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：方毓楚、郑浩、陈伟等（排名不分先后）

学院	地理与海洋科学学院	专业	地理信息科学
学号	191830173	姓名	徐嵩
Email	song.xv@outlook.com	开始/完成日期	2020 年 3 月 13 日

1. 实验名称

Lab 2: Learning Switch

2. 实验目的

In this assignment, you are going to implement the core functionalities of an Ethernet learning switch using the Switchyard framework.

3. 实验内容

Task 1: Preparation

Task 2: Basic Switch

Task 3: Timeouts

Task 4: Least Recently Used

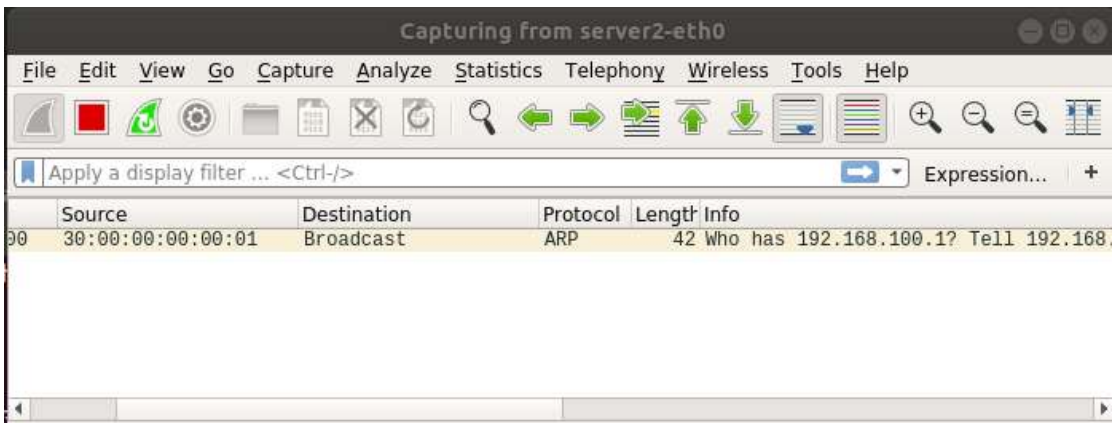
Task 5: Least Traffic Volume

4. 实验结果

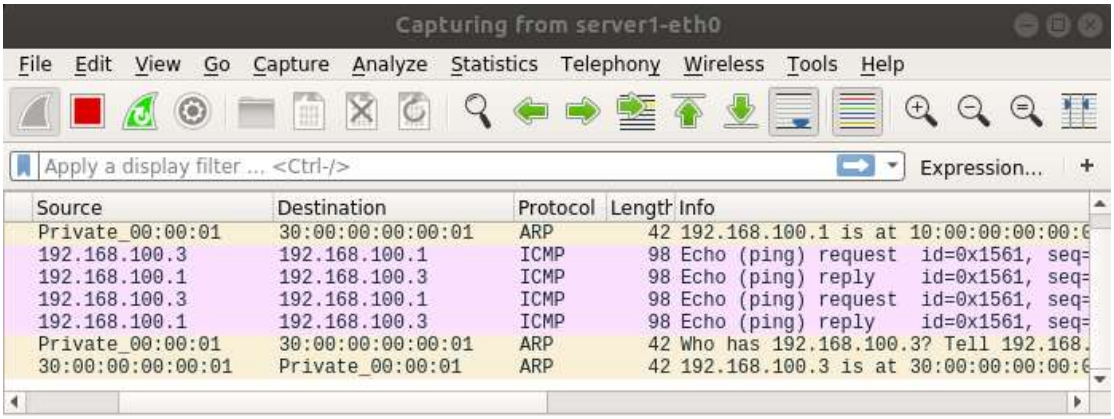
Basic Switch:

Analyze and state the results of the above process in your report with screenshots. Do not explain how you do step by step but focus on the switch's forwarding logic.

答：server2 wireshark 截图：



Server1 wireshark 截图：



从截图中可以看出，使用 client ping -c 2 192.168.100.1 交换机首先广播 ARP 包，在交换机内部维护的表中注册了 client 设备对应的接口。Server1 和 server2 都收到这个包，server1 回复后，在交换机表中注册 server1 对应的接口。

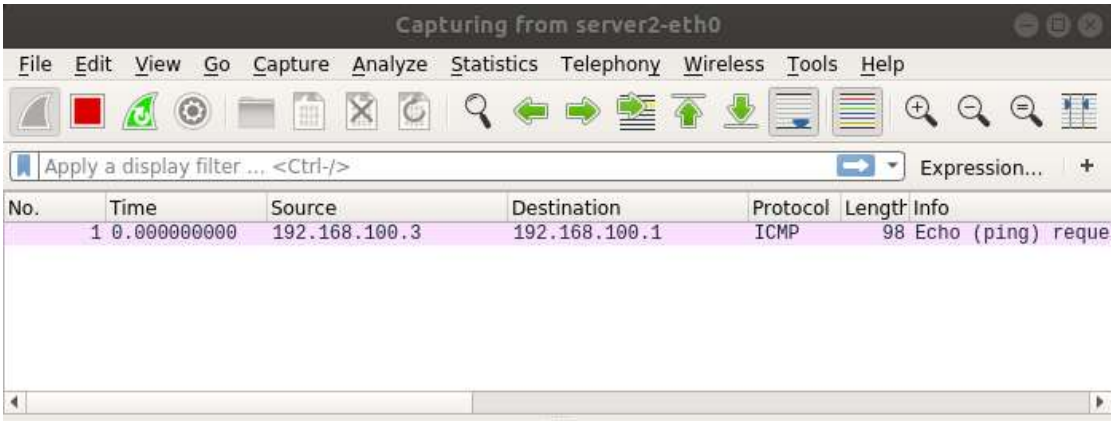
之后 server1 与 client 之间的 ICMP 包等不在像集线器一样广播到所有接口，wireshark 不能在 server2 捕捉到其他包。

Timeouts:

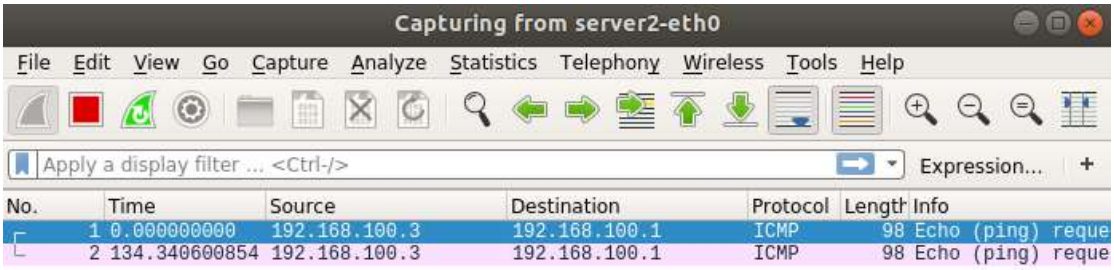
In the report, show the test result of your switch.

```
Results for test scenario switch tests: 9 passed, 0 failed, 0 pending
```

In Mininet, test your timeout mechanism. Prove that the timeout mechanism works with your testing procedure in the report.



使用命令 client ping -c 2 server1 对 server2 打开 wireshark，发现 server2 只收到一个 ICMP 报文，说明 switch 在收到 server1 回复后，将之后目的地址是 server1 的包只发送到 server1 的端口。



一段时间后，再次使用同一条命令，server2 再次收到一条 ICMP 报文，说明交换机之前学习的 server1 对应的接口已经过期，再次从所有接口转发出去。

Least Recently Used:

In the report, show the test result of your switch.

```
Results for test scenario switch tests: 18 passed, 0 failed, 0 pending
```

In Mininet, test the LRU algorithm. Prove that the LRU algorithm works with your testing procedure in the report.

创建 mininet 拓扑，包含 5 个 server 和 1 个 client，1 个 switch。

依次用 client ping -c 2 serverX 命令（X 范围 1~4）得到如下 switchtable：

```
15:42:59 2022/03/15     INFO Get MAC from mytable, MAC: 60:00:00:00:00:01 is at switch-eth4
15:42:59 2022/03/15     INFO Send packet Ethernet 30:00:00:00:00:01->60:00:00:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.100.3 60:00:00:00:00:01:192.168.100.5
 to switch-eth4
[[EthAddr('30:00:00:00:00:01'), 'switch-eth2'], [EthAddr('60:00:00:00:00:01'), 'switch-eth4'], [EthAddr('50:00:00:00:00:01'), 'switch-eth3'], [EthAddr('20:00:00
:00:00:01'), 'switch-eth1'], [EthAddr('10:00:00:00:00:01'), 'switch-eth0']]
```

此时如果使用 client ping -c 2 server5，则 server1 对应的 MAC（10 00 00 00 00 00）和接口 switch-eth0 将被删除，因为 server1 对应最近最少使用的接口。

```
15:50:02 2022/03/15     INFO Get MAC from mytable, MAC: 70:00:00:00:00:01 is at switch-eth5
15:50:02 2022/03/15     INFO Send packet Ethernet 30:00:00:00:00:01->70:00:00:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.100.3 70:00:00:00:00:01:192.168.100.5
 to switch-eth5
[[EthAddr('30:00:00:00:00:01'), 'switch-eth2'], [EthAddr('70:00:00:00:00:01'), 'switch-eth5'], [EthAddr('60:00:00:00:00:01'), 'switch-eth4'], [EthAddr('50:00:00
:00:00:01'), 'switch-eth3'], [EthAddr('20:00:00:00:00:01'), 'switch-eth1']]
```

结果如图，此时末尾是 server2 对应的 MAC 和接口。

Least Traffic Volume:

In the report, show the test result of your switch.

```
Results for test scenario switch tests: 8 passed, 0 failed, 0 pending
```

In Mininet, test the least traffic volume algorithm. Prove that the least traffic volume algorithm works with your testing procedure in the report.

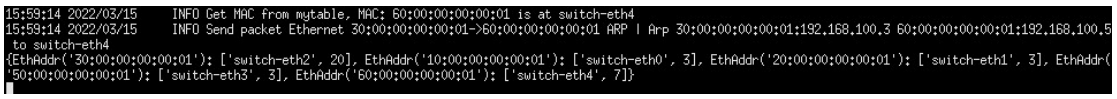
创建 mininet 拓扑，包含 5 个 server 和 1 个 client，1 个 switch。

依次用 client ping -c 2 serverX 命令（X 范围 1~4）得到如下 switchtable：

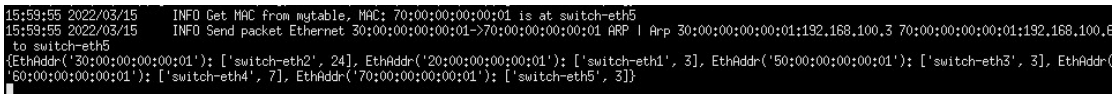
```
15:54:56 2022/03/15     INFO Get MAC from mytable, MAC: 60:00:00:00:00:01 is at switch-eth4
15:54:56 2022/03/15     INFO Send packet Ethernet 30:00:00:00:00:01->60:00:00:00:00:01 ARP | Arp 30:00:00:00:00:01:192.168.100.3 60:00:00:00:00:01:192.168.100.5
 to switch-eth4
{EthAddr('30:00:00:00:00:01'): ['switch-eth2', 16], EthAddr('10:00:00:00:00:01'): ['switch-eth0', 3], EthAddr('20:00:00:00:00:01'): ['switch-eth1', 3], EthAddr(
'50:00:00:00:00:01'): ['switch-eth3', 3], EthAddr('60:00:00:00:00:01'): ['switch-eth4', 3]}
```

此时 client 流量是最多的 16，其他 server1~4 分别有 3 的流量，如果此时使用命令 client ping -c2 server5 将删除表中的最后一项，即 server4 对应的 MAC 地址和接口。

所以，使用命令 client ping -c 2 server4 为 server4 增加一些流量，结果如下图：



此时再使用命令 client ping -c 2 server5 将删除 server3 对应的 MAC 和接口，因为此时 server4 的流量是 7 不再是最小流量。结果如下图：



5. 核心代码

超时算法

```
# Table!!
current = time.time()
totable[eth.src] = [fromIface, current]

# Table!!
dellist = []
for k,v in totable.items():
    if current - v[1] > 10:
        dellist.append(k)
for k in dellist:
    totable.pop(k)

# Table!!
if eth.dst in totable:
    target = totable.get(eth.dst)[0]
    log_info (f"Get MAC from mytable, MAC: {eth.dst} is at {target}")
    log_info (f"Send packet {packet} to {target}")
    net.send_packet(target, packet)
else:
    for intf in my_interfaces:
        if fromIface!= intf.name:
            log_info (f"Flooding packet {packet} to {intf.name}")
            net.send_packet(intf, packet)
```

最近最少使用算法

```
# List!!
isnew = True
num = 0
for item in lrulist:
    if item[0] == eth.src:
        isnew = False
        break
    num += 1
if isnew:
    if len(lrulist) == 5:
        lrulist.pop()
else:
    lrulist.pop(num)
lrulist.insert(0,[eth.src, fromIface])

if eth is None:
    log_info("Received a non-Ethernet packet?!")
    return
if eth.dst in mymacs:
    log_info("Received a packet intended for me")
    continue

# List!!
Exists = False
num = 0
for item in lrulist:
    if item[0] == eth.dst:
        Exists = True
        break
    num += 1
if Exists:
    target = lrulist[num][1]
    log_info (f"Get MAC from mytable, MAC: {eth.dst} is at {target}")
    log_info (f"Send packet {packet} to {target}")
    net.send_packet(target, packet)
```

最小流量算法

```
# Table!!
if (eth.src in trtable) == False:
    if len(trtable) == 5:
        k = min(trtable.items(), key=lambda x: x[1][1])[0]
        trtable.pop(k)
    trtable[eth.src] = [fromIface, 0]


if eth is None:
    log_info("Received a non-Ethernet packet?!")
    return
if eth.dst in mymacs:
    log_info("Received a packet intended for me")
    continue

# Table!!
if eth.dst in trtable:
    target = trtable.get(eth.dst)[0]
    trtable.get(eth.dst)[1] += 1
    log_info (f"Get MAC from mytable, MAC: {eth.dst} is at {target}")
    log_info (f"Send packet {packet} to {target}")
    net.send_packet(target, packet)
else:
    for intf in my_interfaces:
        if fromIface!= intf.name:
            log_info (f"Flooding packet {packet} to {intf.name}")
            net.send_packet(intf, packet)
```

6. 总结与感想

学会了实现交换机的学习功能。