

南京大学本科生实验报告

课程名称：计算机网络

任课教师：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	211220049	姓名	石璐
Email	211220049@smail.nju.edu.cn	开始/完成日期	2023 年 4 月 18 日

1. 实验名称：Respond to ARP

2. 实验目的

完成 IPv4 路由器中响应 ARP 协议请求的功能，建立并维护缓存 ARP 表。

3. 实验内容

Task 1: Handle ARP Requests

➤ 实现代码：

```
12 class Router(object):
13     def __init__(self, net: switchyard.llnetbase.LLNetBase):
14         self.net = net
15         # other initialization stuff here
16         self.interfaces = self.net.interfaces()
17         self.ipaddrs = [intf.ipaddr for intf in self.interfaces]
18         self.arptable = {}
19
20     def handle_packet(self, recv: switchyard.llnetbase.ReceivedPacket):
21         _, ifaceName, packet = recv
22         # TODO: your logic here
23         log_info(f"Received packet {packet} on {ifaceName}")
24         arp = packet.get_header(Arp)
25         if arp is None:
26             log_info("Received a non-ARP packet?!")
27             return
28
29         # Handle ARP request
30         if arp.targetprotoaddr in self.ipaddrs:
31             log_info("Received a packet intended for me")
32             intf = self.net.interface_by_name(ifaceName)
33             arp_reply = create_ip_arp_reply(intf.ethaddr, arp.senderhwaddr,
34                                             arp.targetprotoaddr, arp.senderprotoaddr)
35             log_info(f"Sending packet {arp_reply} to {intf.name}")
36             self.net.send_packet(intf, arp_reply)
```

➤ 测试结果:

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

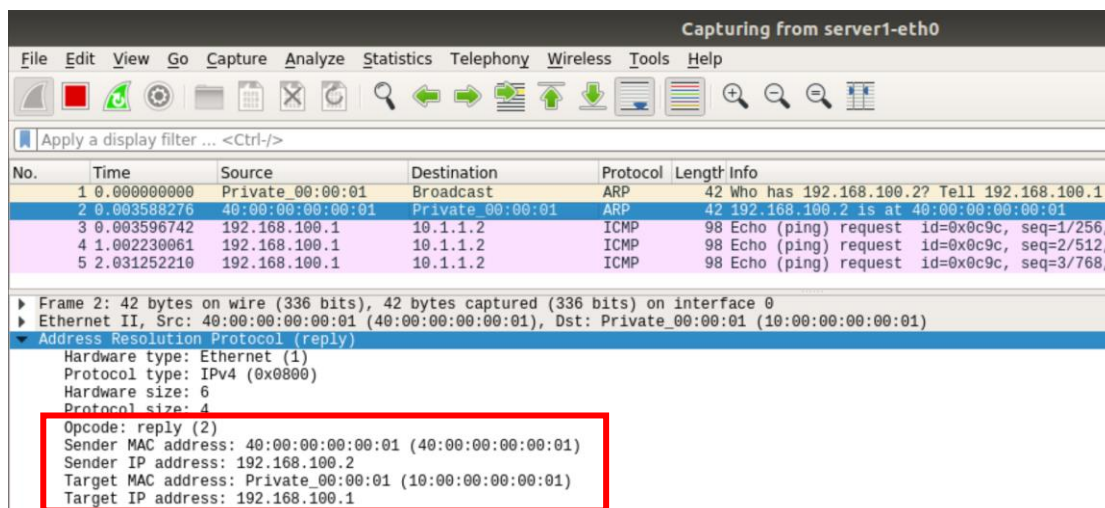
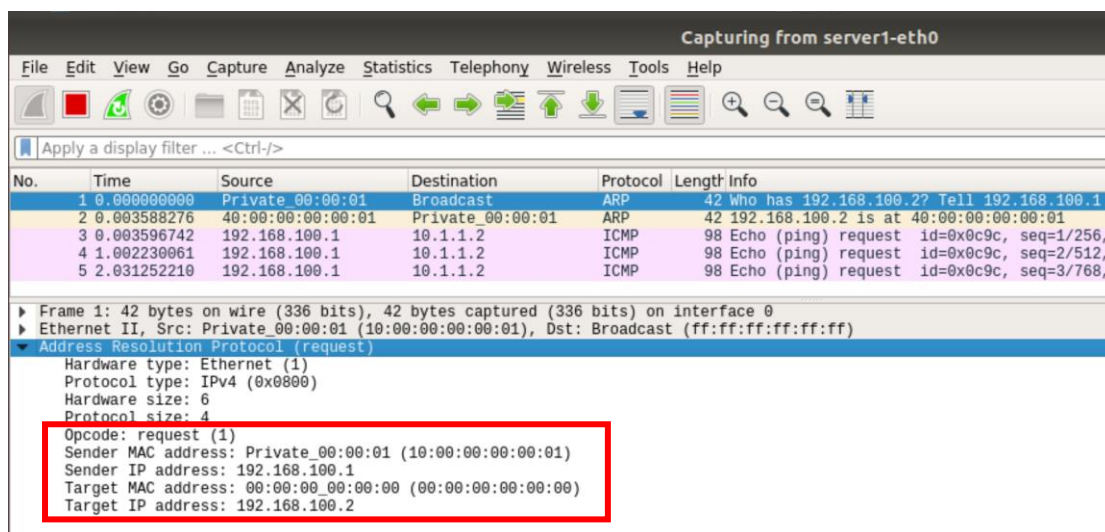
Passed:
1 ARP request for 192.168.1.1 should arrive on router-eth0
2 Router should send ARP response for 192.168.1.1 on router-eth0
3 An ICMP echo request for 10.10.12.34 should arrive on router-eth0, but it should be dropped (router should only handle ARP requests at this point)
4 ARP request for 10.10.1.2 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

All tests passed!
```

➤ 本地测试: 执行 server1 ping -c 3 10.1.1.2

路由器最初应该收到一个关于它自己的 IP 地址的 ARP 请求, 接着它会正确地响应该请求; 然后它应该收到 ICMP 回波请求, 由于路由器还没有被编程为响应 ping 请求, 因此不会响应 ICMP 回波请求。

在 Wireshark 中, 点击捕获窗口第一行的 ARP request packet 时, 可以看到 4 个地址, 其中 "目标 MAC 地址" 目前都是零, 因为这是被请求的地址; 点击捕获窗口第二行的 ARP reply packet 时, 可以看到 ARP 头中的所有地址都被填上了, 而且源地址和目的地址被有效地交换了。



Task 2: Cached ARP Table

➤ 实现代码:

```
12 class Router(object):
13     def __init__(self, net: switchyard.llnetbase.LLNetBase):
14         self.net = net
15         # other initialization stuff here
16         self.interfaces = self.net.interfaces()
17         self.ipaddrs = [intf.ipaddr for intf in self.interfaces]
18         self.arptable = {}
19
20     def handle_packet(self, recv: switchyard.llnetbase.ReceivedPacket):
21         _, ifaceName, packet = recv
22         # TODO: your logic here
23         log_info(f"Received packet {packet} on {ifaceName}")
24         arp = packet.get_header(Arp)
25         if arp is None:
26             log_info("Received a non-ARP packet?!")
27             return
28         # Cached ARP Table
29         if arp.senderhwaddr != "ff:ff:ff:ff:ff:ff":
30             self.arptable[arp.senderprotoaddr] = arp.senderhwaddr
31             log_info("##### ARP Table Updated #####")
32             for ip, mac in self.arptable.items():
33                 log_info(f"# IP({ip})\t\tMAC({mac})")
34             log_info("#####\n")
```

➤ 测试结果:

```
(syenv) njucs@njucs-VirtualBox:~/workspace/lab-3-PolluxyShi$ swyard -t testcases/myrouter1_testscenario.srpy myrouter.
py
22:46:45 2023/04/18 INFO Starting test scenario testcases/myrouter1_testscenario.srpy
22:46:45 2023/04/18 INFO Received packet Ethernet 30:00:00:00:00:01->ff:ff:ff:ff:ff:ff ARP | Arp 30:00:00:00:00:01
:192.168.1.100 ff:ff:ff:ff:ff:ff:192.168.1.1 on router-eth0
22:46:45 2023/04/18 INFO ##### ARP Table Updated #####
22:46:45 2023/04/18 INFO # IP(192.168.1.100) MAC(30:00:00:00:00:01)
22:46:45 2023/04/18 INFO #####
22:46:45 2023/04/18 INFO Received a packet intended for me
22:46:45 2023/04/18 INFO Sending packet Ethernet 10:00:00:00:00:01->30:00:00:00:00:01 ARP | Arp 10:00:00:00:00:01:
192.168.1.1 30:00:00:00:00:00:00:00:01:192.168.1.100 to router-eth0
22:46:45 2023/04/18 INFO Received packet Ethernet ab:cd:ef:00:00:01->10:00:00:00:00:01 IP | IPv4 192.168.1.242->10
.10.12.34 ICMP | ICMP EchoRequest 0 42 (13 data bytes) on router-eth0
22:46:45 2023/04/18 INFO Received a non-ARP packet?!
22:46:45 2023/04/18 INFO Received packet Ethernet 60:00:de:ad:be:ef->ff:ff:ff:ff:ff:ff ARP | Arp 60:00:de:ad:be:ef
:10.10.1.1 ff:ff:ff:ff:ff:ff:10.10.1.2 on router-eth1
22:46:45 2023/04/18 INFO ##### ARP Table Updated #####
22:46:45 2023/04/18 INFO # IP(192.168.1.100) MAC(30:00:00:00:00:01)
22:46:45 2023/04/18 INFO # IP(10.10.1.1) MAC(60:00:de:ad:be:ef)
22:46:45 2023/04/18 INFO #####
22:46:45 2023/04/18 INFO Received packet Ethernet 70:00:ca:fe:c0:de->ff:ff:ff:ff:ff:ff ARP | Arp 70:00:ca:fe:c0:de
:10.10.5.5 ff:ff:ff:ff:ff:ff:10.10.0.1 on router-eth1
22:46:45 2023/04/18 INFO ##### ARP Table Updated #####
22:46:45 2023/04/18 INFO # IP(192.168.1.100) MAC(30:00:00:00:00:01)
22:46:45 2023/04/18 INFO # IP(10.10.1.1) MAC(60:00:de:ad:be:ef)
22:46:45 2023/04/18 INFO # IP(10.10.5.5) MAC(70:00:ca:fe:c0:de)
22:46:45 2023/04/18 INFO #####
22:46:45 2023/04/18 INFO Received a packet intended for me
22:46:45 2023/04/18 INFO Sending packet Ethernet 10:00:00:00:00:02->70:00:ca:fe:c0:de ARP | Arp 10:00:00:00:00:02:
10.10.0.1 70:00:ca:fe:c0:de:10.10.5.5 to router-eth1
```

路由器每次收到一个 ARP 报文时，将其源 MAC 地址和源 IP 地址更新到 ARP 表中，如果收到的不是 ARP 报文，则不会更新。

在该测试中，路由器共收到 4 个报文，其中 3 个为 ARP 报文，1 个非 ARP 报文，因此 ARP 表进行了 3 次更新。