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	2022/3/24

1. 实验名称

Lab 3: Respond to ARP

2. 实验目的

给路由器增加回复 ARP 请求分组的功能，在 ARP 表中记录 IP 地址和对应的 MAC 地址。

3. 实验内容

A. 处理 ARP 请求

接收 ARP 请求：分析 ARP 请求的源地址和目的地址。

接收到目的 IP 地址是自身的，返回 ARP 回复分组。

B. 缓存 ARP 表

当收到 ARP 分组时，更新或将新的 IP 与 MAC 地址对应的表项添加到 ARP 表中。其中 IP 地址作为键，是唯一的。

4. 实验结果

Show how you implement the logic of responding to the ARP request.

通过循环不断探测有无分组到达，若有分组，进入分组处理程序；

检测分组有无 ARP 包头，若无不处理；若有，获取 ARP 包头，更新缓存的 ARP 表并解析 ARP 包头；

若协议目的地址与本机 IP 地址不同，不做处理；若相同，将源 IP 地址和 MAC 地址作为目的地址，将本机 IP 和对应的 MAC 地址作为源地址，构造 ARP 回复报文发送。

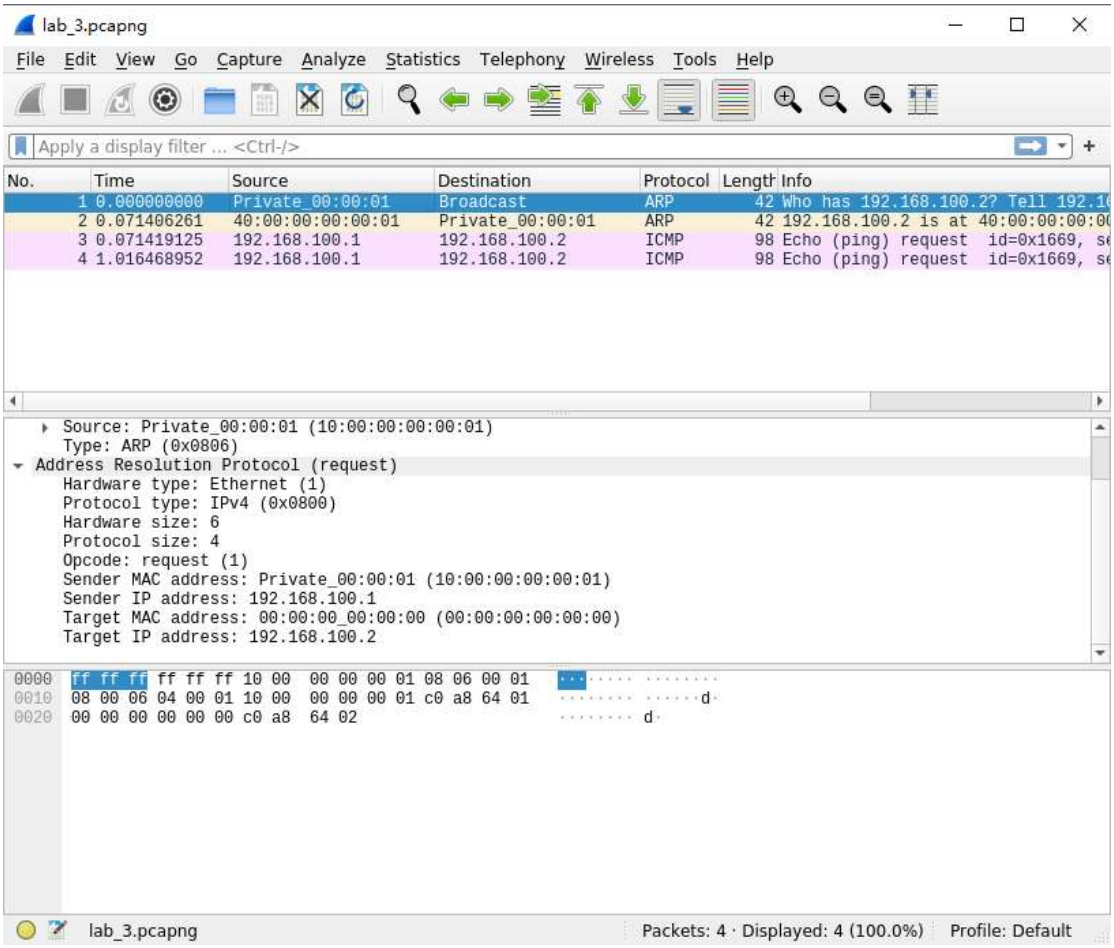
```
if packet.has_header(Arp):
    arp = packet.get_header(Arp)
    self.update_arptable(arp, timestamp)
    if arp.targetprotoaddr in self.ips:
        targethwaddr = arp.senderhwaddr
        targetprotoaddr = arp.senderprotoaddr
        senderprotoaddr = arp.targetprotoaddr
        senderhwaddr = self.find_mac(senderprotoaddr)
        arp_reply_packet = create_ip_arp_reply(senderhwaddr, targethwaddr, senderprotoaddr, targetp
        self.net.send_packet(ifaceName, arp_reply_packet)
```

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

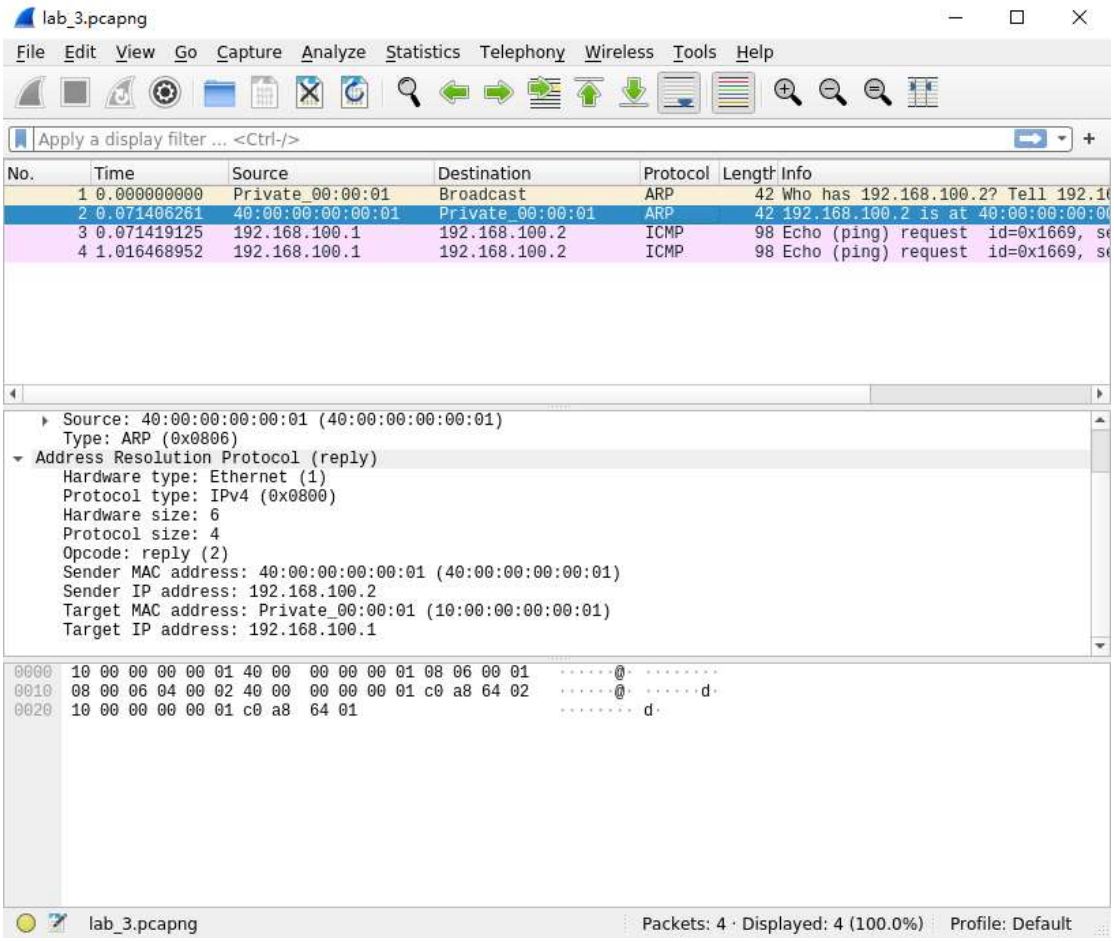
```
Results for test scenario ARP request: 6 passed, 0 failed, 0 pending


Passed:
1   ARP request for 192.168.1.1 should arrive on router-eth0
2   Router should send ARP response for 192.168.1.1 on router-
    eth0
3   An ICMP echo request for 10.10.12.34 should arrive on
    router-eth0, but it should be dropped (router should only
    handle ARP requests at this point)
4   ARP request for 10.10.1.2 should arrive on router-eth1, but
    the router should not respond.
5   ARP request for 10.10.0.1 should arrive on on router-eth1
6   Router should send ARP response for 10.10.0.1 on router-eth1
```

Write the procedure and analysis in your report with screenshots.

在 server1 使用 ping 192.168.100.2，并在 switch 端使用 wireshark 捕获。首先 server1 构建一个目的 MAC 地址全 0 的以太网广播 ARP 分组，目的 IP 地址为需要的 IP 地址。



Switch 收到后，将自身对应的 MAC 地址和 IP 地址作为源地址，将 ARP request 中的源地址作为目的地址，构造 ARP reply 分组向对应的 MAC 地址单独发送。



In your report, show how you construct the ARP table.

首先创建一个哈希表，将收到 ARP 分组的源 IP 地址作为 key，原 MAC 地址和收到的时间构成 list 作为 value；
创建一个删除列表，遍历 ARP 表中的所有项，将时间大于指定时间(如 30s)的表项的 key 加入删除列表中；
删除删除列表中 key 所对应的表项。

```
def update_arptable(self, arp, timestamp):
    self.arptable[arp.senderprotoaddr] = [arp.senderhwaddr, timestamp]
    delete_list = []
    for k,v in self.arptable.items():
        if timestamp - v[1] > 30:
            delete_list.append(k)
    for k in delete_list:
        self.arptable.pop(k)

    log_info(f"current ARP table is {self.arptable}")
```

In your report, show the cached ARP table with screenshots. Explain how entries have changed.

下图先后使用 server1 ping 192.168.100.2， server2 ping 192.168.200.2， server2 ping

192.168.200.2，缓存的 ARP 表项过期时间是 30s。

```
mininet> server1 ping -c2 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.

--- 192.168.100.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1016ms

mininet> server2 ping -c1 192.168.200.2
PING 192.168.200.2 (192.168.200.2) 56(84) bytes of data.

--- 192.168.200.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

mininet> server2 ping -c1 192.168.200.2
PING 192.168.200.2 (192.168.200.2) 56(84) bytes of data.

--- 192.168.200.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

mininet>
```

其中前两条命令间隔时间在 30s 以内，因此可以在第二条命令之后看到缓存的 ARP 表中有两项表项。30s 之后使用第三条命令，这时与 server1 对应的表项已经被移出。

```
(v) root@njucs-VirtualBox:~/Desktop/lab-03-SonicoGO# swyard myrouter.py
14:02:45 2022/03/27     INFO Saving iptables state and installing switchyard rul
es
14:02:45 2022/03/27     INFO Using network devices: router-eth0 router-eth2 rout
er-eth1
14:05:42 2022/03/27     INFO current ARP table is {IPv4Address('192.168.100.1'):
 [EthAddr('10:00:00:00:00:01'), 1648361142.029438]}
14:05:55 2022/03/27     INFO current ARP table is {IPv4Address('192.168.100.1'):
 [EthAddr('10:00:00:00:00:01'), 1648361142.029438], IPv4Address('192.168.200.1')
: [EthAddr('20:00:00:00:00:01'), 1648361155.596653]}
14:06:30 2022/03/27     INFO current ARP table is {IPv4Address('192.168.200.1'):
 [EthAddr('20:00:00:00:00:01'), 1648361190.157132]}
```

5. 核心代码

```
    def find_mac(self, ipaddr):
        for i in self.interfaces:
            if ipaddr == i.ipaddr:
                return i.ethaddr

    def update_arptable(self, arp, timestamp):
        self.arptable[arp.senderprotoaddr] = [arp.senderhwaddr, timestamp]
        delete_list = []
        for k,v in self.arptable.items():
            if timestamp - v[1] > 30:
                delete_list.append(k)
        for k in delete_list:
            self.arptable.pop(k)

        log_info(f"current ARP table is {self.arptable}")

    def handle_packet(self, recv: switchyard.llnetbase.ReceivedPacket):
        timestamp, ifaceName, packet = recv
        # TODO: your logic here
        if packet.has_header(Arp):
            arp = packet.get_header(Arp)
            self.update_arptable(arp, timestamp)
            if arp.targetprotoaddr in self.ips:
                targethwaddr = arp.senderhwaddr
                targetprotoaddr = arp.senderprotoaddr
                senderprotoaddr = arp.targetprotoaddr
                senderhwaddr = self.find_mac(senderprotoaddr)
                arp_reply_packet = create_ip_arp_reply(senderhwaddr, targethwaddr,
senderprotoaddr, targetprotoaddr)
                self.net.send_packet(ifaceName, arp_reply_packet)
```

6. 总结与感想

在实验过程中没有注意区分 ARP 包的两种类型，应该只对 ARP request 回复 ARP reply 报文。验收的时候助教指出了我的错误。