

计算机网络实验报告

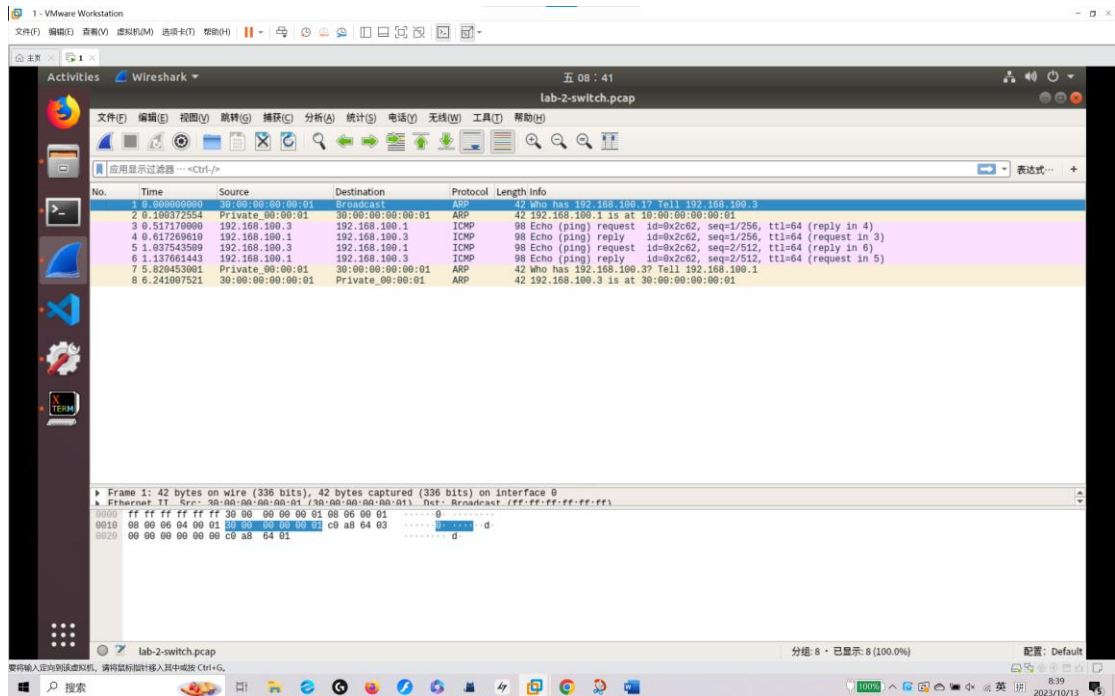
一. 实验名称

Lab2: learning switch

二 实验内容

1.basic——switchyard

首先根据手册要求，我们由 client ping 到 server1，再对 server1 节点进行抓包，得到的抓包结果如下图所示。



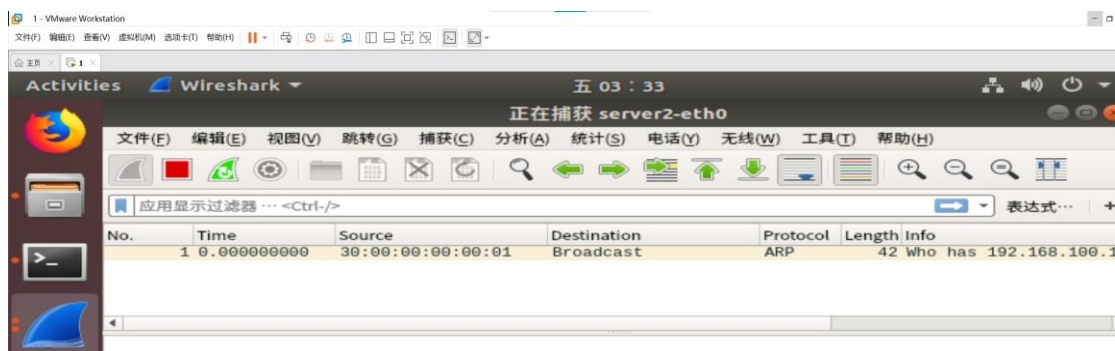
其中总共有 8 个包，

当处理第一个包：client 节点中 arp 为空，通过广播来询问 server1 的 MAC 地址，此时 switch 会将 client 的 mac 和端口对应好并记录。此时第二个包中，server1 听到该 arp 包，将自己的 mac 地址发送给 client，此时 switch 也会学习 server1 的 mac 地址和端口的对应关系，此时 client 和 server1 构建好了联系。

3, 4 相当于一个发送和回复的功能，由于在第一个和第二个包的处理使得其知晓了 client 和 server1，因此因此直接通过已知的 server1 和 client 接口来互相传递 ICMP 包。

5, 6 同上。

7 和 8 的包是为了确认 client 的 mac 地址没有发生变化，任然和原来一样在发送一个 arp 包来确认 client 的位置



Server2 唯一的一个包来自于一开始询问 server1 的 mac 地址的广播。

```

(cymon) njucen@juc-VirtualBox:~/workspace/lab-kayasi20215$ swyard -t testcases/nyawtch-to-testscenario.srpy nyawtch-to-py
03:36:10 2023/10/13 INFO Starting test scenario testcases/nyawtch-to-testscenario.srpy
03:36:10 2023/10/13 INFO Flooding packet Ethernet 30:00:00:00:00:02 -ffff:ffff:ffff:ffff IP 1Pv4 172.16.42.2 -255.255.255.255 ICMP 1CMP EchoRequest 0 0 (0 data bytes) to eth0
03:36:10 2023/10/13 INFO Flooding packet Ethernet 30:00:00:00:00:02 -ffff:ffff:ffff:ffff IP 1Pv4 172.16.42.2 -255.255.255.255 ICMP 1CMP EchoRequest 0 0 (0 data bytes) to eth2
03:36:10 2023/10/13 INFO Flooding packet Ethernet 20:00:00:00:00:01 -30:00:00:00:00:02 IP 1Pv4 192.168.1.100 -172.16.42.2 ICMP 1CMP EchoRequest 0 0 (0 data bytes) to eth1
03:36:10 2023/10/13 INFO Flooding packet Ethernet 20:00:00:00:00:01 -30:00:00:00:00:02 IP 1Pv4 192.168.1.100 -172.16.42.2 ICMP 1CMP EchoRequest 0 0 (0 data bytes) to eth1
03:37:13 2023/10/13 INFO Flooding packet Ethernet 20:00:00:00:00:01 -30:00:00:00:00:02 IP 1Pv4 192.168.1.100 -172.16.42.2 ICMP 1CMP EchoRequest 0 0 (0 data bytes) to eth2
03:37:13 2023/10/13 INFO Received a packet intended for me
Results for test scenario switch tests: 9 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 eth0 and eth2
3 An Ethernet frame from 20:00:00:00:00:01 to 20:00:00:00:00:02 should arrive on eth0
4 Ethernet frame destined for 30:00:00:00:00:02 should arrive on eth1 after self-learning
5 Timeout for 60s
6 An Ethernet frame from 20:00:00:00:00:01 to 30:00:00:00:00:02 should arrive on eth0
7 Ethernet frame destined for 30:00:00:00:00:02 should be flooded out eth1 and eth2
8 Ethernet frame should arrive on eth2 with destination address the same as eth2's MAC address
9 The hub should not do anything in response to a frame arriving with a destination address referring to the hub itself.

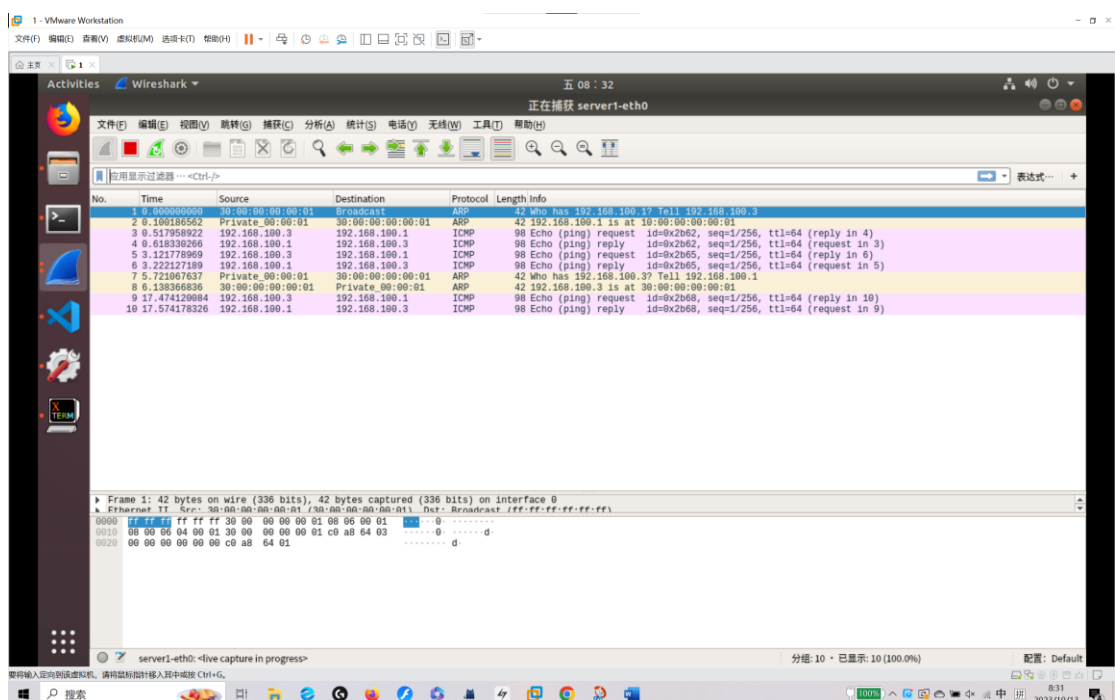
```

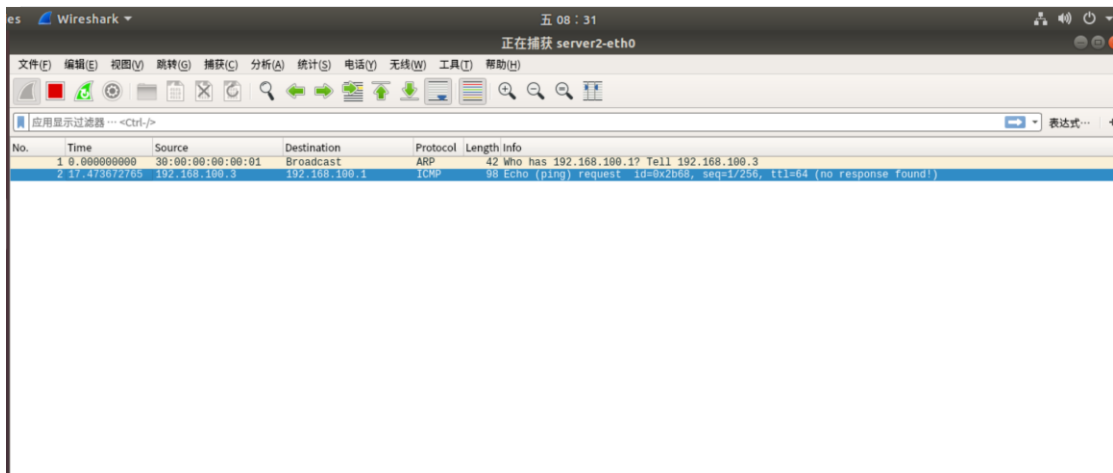
```
mininet> server2 wireshark &
mininet> client ping -c 1 server1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=477 ms

--- 192.168.100.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 477.830/477.830/477.830/0.000 ms
mininet> client ping -c 1 server1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=496 ms

--- 192.168.100.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 496.430/496.430/496.430/0.000 ms
mininet> client ping -c 1 server1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=470 ms

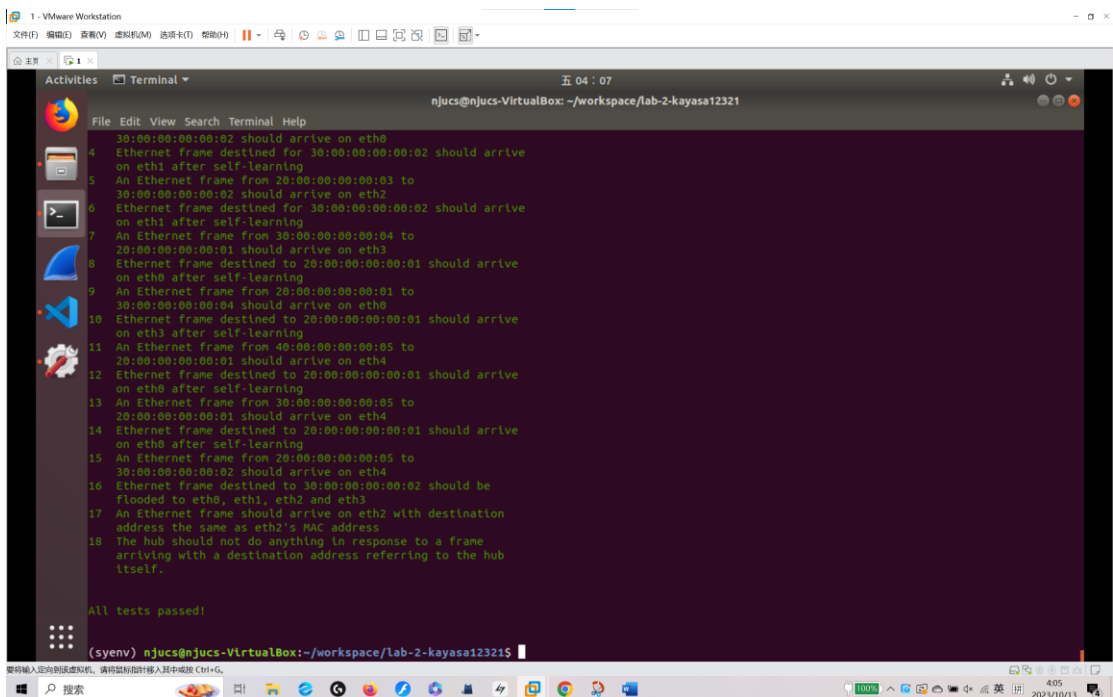
--- 192.168.100.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 470.085/470.085/470.085/0.000 ms
mininet>
```

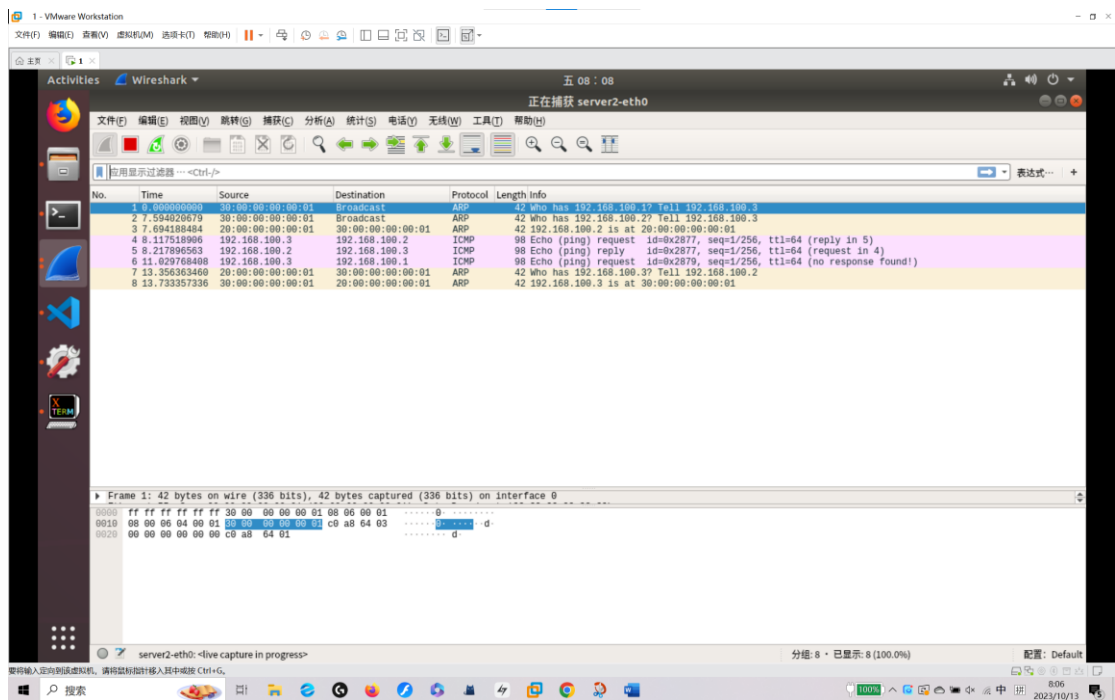




我们可以看出在 6 秒多的时候，server1 会通过 arp 包来确认 client 的位置，之后在超过 10 秒的时候，我们发现由于接口被删除，所以数据包只能被 flood，因而在 server2 中我们可以看到一个新的没有用的 ICMP 包

Task 4: LRU



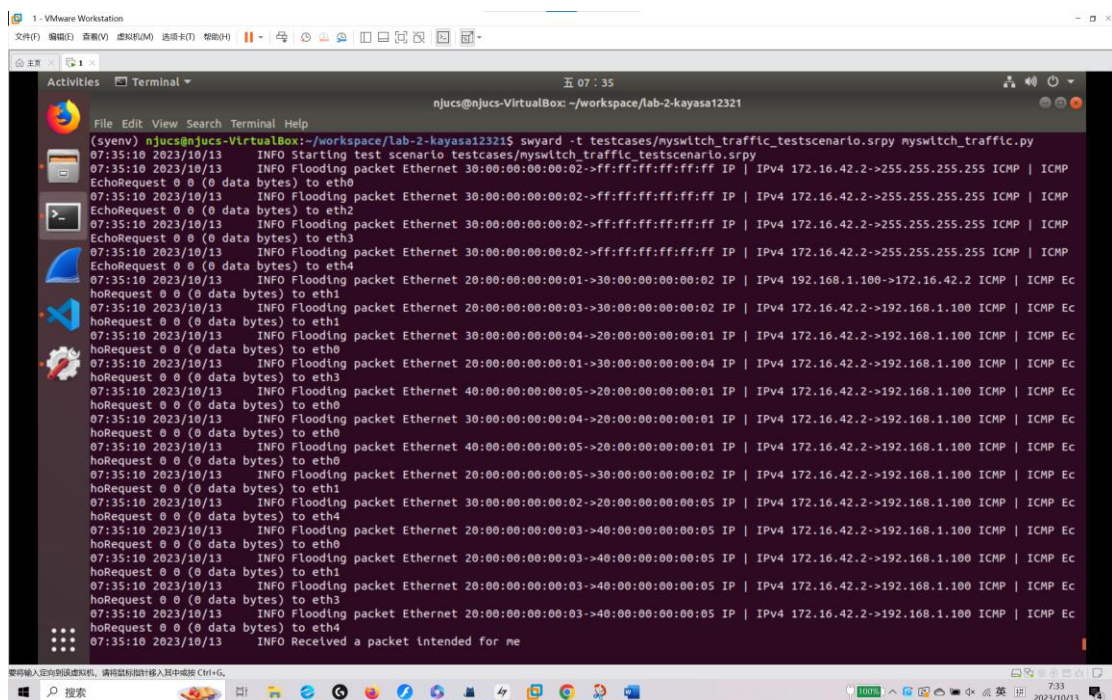


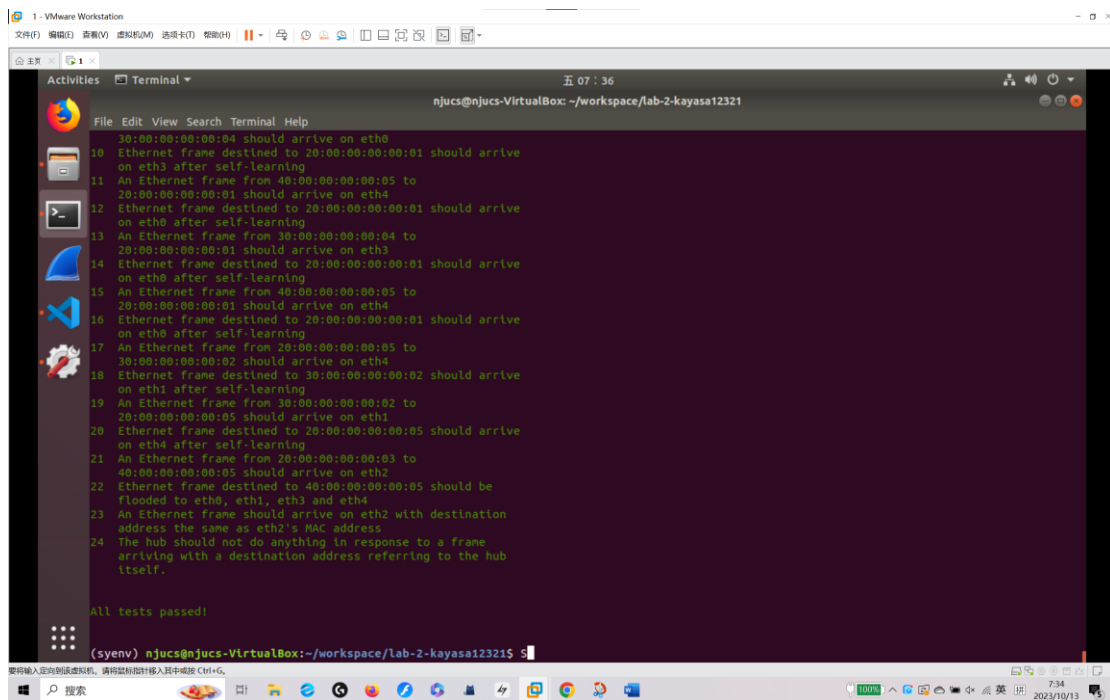
第一次 ping 到 server1 时，server1 中前四个包和 server2 中的第一个包已经同上面对于基本的 switch，此时映射表中有 client 和 server1，

然后由于 client 不知道 server2 的地址，他通过轮询广播来获得 server2 的 mac 地址，捕获文件中 server2 的 2, 3，同时 server1 中 5，然后同构该地址能够发送信息，这时表中更新为 client 和 server2，这时，server1 定期确认 client 位置，再次将 server1 加入关系中，此时更具 lru 删除 server2，因此可以直接从其中进行第二次和第三次的 ping server1

Task5 traffic

测试结果





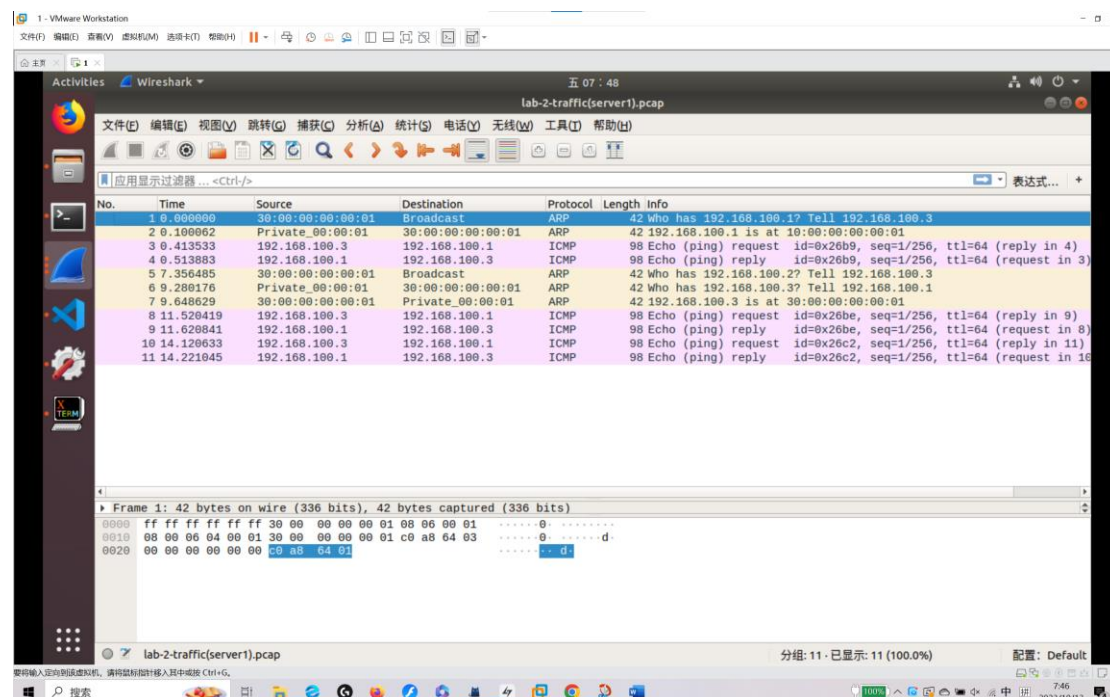
同上面的方法，我们可以暂时修改容量为 2，然后采用和 4 一样的做法，先 pingserver1 和 server2，然后再 ping 两次 server。

Server2:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	30:00:00:00:00:01	Broadcast	ARP	42	Who has 192.168.100.1? Tell 192.168.100.3
2	7.356485972	30:00:00:00:00:01	Broadcast	ARP	42	Who has 192.168.100.2? Tell 192.168.100.3
3	7.456756924	20:00:00:00:00:01	30:00:00:00:00:01	ARP	42	192.168.100.2 is at 20:00:00:00:00:01
4	7.876747796	192.168.100.3	192.168.100.2	ICMP	98	Echo (ping) request id=0x26bb, seq=1/256, ttl=64 (reply in 5)
5	7.977093221	192.168.100.2	192.168.100.3	ICMP	98	Echo (ping) reply id=0x26bb, seq=1/256, ttl=64 (request in 4)
6	13.119993138	20:00:00:00:00:01	30:00:00:00:00:01	ARP	42	Who has 192.168.100.3? Tell 192.168.100.2
7	13.600514451	30:00:00:00:00:01	20:00:00:00:00:01	ARP	42	192.168.100.3 is at 30:00:00:00:00:01
8	14.120631612	192.168.100.3	192.168.100.1	ICMP	98	Echo (ping) request id=0x26c2, seq=1/256, ttl=64 (no response found!)

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0	
0000	ff ff ff ff ff 30 00 00 00 01 08 06 00 01
0010	08 00 06 04 00 01 30 00 00 00 01 c0 ab 64 01
0020	00 00 00 00 00 c0 a8 64 01

Server1:



首先 client 在第一次捕获 server1, server1 捕获的前四个包和 server2 捕获的第一个包就是和一开始解释的相同, 此时 switch 中记录为{ client-2|server1-1 }

在 client 第一次 ping server2 前, 需要通过 2, 3 两个 packets 去获得 server2 的地址, 同时因为是广播轮询, server1 收到第五个 packet, 在这时 server2 会得到一个端口映射, 此时根据要求, 移除 server1, 此时记录为{client-4|server2-1}

在第二次 ping server1 前, 由于 server1 会发出 arp 包定时查询 client 的地址, switch 会再次习得 server1 的端口映射, 此时需要移除 server2, {client-5|server1-1}

此时在第二次 ping server1 时, 表中已经具有该信息, 直接发送即刻{client-6|server1-2}

在第三次 ping 时, 由于 server2 再次查询 client 地址, 导致 server1 被移除, 加入 server2, 因此, 表中不具有 server1 的接口, 所以直接 broadcast