[2015 Advanced Computer Networks Homework 6]

Motivation

To implement a PING program which provides IP source route ability.

In normal case:
140.117.171.225 -> 140.117.171.226

In source routing, we redirect the packet through 140.117.171.254:
140.117.171.225 -> 140.117.171.140 -> 140.117.171.226

Attention: Generally, We don't use source routing like this, because most of ISP doesn't support. So you can use LAN network to test source routing. How to confirm your network environment supports source routing or not, you can use the Windows ping.

System Configure:

Ubuntu disables source routing by default. You can type the following command to turn it on:

```
echo 1 > /proc/sys/net/ipv4/conf/all/accept_source_route
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Attention: Need root privilege.

Request:

You need to use IP source routing (Section 8) and assign one source route.
Please Use **libcap** library to implement packet receiver. For testing, you should set the **system configure** on both routed hosts (gateways) and the destination host.

Before you start writing your code, you must install **pcap** library in your PC:

```
sudo apt-get install libpcap-dev
```

Then, include <pcap/pcap.h> in your program file.
When you compile your program, you must pass **–lpcap** option to link to pcap library.

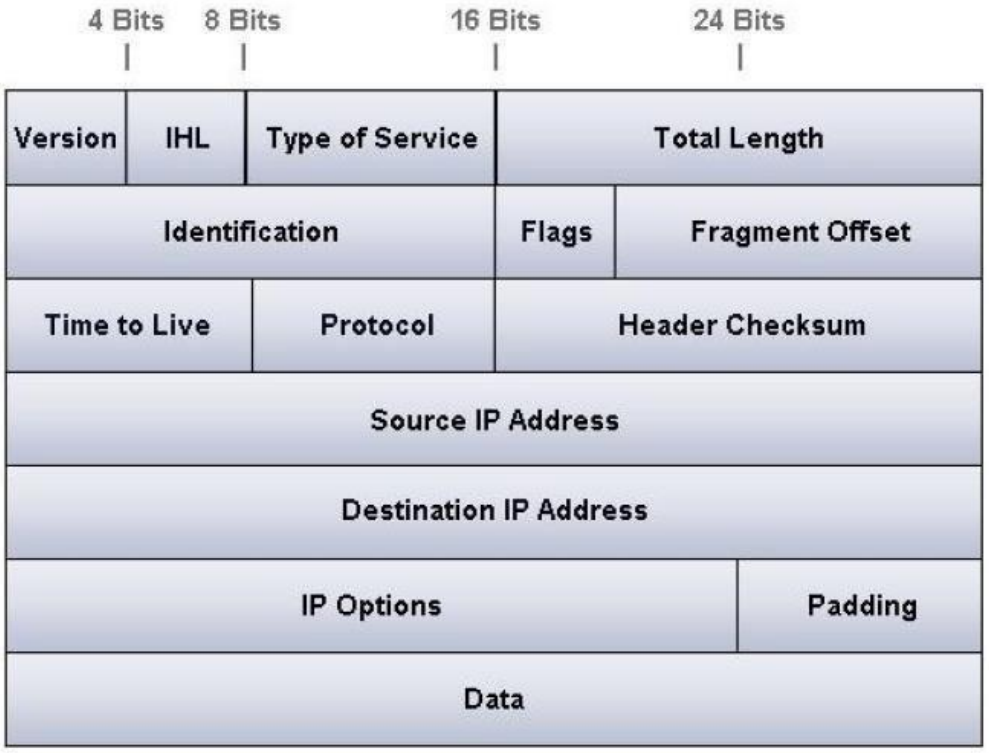

Figure 1. IP header

Fill the IP header according to the following format:

1. Header length = 28 bytes
2. Total length = 92 bytes
3. Id = 0
4. Flag = don't fragment
5. TTL = 64
6. Protocol = icmp
7. Checksum (You can let OS do it for you.)

IP option uses 8 bytes, and no need to fill up the padding, just fill with ICMP header.

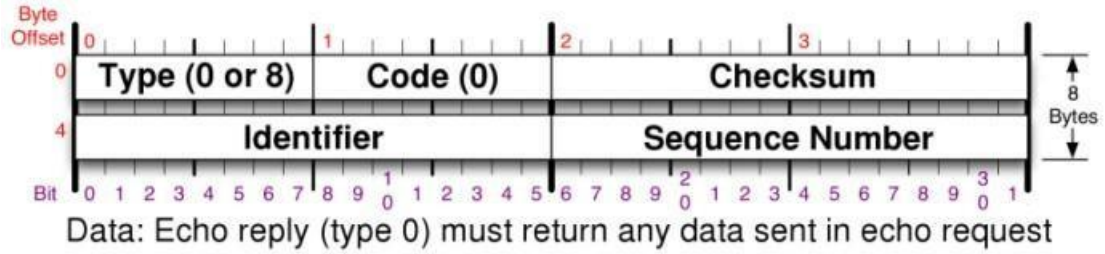

Figure 2. ICMP echo format

Please fill the ICMP packet according to the following format:

1. Checksum (same as HW4)
2. ID: process id
3. Sequence number: Starting from 1, increase one by one.
4. Payload: Random, but don't fill all 0. Note that data size must correspond with IP header.

According to the ping program in Ubuntu doesn't support IP source routing, so you can refer to the windows ping result.

Figure 3. Windows ping. Using IP source route.

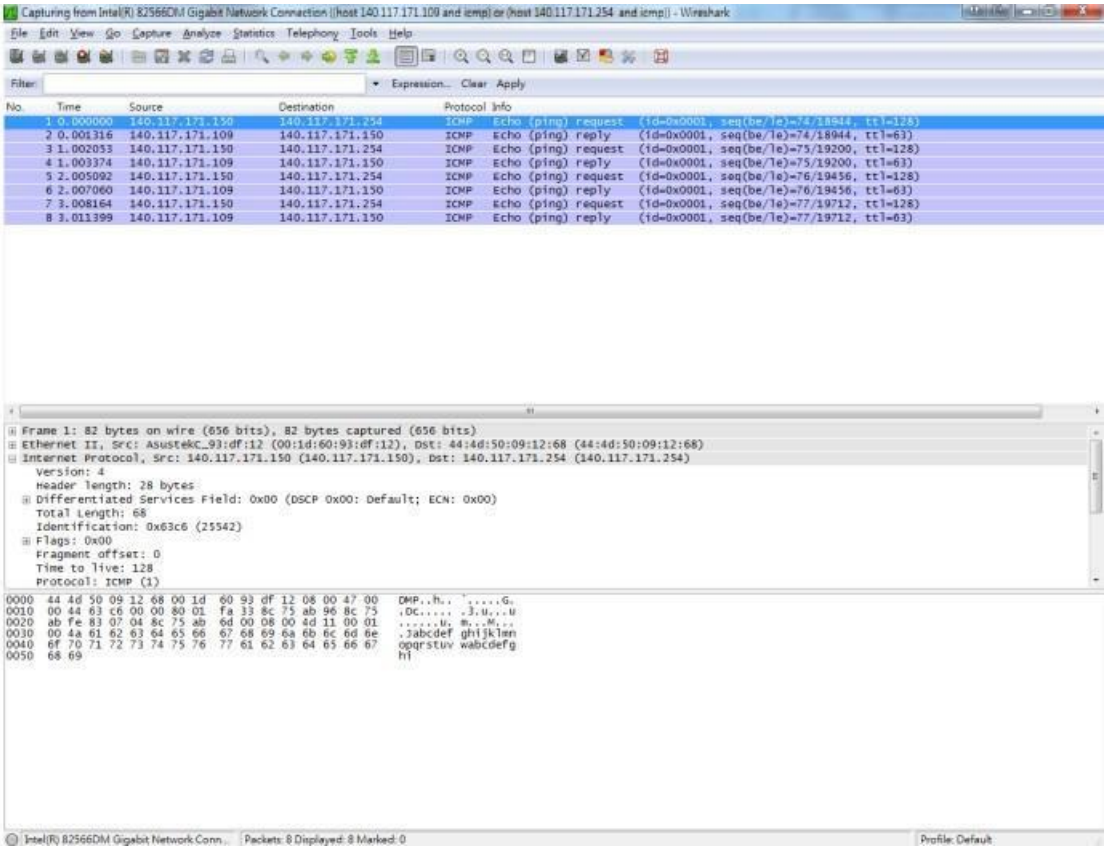

Figure 4. Windows ping. Example of using IP source route.

Packet receiver DO NOT uses the standard socket as before. In this homework, you must use **libpcap** to implementation. **libpcap** likes *Wireshark* and *tcpdump*, you can set filters to get expected packets from incoming packets.
You can use **tcpdump** to test your filter rules. The rules you use on tcpdump basically can also use on *libcap*. Maybe a little different, but similar.
You should use the filter to minimize the packet received, then to determine whether the packet is expected.

For ICMP echo reply, please determine the following fields. (As possible to filter packet):

1. The source in IP header
2. ICMP type
3. The ID in ICMP packet is what you set.
4. The sequence number in ICMP packet is the same as echo request.

Part of libcap initial procedure is already in the file we provided. You just need to set the appropriate filter rule.

Input Usage

usage: sudo ./myping -g gateway [-w timeout (in msec)] [-c count] target_ip

You can set the *timeout* in libpcap, by default, please give **2000ms**.

If *count* is not specified, please give **3 times** by default.

Please implement the timer in local side(client), and accurate to the third decimal place.

```
root@wei:/home/wei/Download/hw6# ./myping -g 140.117.171.140 140.117.171.171 -c
8 -w 1500

Ping 140.117.171.171 (data size = 56, id = 0x302c, timeout = 1500 ms, count = 8)
:
Reply from 140.117.171.171: time = 0.760ms
        Router: 140.117.171.140
Reply from 140.117.171.171: time = *
Reply from 140.117.171.171: time = 0.650ms
        Router: 140.117.171.140
Reply from 140.117.171.171: time = 0.030ms
        Router: 140.117.171.140
Reply from 140.117.171.171: time = *
Reply from 140.117.171.171: time = 0.030ms
        Router: 140.117.171.140
Reply from 140.117.171.171: time = 0.020ms
        Router: 140.117.171.140
Reply from 140.117.171.171: time = 0.300ms
        Router: 140.117.171.140
root@wei:/home/wei/Download/hw6#
```

Figure 5. Correct result

```
root@wei:/home/wei/Download/hw6# ./myping -g 140.117.171.254 140.117.171.171 -c
2 -w 1500

Ping 140.117.171.171 (data size = 56, id = 0x2e15, timeout = 1500 ms, count = 2)
:
Reply from 140.117.171.171: time = *
Reply from 140.117.171.171: time = *
root@wei:/home/wei/Download/hw6#
```

Figure 6. When timeout occur, replace time by “*”.

Rules:

1. Please do this homework in **C language** and run your program on **Ubuntu 14.04.**

2. Please provide **Makefile** to compile your homework; otherwise, you will get **ZERO** point.

3. **Do not copy homework** from others (classmates, senior etc…). If this happened, you will get **ZERO** point, whether you are either the owner of the homework or the copycat.

4. You have to deeply understand what your program do because TA will ask you about your program during demo.

5. If you have any question, please send email to（net_ta@net.nsysu.edu.tw） or come to EC5018, but TA will not help debugging.

6. **No delay assignment is accepted**. If you have trouble, please notify us in advance by email.

Upload:

1. Please compress your homework into **zip** or **tar** archive.

2. Naming rules: "**StudentID_TCP_HW8.zip**".

   For example: M033040001_TCP_HW8.zip

3. Upload your homework to **National Sun Yat-sen Cyber University.**

4. **Deadline: 2015/12/16 (Wed.) 23:59**