

# Computer Network lab3实验报告

姓名	学号
张洋彬	191220169
邮箱	完成日期
<a href="mailto:1016466918@qq.com">1016466918@qq.com</a>	2021.4.18

## Computer Network lab3实验报告

### 1、实验名称

Lab 3: Respond to ARP

### 2.实验目的

### 3、实验进行

3.1 Preparation

3.2 Handle ARP Requests

3.3 Cached ARP Table

### 4、实验感想

## 1、实验名称

### Lab 3: Respond to ARP

## 2.实验目的

- 1、实现IPv4路由器的第一步
- 2、对ARP请求进行回复
- 3、模拟实现ARP缓存表

## 3、实验进行

### 3.1 Preparation

清楚整个实验的结构：

bbzunyi Delete forwarding_table.txt 92561bd 5 hours ago 7 commits		
.github	GitHub Classroom Feedback	yesterday
testcases	Initial commit	yesterday
.gitignore	Initial commit	yesterday
README.md	Initial commit	yesterday
lab_3_task2_server2.pcapng	no report	5 hours ago
myrouter.py	no report	5 hours ago
start_mininet.py	Initial commit	yesterday

## 3.2 Handle ARP Requests

1、利用arp中的get\_header()函数获取包的包头

```
1 arp=packet.get_header(Arp)
```

2、判断是不是arp包，如果不是就不处理，如果是就在端口中找ip地址对应的端口

```
1         if not arp is None:
2             if arp.operation == 1:
3                 targetprotoaddr = arp.targetprotoaddr
4                 target = None
5                 for intf in interfaces:
6                     if intf.ipaddr == targetprotoaddr:#根据ip地址找到mac地址
7                         target = intf
8                         break
```

3、如果找到了，就用target端口的ip和MAC地址以及发送方的ip和MAC地址创建一个ARP reply包，并发送给发送方的端口

```
1         if not target is None:
2             arp_reply=create_ip_arp_reply(target.ethaddr,arp.senderhwaddr,target.ipaddr
,arp.senderprotoaddr)
3             for intf in interfaces:
4                 if intf.name == ifaceName:
5                     self.net.send_packet(intf,arp_reply)
6                     break
```

用提供的 `myrouter1_testscenario.srpy` 进行测试，通过了全部测试

```
(syenv) njucs@njucs-VirtualBox:~/lab-3-bbzunyi$ swyard -t testcases/myrouter1_testscenario.srpy myrouter.py
16:39:27 2021/04/18      INFO Starting test scenario testcases/myrouter1_testscenario.srpy

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

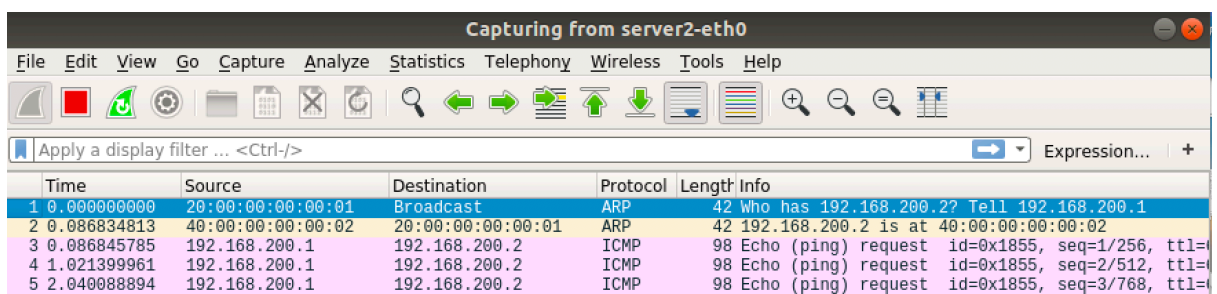
Passed:
1  ARP request for 192.168.1.1 should arrive on router-eth0
2  Router should send ARP response for 192.168.1.1 on router-eth0
3  An ICMP echo request for 10.10.12.34 should arrive on router-eth0, but it should be dropped (router should only handle ARP requests at this point)
4  ARP request for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

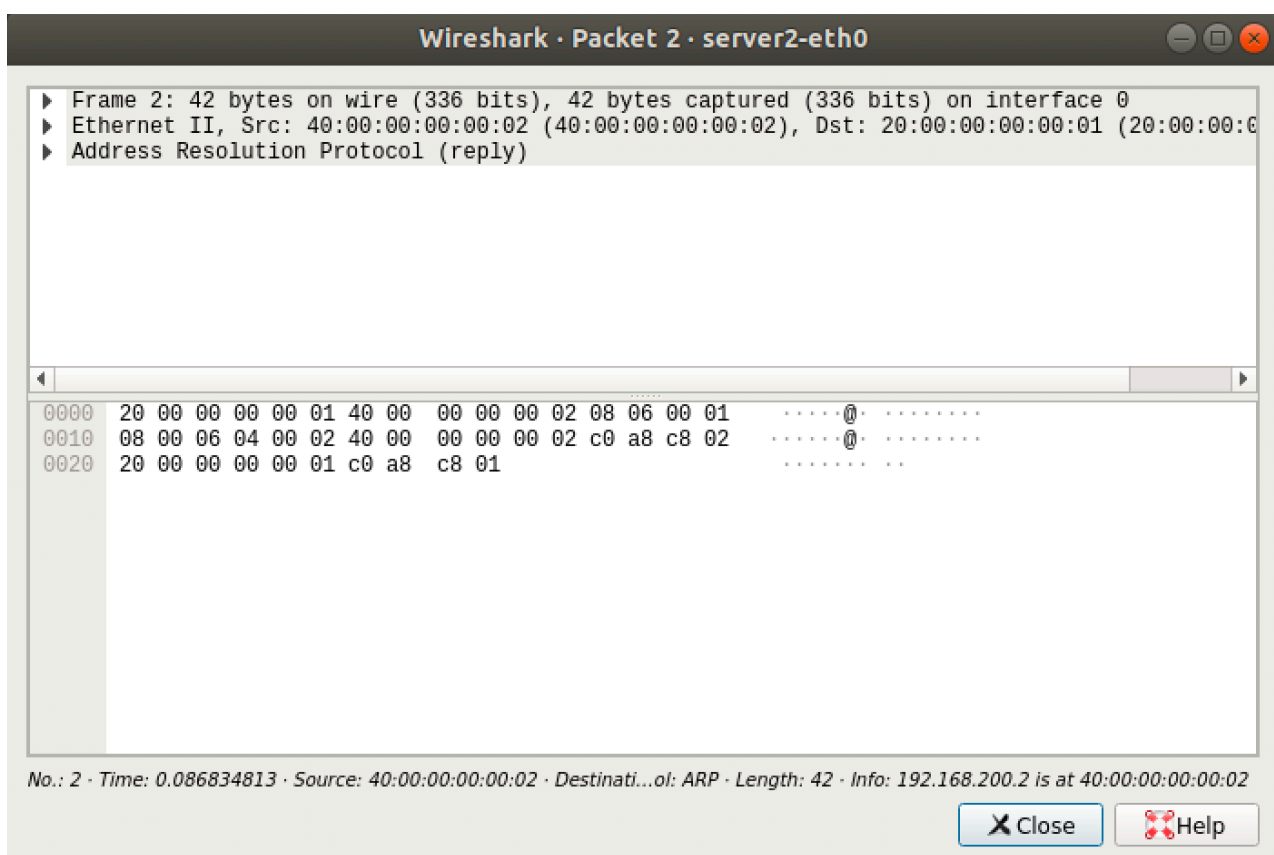
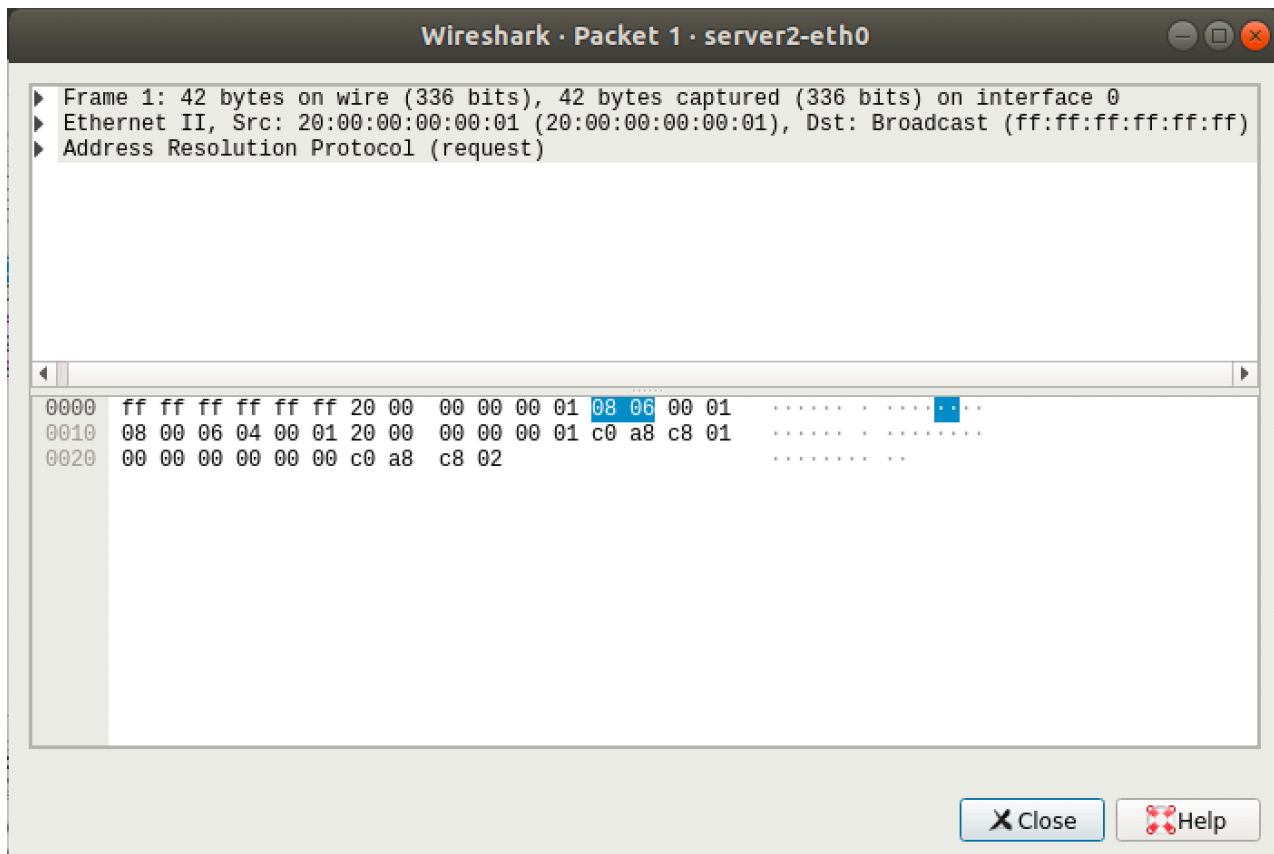
用以下指令，在mininet中进行测试：

- `sudo python start_mininet.py`
- `xterm router swyard myrouter.py`
- `server2 wireshark -k &`
- `server2 ping -c3 192.168.200.2`

server2的抓包结果如下：



Time	Source	Destination	Protocol	Length	Info
1 0.000000000	20:00:00:00:00:01	Broadcast	ARP	42	Who has 192.168.200.2? Tell 192.168.200.1
2 0.086834813	40:00:00:00:00:02	20:00:00:00:00:01	ARP	42	192.168.200.2 is at 40:00:00:00:00:02
3 0.086845785	192.168.200.1	192.168.200.2	ICMP	98	Echo (ping) request id=0x1855, seq=1/256, ttl=
4 1.021399961	192.168.200.1	192.168.200.2	ICMP	98	Echo (ping) request id=0x1855, seq=2/512, ttl=
5 2.040088894	192.168.200.1	192.168.200.2	ICMP	98	Echo (ping) request id=0x1855, seq=3/768, ttl=



第一个包是ARP request包，目标方的MAC地址并不知道（FF: FF: FF: FF: FF: FF），所以就会访问所有端口，找到ip地址对应的MAC地址，确定MAC地址后，即路由器收到ARP request包后，会回复一个ARP reply 包，发送方MAC地址对应之前接收方的ip地址，接收方对应之前的发送方。之后server2会向路由器发送ICMP包，本实验阶段路由器不能进行回复。

## 3.3 Cached ARP Table

1、创建一个 `arp_table` 用于存储arp包提供的信息

```
1 self.arp_table={}#key是ip地址, value[0]是对应的MAC地址, value[1]是datetime.now()  
    存储的时间
```

2、利用lab2中timeout机制（超过10s就清除）对 `arp_table` 中的表项进行清理

```
1         now=datetime.now()  
2         for key in list(self.arp_table.keys()):  
3             elapsed_time=now.timestamp() - self.arp_table[key]  
                [1].timestamp()  
4             if elapsed_time>=10:  
5                 del self.arp_table[key]  
6                 log_info (f"table entry with IP Address {key} has been  
                    removed from arp_table ")
```

3、当ARP包不是None的时候，更新 `arp_table` 并打印

```
1         self.arp_table[arp.senderprotoaddr] =  
            [arp.senderhwaddr,datetime.now()]  
2         print(self.arp_table)
```

test:

在mininet中进行测试，每隔15s，依次由client，server1，server2向路由器发包，由下图可以发现，由于间隔超过10s，在下一个表项进入之前，`arp_table`中已经清空。

```
Edit View Search Terminal Help
"Node: router"
root@njucs-VirtualBox:~/lab-3-bbzunyi# cd ../
root@njucs-VirtualBox:~# source switchyard/syenv/bin/activate
(syenv) root@njucs-VirtualBox:~# cd lab-3-bbzunyi/
(syenv) root@njucs-VirtualBox:~/lab-3-bbzunyi# swyard myrouter.py
17:56:44 2021/04/18 INFO Saving iptables state and installing switchyard rules
17:56:44 2021/04/18 INFO Using network devices: router-eth0 router-eth1 router-eth2
c{IPv4Address('10.1.1.1'): [EthAddr('30:00:00:00:00:01'), datetime.datetime(2021, 4, 18, 17, 56, 51, 58037)]}
17:57:06 2021/04/18 INFO table entry with IP Address 10.1.1.1 has been removed from arp_table
c{IPv4Address('192.168.100.1'): [EthAddr('10:00:00:00:00:01'), datetime.datetime(2021, 4, 18, 17, 57, 6, 201939)]}
17:57:26 2021/04/18 INFO table entry with IP Address 192.168.100.1 has been removed from arp_table
c{IPv4Address('192.168.200.1'): [EthAddr('20:00:00:00:00:01'), datetime.datetime(2021, 4, 18, 17, 57, 26, 237847)]}
1
```

## 4、实验感想

因为是系列实验，第一阶段比较简单，代码量很少，但更重要的是整个实现过程和测试的过程。目前已经大概理解了如何去回复一个ARP包。在整个实验的过程中，先开始困惑于为啥我写的arp已经是ARP包的header了，arp.op打上去的时候，里面居然没有可以填充的operation，后来问了下同学，大概是API里的东西被封装起来了，收获还是满满的，赶紧进入下一阶段。