

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	电子科学与工程学院	专业（方向）	电子信息科学与技术
学号	191180102	姓名	任晟昊
Email	191180102@smail.nju.edu.cn	开始/完成日期	2022.03.06

1. 实验名称

Lab1

2. 实验目的

对实验环境进行搭建，对实验工具进行熟悉

3. 实验内容

Step1

对 `start_mininet.py` 文件进行修改，删除 `server2`

Step2

在 `myhub.py` 文件中进行修改，加入 `in`, `out` 计数功能

Step3

在 `myhub_testscenario.py` 文件中修改，增加一个自己的 `new_packet`

Step4

在 `mininet` 中用自己的 `topology` 试验自己写的 `hub`

Step5

综合以上几步，利用 `wireshark` 保存数据包，并查看对应的数据。

4. 实验结果

Step1

由于代码中是统一列出了 `server1`, `server2` 和 `client` 三个节点，通过注释掉 `server2` 相关的 `mac` 和 `ip` 就可以删除 `server2`

```

*** Creating network
*** Adding hosts:
client hub server1
*** Adding switches:

*** Adding links:
(10.00Mbit 100ms delay) (10.00Mbit 100ms delay) (client, hub) (10.00Mbit 100ms d
elay) (10.00Mbit 100ms delay) (server1, hub)
*** Configuring hosts
client hub server1
('client', <TCIntf client-eth0>, '30:00:00:00:00:01')
('server1', <TCIntf server1-eth0>, '10:00:00:00:00:01')
('hub', <TCIntf hub-eth0>, '40:00:00:00:00:01')
('hub', <TCIntf 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

```

```

mininet> nodes
available nodes are:
client hub server1
mininet> net
client client-eth0:hub-eth0
hub hub-eth0:client-eth0 hub-eth1:server1-eth0
server1 server1-eth0:hub-eth1

```

Step2

由于hub在实验中起到一个收发的功能，所以为了统计进hub和出hub的包数，我们需要在入hub和出hub函数出，加入计数功能。具体见核心代码。

```

mininet> xterm hub
mininet> pingall
*** Ping: testing ping reachability
client -> X server1
hub -> X X
server1 -> client X
*** Results: 66% dropped (2/6 received)

```

```

root@njucs-VirtualBox:~/switchyard/lab1/lab-01-wjrz# source /home/njucs/switchyard/syenv/bin/activate
(syenv) root@njucs-VirtualBox:~/switchyard/lab1/lab-01-wjrz# swyard myhub.py
15:02:08 2022/03/06      INFO Saving iptables state and installing switchyard rules
15:02:08 2022/03/06      INFO Using network devices: hub-eth0 hub-eth1
15:02:16 2022/03/06      INFO Flooding packet Ethernet 30:00:00:00:00:01->ff:ff:ff:ff:ff:ff ARP | Arp 30:00:00:00:00:01:192.168.100.3 00:00:00:00:00:00:192.168.100.1 to hub-eth1
15:02:16 2022/03/06      INFO in: 1 out: 1
15:02:17 2022/03/06      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 30:00:00:00:00:01:192.168.100.3 to hub-eth0
15:02:17 2022/03/06      INFO in: 2 out: 2
15:02:17 2022/03/06      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 EchoRequest 5165 1 (56 data bytes) to hub-eth1
15:02:17 2022/03/06      INFO in: 3 out: 3
15:02:17 2022/03/06      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 5165 1 (56 data bytes) to hub-eth0
15:02:17 2022/03/06      INFO in: 4 out: 4
15:02:17 2022/03/06      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 5168 1 (56 data bytes) to hub-eth0
15:02:17 2022/03/06      INFO in: 5 out: 5
15:02:18 2022/03/06      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 5168 1 (56 data bytes) to hub-eth1
15:02:18 2022/03/06      INFO in: 6 out: 6
15:02:22 2022/03/06      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-eth0
15:02:22 2022/03/06      INFO in: 7 out: 7
15:02:23 2022/03/06      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-eth1
15:02:23 2022/03/06      INFO in: 8 out: 8

```

Step3

增加一个发向eth0节点的包，使得包最后不会从其他节点出去

```

(syenv) njucs@njucs-VirtualBox:~/switchyard/lab1/lab-01-wjrz$ swyard -t testcases/myhub_testscenario.py myhub.py
17:06:58 2022/03/06      INFO Starting test scenario testcases/myhub_testscenario.py
17:06:58 2022/03/06      INFO Flooding packet Ethernet 30:00:00:00:00:02->ff:ff:ff:ff:ff:ff IP | IPv4 172.16.42.2->255.255.255.255 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth0
17:06:58 2022/03/06      INFO in: 1 out: 1
17:06:58 2022/03/06      INFO Flooding packet Ethernet 30:00:00:00:00:02->ff:ff:ff:ff:ff:ff IP | IPv4 172.16.42.2->255.255.255.255 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth2
17:06:58 2022/03/06      INFO in: 1 out: 2
17:06:58 2022/03/06      INFO Flooding packet Ethernet 20:00:00:00:00:01->30:00:00:00:00:02 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth1
17:06:58 2022/03/06      INFO in: 2 out: 3
17:06:58 2022/03/06      INFO Flooding packet Ethernet 20:00:00:00:00:01->30:00:00:00:00:02 IP | IPv4 192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 0 (0 data bytes) to eth2
17:06:58 2022/03/06      INFO in: 2 out: 4
17:06:58 2022/03/06      INFO Flooding packet Ethernet 30:00:00:00:00:02->20:00:00:00:00:01 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 data bytes) to eth0
17:06:58 2022/03/06      INFO in: 3 out: 5
17:06:58 2022/03/06      INFO Flooding packet Ethernet 30:00:00:00:00:02->20:00:00:00:00:01 IP | IPv4 172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 0 (0 data bytes) to eth2
17:06:58 2022/03/06      INFO in: 3 out: 6
17:06:58 2022/03/06      INFO Received a packet intended for me
17:06:59 2022/03/06      INFO Received a packet intended for me

Results for test scenario hub tests: 10 passed, 0 failed, 0 pending

```

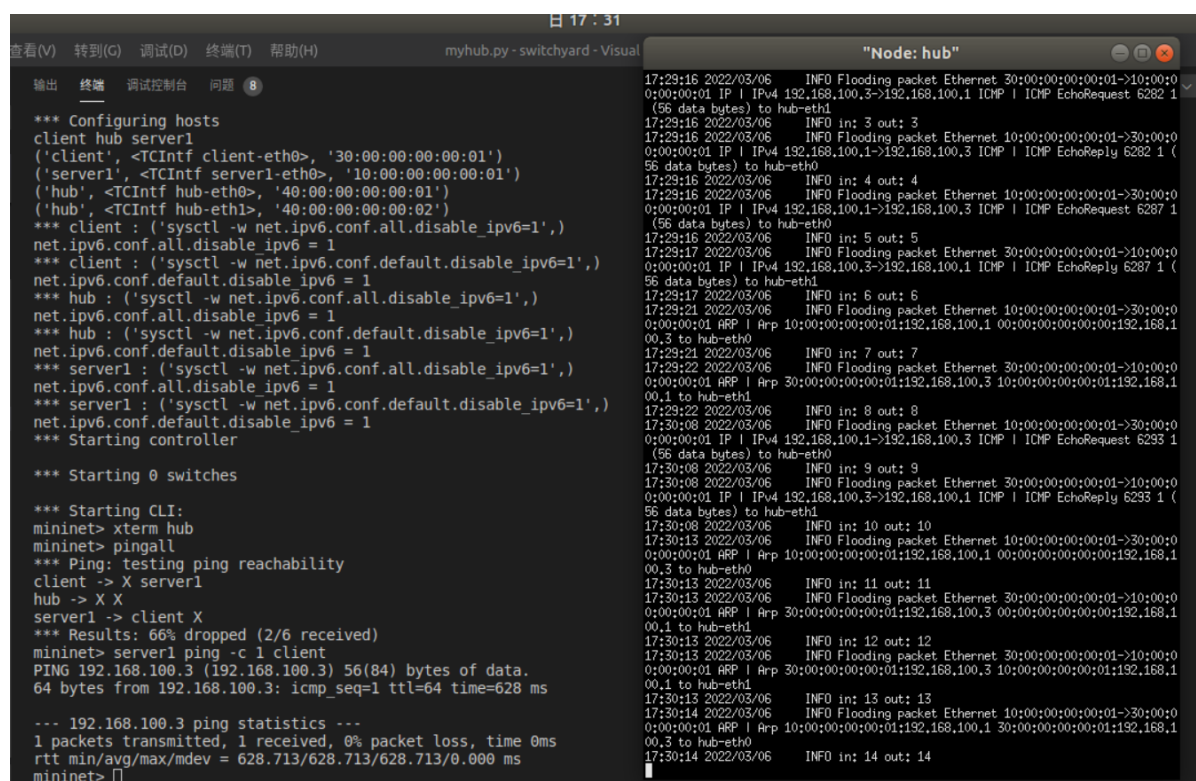
Passed:

- 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 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 should arrive on eth2 with destination address the same as eth0'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!

Step4

在终端中 `sudo python start_mininet.py` 配置了三个节点，在mininet中 `xterm hub` 在nodehub中，进入虚拟环境，再 `swyard myhub.py`，对这个hub进行配置，写好了如何进行收包和出包的工作。这时相当于有了一个中间节点，再进行pingall操作，得到了对应的信息。



```
myhub.py - switchyard - Visual
17:29:16 2022/03/06 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoRequest 6282 1 (56 data bytes) to hub-eth1
17:29:16 2022/03/06 INFO in: 3 out: 3
17:29:16 2022/03/06 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoReply 6282 1 (56 data bytes) to hub-eth0
17:29:16 2022/03/06 INFO in: 4 out: 4
17:29:16 2022/03/06 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoRequest 6287 1 (56 data bytes) to hub-eth0
17:29:16 2022/03/06 INFO in: 5 out: 5
17:29:17 2022/03/06 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoReply 6287 1 (56 data bytes) to hub-eth1
17:29:17 2022/03/06 INFO in: 6 out: 6
17:29:21 2022/03/06 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 00:00:00:00:00:192.168.100.3 to hub-eth0
17:29:21 2022/03/06 INFO in: 7 out: 7
17:29:22 2022/03/06 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:01 ARP | Arp 30:00:00:00:01:192.168.100.3 10:00:00:00:01:192.168.100.1 to hub-eth1
17:29:22 2022/03/06 INFO in: 8 out: 8
17:30:08 2022/03/06 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:01 IP | IPv4 192.168.100.1->192.168.100.3 ICMP | ICMP EchoRequest 6293 1 (56 data bytes) to hub-eth0
17:30:08 2022/03/06 INFO in: 9 out: 9
17:30:08 2022/03/06 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:01 IP | IPv4 192.168.100.3->192.168.100.1 ICMP | ICMP EchoReply 6293 1 (56 data bytes) to hub-eth1
17:30:08 2022/03/06 INFO in: 10 out: 10
17:30:13 2022/03/06 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 00:00:00:00:00:192.168.100.3 to hub-eth0
17:30:13 2022/03/06 INFO in: 11 out: 11
17:30:13 2022/03/06 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:01 ARP | Arp 30:00:00:00:01:192.168.100.3 10:00:00:00:01:192.168.100.1 to hub-eth1
17:30:13 2022/03/06 INFO in: 12 out: 12
17:30:13 2022/03/06 INFO Flooding packet Ethernet 30:00:00:00:01->10:00:00:00:01 ARP | Arp 30:00:00:00:01:192.168.100.3 10:00:00:00:01:192.168.100.1 to hub-eth1
17:30:13 2022/03/06 INFO in: 13 out: 13
17:30:14 2022/03/06 INFO Flooding packet Ethernet 10:00:00:00:01->30:00:00:00:01 ARP | Arp 10:00:00:00:00:01:192.168.100.1 00:00:00:00:00:192.168.100.3 to hub-eth0
17:30:14 2022/03/06 INFO in: 14 out: 14

*** Configuring hosts
client hub server1
('client', <TCIntf client-eth0>, '30:00:00:00:00:01')
('server1', <TCIntf server1-eth0>, '10:00:00:00:00:01')
('hub', <TCIntf hub-eth0>, '40:00:00:00:00:01')
('hub', <TCIntf 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

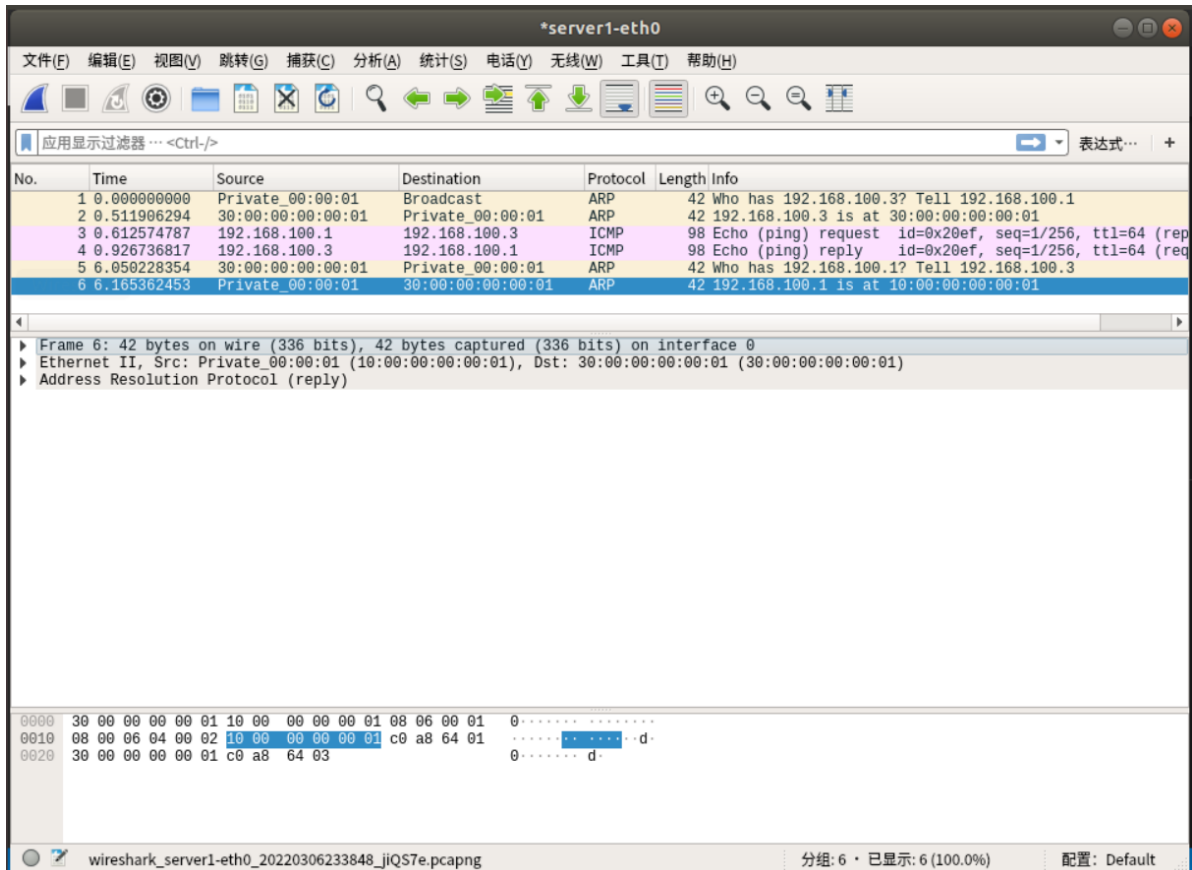
*** Starting 0 switches

*** Starting CLI:
mininet> xterm hub
mininet> pingall
*** Ping: testing ping reachability
client -> X server1
hub -> X X
server1 -> client X
*** Results: 66% dropped (2/6 received)
mininet> server1 ping -c 1 client
PING 192.168.100.3 (192.168.100.3) 56(84) bytes of data.
64 bytes from 192.168.100.3: icmp_seq=1 ttl=64 time=628 ms

--- 192.168.100.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 628.713/628.713/628.713/0.000 ms
mininet>
```

Step5

输入server1 wireshark &启动wireshark, 然后server1 ping -c 1 client得到了如下结果

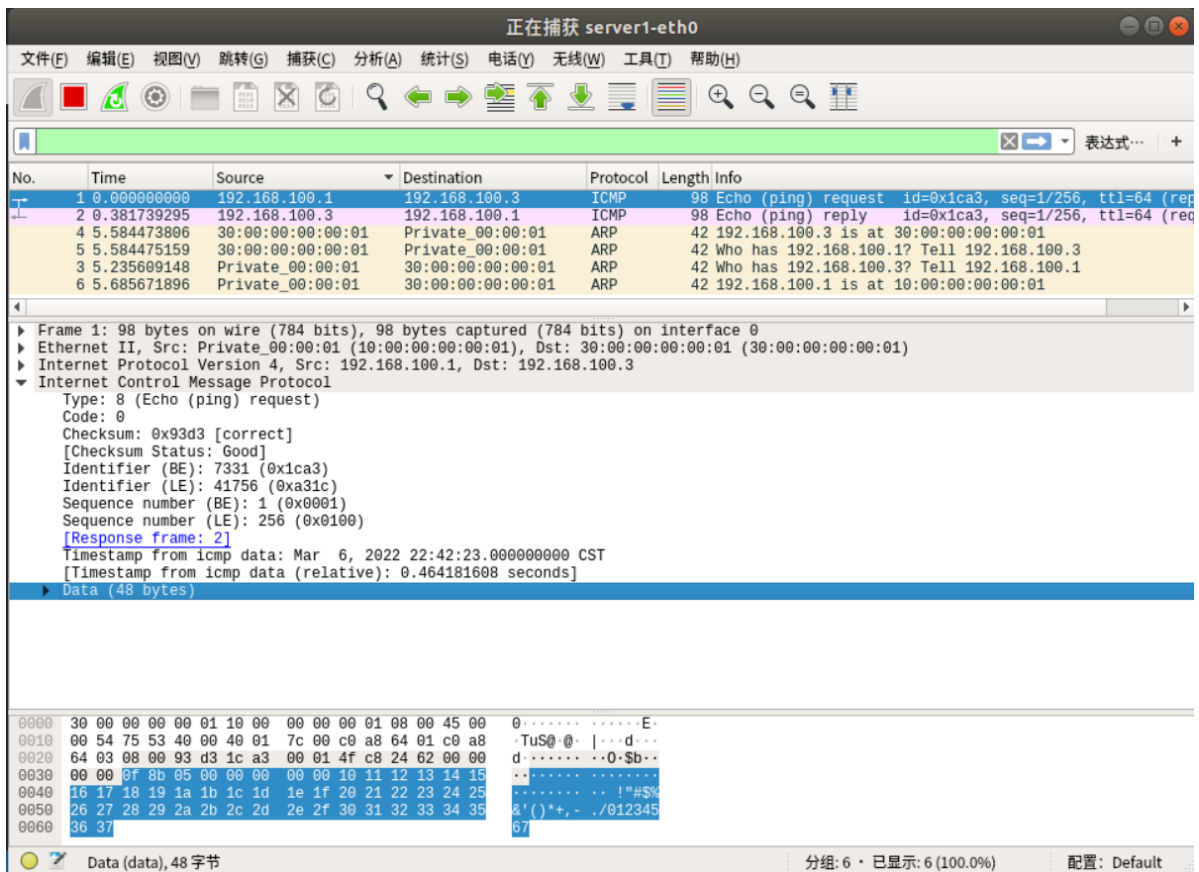


Frame1是指本次发送的数据块有98字节, 里面有关于数据的众多参数。

Ethernet2是数据链路层, 是指从10:00:00:00:00:01发送到30:00:00:00:00:01, 类型是IPv4。

IPv4协议, 是网络层, 是从IP地址为192.168.100.1发送到192.168.100.3, 以及众多网络层参数。

ICMP协议介绍了其中的参数



5. 核心代码

Step1

```
# "server2": {
#     "mac": "20:00:00:00:00:{:02x}",
#     "ip": "192.168.100.2/24"
# },
```

Step2

```
while True:
    try:
        _, fromIface, packet = net.recv_packet()
        incnt += 1
    except NoPackets:
        continue
    except Shutdown:
        break

    log_debug (f"In {net.name} received packet {packet} on {fromIface}")
    eth = packet.get_header(Ethernet)
    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")
    else:
        for intf in my_interfaces:
            if fromIface != intf.name:
                log_info (f"Flooding packet {packet} to {intf.name}")
```

```
net.send_packet(intf, packet)
outcnt += 1
log_info (f"in: {incnt} out: {outcnt}")
```

Step3

```
# my test case
testpkt = new_packet(
    "20:00:00:00:00:01",
    "10:00:00:00:00:01",
    '192.168.1.100',
    '172.16.42.2'
)
s.expect(
    PacketInputEvent("eth0", testpkt, display=Ethernet),
    ("An Ethernet frame should arrive on eth0 with destination address "
     "the same as eth0's MAC address")
)
s.expect(
    PacketInputTimeoutEvent(1.0),
    ("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

由于之前接触过Linux系统，外加一直是VScode的忠实用户，所以在配置环境方面基本没有遇到问题，只是在配置虚拟机时，导入Extension-Pack，遇到了困难，不过通过强大的Google搜索成功解决了问题。

在解决每一个问题时，由于一些理论知识没有学到，所以自己在理解理论知识上花费了一些时间，由于本实验主要目的是让我们熟悉实验环境，把整个实验流程走一遍。所以其实难度不大，重要的熟悉 `mininet`，`wireshark`，`switchyard` 等工具。