9주차_RawSocket_PacketCapture

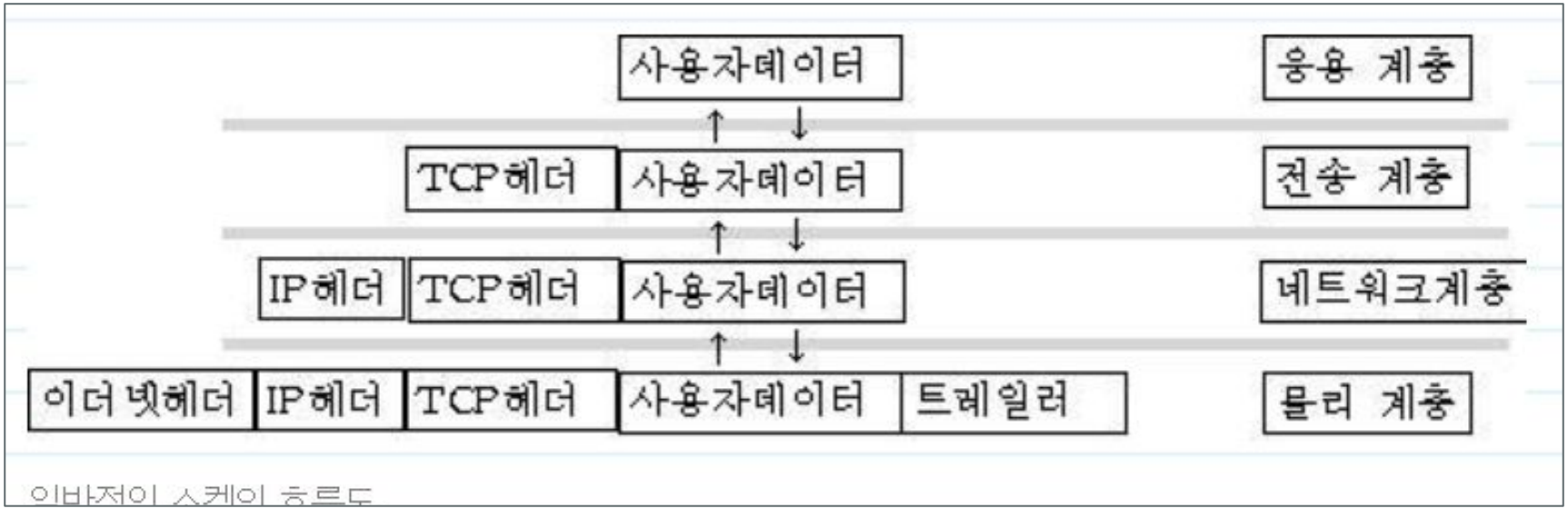
데이터 네트워크연구실
이현호
lee075@cs-cnu.org

Asignment

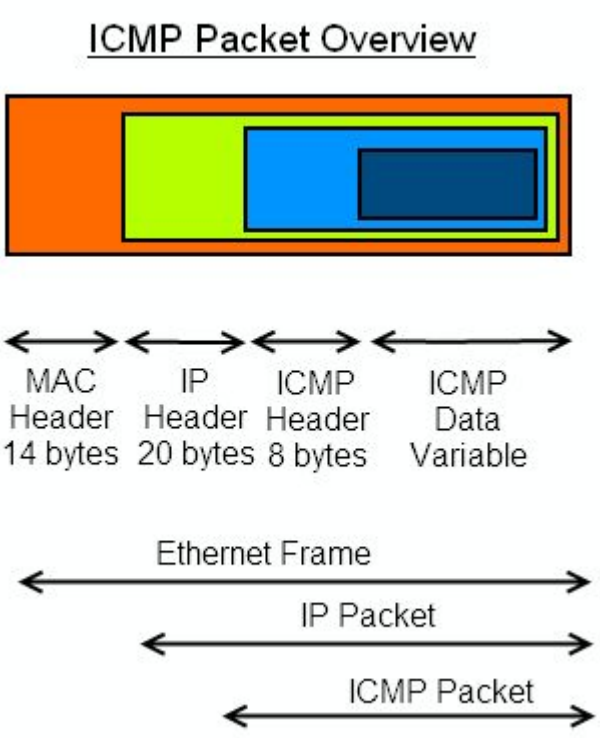
- Raw Socket을 이용한 Packet capture 프로그램 만들기
- TCP / UDP / ICMP Header를 파싱
- 결과 분석 후 출력 해주는 프로그램
- 출력 결과물은 파일로 저장

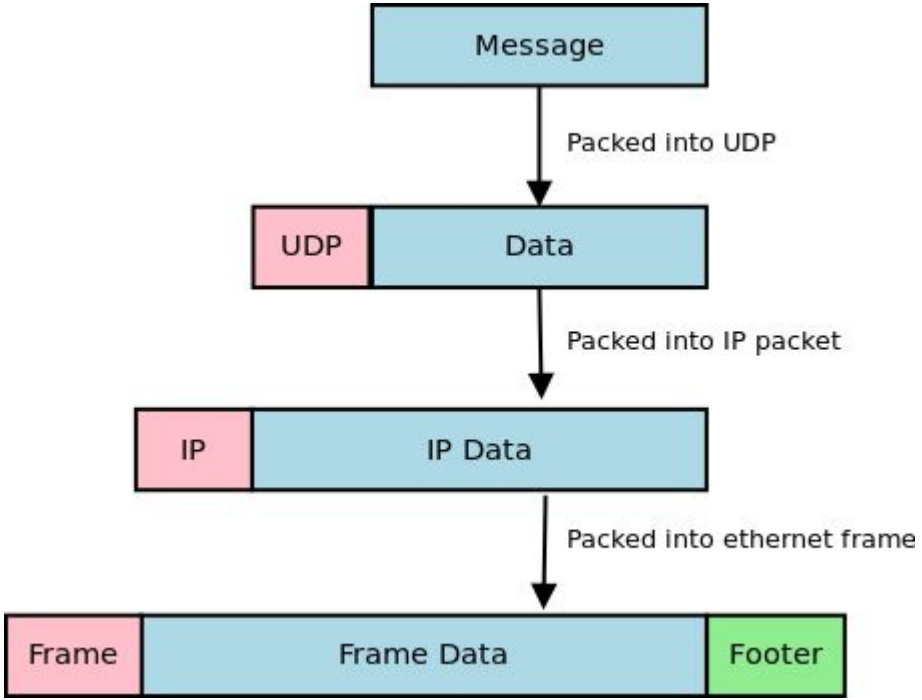
패킷 도식화

- 각 층의 헤더들이 추가됨
- 헤더 사이즈가 offset이 됨



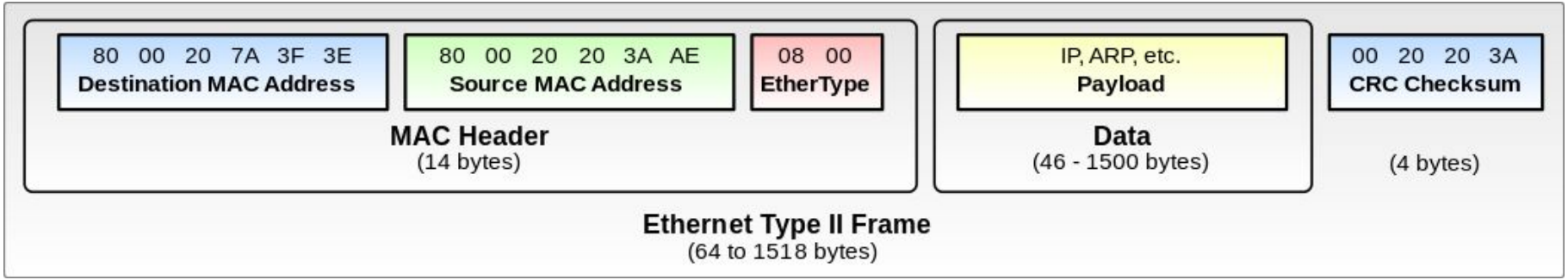
각 패킷의 구조


ICMP Packet Overview
MAC Header 14 bytes
IP Header 20 bytes
ICMP Header 8 bytes
ICMP Data Variable
Ethernet Frame
IP Packet
ICMP Packet
OPENMANIAK.COM


Message
Packed into UDP
UDP
Data
Packed into IP packet
IP
IP Data
Packed into ethernet frame
Frame
Frame Data
Footer

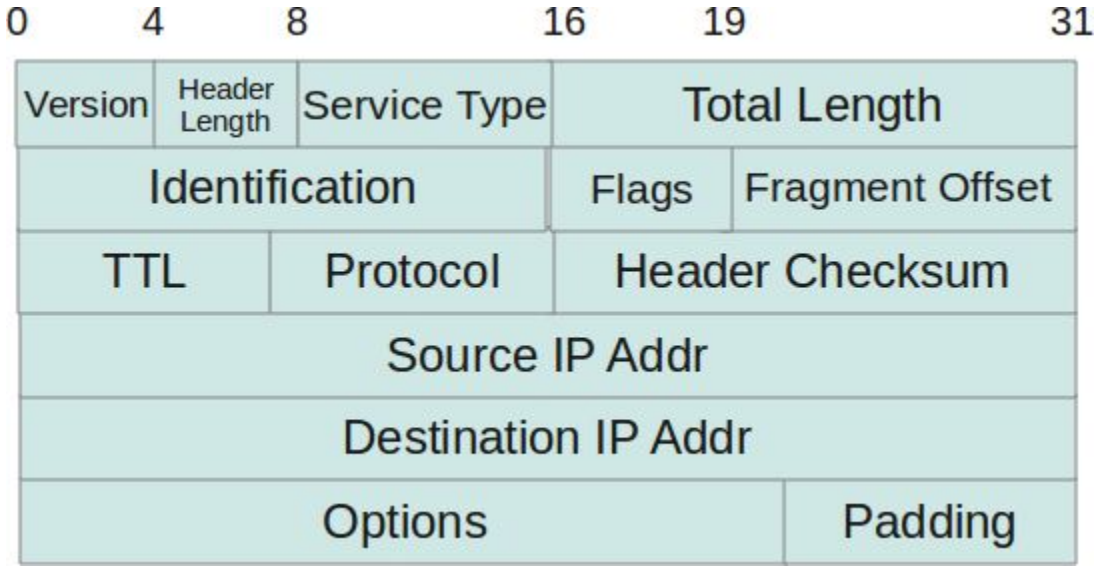
Ethernet Frame

- 이더넷 프레임



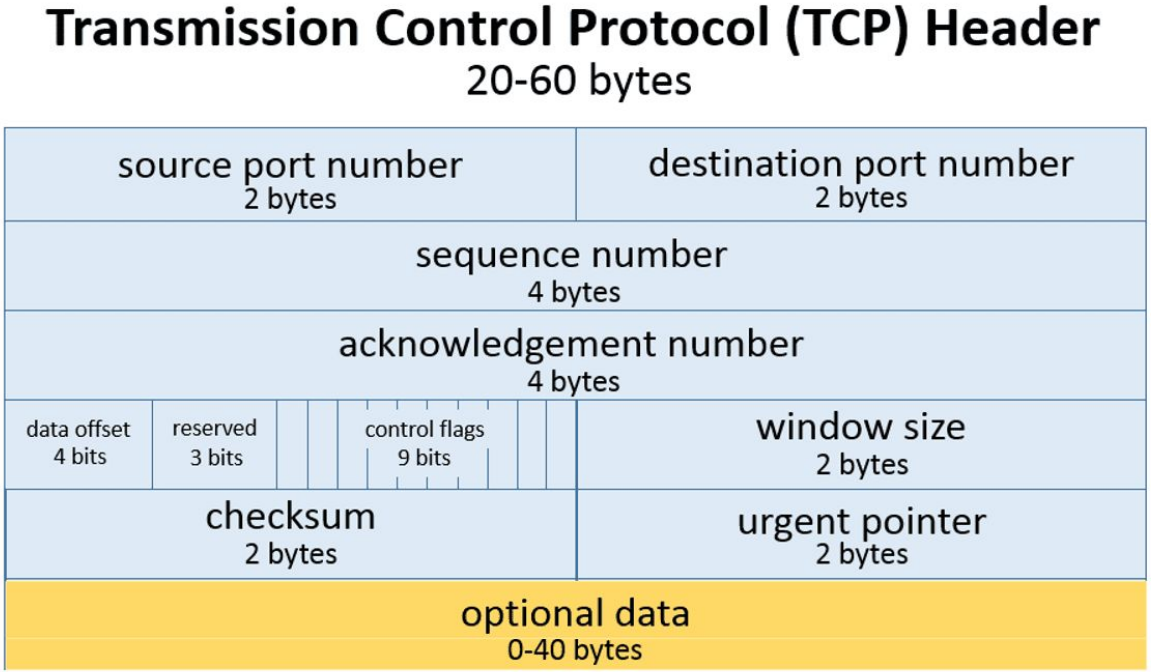
IP Header

- IP헤더의 구조



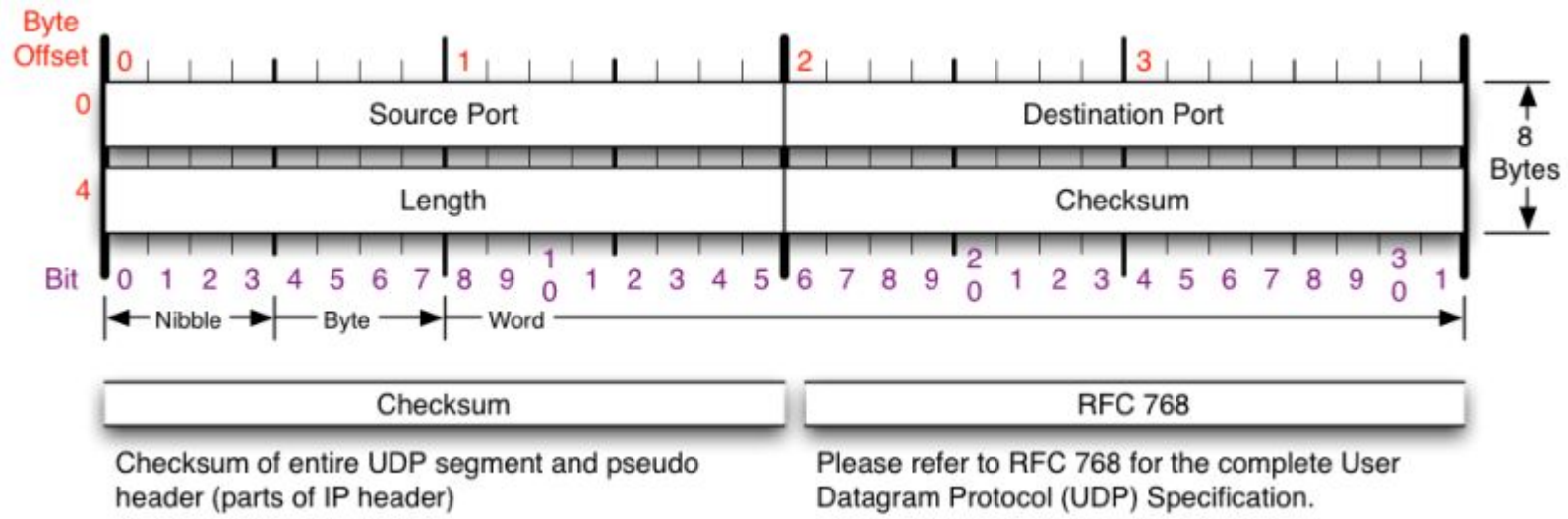
TCP Header

- TCP헤더의 구조

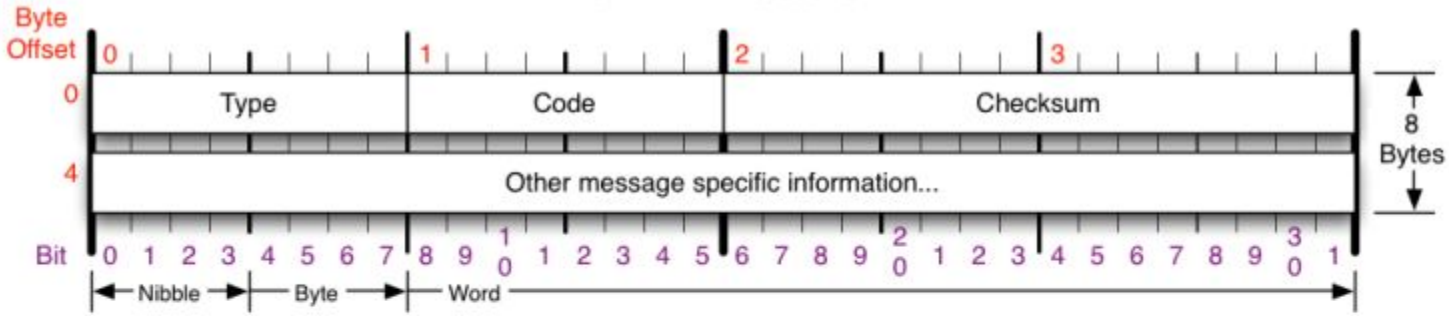


UDP Header

- UDP 헤더의 구조



ICMP Header



ICMP Message Types

Type	Code/Name
0	Echo Reply
3	Destination Unreachable
	0 Net Unreachable
	1 Host Unreachable
	2 Protocol Unreachable
	3 Port Unreachable
	4 Fragmentation required, and DF set
	5 Source Route Failed
	6 Destination Network Unknown
	7 Destination Host Unknown
	8 Source Host Isolated
	9 Network Administratively Prohibited
	10 Host Administratively Prohibited
	11 Network Unreachable for TOS

Type	Code/Name
3	Destination Unreachable (continued)
	12 Host Unreachable for TOS
	13 Communication Administratively Prohibited
4	Source Quench
5	Redirect
	0 Redirect Datagram for the Network
	1 Redirect Datagram for the Host
	2 Redirect Datagram for the TOS & Network
	3 Redirect Datagram for the TOS & Host
8	Echo
9	Router Advertisement
10	Router Selection

Type	Code/Name
11	Time Exceded
	0 TTL Exceeded
	1 Fragment Reassembly Time Exceeded
12	Parameter Problem
	0 Pointer Problem
	1 Missing a Required Operand
	2 Bad Length
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
30	Traceroute

Checksum

Checksum of ICMP header

RFC 792

Please refer to RFC 792 for the Internet Control Message protocol (ICMP) specification.

Packet capture

- 패킷 캡쳐를 위한 로우 소켓 생성 (프로그램 실행시 관리자 권한 필요)
- 모든 프로토콜을 받겠다는 의미

```
sock_raw = socket( AF_PACKET , SOCK_RAW , htons(ETH_P_ALL))
```

Include Header

- 각 헤더의 구조체를 Include 해줘야 함

```
#include<netinet/ip_icmp.h>
#include<netinet/udp.h>
#include<netinet/tcp.h>
#include<netinet/ip.h>
#include<netinet/if_ether.h>
#include<net/ethernet.h>
```

UDP Header Structure

- sudo find / -name udp.h
- ethernet.h / tcp.h / icmp.h/ ip.h 모든 헤더의 구조를 파악 해야함
- 구조체 분석 후 작성

```
struct udphdr
{
  __extension__ union
  {
    struct
    {
      u_int16_t uh_sport;               /* source port */
      u_int16_t uh_dport;               /* destination port */
      u_int16_t uh_ulen;                /* udp length */
      u_int16_t uh_sum;          /* udp checksum */
    };
    struct
    {
      u_int16_t source;
      u_int16_t dest;
      u_int16_t len;
      u_int16_t check;
    };
  };
```

Packet Capture Part

- 패킷을 받은 후 IP 헤더를 확인

```
void PacketCapture(unsigned char* buffer, int size){
    //이더넷 헤더를 제외한, 패킷의 IP 헤더를 받아온다
    struct iphdr *iph =                                    ;
    ++total; //패킷의 총 갯수
    switch (iph->protocol) //프로토콜을 체크
    {
        case 1:  //ICMP 프로토콜
            icmp_packet(buffer , size);
            break;
        case 6:  //TCP 프로토콜
            tcp_packet(buffer , size);
            break;
        case 17: //UDP 프로토콜
            udp_packet(buffer , size);
            break;
        default: //다른 프로토콜
            break;
    }
}
```

Result

```
Ethernet Header
   |-Destination Address : 1C-6A-7A-1F-4C-3F
   |-Source Address      : 74-27-EA-0F-A7-87
   |-Protocol            : 8

IP Header
   |-IP Version         : 4
   |-IP Header Length   : 5 DWORDS or 20 Bytes
   |-Type Of Service    : 16
   |-IP Total Length    : 92  Bytes(Size of Packet)
   |-Identification     : 63083
   |-TTL      : 64
   |-Protocol : 6
   |-Checksum : 62203
   |-Source IP         : 168.188.125.218
   |-Destination IP    : 168.188.129.209
```

```
TCP Header
   |-Source Port      : 22
   |-Destination Port : 3456
   |-Sequence Number    : 761379570
   |-Acknowledge Number : 2028658082
   |-Header Length      : 5 DWORDS or 20 BYTES
   |-Urgent Flag          : 0
   |-Acknowledgement Flag : 1
   |-Push Flag            : 1
   |-Reset Flag           : 0
   |-Synchronise Flag     : 0
   |-Finish Flag          : 0
   |-Window         : 1452
   |-Checksum       : 34061
   |-Urgent Pointer : 0
```

Result

```
Ethernet Header
   |-Destination Address : FF-FF-FF-FF-FF-FF
   |-Source Address      : 50-B7-C3-AD-BC-F9
   |-Protocol            : 8

IP Header
   |-IP Version        : 4
   |-IP Header Length  : 5 DWORDS or 20 Bytes
   |-Type Of Service   : 0
   |-IP Total Length   : 96  Bytes(Size of Packet)
   |-Identification    : 19865
   |-TTL      : 128
   |-Protocol : 17
   |-Checksum : 40868
   |-Source IP        : 168.188.125.215
   |-Destination IP   : 168.188.125.255
```

```
UDP Header
   |-Source Port      : 137
   |-Destination Port : 137
   |-UDP Length       : 76
   |-UDP Checksum     : 37759
```

Result

```
Ethernet Header
   |-Destination Address : 00-00-00-00-00-00
   |-Source Address      : 00-00-00-00-00-00
   |-Protocol            : 8

IP Header
   |-IP Version        : 4
   |-IP Header Length  : 5 DWORDS or 20 Bytes
   |-Type Of Service   : 0
   |-IP Total Length   : 84  Bytes(Size of Packet)
   |-Identification    : 41654
   |-TTL      : 64
   |-Protocol : 1
   |-Checksum : 39408
   |-Source IP        : 127.0.0.1
   |-Destination IP   : 127.0.0.1
```

```
ICMP Header
   |-Type : 8   |-Code : 0
   |-Checksum : 28952
```

Question

- 질문사항을 leeo75@cs-cnu.org 로 메일 보내주세요
- 과제 보고서 및 결과 화면 제출 (TCP / UDP / ICMP)