CSED353: Computer Networks

Assignment #6: Programming using libpcap

Submission Due Date: 11:59 pm, May 19th (Sun.), 2019

In this assignment, you will develop three simple programs using libpcap to learn how to use it and to understand the Ethernet packet structure.

This assignment should be done individually and is worth a total **10%** of the final mark. The source code should be written in C or C++ using libpcap library. The use of any supplemental library is prohibited and all programming logic and details should be implemented by yourself. Plagiarism is not tolerated.

Remark

- Before you start programming, please construct the libpcap executable environment. Any Linux distribution will be fine, but we highly recommend using Ubuntu or CentOS distribution.
- Make sure that you have read the presentation material on libpcap provided in the course web page (http://dpnm.postech.ac.kr/cs353/2019/src/Programming-using-Libpcap.pdf)
- You can freely define any functions or classes using C or C++ language (do not use other languages such as Python, Java), but do not use any external libraries except standard I/O library.

1. Capturing packets using libpcap (Part 1)

Write a program that captures packets on your NIC (Network Interface Card). The program needs to capture TCP, UDP and ICMP traffic only. (Using **pcap_setfilter** API)

Using the program, capture packets more than 30 seconds. While capturing packets, generate more than three different application traffic which use well-known port number (TCP/UDP 1-1024) (e.g., HTTP(S), SSH, FTP)

2. Packet analysis using libpcap (Part 2)

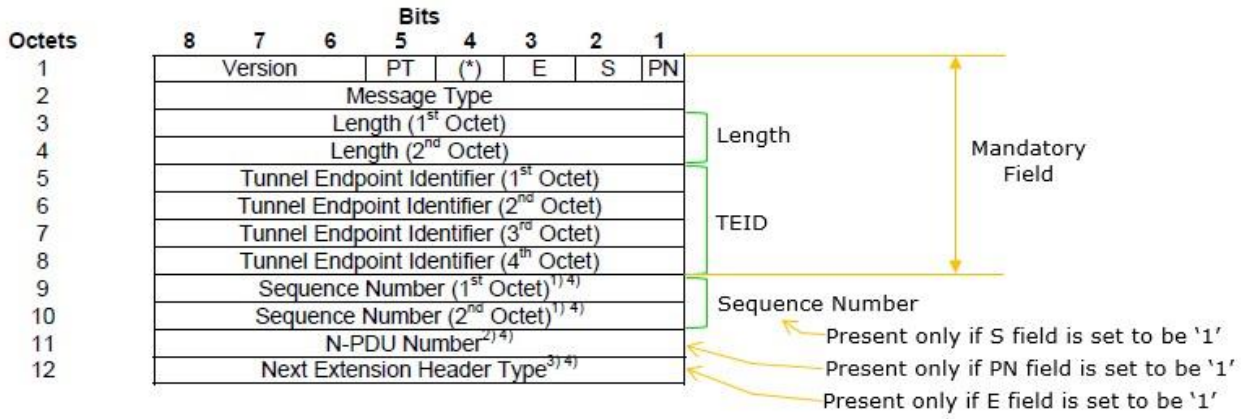
Write a traffic analysis program which analyzes traffic traces in terms of following categories:

- Number of total packets and total bytes
- The time difference between the first and the last packet
- The number of packet and total bytes of TCP, UDP and ICMP traffic
- The number of packet and total bytes of each end host
- The number of packet and total bytes of FTP, SSH, DNS, and HTTP
- Enumerate the average packet size, average packet inter-arrival time

Analyze traffic traces that you captured in the previous part using your analysis program and attach the analysis result in your report.

3. Packet decapsulation using libpcap (Part 3)

IP packets can be encapsulated in other IP packets, tunneling for example. In the mobile network, GTP (GPRS Tunneling Protocol) is used to establish a tunnel through the network and transmit packets. GTP header is shown as bellow.



Write a program that decapsulates given GTP packets and analyzes given traffic traces provided in the course web page (http://dpnm.postech.ac.kr/cs353/2019/src/internet_trace.pcap). Attach **the number of packets of TCP, UDP, ICMP, and HTTP traffic after decapsulation as analysis result** in your report.

(Hint) The length of GTP header depends on the flag bits (E,S and N). All of GTP packets in the given traffic traces set only S field to '1'. Moreover, you can check detailed GTP packet information by using wireshark. For analysis, you can extend your program that is developed in the part 2 by adding GTP decapsulation function.

4. Submission

The final submission files should include all source files and report which describes how to compile and execute your program. The report should also include the detailed description of your program. The report should be formatted as ".PDF".

Have fun!