8주차_
RawSocket

데이터 네트워크연구실
이현호
lee075@cs-cnu.org

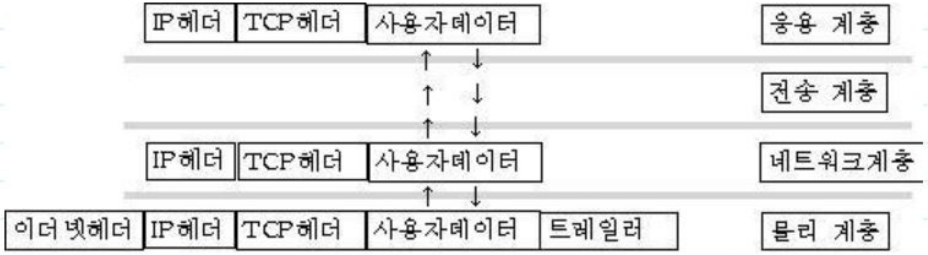
Goals

- Raw Socket 이해하기
- ICMP 프로토콜 구현하기
- 패킷 캡쳐 프로그램

Raw Socket

- 일반 소켓을 이용하는 것 보다 세부적인 조작 가능
- IP 데이터 그램, ICMP, IGMP를 읽고 쓰는 것이 가능

일반적인 소켓의 흐름도



RAW소켓의 데이터 흐름도

Raw Socket 생성

```
SOCKET sock = socket(AF_INET, SOCK_RAW, protocol);
if(sock == INVALID_SOCKET) err_quit("socket()");
```

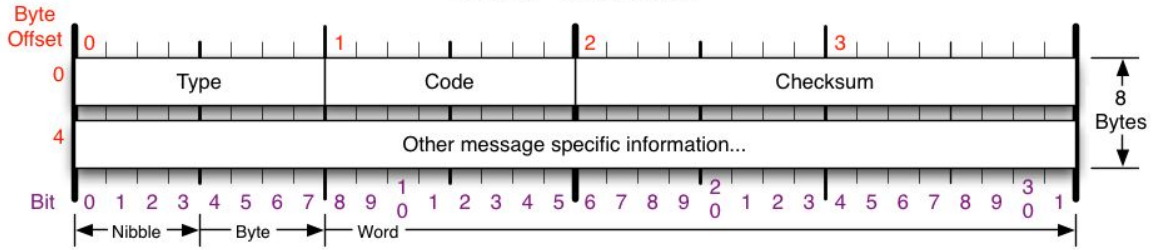
- Socket 함수 호출 시 두 번째 매개변수 값
  - SOCK_RAW로 설정
- Protocol 입력

IP (dummy)	IPPROTO_IP	0
ICMP	IPPROTO_ICMP	1
IGMP	IPPROTO_IGMP	2
Gateway	IPPROTO_GGP	3
TCP	IPPROTO_TCP	6

PUP	IPPROTO_PUP	12
UDP	IPPROTO_UDP	17
XND IDP	IPPROTO_IDP	22
Net Disk	IPPROTO_ND	77
Raw IP	IPPROTO_RAW	255

간단한 ICMP 체크 프로그램

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
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
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.

```
1 #include <stdlib.h>
2 #include <string.h>
3 #include <netinet/ip.h>
4 #include <netinet/ip_icmp.h>
5 #include <arpa/inet.h>
6 #include <errno.h>
7 #include <sys/socket.h>
8 #include <stdio.h>
9 #include <unistd.h>
10
11 int in_cksum(u_short *p, int n);
12
13 int main(int argc, char **argv)
14 {
15         int icmp_socket;
16         int ret;
17         struct icmp *p, *rp;
18         struct sockaddr_in addr, from;
```

```
19          struct ip *ip;
20          char buffer[1024];
21          socklen_t sl;
22          int hlen;
23
24          icmp_socket = socket(                                    );
25          if(icmp_socket < 0)
26          {
27                  perror("socket error : ");
28                  exit(0);
29          }
30
31          memset(buffer, 0x00, 1024);
32
33          p = (struct icmp *)buffer;
34          p->icmp_type=ICMP_ECHO;
35          p->icmp_code=0;
36          p->icmp_cksum=0;
```

```
        p->icmp_seq=15;
        p->icmp_id=getpid();

        p->icmp_cksum = in_cksum((u_short *)p, 1000);
        memset(&addr, 0, sizeof(addr));
        addr.sin_addr.s_addr = inet_addr(argv[1]);
        addr.sin_family = AF_INET;

        ret=sendto(icmp_socket,p,sizeof(*p),MSG_DONTWAIT,(struct sockaddr *)&addr, sizeof(addr)
);
        if (ret< 0)
        {
                perror("sendto error : ");
        }

        sl=sizeof(from);
        ret = recvfrom(icmp_socket,buffer, 1024, 0, (struct sockaddr *)&from, &sl);
        if (ret < 0)
```

```
54          {
55                  printf("%d %d %d\n", ret, errno, EAGAIN);
56                  perror("recvfrom error : ");
57          }
58
59          ip = (struct ip *)buffer;
60          hlen = ip->ip_hl*4;
61          rp = (struct icmp *)(buffer+hlen);
62          printf("reply from %s\n", inet_ntoa(from.sin_addr));
63          printf("Type : %d \n", rp->icmp_type);
64          printf("Code : %d \n", rp->icmp_code);
65          printf("Seq  : %d \n", rp->icmp_seq);
66          printf("Iden : %d \n", rp->icmp_id);
67          return 1;
68  }
```

```
70 int in_cksum( u_short *p, int n ){
71         register u_short answer;
72         register long sum = 0;
73         u_short odd_byte = 0;
74         while( n > 1 )  {
75                 sum += *p++;
76                 n -= 2;
77         }
78         if( n == 1 )    {
79                 *( u_char* )( &odd_byte ) = *( u_char* )p;
80                 sum += odd_byte;
81
82         }
83         sum = ( sum >> 16 ) + ( sum & 0xffff );
84         sum += ( sum >> 16 );
85         answer = ~sum;
86         return ( answer );
87 }
```

패킷 캡처 프로그램 개발

```
hyunholee@DNLAB:~/temp/RAW_SOCKET$ wget computer.cnu.ac.kr
--2017-11-06 16:25:10--  http://computer.cnu.ac.kr/
Resolving computer.cnu.ac.kr (computer.cnu.ac.kr)... 168.188.25
4.50
Connecting to computer.cnu.ac.kr (computer.cnu.ac.kr)|168.188.2
54.50|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘index.html.1’
```

패킷 캡처 프로그램 개발

```
HTTP/1.1 200 OK^M
Date: Mon, 06 Nov 2017 07:22:06 GMT^M
Server: Apache/2.4.16 (Fedora) OpenSSL/1.0.1k-fips PHP/5.6.23 mod_perl/2.0.9 Perl/v5
.20.3^M
Last-Modified: Fri, 30 Oct 2015 07:39:37 GMT^M
ETag: "47-5234d87946352"^M
Accept-Ranges: bytes^M
Content-Length: 71^M
Keep-Alive: timeout=5, max=100^M
Connection: Keep-Alive^M
Content-Type: text/html^M
^M
<meta http-equiv="refresh" content="0;url=http://computer.cnu.ac.kr">
```

패킷 캡처 프로그램 개발

- TCP만을 처리 -> protocol : IPPROTO_TCP

```
1 #include <netinet/in.h>
2 #include <stdio.h>
3 #include <netinet/ip_icmp.h>
4 #include <netinet/udp.h>
5 #include <netinet/tcp.h>
6 #include <netinet/ip.h>
7 #include <sys/socket.h>
8 #include <arpa/inet.h>
9 #include <sys/ioctl.h>
10 #include <sys/types.h>
11
12 #include <stdlib.h>
13 #include <string.h>
14 #include <unistd.h>
15
16 #define PACKET_LENGTH  65536
17
18 void PrintPacket(unsigned char* , int);
```

```
19 void PrintTcp(unsigned char *, int size);
20 void PrintData (unsigned char *, int Size);
21
22 int main(int argc, char **argv)
23 {
24         int readn;
25         socklen_t addrlen;
26         int sock_raw;
27         struct sockaddr_in saddr;
28
29         unsigned char *buffer = (unsigned char *)malloc(PACKET_LENGTH);
30
31         sock_raw = socket(                                    );
32         if(sock_raw < 0)
33         {
34                 return 1;
35         }
36        while(1)
```

```
        while(1)
        {
                addrlen = sizeof(saddr);
                memset(buffer, 0x00, PACKET_LENGTH);
                readn = recvfrom(sock_raw , buffer , PACKET_LENGTH , 0 , (struct sockaddr *)&saddr , &addrlen);
                if(readn <0 )
                {
                        return 1;
                }
                PrintPacket(buffer , readn);
        }
        close(sock_raw);
        return 0;
}
```

```
51 void PrintPacket(unsigned char* buffer, int size)
52 {
53         struct iphdr *iph = (struct iphdr*)buffer;
54         printf("protocol : %d\n",                  );
55         switch (                  )
56         {
57
58
59
60
61
62
63
64
65
66                 default:
67                         break;
68         }
```

IP_header structure

- 참고 http://tmdgus.tistory.com/124

```
struct iphdr {
  unsigned char     ihl:4,          // 헤더 길이      // header length
  unsigned int      version:4;      // 버전           // version
  unsigned char     tos;            // 서비스 타입    // type of service
  unsigned short    tot_len;        // 전체 길이      // total length
  unsigned short    id;             // identification
  unsigned short    frag_off;       // fragment offset field
  unsigned char     ttl;            // time to live
  unsigned char     protocol;       // protocol
  unsigned short    check;          // check sum
  unsigned long     saddr;          // source address
  unsigned long     daddr;          // dest address
  /* The options start here. */
};
```

```
69 }
70
71 void PrintTcp(unsigned char* buf, int size)
72 {
73         unsigned short iphdrlen;
74         unsigned char *data;
75
76         struct iphdr *iph = (struct iphdr *)buf;
77         iphdrlen = iph->ihl*4;
78         struct tcphdr *tcph=(struct tcphdr*)(buf + iphdrlen);
79
80         data = (unsigned char *)(buf + (iph->ihl*4) + (tcph->doff*4));
81         printf("%s", data);
82 }
83
84
```

```
hyunholee@DNLAB:~/temp/RAW_SOCKET$ wget computer.cnu.ac.kr
--2017-11-07 15:08:30--  http://computer.cnu.ac.kr/
Resolving computer.cnu.ac.kr (computer.cnu.ac.kr)... 168.188.254.50
Connecting to computer.cnu.ac.kr (computer.cnu.ac.kr)|168.188.254.50
|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘index.html.1’
```

Asignment

- ICMP 체크 프로그램 결과화면
- 패킷 캡처 프로그램 결과화면
- 소스코드 주석 및 설명 보고서 제출