

[2018 Advanced Computer Networks Homework 4]

Rules

1. Please do your homework by using C language and ensure your homework can be compiled under **Ubuntu 18.04**.
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** (11:00 ~ 17:00)

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: **M043040032_TCPIP_HW4.zip**

Deadline

You should upload your assignment before **2018/11/07(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.

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> <target_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. [\(target ip\)](#)

```
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.

```
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/tcplp_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/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
```

3. Get the ARP packet

Reconfirm the request packets you send.

The image displays a Wireshark packet capture of an ARP request. The packet list shows a broadcast ARP request from the source interface to the destination IP 140.117.171.172. The packet details pane shows the Ethernet II header, Internet Protocol Version 4 header, and the ARP (Address Resolution Protocol) request. The ARP request is for the target IP 140.117.171.172 and the sender IP 140.117.171.172. The packet bytes pane shows the raw data of the packet.

Terminal output shows the execution of the ARP sniffer and spoof program. The program is running in 'sniffer mode' and is listening for ARP packets. It has received a packet from 140.117.171.172 and is displaying the MAC address 70:f3:95:1b:8c:55.

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 three ARP reply packets (opcode 2) from various sources to the destination `140.117.171.172`. The selected packet (No. 5176) is an ARP reply from `140.117.171.172` to `140.117.171.172`. The packet details pane shows the Ethernet II header, Internet Protocol Version 4 header, and ARP (Request/Reply) section. The ARP section indicates the sender MAC address is `70:f3:95:1b:8c:55` and the target MAC address is `40:a8:f0:4f: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 is `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	Universa_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:

- A red box highlights the MAC address **00:11:22:33:44:55** in the first ARP packet's info field.
- A red arrow points from the text "real MAC address" to this box.
- Another red box highlights the MAC address **70:f3:95:1b:8c:55** in the second ARP packet's info field.
- A red arrow points from the text "fake MAC address we made" to this box.

Terminal Output:

```
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF: ~/Desktop/tcpip_HW4
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$ sudo ./arp
00:11:22:33:44:55 140.117.171.172
[sudo] password for ubuntu:
[ ARP sniffer and spoof program ]
### ARP spoof mode ###
Get ARP packet - Who has 140.117.171.172 ?      tell 140.117.171.172
Sent ARP Reply : 140.117.171.172 is 0:11:22:33:44:55
Send successfull.
ubuntu@ubuntu-HP-ProDesk-600-G1-SFF:~/Desktop/tcpip_HW4$
```

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 field 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>`