

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	专业（方向）	计算机科学与技术
学号	191220008	姓名	陈南瞳
Email	924690736@qq.com	开始/完成日期	4.15~4.21

一、实验名称

Lab 3: IPv4 Router

二、实验目的

本次实验是创建 IPv4 路由器的第一阶段。需要实现：对 ARP 报文请求进行进行回应（当 dest 为路由器的某个端口时）。此外，还可以创建一个 cached table，存储 IP 地址和 MAC 地址的一一对应关系，采用 timeout 策略。

三、实验内容

Task 1: Preparation

Task 2: Handle ARP Requests

Task 3: Cached ARP Table

四、实验结果

Task 2: Handle ARP Requests

1、思路分析

对于接收的包，取它的 ARP 包头：如果包头为空，则说明不是 ARP 报文，不做处理；如果是 ARP Reply 包，则也不做处理。

ARP 包头中有四种地址：源 IP 地址，源 MAC 地址，目标 IP 地址，目标 MAC 地址。其中，目标 MAC 地址未被填入，属于请求查询的内容。

当路由器接收到 ARP 报文后，根据 ARP 包头中的目的 IP 地址，判断是否为自己的某个端口。若不属于自己的端口，则不做处理。若属于自己的端口，则创建一个回应的 ARP 报文，将自己的 MAC 地址传给它。

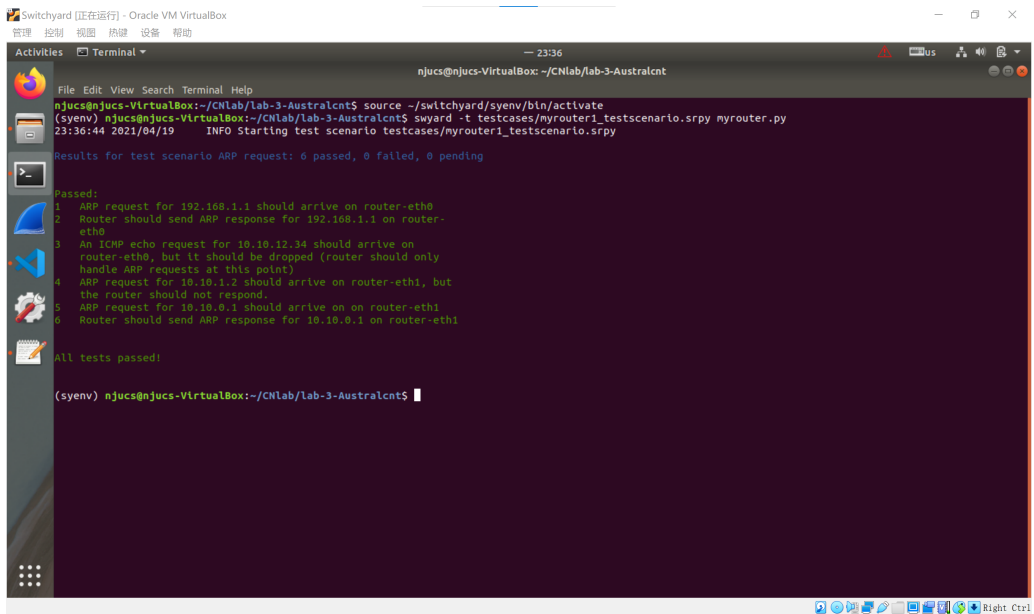
2、核心代码

```
class Router(object):
    ...
    def handle_packet(self, recv: switchyard.llnetbase.ReceivedPacket):
        ...
        # TODO: your logic here
        arp = packet.get_header(Arp)
        if arp is None:
            pass
        elif arp.operation == ArpOperation.Reply:
            pass
        elif arp.operation == ArpOperation.Request:
            for intf in self.net.interfaces():
                if intf.ipaddr == arp.targetprotoaddr:
                    new_pkt = create_ip_arp_reply(intf.ethaddr,
                    arp.senderhwaddr, intf.ipaddr, arp.senderprotoaddr)
                    self.net.send_packet(ifaceName, new_pkt)
                    break
            ...
```

3、测试结果

(1) 测试样例

用提供的已有测试文件进行测试，结果如下：



(2) Mininet下测试

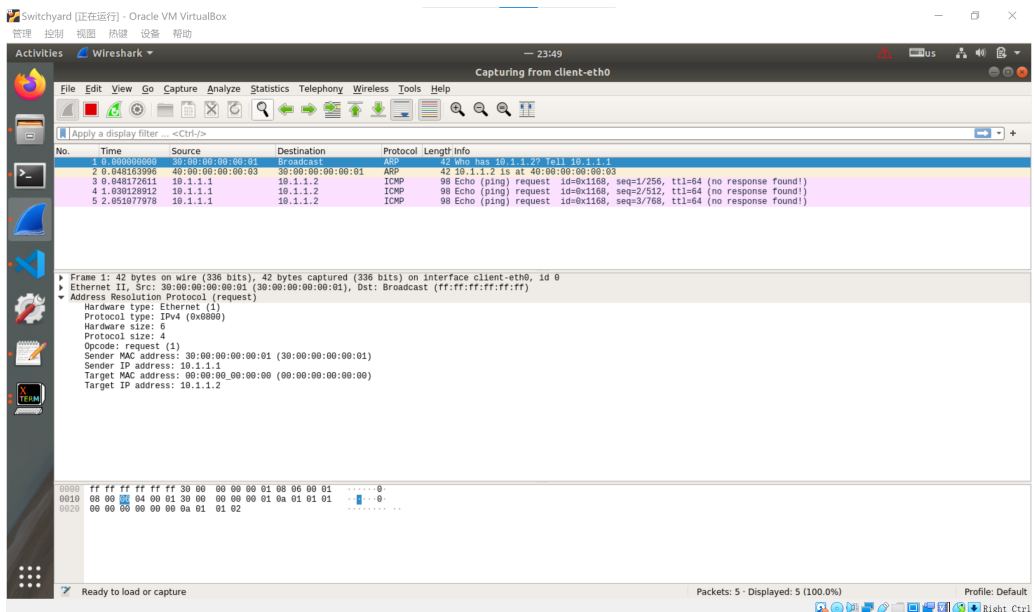
Example:

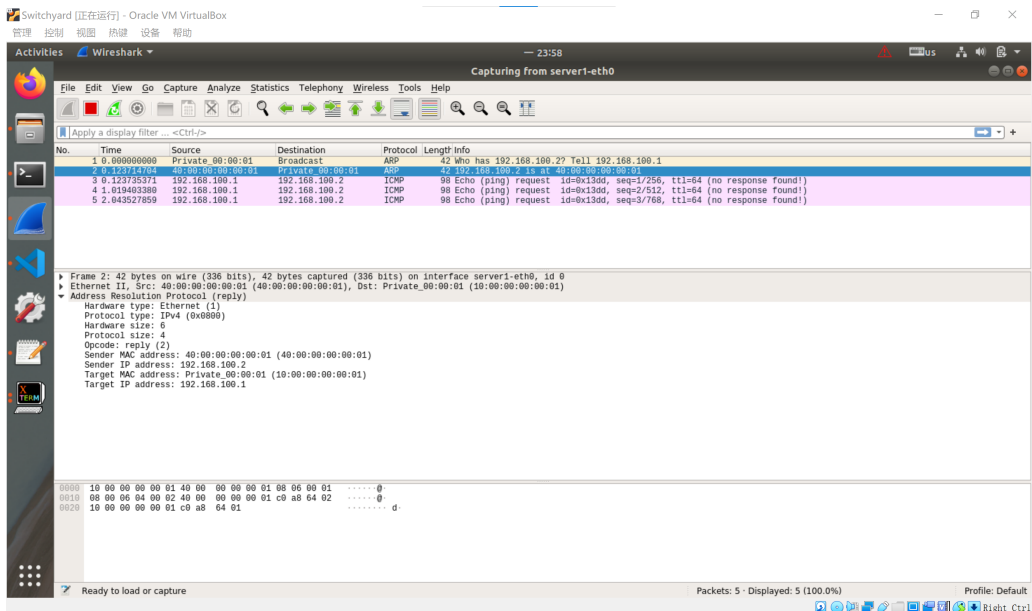
在mininet 中打开 router 的 xterm，在实模式下运行 myswitch.py。

再打开 client 的 xterm，打开 client 的 wireshark。

在 client 的 xterm 中输入 ping -c3 10.1.1.2。

观察抓包和实验结果：





过程分析：与上面的例子类似，在第一个 ARP 报文中，target MAC address 为零，因为是请求询问的 MAC 地址。在第二个回应的 ARP 报文中，所有地址均被填入，同时，发送方和接收方的地址交换。此后，server1 发送 ICMP 报文能够成功到达 router，但由于没有编写 response 的相关代码，所以不会对此进行回应。

(server1 的 IP 地址是 192.168.100.1，MAC 地址是 10:00:00:00:00:01，client 的 IP 地址是 192.168.100.2，MAC 地址是 40:00:00:00:00:01。)

Task 3: Cached ARP Table

1、思路分析

为了在将来主机设备之间的传送 IP 包时，进行不必要的 ARP 报文请求，可以再路由器内创建一个 cached table，保存每个发送方的 IP 地址和 MAC 地址的一一对应关系。因此，可以考虑采用和 Lab3 类似的方法，使用字典 (dict) 结构，以 IP 地址为 key，MAC 地址为 value，保证 IP 唯一，并与 MAC 地址一一对应。此外，因为需要采用了 timeout 的策略，便可以将字典的 value 改为一个列表 [MAC 地址, timestamp]，记录接收 ARP 报文的时间，当时间超过 timeout 规定的时长，便从表中删除该表项。

2、核心代码

```
cached_table = dict() # cache表
timeout_value = 10 # 超时时间设置

class Router(object):
    ...
    def handle_packet(self, recv: switchyard.llnetbase.ReceivedPacket):
        ...
        arp = packet.get_header(Arp)
        for ipaddr in list(cached_table):
            if timestamp - cached_table[ipaddr][1] > timeout_value:
                cached_table.pop(ipaddr)
                log_info("\ncached_table_delete: {}\n".format(cached_table))
        if arp is None:
            pass
```

```

elif arp.operation == ArpOperation.Reply:
    cached_table[arp.senderprotoaddr] = [arp.senderhwaddr, timestamp]
    log_info("\ncached_table_add/update: {}\n".format(cached_table))
elif arp.operation == ArpOperation.Request:
    for intf in self.net.interfaces():
        if intf.ipaddr == arp.targetprotoaddr:
            cached_table[arp.senderprotoaddr] = [arp.senderhwaddr,
timestamp]
            log_info("\ncached_table_add/update:
{}\n".format(cached_table))
            new_pkt = create_ip_arp_reply(intf.ethaddr,
arp.senderhwaddr, intf.ipaddr, arp.senderprotoaddr)
            self.net.send_packet(ifaceName, new_pkt)
            break
...

```

3、测试结果

(1) Mininet下测试

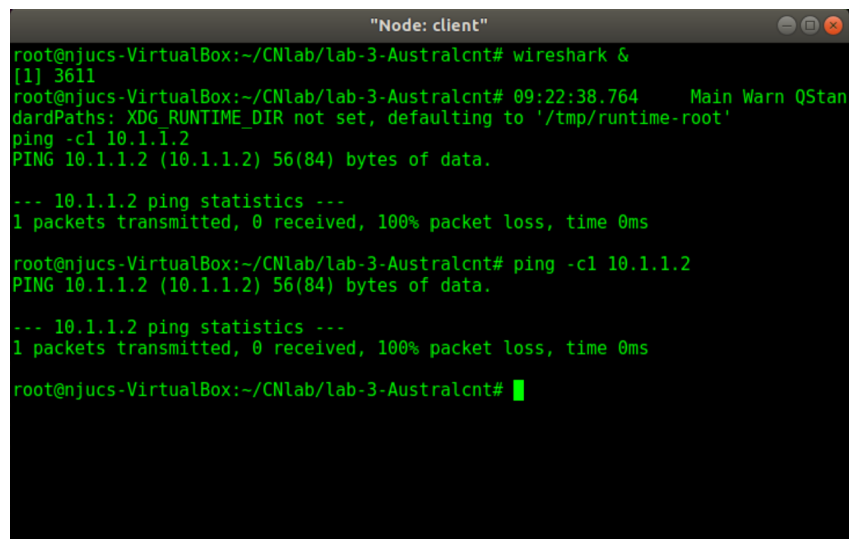
在mininet中打开router的xterm，在实模式下运行myswitch.py。

再打开client的xterm，打开client的wireshark。

在client的xterm中输入ping -c3 10.1.1.2。

等待一段时间后（大于10秒），再次输入ping -c3 10.1.1.2。

观察抓包和实验结果：



```

"Node: client"
root@njucs-VirtualBox:~/CNlab/lab-3-Australcnt# wireshark &
[1] 3611
root@njucs-VirtualBox:~/CNlab/lab-3-Australcnt# 09:22:38.764 Main Warn QStan
dardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
ping -c1 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 56(84) bytes of data.

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

root@njucs-VirtualBox:~/CNlab/lab-3-Australcnt# ping -c1 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 56(84) bytes of data.

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

root@njucs-VirtualBox:~/CNlab/lab-3-Australcnt#

```

```
"Node: router"
root@njucs-VirtualBox:~/CNlab/lab-3-Australcnt# source ~/switchyard/syenv/bin/activate
(syenv) root@njucs-VirtualBox:~/CNlab/lab-3-Australcnt# swyard myrouter.py
09:23:19 2021/04/20 INFO Saving iptables state and installing switchyard rules
09:23:19 2021/04/20 INFO Using network devices: router-eth1 router-eth0 router-eth2
09:23:37 2021/04/20 INFO
cached_table_add/update: {IPv4Address('10.1.1.1'): [EthAddr('30:00:00:00:00:01'), 1618881817.896257]}
09:24:46 2021/04/20 INFO
cached_table_delete: {}
09:24:51 2021/04/20 INFO
cached_table_add/update: {IPv4Address('10.1.1.1'): [EthAddr('30:00:00:00:00:01'), 1618881891.108681]}
█
```

通过在 cached table 的添加/更新和删除的语句后 log 出表中的信息，可以发现在发送第一个 ARP 报文时，client 的 IP 地址和 MAC 地址被 router 记录下来。当等待一段时间再次发送 ARP 报文时，该表项被删除，说明 timeout 策略生效。紧接着，client 的 IP 地址和 MAC 地址重新被加入router。

五、总结与感想

本次实验内容不多，难度也不大，但是没有了 Lab2 中的流程图后，不能够直接编写代码了，需要自己去对实验的流程进行概括，当绘制出简单的流程图后，代码的编写就相对容易了。

此外，本实验的能进行的测试并不多，可以直接在 mininet 下简单测试，无太大必要编写测试文件。