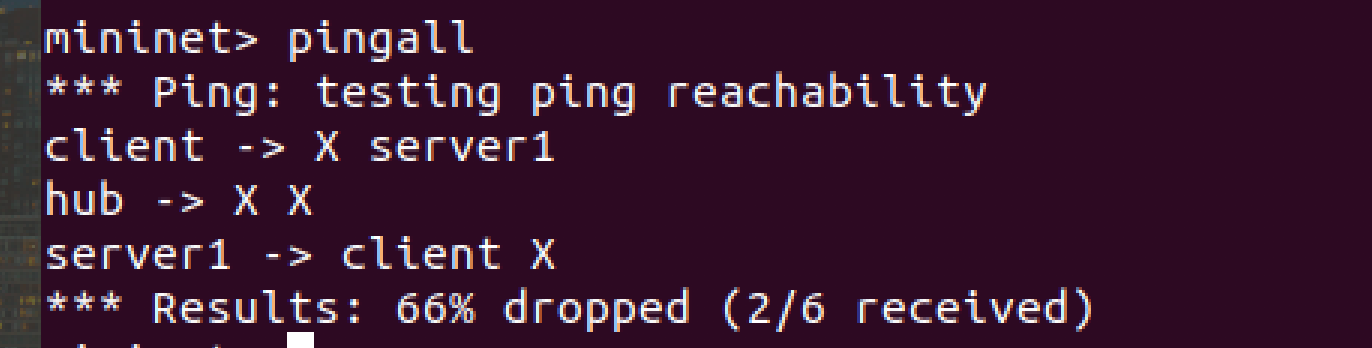


## Step4：在mininet中运行设备

1. 输入sudo python start\_mininet.py，打开mininet，创建拓扑结构
2. 输入xterm打开hub节点，并在xterm中激活虚拟环境，在用switchyard运行myhub.py

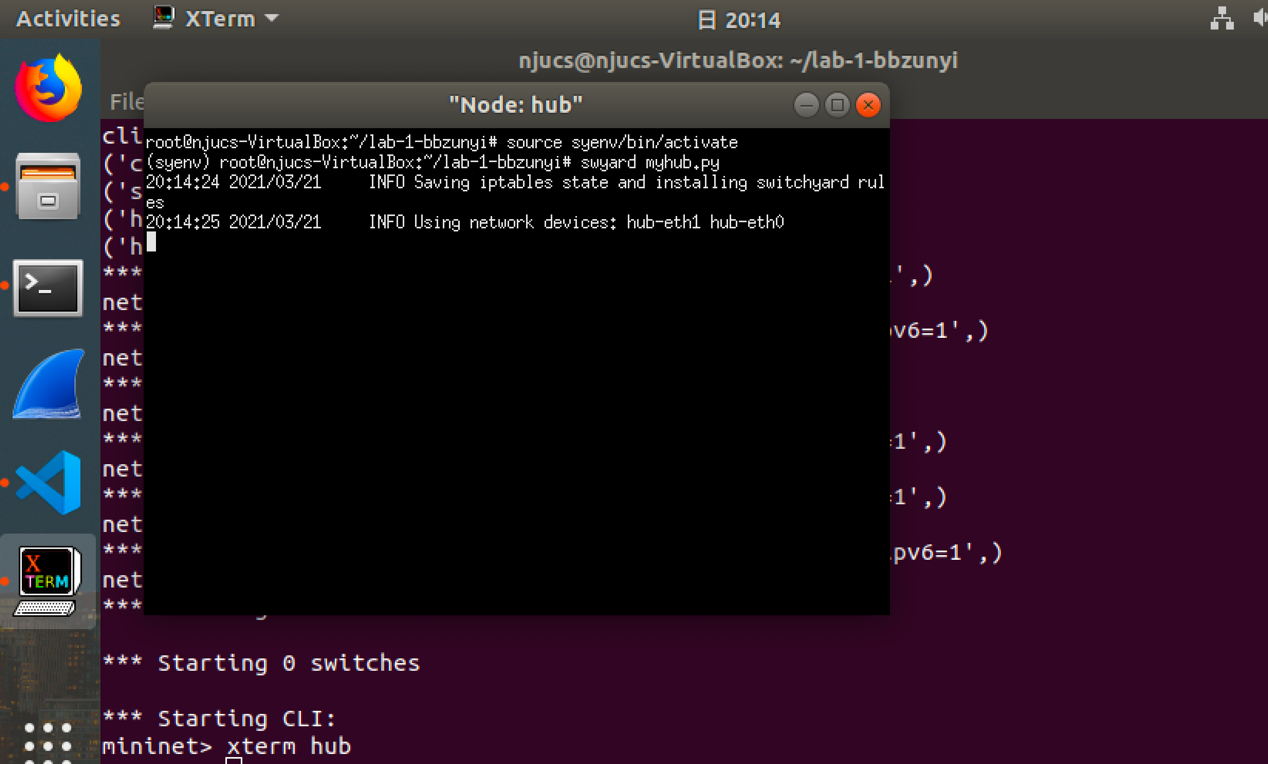


3、在mininet中输入pingall指令，检查各个节点的连通性，hub工作在网络层之下，没有ip地址，所以无法与其他节点连接。

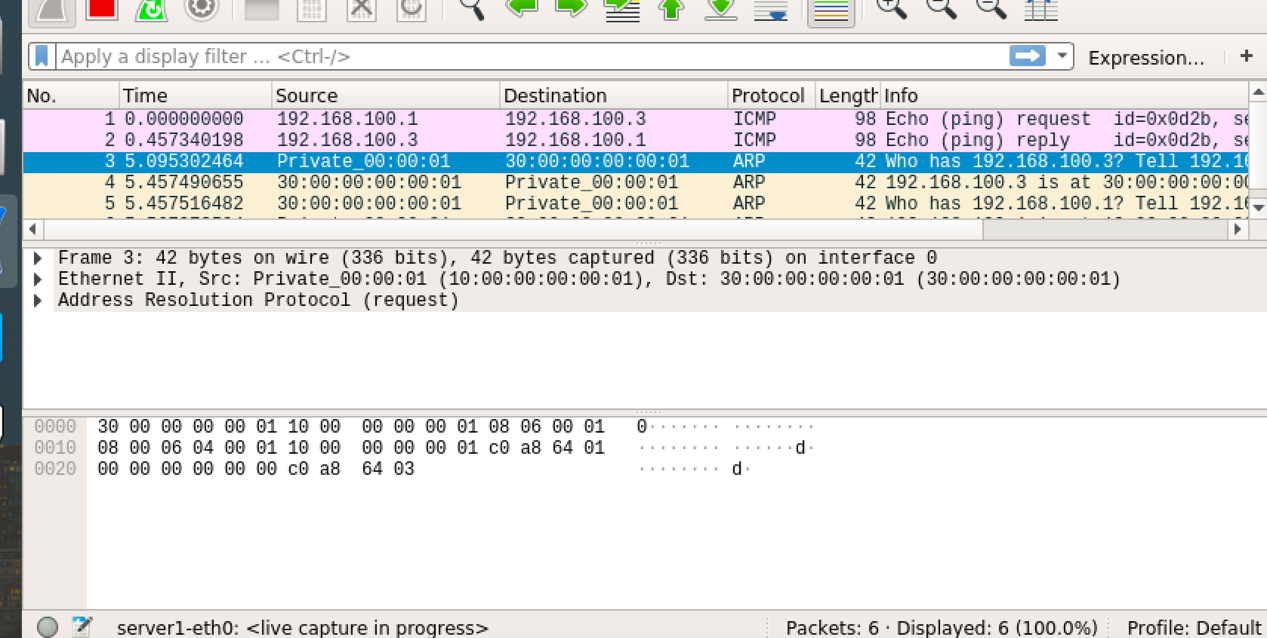


## Step5 用wireshark抓包

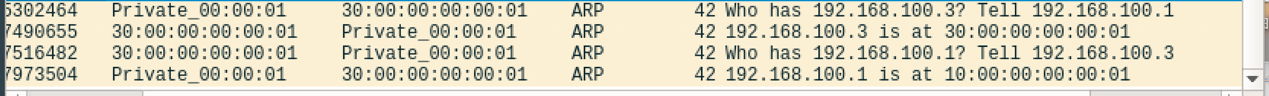
1. 同step4，先用myhub.py进行hub实例化。



1. 在mininet中输入server1 wireshark &指令，对server抓包
2. 输入server1 ping -c 1 client 让server1和client进行通信



如这两张图所示，server1和client之间先通过ICMP协议通信一次，server1先向client发送一次请求，然后client收到后给予回复server1。下图中的四个包，他们之间以问答的形式，询问IP地址对应的MAC地址并得到了地址。



## 感想与总结

本次实验内容比较简单，主要是基础知识的学习，在做实验的过程中学习到了很多知识，比如之前了解比较少的git和python，现在基本了解了git的全部内容，比如删除分支、合并分支之类的操作，以及多人一起在github上工作的各种技巧，python也学习了一些基础的语言操作。

感觉助教在问题网站上给予的回答有很大的参考性，基本了解了本次实验中之前会迷惑的内容：比如hub为什么与其他几个点不能传包，mininet和switchyard的概念等等。