**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：

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | **191220154** | 姓名 | **张涵之** |
| Email | **1683762615@qq.com** | 开始/完成日期 | **2021/3/19 – 2021/3/21** |

1. **实验名称：Lab 1: Switchyard & Mininet**
2. **实验目的：**
   1. Setting up the environment;
   2. Getting familiar with the tools: Switchyard, Mininet, Wireshark, Git, etc.;
   3. Acquiring a general impression on the virtual network structures, learning how to construct nodes, write testcases and capture files.
3. **实验内容**
   1. Step 1: Modify the Mininet topology

(I choose to delete server2 in the topology);

* 1. Step 2: Modify the logic of a device

Count how many packets pass through a hub in and out, log the statistical result every time one packet is received;

* 1. Step 3: Modify the test scenario of a device

(I choose to create a test case using new\_packet with different arguments);

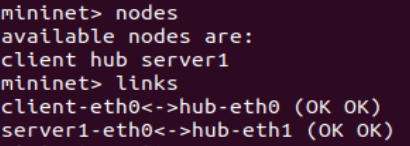
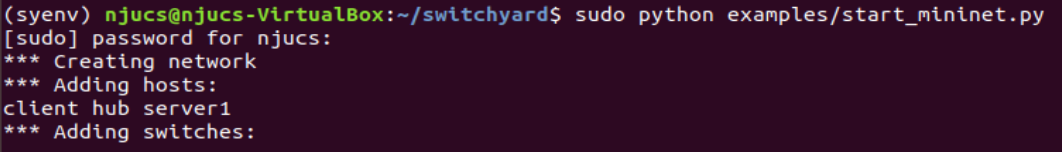
* 1. Step 4: Run your device in Mininet

Run the new hub in the new topology and make sure it works;

* 1. Step 5: Capture using Wireshark

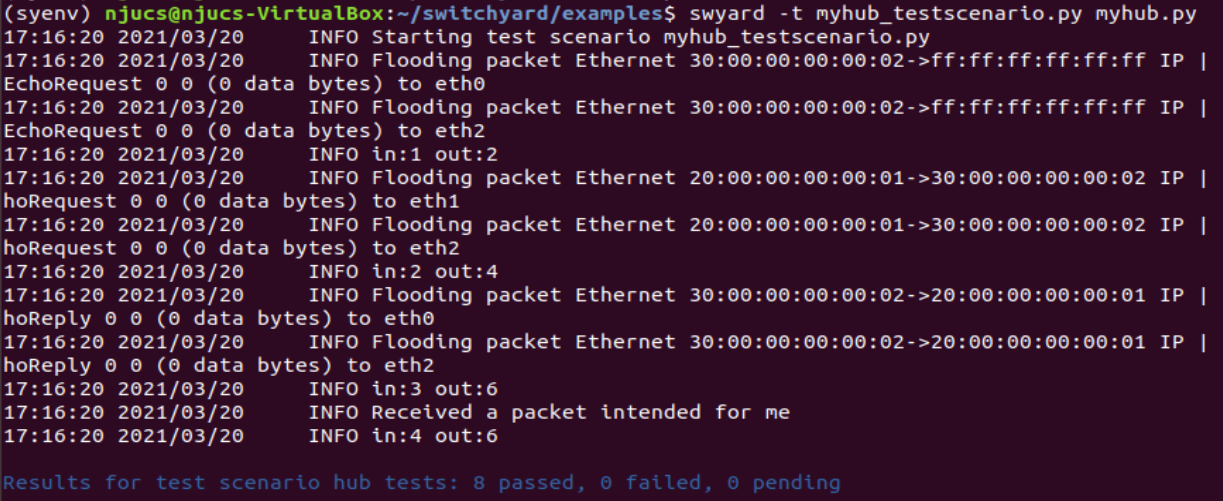
Save the capture file and describe the details of it capture file;

1. **实验结果**
   1. Step 1: Deleted server2 in the topology

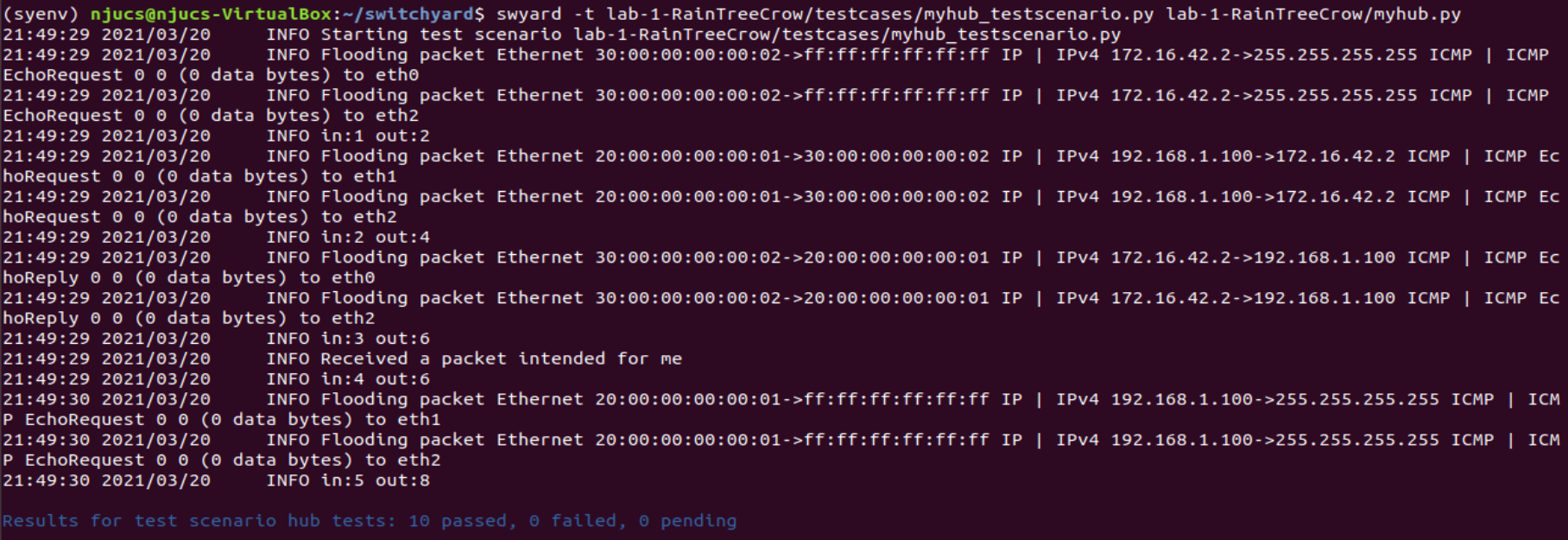


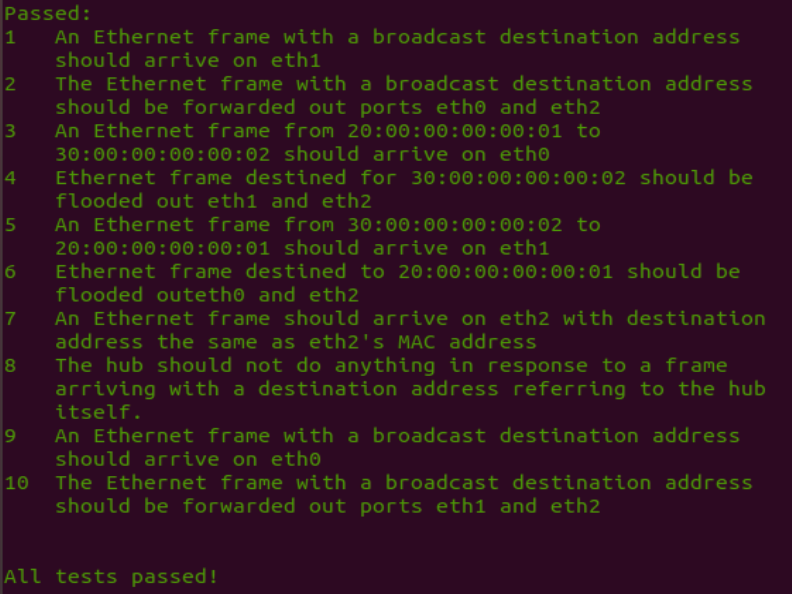
* 1. Step 2: Counting packets and logging statistics

(See more results of log information in Step 4)

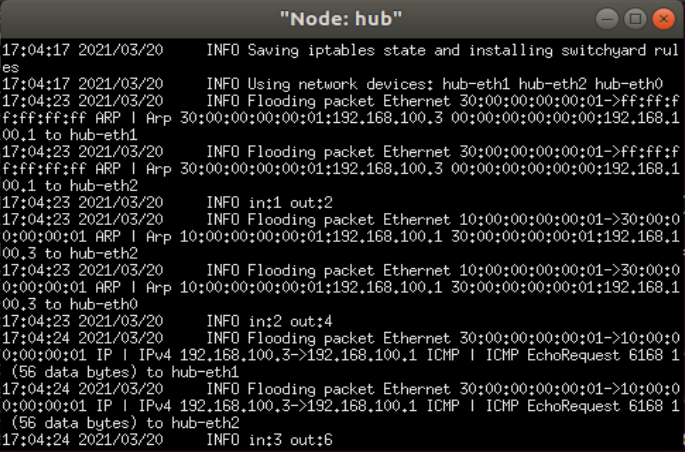


* 1. Step 3: Created new test case and passed successfully



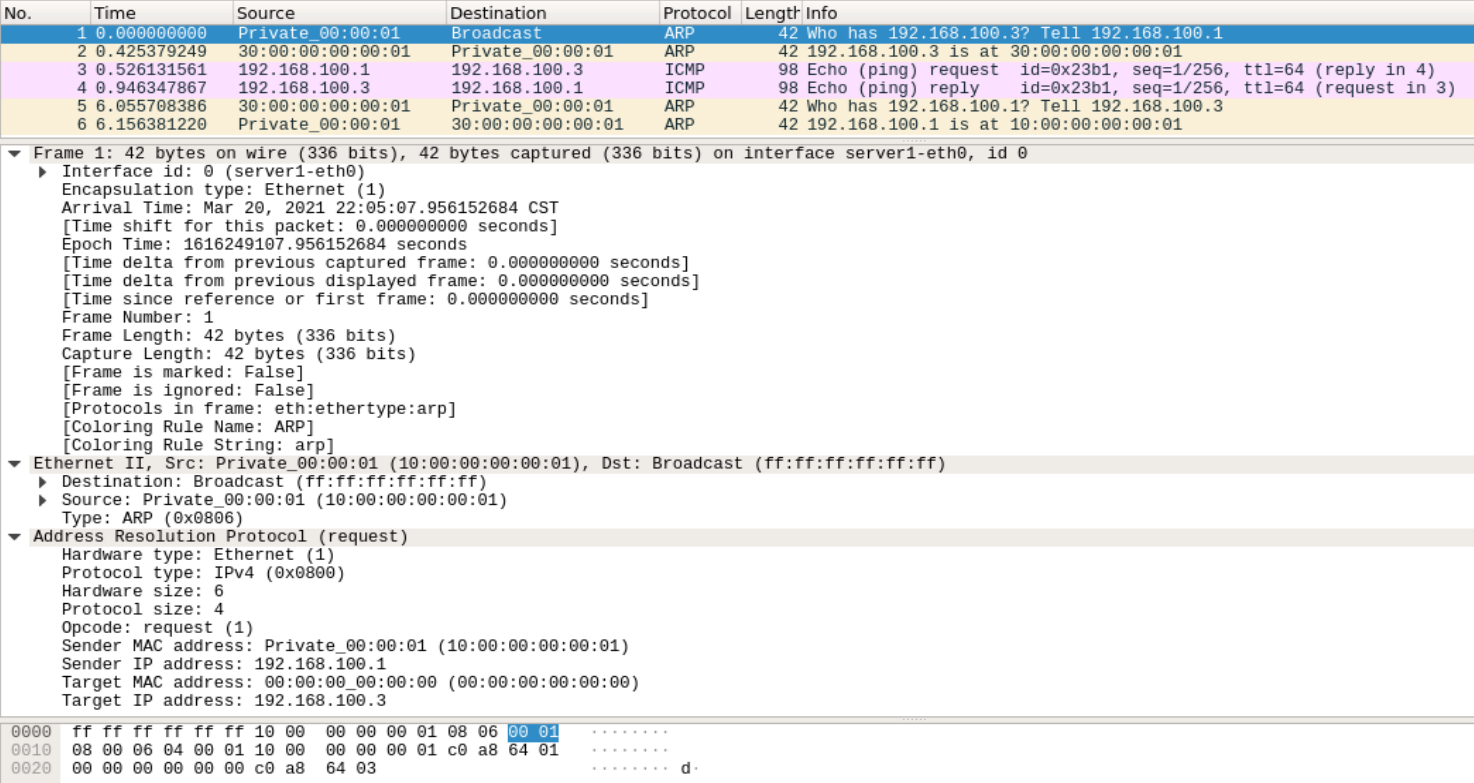


* 1. Step 4: Running the new hub in Mininet



The testing process involves running start\_mininet.py, which set up the nodes client, hub and server1, if try to ping at once, the drop rate would be 100%, as the links are not established yet. If ping after running myhub.py in xterm hub window, however, only the links between nodes EXCEPT hub can transfer file successfully, for hub does noy have an IP address and cannot be pinged.

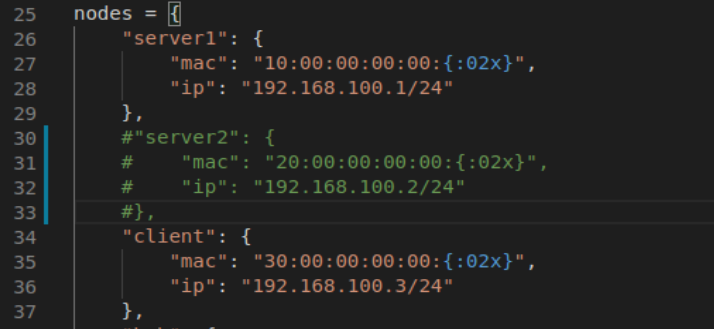
* 1. Step 5: Capturing files with Wireshark



This is the file captured after running “server1 ping -c client”, From the file I can see the interface id, arrival time, capture length of the frame, as well as the source and destination, the MAC and IP address of the sender and the target, I may as well infer that the packet is sent using broadcast.

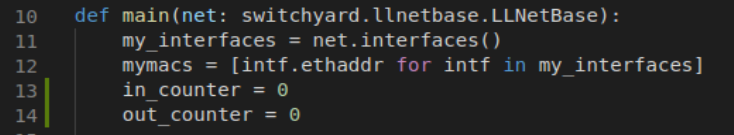
1. **核心代码**
   1. Step 1: Delete server2 by annotating relevant code;

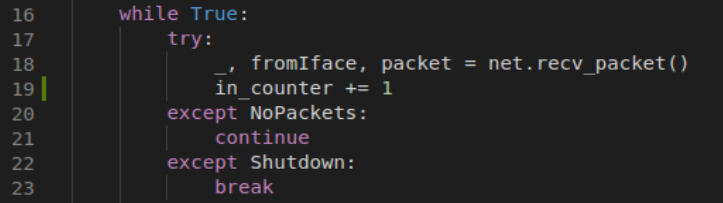
This is the only place where “server2” is mentioned, when the nodes are used the code is often written “for node in nodes”, so I assume deleting it from the array is enough. Further testing shows it IS enough.

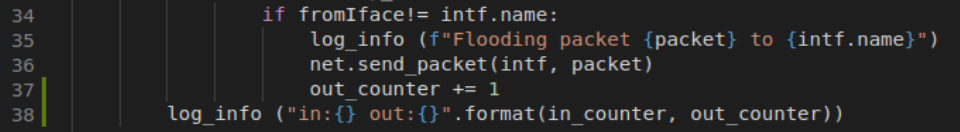


* 1. Step 2: Adding counters to count packets and logging statistics;

The statistical result is logged every time a packet is received with the format of each line in:<ingress packet count> out:<egress packet count>.

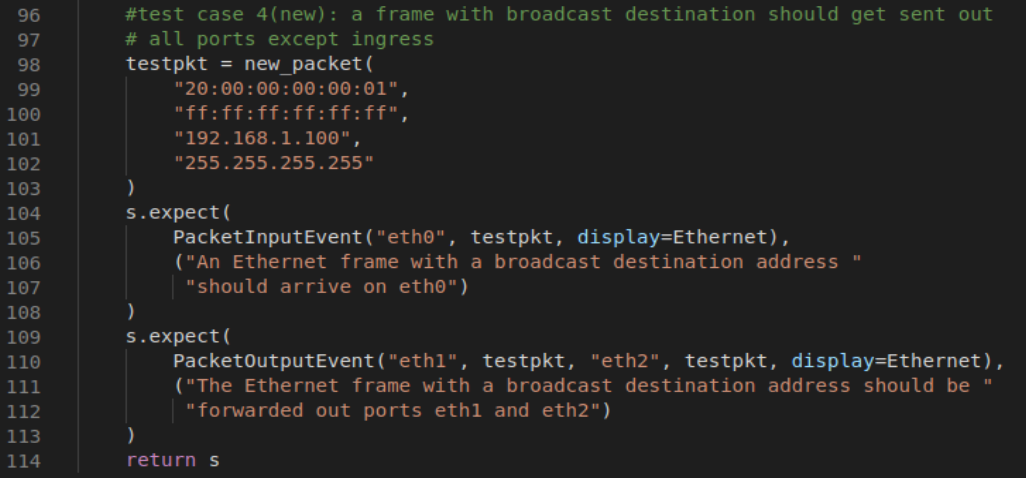






* 1. Step 3: Creating new test case

From the example testcases I can infer there are three types of cases, they are broadcast, any unicast address and a dest address among the interfaces. I chose to write another testcase using broadcast destination, which arrives on eth0 and was forwarded out through ports eth1 and eth2.



1. **总结与感想**
   1. When trying to understand the structure of the virtual network and how the test cases work, it is useful to read the examples carefully, analyze the similarities and differences between them, and test the guessing through running.
   2. Certain phenomenon deserve attention for there are rationales behind them, for example, the drop rate of command “pingall” differs before and after running myhub.py because of the links between nodes. Moreover, the rate itself has to do with the features of hub, that it works on a second level and does not have an IP address to make a packet. When coming across this kind or results during testing, I should pause to think about the logic behind them.