CSED 353 NetWork Term Project 보고서

Topic_dif_B : Implement packet capturing program using libpcap

20180038 박형규

1. 목적

   이번 Network Term Project의 주제로는 libpcap을 이용하여 Packet capturing program을 직접 구현해보는 것을 목표로 하였다. 강의에서 배운 Adapter, IP, TCP, UDP, ICMP등의 지식을 활용할 것이다.

2. 구현 방법

   - 여러 구조체들

     이 프로그램의 구현에 필요한 여러 구조체들을 정의하였다. 이 구조체들에는 mac_address, ether_header, ip_address, ip_header, tcp_header, udp_header, icmp_header가 있다. 각 구조체에는 각각의 구조체가 나타내야 할 정보가 포함되어있다. 대표적으로 mac_address에는 mac_address로 나타날 6개의 byte 정보가 포함되어 있다.

   - Network의 모든 interface들을 나타낸다.

     Network의 모든 interface들을 나타내는 것은 pcap_findalldevs() 함수를 이용하여 나타내었다. 이 함수를 호출함으로써 alldevs에 모든 interface들이 리스트 형태로 저장되어졌다. 그 이후에 for문을 이용하여 모든 interfaces들을 차례대로 출력하고, 사용자가 원하는 interface를 선택할 수 있도록 입력받았다.

   - 특정 device 열기

     Pcap_open_live()함수를 이용하여 사용자가 선택한 디바이스를 open하도록 하였다. 그 이후에 pcap_compile() 함수와 pcap_setfilter()함수를 이용하여 패킷 필터링을 수행하여 특정 프로토콜 패킷만 도달하도록 하였다.

   - 패킷 캡처하기

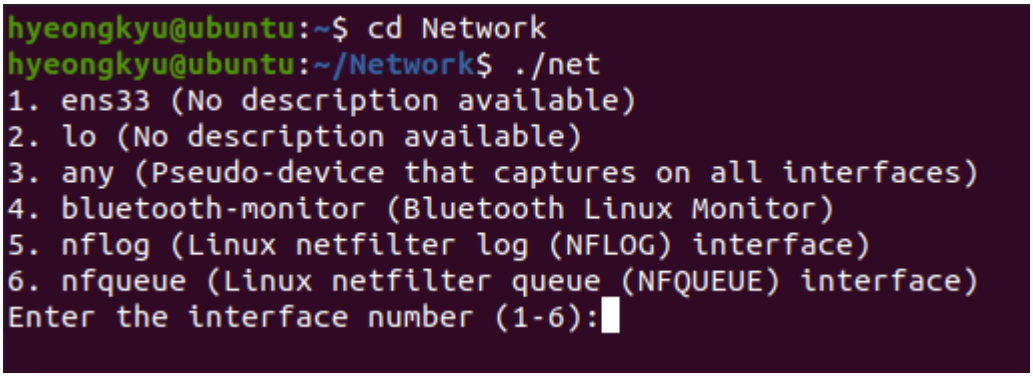
     Pcap_loop()함수를 이용하여 선택한 디바이스에서 무한으로 패킷을 캡처하도록 하였다. 패킷이 도달할 때마다 packet_handler() 함수가 호출되도록 하였다.

   - 패킷 내용 출력하기

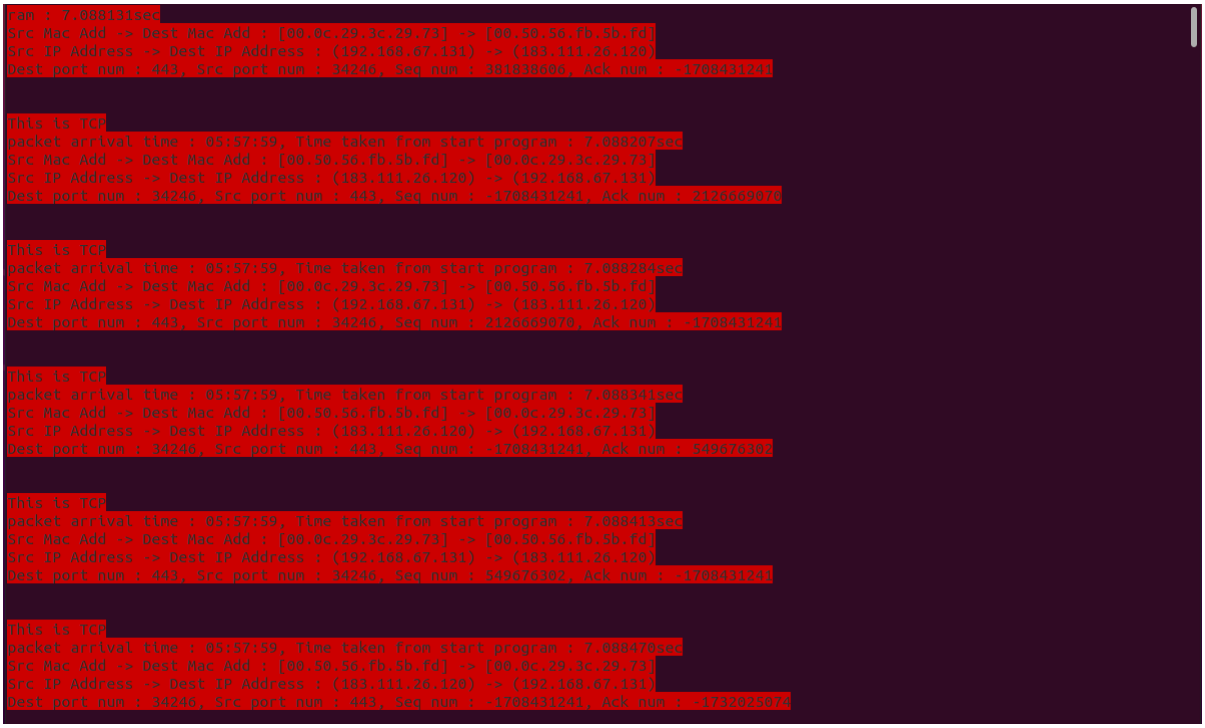
     Packet_handler() 함수에서는 실제 패킷이 도달하였을 때 실행되는 내용을 담은 함수

이다. 먼저 ethernet header에 있는 MAC address를 srcmac, destmac 변수에 각각 저장한다. 이와 마찬가지로 ip_header 구조체의 포인터 변수 ih를 선언하여 ip header의 정보를 저장하였다. 우리의 목적은 TCP, UDP, ICMP header를 각각 분리하는 것이므로 IP_header일 경우에 ih->proto의 값에 따라 TCP, UDP, ICMP로 case를 분류하였다. TCP일 경우에는 빨간색 배경에 Src Mac Address, Dest Mac Address, Src IP Address, Dest IP Address, Dest port num, Src port num, Seq num, Ack num이 출력되도록 하였다. UDP일 경우에는 노란색 배경에 Src Mac Address, Dest Mac Address, Src IP Address, Dest IP Address, Dest port num, Src port num이 출력되도록 하였다. ICMP일 경우에는 핑크색 배경에 Src Mac Address, Dest Mac Address, Src IP Address, Dest IP Address, Type, Code가 출력되도록 하였다. IP header를 가지면서 TCP, UDP, ICMP가 모두 아닌 경우에는 초록색 배경에 Src Mac Address, Dest Mac Address가 출력되도록 하였다. 마지막으로 IP header를 가지고 있지 않은 경우에는 하늘색 배경에 Src Mac Address, Dest Mac Address이 출력되도록 하였다. 그리고 각 경우에서 패킷이 캡쳐된 시간과, 프로그램이 시작된 시간부터 패킷이 시작된 시간 사이의 간격이 출력되도록 하였는데, 이는 clock() 함수를 이용하여 구현하였다.

3. 구현 모습



위 사진은 처음 network interface들이 모두 표시된 것이다.



위 사진은 TCP packet이 도착한 모습들을 보여준다.

위 사진은 TCP packet들과 UDP packet들이 도착한 모습을 보여준다.

위 사진은 TCP packet, UDP packet과 함께 Unknown packet(not IP header)이 함께 있는 경우이다.

4. 결론

처음에 목표하였던 simple Wireshark를 성공적으로 구현하는데 성공하였다. 이를 구현하면서 네트워크 계층의 각 체계에 대하여 조금 더 자세히 알게 되었으며 패킷이 구성되는 방법에 대해서도 구체적으로 이해할 수 있었다. 처음에 Ubuntu 18.04를 사용했을 때 pcap_findalldevs() 함수를 호출하면 socket: Socket type not supported 라는 에러가 발생하여 구현에 어려움을 겪었다. 하지만 VM을 이용하여 Ubuntu를 설치하고 이 환경에서 실행하니 정상적으로 실행되어 문제를 해결할 수 있었다. 또한 pcap_open_live() 함수를 실행할 때 Ubuntu 환경 상에서 관리자 권한을 이용해야만 제대로 접근할 수 있었다. 이는 'sudo su'의 명령어를 이용하여 관리자 권한으로 접근하여서 문제를 해결할 수 있었다.