

[2019 Advanced Computer Networks Homework 3]

Rules

1. 請在 **Ubuntu 18.04** 下完成本次作業。
2. 請使用 **C** 或是 **Python** 語言完成本次作業，如果使用C語言實作，請提供 **Makefile** 來編譯你的程式。
3. 禁止抄襲任何人的作業。
4. 請將作業壓縮成 zip 或 tar 檔案，並於期限內上傳至中山網路大學 (<http://cu.nsysu.edu.tw/>)，命名規則為“ **Student ID_TCPIP_HW 3** ”。

Example： “D073040002_TCPIP_HW 3”

5. 如果不遵守上述規則，**作業以 0 分計算**。
6. 有任何問題請 email 至 net_ta@net.nsysu.edu.tw，或於 11:00 A.M.–5:00 P.M. 到網路系統實驗室(F5018)詢問。

Deadline： 2019/10/30(Wed.) 23:59

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.

Motivation

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

Part 1

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.

Request

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.

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$ ./arp
ERROR: You must be root to use this tool!
```

Use “./arp -help” to show all commands.

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$ sudo ./arp -help
[ ARP sniffer and spoof program ]
Format :
1) ./arp -l -a
2) ./arp -l <filter_ip_address>
3) ./arp -q <query_ip_address>
4) ./arp <fake_mac_address> <taget_ip_address>
```

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

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$ sudo ./arp -l -a
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.169.254 ?      Tell 140.117.169.40
Get ARP packet - Who has 140.117.169.254 ?      Tell 140.117.169.40
Get ARP packet - Who has 140.117.174.60 ?        Tell 140.117.174.254
Get ARP packet - Who has 140.117.172.158 ?        Tell 140.117.172.254
Get ARP packet - Who has 140.117.175.47 ?        Tell 140.117.175.254
Get ARP packet - Who has 140.117.172.196 ?        Tell 140.117.172.254
Get ARP packet - Who has 140.117.172.189 ?        Tell 140.117.172.254
Get ARP packet - Who has 140.117.174.79 ?        Tell 140.117.174.254
Get ARP packet - Who has 140.117.169.50 ?        Tell 140.117.169.248
Get ARP packet - Who has 140.117.169.50 ?        Tell 140.117.169.248
Get ARP packet - Who has 140.117.169.51 ?        Tell 140.117.169.254
Get ARP packet - Who has 140.117.168.84 ?        Tell 140.117.168.254
Get ARP packet - Who has 140.117.168.94 ?        Tell 140.117.168.254
Get ARP packet - Who has 140.117.176.109 ?       Tell 140.117.176.254
Get ARP packet - Who has 140.117.174.250 ?       Tell 140.117.174.254
Get ARP packet - Who has 140.117.168.122 ?       Tell 140.117.168.104
```

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

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$ sudo ./arp
-l 140.117.171.172
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.171.172 ?      Tell 140.117
.171.173
^C
```

Part 2

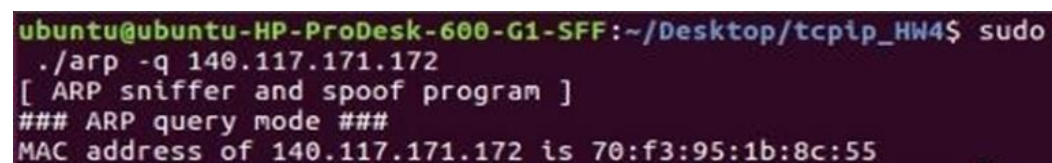
Send an ARP request and receive the ARP reply to analyze the packet and find the MAC address of the specific IP.

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.

Request

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

A terminal window with a dark background and light-colored text. The prompt is 'ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4\$'. The command 'sudo ./arp -q 140.117.171.172' has been executed. The output shows '[ARP sniffer and spoof program]', '### ARP query mode ###', and 'MAC address of 140.117.171.172 is 70:f3:95:1b:8c:55'.

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$ sudo
./arp -q 140.117.171.172
[ ARP sniffer and spoof program ]
### ARP query mode ###
MAC address of 140.117.171.172 is 70:f3:95:1b:8c:55
```

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.

1.Listen the packets

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpi_HW4$ sudo ./arp
-l 140.117.171.172
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.171.172 ?          Tell 140.117
.171.173
^C
```

2.Query the mac address of specific IP (send ARP request packet)

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpi_HW4$ sudo
./arp -q 140.117.171.172
[ ARP sniffer and spoof program ]
### ARP query mode ###
MAC address of 140.117.171.172 is 70:f3:95:1b:8c:55
```

3.Get the ARP packet

Reconfirm the request packets you send.

The image displays a Wireshark packet capture of ARP traffic. The packet list shows several ARP requests (type 1) from various sources to the broadcast address (ff:ff:ff:ff:ff:ff). The packet details pane for the selected packet (No. 3848) shows the Ethernet II header and the ARP request structure. The ARP request is for the IP address 140.117.171.172, with the sender's MAC address being 40:a8:f0:4f:6b:66 and the target's MAC address being 00:00:00:00:00:00. The packet bytes pane shows the raw data of the packet, including the Ethernet II header and the ARP request structure.

Below the Wireshark window, a terminal window shows the output of the ARP sniffer program. The program is running in 'sniffer mode' and has received an ARP request from 140.117.171.172. The terminal output is as follows:

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpi_HW4$ sudo ./arp
-l 140.117.171.172
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.171.172 ?          Tell 140.117
.171.173
^C
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpi_HW4$
```

Reconfirm the reply packets you receive.

The image displays a Wireshark packet capture window with a filter set to `arp.opcode==2`. The packet list shows several ARP requests and one reply. The selected packet (No. 3849) is an ARP reply from `Univera_1b:8c:55` to `HewlettP_4f:6b:66`. The packet details pane shows the Ethernet II header, Internet Protocol Version 4 header, and the ARP (Reply) payload. The ARP payload shows the sender's MAC address as `Univera_1b:8c:55` and the target's MAC address as `HewlettP_4f:6b:66`.

Below the Wireshark window, a terminal window shows the execution of the `arp` command to query the MAC address of `140.117.171.172`. The output shows the MAC address as `70:f3:95:1b:8c:55`.

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF: ~/Desktop/tcplp_HW4
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcplp_HW4$ sudo
./arp -q 140.117.171.172
[ ARP sniffer and spoof program ]
### ARP query mode ###
MAC address of 140.117.171.172 is 70:f3:95:1b:8c:55
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcplp_HW4$
```


Part 3

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

Request

You **CANNOT** use example IP (140.117.171.172) 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 140.117.171.172 (this is example IP), send a 00:11:22:33:44:55 reply.

The image displays a Wireshark packet capture and a terminal window illustrating an ARP spoofing attack.

Wireshark Packet Capture:

No.	Time	Source	Destination	Protocol	Length	Info
5872	29.844365160	HewlettP_4f:6b:66	HewlettP_4f:6b:66	ARP	42	140.117.171.172 is at 00:11:22:33:44:55
5873	29.844512490	Univera_1b:8c:55	HewlettP_4f:6b:66	ARP	60	140.117.171.172 is at 70:f3:95:1b:8c:55

Annotations in the Wireshark packet list:

- real MAC address:** Points to the MAC address 00:11:22:33:44:55 in the first ARP packet.
- fake MAC address we made:** Points to the MAC address 70:f3:95:1b:8c:55 in the second ARP packet.

Terminal Output:

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF: ~/Desktop/tcplp_HW4
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcplp_HW4$ sudo ./arp
[sudo] password for ubuntu:
[ ARP sniffer and spoof program ]
## ARP spoof mode ##
Get ARP packet - Who has 140.117.171.172 ?          tell 140.117.171.173
Sent ARP Reply : 140.117.171.172 is 00:11:22:33:44:55
Send successfull.
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcplp_HW4$
```

Packet Details:

Frame 5872: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: HewlettP_4f:6b:66 (40:a8:f0:4f:6b:66), Dst: HewlettP_4f:6b:66 (40:a8:f0:4f:6b:66)
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: Cmsys 33:44:55 (00:11:22:33:44:55)
Sender IP address: 140.117.171.172
Target MAC address: HewlettP_4f:6b:66 (40:a8:f0:4f:6b:66)
Target IP address: 140.117.171.173

Packet Bytes:

```
0000 40 a8 f0 4f 6b 66 40 a8 f0 4f 6b 66 08 06 00 01  0..0kf0. .0kf...
0010 08 00 06 04 00 02 00 11 22 33 44 55 8c 75 ab ac  .....000U..
0020 40 a8 f0 4f 6b 66 8c 75 ab ad                0..0kf.u ..
```

You can use another computer and ping 140.117.171.172 (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 (70:f3:95:1b:8c:55), another is fake (00:11:22:33:44:55).

Notice

1. In the Part 2 and Part 3, 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.



```

b [Duplicate IP address detected for 140.117.171.105 (00:11:22:33:44:55) - also in use by 00:13:72:a1:ee:7c (frame 2948)]
Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0000)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Cinsys_33:44:55 (00:11:22:33:44:55)
  Sender IP address: 140.117.171.105 (140.117.171.105)
  Target IP address: 0.0.0.0 (0.0.0.0)

```

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 140.117.171.172.
5. This homework require superuser privileges, so you should build your own Ubuntu Linux 18.04 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-help

./arp-l -a

./arp-l <filter_ip_address>

./arp-q <query_ip_address >

./arp<fake_mac_address> <target_ip_address>