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	211220079	姓名	谷石磊
Email	211220079@smail.nju.edu.cn	开始/完成日期	2023.3.17

1. 实验名称

Lab 1: Switchyard & Mininet

2. 实验目的

In this lab assignment, you will gradually master our experimental environment.

3. 实验内容

Task 1: Get Ready

Meet the requirements for using Linux, Python and Git. We assume that you have a basic understanding of these contents.

Task 2: An Example

Meet the requirements for using Mininet, Wireshark and Switchyard. At the meantime you will learn how to complete this assignment.

Task 3: GitHub Classroom

Sign up on GitHub and prepare for submission.

Task 4: Modification

Step 1: Modify the Mininet topology

In the section Mininet, we introduced how to construct a topology. So here we have two options for you, choose one to implement. Then show the details of how you built the topology in your report.

Delete server2 in the topology,

Or create a different topology containing 6 nodes using hosts and hubs (don't use other kinds of devices).

The file you need to modify is start_mininet.py.

Step 2: Modify the logic of a device

In the section Switchyard, we introduced how to program a device. Your task is to count how many packets have passed through a hub in and out. You need to record the statistical result every time you receive one packet with the format of each line in:<ingress packet count> out:<egress packet count>. For example, if there is a packet that is not addressed to the hub itself, then the hub may log in:1 out:2. Then show the log of your hub when running it in Mininet and how you implemented it in your report.

The file you need to modify is myhub.py.

Step 3: Modify the test scenario of a device

In the section Switchyard, we introduced how to write the test case. So here we have two options for you, choose one to implement. Then show the details of your test cases in your report.

Create one test case by using the given function new_packet with

different arguments,

Or create one test case with your handmade packet.

The file you need to modify is testcases/myhub_testscenario.py.

Step 4: Run your device in Mininet

In the section Switchyard, we introduced how to run Switchyard programs in Mininet. So run your new hub in your new topology and make sure it works. Show the procedure in your report.

Step 5: Capture using Wireshark

Both in section Wireshark and Switchyard, we introduced how to capture packets. In your own topology, capture packets on one host (or the hub) while creating some traffic. Save your capture file and submit it with your report and code. Also you need to describe the details of your capture file.

4. 实验结果

Task4: Modification

Step 1: Modify the Mininet topology

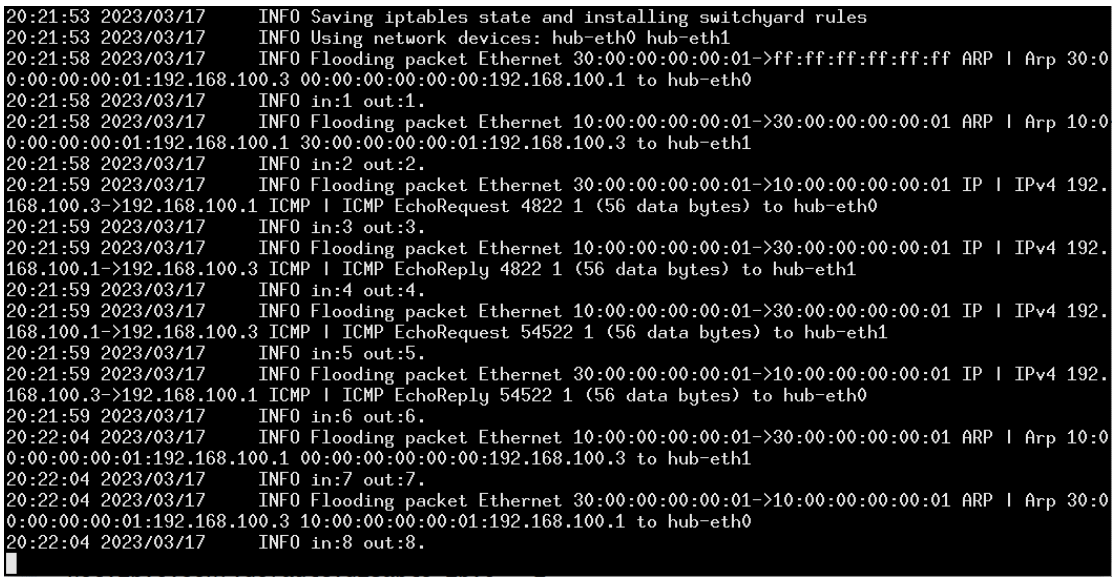
I delete server2 in the topology, this is the new topology.

server1----hub----client

Step 2: Modify the logic of a device

I add two variables in this code, the one is used to keep track of the

number of packets received, and the other is used to keep track of the number of packets sent. Every time I receive one packet, I will print them in the given format. The following figure shows the execution result.



Step 3: Modify the test scenario of a device

I create a new test case. In this test case. I send a packet to eth1. The following figure shows the test result.

```
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 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 should arrive on eth1 with destination
    address the same as eth1's MAC address
10  The hub should not do anything in response to a frame
    arriving with a destination address referring to the hub
    itself.


All tests passed!
```

Step 4: Run your device in Mininet

First, start the mininet.

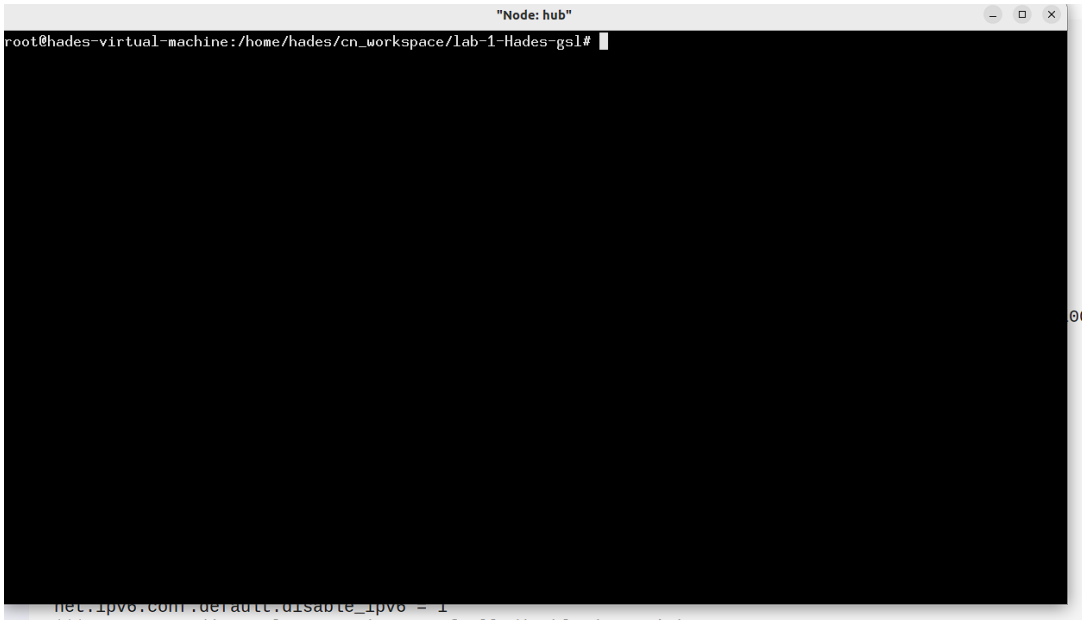
```
(syenv) hades@hades-virtual-machine:~/cn_workspace/lab-1-Hades-gsl$ sudo python3 start_mininet.py
[sudo] password for hades:
*** Creating network
*** Adding hosts:
client hub server1
*** Adding switches:

*** Adding links:
(10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (client, hub) (10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (server1, hub)
*** Configuring hosts
client hub server1
server1 server1-eth0 10:00:00:00:00:01
client client-eth0 30:00:00:00:00:01
hub hub-eth0 40:00:00:00:00:01
hub hub-eth1 40:00:00:00:00:02
*** client : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** client : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
net.ipv6.conf.default.disable_ipv6 = 1
*** hub : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** hub : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
net.ipv6.conf.default.disable_ipv6 = 1
*** server1 : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** server1 : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
net.ipv6.conf.default.disable_ipv6 = 1
*** Starting controller

*** Starting 0 switches

*** Starting CLI:
mininet>
```

Second, start the xterm in mininet.

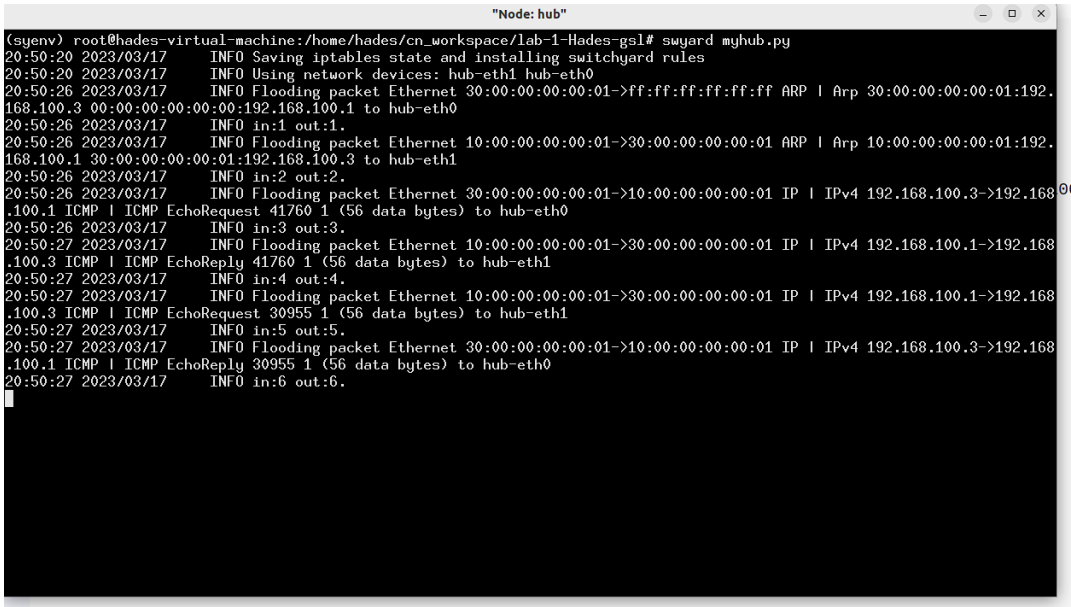



```
net.ipv6.conf.default.disable_ipv6 = 1
*** server1 : ('sysctl -w net.ipv6.conf.all.disable_ipv6=1',)
net.ipv6.conf.all.disable_ipv6 = 1
*** server1 : ('sysctl -w net.ipv6.conf.default.disable_ipv6=1',)
net.ipv6.conf.default.disable_ipv6 = 1
*** Starting controller

*** Starting 0 switches

*** Starting CLI:
mininet> xterm hub
```

Third, run swyard program in the xterm.




```
*** 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)
```

Step 5: Capture using Wireshark

I use ping to get some data packets. In this capture file, there are four ICMP protocol data packets and four ARP protocol data packets. Two ICMP protocol packets are ping request and the other two are corresponding replies. Two ARP protocol packets are queries about MAC addresses and the other two are corresponding responses.

5. 核心代码

Step 3: Modify the test scenario of a device

```
MYPKT = NEW_PACKET(
    "20:00:00:00:00:03",
    "10:00:00:00:00:02",
    "192.168.1.101",
    "172.16.42.1"
)
S.EXPECT(
    PACKETINPUTEVENT("ETH1", MYPKT, DISPLAY=ETHERNET),
    ("AN ETHERNET FRAME SHOULD ARRIVE ON ETH1 WITH DESTINATION ADDRESS "
     "THE SAME AS ETH1'S MAC ADDRESS")
)
S.EXPECT(
    PACKETINPUTTIMEOUTEVENT(2.0),
    ("THE HUB SHOULD NOT DO ANYTHING IN RESPONSE TO A FRAME ARRIVING WITH"
     " A DESTINATION ADDRESS REFERRING TO THE HUB ITSELF.")
)
```

6. 总结与感想

This lab is a challenge for me. There are so many things for me to learn. In the process of completing it, I learned a lot.