Programming using libpcapk

Tan Wei Xuan (49003140)
tanweixuan@postech.ac.kr

May 19, 2019

1 Capturing Packets using libpcap

The file **"question1.c"** contains the source code required to capture **TCP, UDP and ICMP traffic** on my Network Interface Card for a period of 40 seconds. The key functions that I have utilised to capture these traffic are as follow:

1. **pcap_lookupdev()**
2. **pcap_openlive()**
3. **pcap_compile() and pcap_setfilter()**
4. **pcap_dump_open()**

1.1 Total Number of Packets and Bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	4137680	4137680 (100.0%)	—
Time span, s	59.088	59.088	—
Average pps	70026.3	70026.3	—
Average packet size, B	754	754	—
Bytes	3120951509	3120951509 (100.0%)	0
Average bytes/s	52 M	52 M	—
Average bits/s	422 M	422 M	—

Figure 1: *Capture Statistics (Statistics → Capture File Properties)*

The total number of packets being captured between a **59 second period** is **4137680**. The total number of Bytes between captured is **3120951509**. This information can be obtained thorugh *Stastics → Capture File Properties*.

1.2 Time Difference between First and Last Packet

frame.number == 1 || frame.number == 4137680

No.	Time	Source	Destination	Protocol
1	0.000000	141.223.170.141	112.162.88.78	TCP
4137680	59.087530	95.39.36.34	141.223.60.4	SIP

We know that the total number of packets being captured is **4137680**. As such, the first frame be captured will be **1** and the last frame being captured will be **4137680**. We can filter out these two frames by applying the filter, ***(frame.number == 1)*** || ***(frame.number == 4137680)***. From the filtered results, we can see that the first packet is being transmitted at **0.0** seconds while the last packet is being transmitted at **59.087530** seconds. As such, the **time difference** between the **first** and **last** packet is **59.087530 seconds**

1.3 The number of packet and total bytes of TCP, UDP and ICMP traffic

Protocol	Percent Packets	Packets	Percent Bytes	Bytes
Frame	100.0	4137680	8.6	269186417
Ethernet	100.0	4137782	1.9	57928948
Internet Protocol Version 4	100.0	4137680	2.7	82753600
User Datagram Protocol	61.2	2533291	0.6	20266328
Transmission Control Protocol	37.9	1568769	1.4	43878139
Internet Protocol Version 6	0.1	3870	0.0	112314
Internet Control Message Protocol	0.8	31256	0.0	978571

Figure 2: *TCP,UDP and ICMP Proticol Hierarchy (Statistics → Protocol Hierarchy)*

The entirety of the network traffic is being transmitted through **IPv4** as it takes up **100%** of the total packets. The total number of packet and total bytes of IPv4 TCP, UDP and Internel Control Message Protocol (ICMP) traffic are as follow:

1. **TCP**
   The total number of packets being transmitted using TCP is **1568769** and the total number of bytes being transmitted is **43878139**. TCP takes up **37.9%** of total network traffic.

2. **UDP**
   The total number of packets being transmitted using UDP is **2533291** and the total number of bytes being transmitted is **20266328**. UDP takes up **61.2%** of total network traffic.

3. **ICMP**
   The total number of packets being transmitted using ICMP is **31256** and the total number of bytes being transmitted is **978571**. ICMP takes up **0.8%** of total network traffic.

For this captured netwrok traffic, IPv4 TCP and UDP take the majority of the percent of total packets, with **UDP taking up most of the traffic *(61%)***. Most of the traffic is probably allocated for UDP services such as Media Streaming, VoIP, etc. This information can be obtained thorugh *Statistics → Protocol Hierarchy*.

2 Packet Analysis using libpcap

3 Packet Decapsulation using libpcap