2016 Advanced Computer Networks
Homework 4

Motivation

To learn how to receive, build and send Ethernet packets. You will know how ARP works by this homework.

Specification

First part:

Use the main.c which is included in attachment to make an ARP packet capture program. In order to make program in a common format, please refer to “arp.h” when you do this homework. You can consult your book on page 170 for ARP packet format. Besides, you should implement the filter in this part as well.

Second part:

Send an ARP request and receive the ARP reply to analyze the packet and find the MAC address of the specific IP. Generally, we usually find the MAC address by cleaning the ARP cache, pinging the IP, capturing the packets with something like Wireshark and analyze the packet by yourself. In this part, you should do the same thing by programming.

Third part:

Make an ARP daemon, it can reply a MAC address when it receive specific IP address.

Request

First part:

Show usage when the command with insufficient or excessive parameters. You need to validate IP and MAC address format. You also need to show error message when the program isn't executed by superuser privileges.

```
jth@jth-XS35:~/桌面/tcpip_tw4$ ./arp
ERROR: You must be root to use this tool!
jth@jth-XS35:~/桌面/tcpip_tw4$ sudo ./arp -h
[sudo] password for jth:
Format:
1) ./arp -l -a
2) ./arp -l <filter_ip_address>
3) ./arp -q <query_ip_address>
4) ./arp <fake_mac_address> <target_ip_address>
jth@jth-XS35:~/桌面/tcpip_tw4$
```

Use "./arp -l -a" command to show all of the ARP packets.

```
2) ./arp -l <filter_ip_address>
3) ./arp -q <query_ip_address>
4) ./arp <fake_mac_address> <target_ip_address>
jth@jth-XS35:~/桌面/tcpip_tw4$ sudo ./arp -l -a
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.169.237?              Tell 140.117.169.238
Get ARP packet - Who has 140.117.169.237?              Tell 140.117.169.238
Get ARP packet - Who has 140.117.172.199?              Tell 140.117.172.254
Get ARP packet - Who has 140.117.162.247?              Tell 140.117.162.254
Get ARP packet - Who has 140.117.171.204?              Tell 140.117.171.254
Get ARP packet - Who has 140.117.172.123?              Tell 140.117.172.254
Get ARP packet - Who has 192.168.0.1?                  Tell 192.168.0.1
Get ARP packet - Who has 192.168.0.1?                  Tell 192.168.0.1
Get ARP packet - Who has 140.117.168.255?              Tell 140.117.169.239
Get ARP packet - Who has 140.117.168.255?              Tell 140.117.169.239
Get ARP packet - Who has 140.117.170.156?              Tell 140.117.170.254
Get ARP packet - Who has 140.117.175.210?              Tell 140.117.175.254
Get ARP packet - Who has 140.117.162.193?              Tell 140.117.162.254
Get ARP packet - Who has 140.117.170.157?              Tell 140.117.170.254
Get ARP packet - Who has 140.117.162.125?              Tell 140.117.162.125
Get ARP packet - Who has 140.117.162.125?              Tell 140.117.162.125
Get ARP packet - Who has 140.117.175.96?               Tell 140.117.175.254
Get ARP packet - Who has 140.117.162.8?                Tell 140.117.162.254
^C
jth@jth-XS35:~/桌面/tcpip_tw4$
```

Use "./arp -l <ip address> "command to implement the filter work. Thus, it should show specific ARP packets.

```
jth@jth-XS35:~/桌面/tcpip_tw4$ sudo ./arp -l 140.117.171.181
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.171.181?              Tell 140.117.171.180
^C
jth@jth-XS35:~/桌面/tcpip_tw4$
```

Second part:

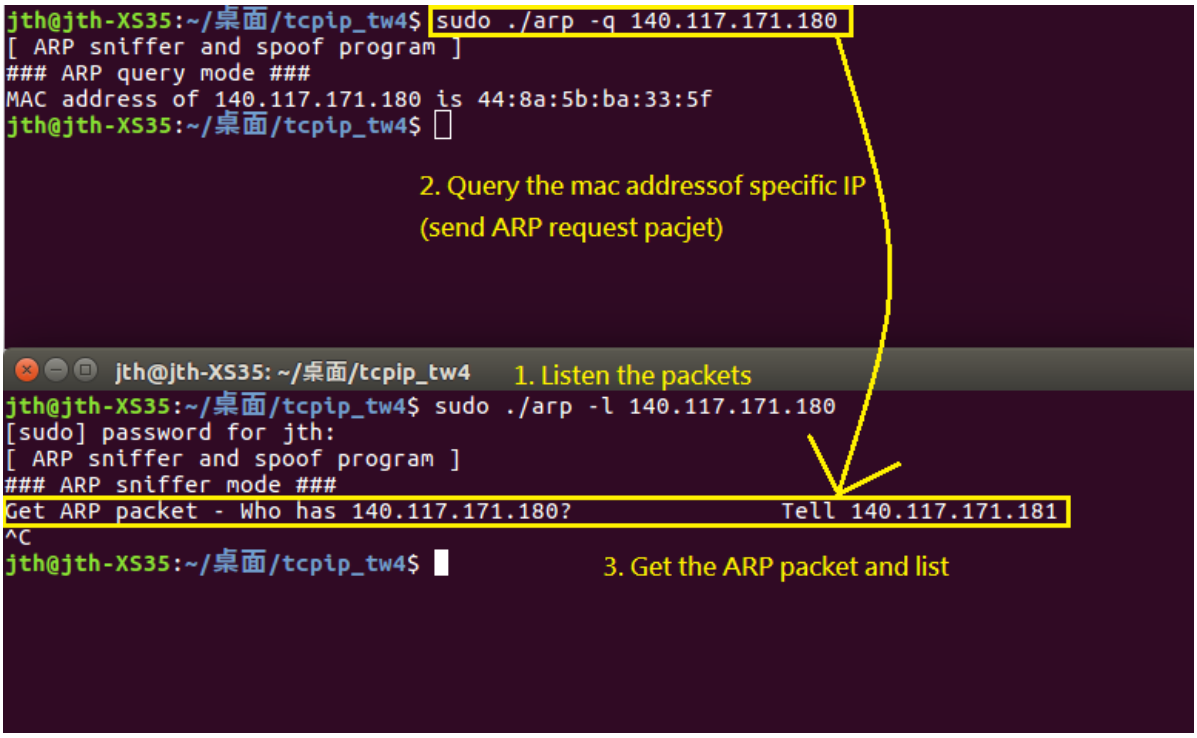
Fill an ARP request packet and send it by broadcast to query the MAC address of the specific IP address.

```
jth@jth-XS35:~/桌面/tcpip_tw4$ sudo ./arp -q 140.117.171.180
[ ARP sniffer and spoof program ]
### ARP query mode ###
MAC address of 140.117.171.180 is 44:8a:5b:ba:33:5f
jth@jth-XS35:~/桌面/tcpip_tw4$
```

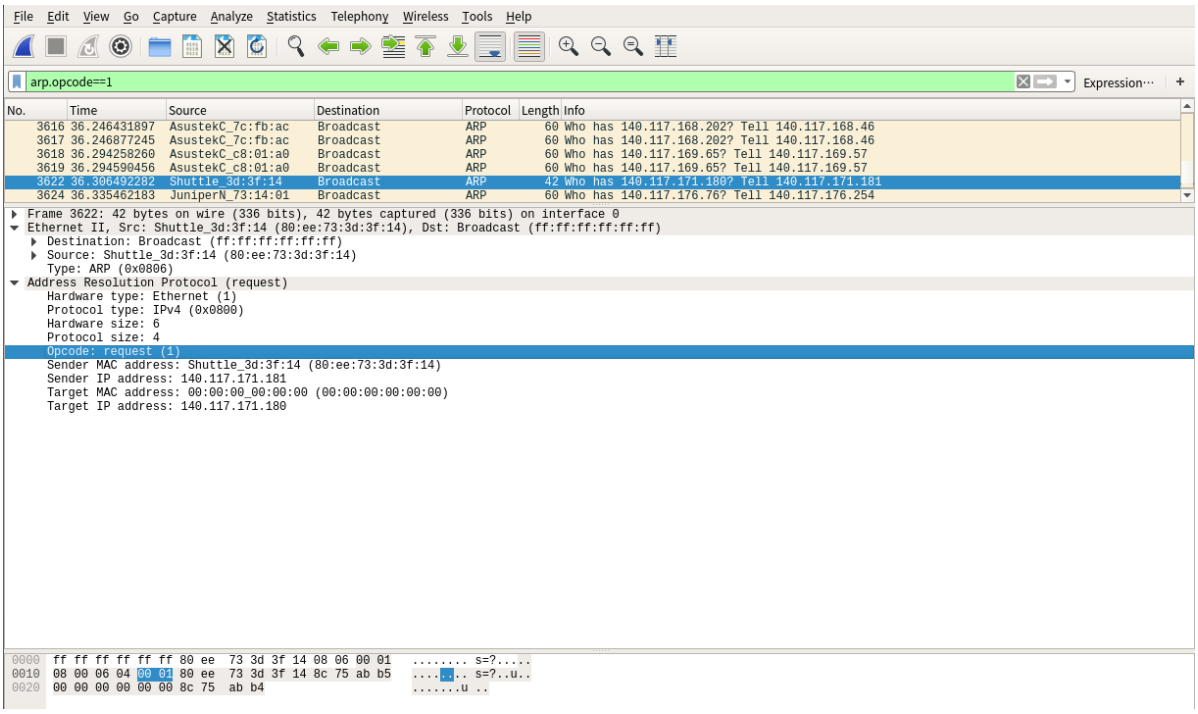
If the IP is offline, you might not find its MAC address, so you have to check the connection before your homework executed. You can use **ifconfig** on Linux or **ipconfig /all** on Windows to check the MAC address of the computer.

Also, you have to install the Wireshark to reconfirm your packets sent and received.

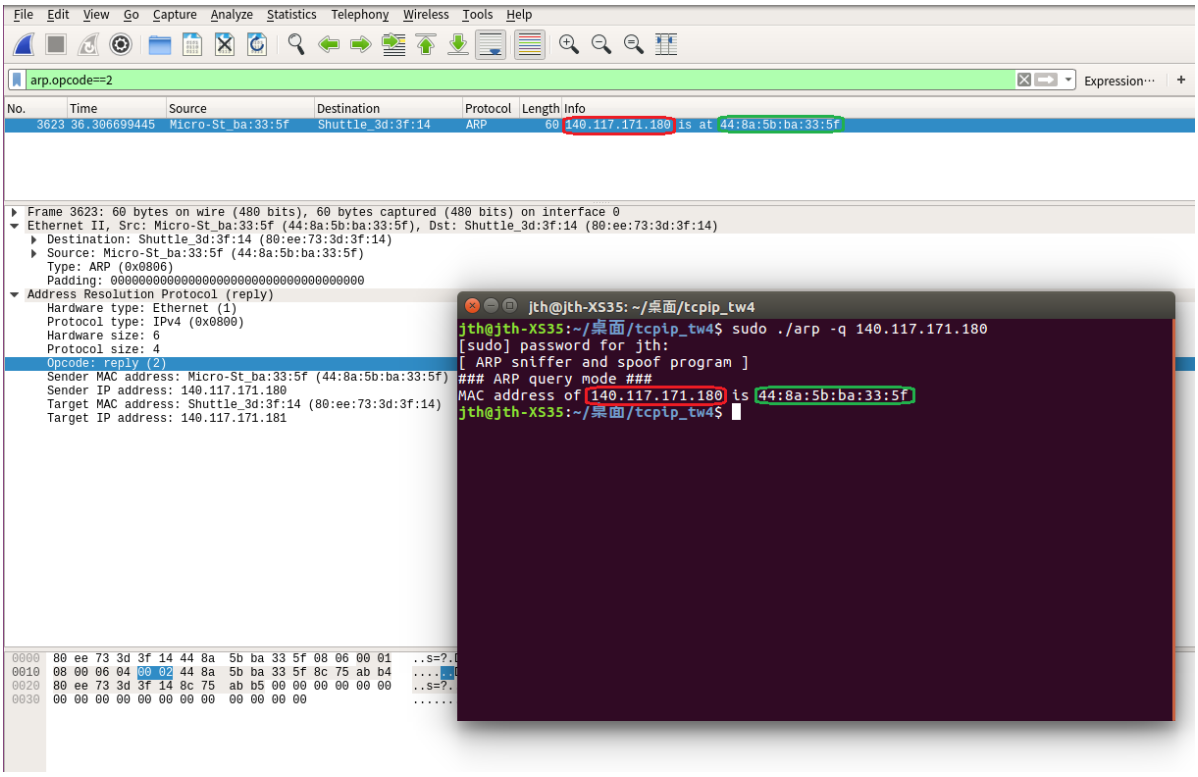
If you obey the order of the homework part, you can use the filter ARP list of the part 1 to detect whether the request packet which part 2 sends is sent successfully or not.



Reconfirm the request packets you send.



Reconfirm the reply packets you receive.

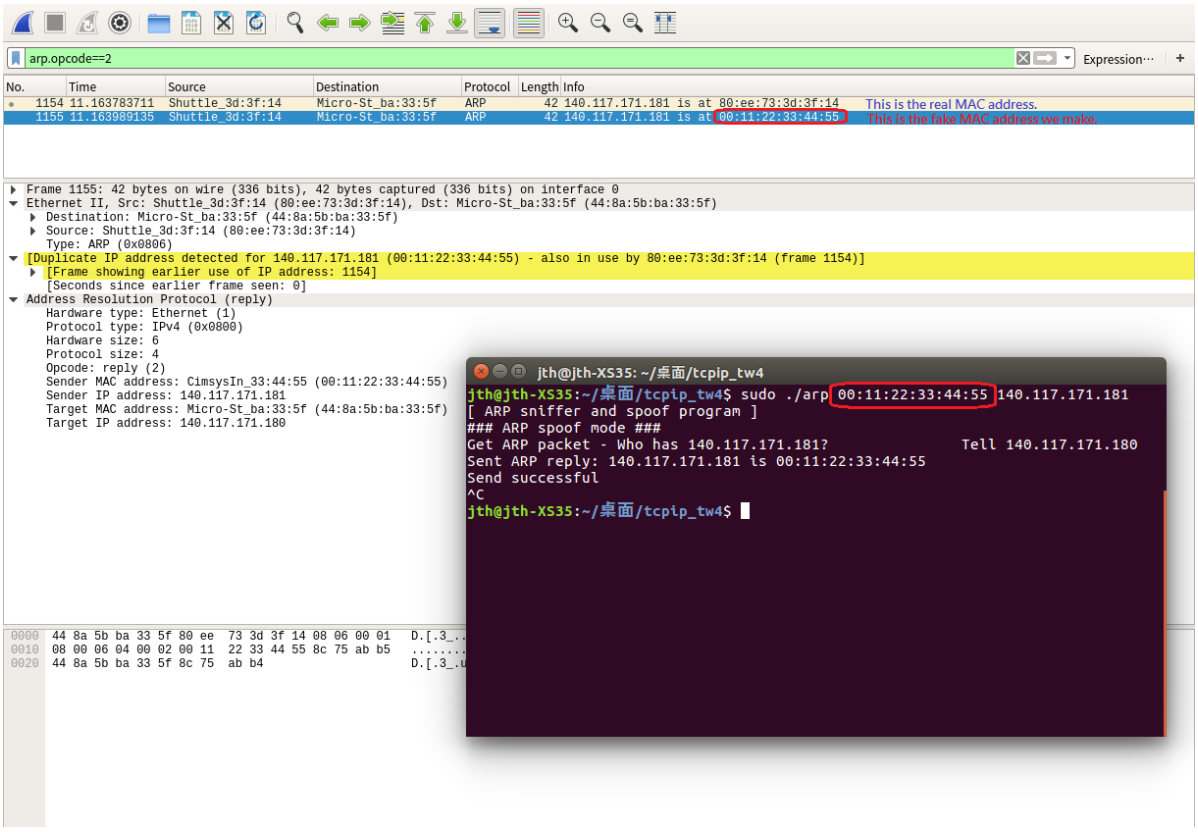


Third part:

You CANNOT use example IP (140.117.171.181) when you test your homework.

Please check out the notice first when you start third part, it is very important.

When program receive an ARP request for 140.117.171.181 (this is example IP), send a 00:11:22:33:44:55 reply.



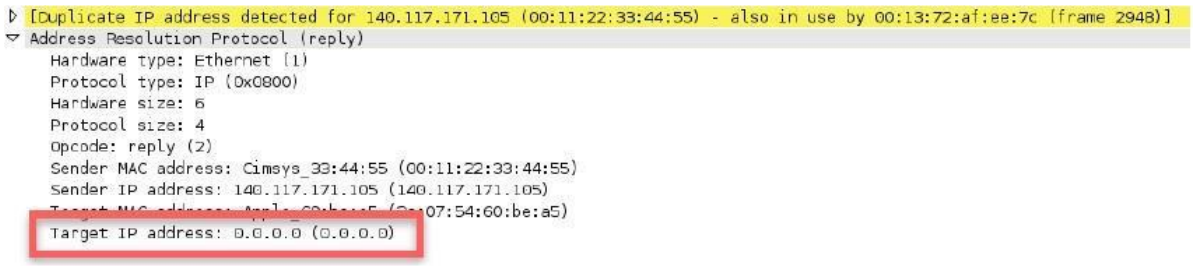
You can use another computer and ping 140.117.171.181 (this is example IP), it will send an ARP request packet. Your program will send an ARP reply in the same time. (If it's not work, you can clear your ARP cache first.)

You can use Wireshark tool to capture the packet you made. There have two ARP packets, one is from true target (80:ee:73:3d:3f:14), another is fake (00:11:22:33:44:55).

Notice

1. In the second and third part, TAs will use Wireshark to verify the ARP reply you made, so make sure your ARP format is as same as the above picture.

2. The packets you send should fully follow the ARP packet standard, every filed should be correct and not be empty.



The above example is not correct, because of missing target IP address.

3. ARP spoofing is **illegal!** Do not attack device of others!

4. You should build an ARP spoofing target **by yourself**. For the above example, spoofing target is 140.117.171.181.

5. This homework require superuser privileges, so you should build your own Ubuntu Linux 16.04 by yourself, we will not provide server's superuser privileges.

6. In order to make program in a common format, please make your input as follow:

   ./arp –l –a
   ./arp –l <filter_ip_address>
   ./arp –q <query_ip_address >
   ./arp <fake_mac_address> <target_ip_address>

Rules

1. Please do your homework by using C language and ensure your homework can be compiled under Ubuntu 16.04.

2. You have to upload your own Makefile to compile your program.

3. Deducting points if you do not follow above restrictions.

4. Do not copy assignment from anyone. All participants will get **ZERO**.

5. We will notice your demonstration time later by email.

6. You can ask TAs any questions about this assignment except debugging.

TAs email: **net_ta@net.nsysu.edu.tw**
Lab: Network & System Laboratory-**EC5018**(10:00a.m. -5:30p.m.)

Upload

Please compress your homework to zip or tar and upload to National Sun Yat-Sen Cyber University. Name your homework to “Student ID_TCPIP_HW4”.
Example: **M013040001_TCPIP_HW4.zip**

Deadline

You should upload your assignment before 2016 /11/ 09(Wed.) 23:59.
Any late submission will not be entertained.

Hint

It is important:
1. structure of **arp_packet** in “arp.h” .
2. **ioctl()** and **structure of ifreq.**
3. **htons()**and **ntohs().**
4. **Wireshark** can help you know what the packet fields are.