How to implement IPRAW for W5500

Version 1.0

For more information, visit our website at http://www.wiznet.co.kr

Table of Contents

1 IPRAW Introduction

Internet Protocol (IP) RAW는 Transmission Control Protocol (TCP) 와 User Datagram Protocol (UDP) 의 하위 protocol 계층인 IP layer를 이용한 Data 통신이다. Figure 1은 Application Data가 각 하위 layer로 전달되는 Data Encapsulation과정을 도식화 한 것이다. W5500은 Link layer부터 Transport layer까지의 과정이 Hardwired로 구현 되어 있다. IP layer의 Data를 처리하기 위한 IPRAW mode를 지원하며, IPRAW는 protocol 에 따라 ICMP(0x01), IGMP(0x02)와 같은 IP layer의 protocol을 지원한다. 하지만 필요에 따라 Host는 SOCKET을 IPRAW mode로 open하여 이를 직접 구현하여 처리할 수 있다. 여기서는 IP Layer Protocol인 ICMP를 알아보고, ICMP를 이용한 간단한 Ping Application을 구현해 본다.

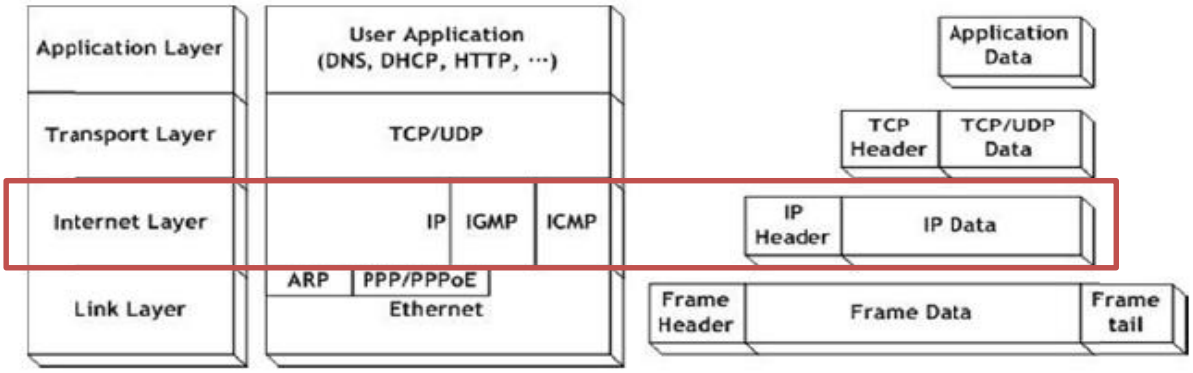

Figure 1
Encapsulation of data as it goes down the protocol stack

2 IPRAW SOCKET

W5500은 8개의 SOCKET을 지원하며, 모든 SOCKET은 IPRAW모드를 지원한다. IPRAW mode로 SOCKET n(the n th SOCKET)을 사용할 경우에는 어떤 protocol을 사용할지 반드시 IP header의 protocol number field를 설정해야 한다. protocol number의 경우 socket open 이전에 SOCKET n protocol register(Sn_PROTO)에 반드시 설정해야 한다.

Protocol	Number	Semantic	W5500 Support
-	0	Reserved	O
ICMP	1	Internet Control Message Protocol	O
IGMP	2	Internet Group Management Protocol	O
TCP	6	Transmission Control Protocol	X
EGP	8	Exterior Gateway Protocol	O
UDP	17	User Datagram Protocol	X
Others	-	Another Protocols	O

Table 1 Key Protocol in IP layer

Table 1은 IP layer의 protocol를 보여준다. IPRAW모드로 open된 socket은 TCP(0x06)나 UDP(0x11)을 지원하지 않는다. 또한 ICMP로 설정된 IPRAW mode socket은 IGMP나 그 외 프로토콜의 데이터를 수신할 수 없다. W5500은 칩의 Initialization이 끝난 후, Ping Request에 대한 Ping Reply를 자동으로 처리한다. 그러나 IPRAW SOCKET n을 ICMP protocol로 OPEN한 경우 Hardwired Ping Reply Logic은 Disable 된다는 것에 유의하기 바란다.

IPRAW Data의 구조는 Figure 2와 같다. IPRAW Data는 6 bytes의 PACKET-INFO와 DATA packet으로 이루어지며, PACKET-INFO는 송신자의 정보(IP address)와 DATA packet의 길이가 포함된다. IPRAW mode의 Data 수신은 UDP의 PACKET-INFO에서 송신자의 Port number 처리를 제외하고는 UDP data 수신과 모두 동일하다.

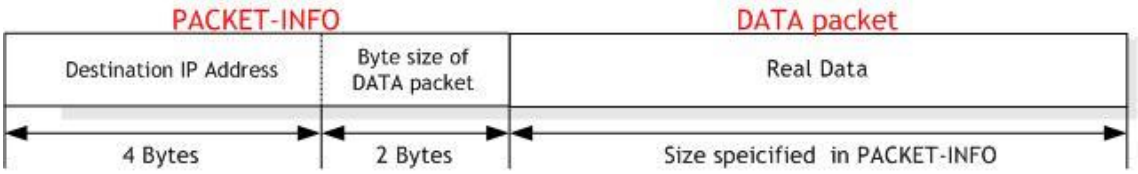

Figure 2 received IPRAW data format

IPRAW의 SOCKET Lifecycle은 OPEN, SEND, RECEIVE, CLOSE로 이루어 진다. SOCKET의 Lifecycle의 구현 방법에 대해서 살펴보도록 한다.

2.1 OPEN

Socket number를 S로 선택한 후, Sn_PROTO에 Protocol number(ICMP)를 설정하고 socket()를 사용하여 IPRAW mode로 설정되어 있는 SOCKET n을 호출한다. 그 후 Sn_SR를 체크하여 SOCK_IPRAW 상태를 확인하고 Sn_SR이 SOCK_IPRAW(0x32)로 변경되면 SOCKET n OPEN이 완료된다.

```
/* Create Socket */
  IINCHIP_WRITE(Sn_PROTO(s), IPPROTO_ICMP);    // set ICMP Protocol
  if(socket(s,Sn_MR_IPRAW,port,0)!=s){ // open the SOCKET with IPRAW mode, if fail then
Error
        printf( "\r\n socket %d fail r\n",   (s)) ;
}
/* Check socket register */
while(getSn_SR(s)!=SOCK_IPRAW);
```

Example 2.1.1 Socket Open

2.2 SEND

Sendto()를 이용하여 PingRequest에 저장된 정보들을 Destination address로 전송한다. IPRAW mode로 설정된 Socket을 사용한다.


```
/* sendto ping_request to destination */
// Send Ping-Request to the specified peer.
if(sendto(s,(uint8_t *)&PingRequest,sizeof(PingRequest),addr,port)==0){
printf( "\r\n Fail to send ping-reply packet  r\n") ;
}
```

Example 2.2.1 Send Data

2.3 RECEIVE

Recvfrom()을 이용하여 Destination address로부터 수신받은 Data를 data_buf에 저장한다.

IPRAW mode로 설정된 Socket을 사용한다.

```
if ( (rlen = getSn_RX_RSR(s) ) > 0){
        /* receive data from a destination */
 len = recvfrom(s, (uint8_t *)data_buf,rlen,addr,&port);
}
```

Example 2.3.1 Receive Data

2.4 CLOSE

Close()를 이용하면 되며, IPRAW socket이 필요하지 않을 경우 사용하면 된다.

3 ICMP(Internet Control Message Protocol)

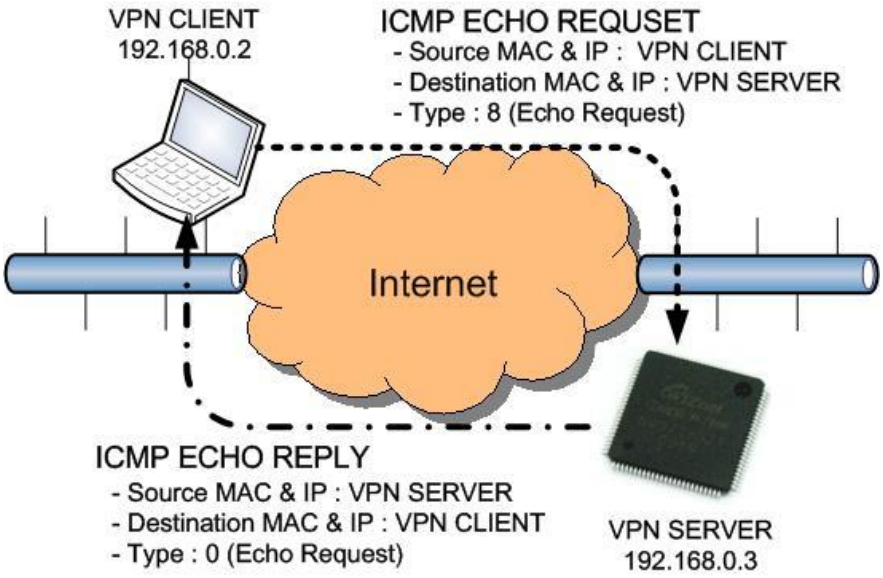

Figure 3 ICMP ECHO REQUSET/REPLY

ICMP Echo은 주로 troubleshooting에 사용된다. 통신의 문제를 가지는 2개의 host가 존재 할 경우, ICMP Echo로 두 Host의 TCP/IP stack의 설정이 정확한지의 여부를 알 수 있다. Figure 3 은 잘 알려진 ping command를 보여준다. ICMP Echo Request(Ping) Packet일 경우 Type filed는 8의 값을 가지며, ICMP ECHO Reply packet일 경우 0의 값을 가진다.

Table 3.1과 Table 3.2은 각각 Message Format과 Message Type을 나타낸다.

1Byte	1Byte
Type	Code
Check Sum	
Type dependent	
Data	

Table 3.1 ICMP Message Format

Type	Semantic
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

Table 3.2 ICMP Message Type

"Ping" command을 실행하면 Figure 3과 같이 Source(VPN client)에서 Destination(VPN server)에 대한 Ping Echo Request메시지가 송신된다. Request메시지를 수신한 Destination은 Source에 대해 Echo Reply를 송신한다. Echo Reply는 Request 메시지와 동일한 ID, 시퀀스 번호 필드, 데이터로 구성된다. 따라서, Source는 Destination으로부터 수신한 Echo Reply와 Echo Request의 ID, 시퀀스 번호, 데이터를 비교하여 특정 Destination의 접속을 확인 할 수 있다.

3.1 Ping Implementation

Ping Message의 ICMP Type field는 '0'(Ping Reply) or '8'(Ping Request)을 가지며, code field는 '0'만을 가진다. 또, Check Sum, ID, Sequence Number Field로 2Byte씩 각각 나뉘어 재정의되어 값을 가진다. Ping Data는 가변길이를 가진다. Ping message format는 Table 3.1.1과 같다.

1Byte	1Byte
8 (0)	0
Check Sum	
ID	
Sequence Number	
Ping Data	

Table 3.1.1 Ping Message Format

Ping Message를 쉽게 구현하기 위해 구조체를 사용했으며, 이는 Example 3.1에 정의하였다.

```
#define BUF_LEN 32
#define PING_REQUEST 8
#define PING_REPLY 0
#define CODE_ZERO 0

typedef struct pingmsg
{
  uint8_t  Type;            // 0 - Ping Reply, 8 - Ping Request
  uint8_t  Code;            // Always 0
  int16_t  CheckSum;        // Check sum
  int16_t  ID;              // Identification
  int16_t  SeqNum;          // Sequence Number
  int8_t   Data[BUF_LEN];// Ping Data  : 1452 = IP RAW MTU - sizeof(Type+Code+CheckSum+ID+SeqNum)
} PINGMSGR;
```

Example 3.1 Ping Message Structure

Ping Application은 W5500 Driver Program인 Socket API중 UDP관련 API를 이용하여 구현할 수 있다. Table 3.1.2는 Ping을 위해 사용되는 주요 API Function들을 나타낸다.

API Function Name	Semantic
Socket	Open socket with IPRAW Mode
Sendto	Send Ping Request to Peer
Recvfrom	Receive Ping Reply from Peer
Close	Close Socket

Table 3.1.2 Socket API Functions

구현된 Ping Application은 IPRAW mode에서 사용할 socket과 Destination Address 를 parameter 로 설정한다. 그런 후에 user는 특정한 peer에 특정한 갯수의 ping request를 요청하고, peer로부터 Ping Reply를 받을 수 있다.

uint8 ping_auto(SOCKET s, uint8 *addr)

Function Name	Ping
Arguments	s - socket number addr - Peer IP Address

Table 3.1.3 ping_auto function

uint8 ping_request(SOCKET s, uint8 *addr)

Function Name	ping_request
Arguments	s - socket number addr - Peer IP Address

Table 3.1.4 ping_request function

uint8 ping_reply (SOCKET s, uint8 *addr, uint16 len)

Function Name	ping_reply
Arguments	s - socket number addr - Peer IP Address len - packet length

Table 3.1.5 ping_reply function

uint16 checksum(uint8 * data_buf, uint16 len)

Function Name	Checksum
Arguments	data_buf - ping message len - ping message length

Table 3.1.6 checksum function

Figure 3.1.1 는 간단한 ping application 을 보여준다. Ping test 를 위해 구현된 Checksum 검사, PingReply Message 와 Ping request Message 등을 처리하는 순서를 보여준다.

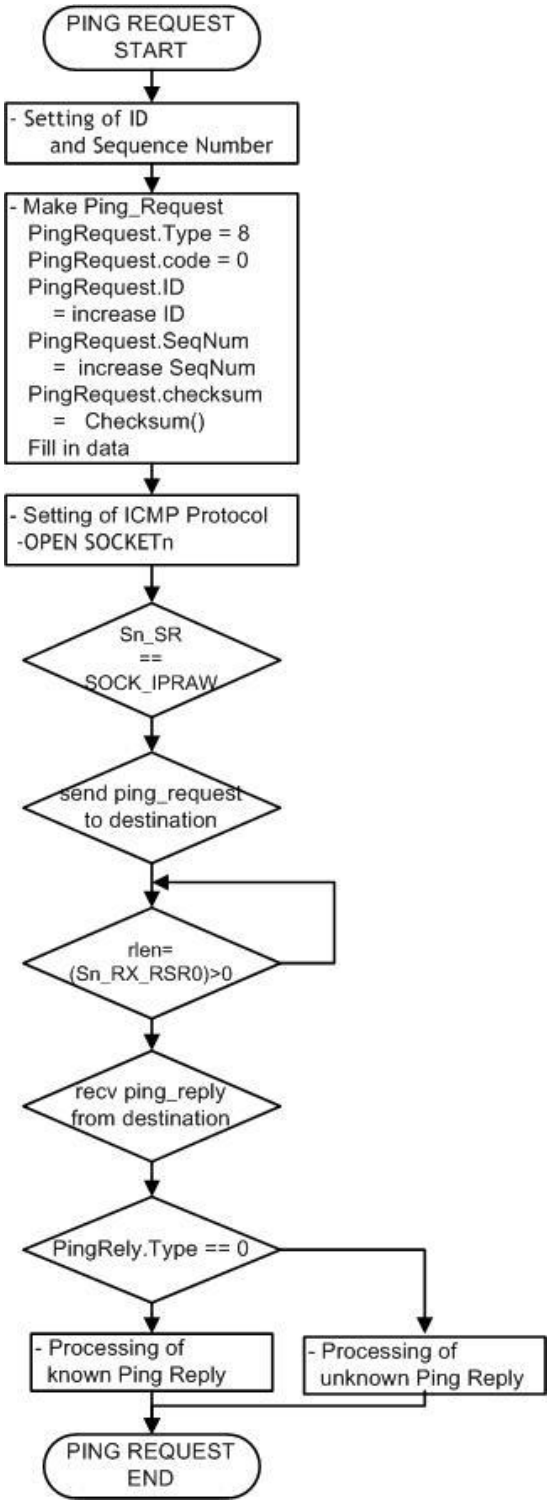

Figure 3.1.1 Flow chart of Ping Application

- Calling Ping Function

Ping application Function은 Destination IP 를 설정해야 하며, Ping Request Function은 Network Configuration 과 Ping request의 Parameter 설정 이후 호출 된다. Ping Request Function의 설정 과정을 Example 3.1.1에 정리하였다.

```
/* main.c */
/* setting of Destination IP address */
  pDestaddr[4]= {192,168,0,200};
/* Calling ping_request function */
/* pring_request( SOCKETn, COUNT, DESTINATION_IP, PORT) */
ping_auto(0,pDestaddr);
```

Example 3.1.1 Setting of Ping Request Function

- Ping Request

Ping Request Processing은 Ping Request의 Header와 Data의 작성, Protocol설정, SOCKET n OPEN, 전송으로 구성된다. Example 3.1.2는 Ping Request Processing을 보여준다. Ping Request의 헤더 및 Data를 작성 후 Checksum를 실행한다. Protocol을 ICMP로 설정하여 SOCKET를 IPRAW mode로 OPEN한 뒤 SENDTO함수를 이용하여 Ping Request를 Host PC로 전송한다.

```
/* ping_request.c */
/* make header of the ping-request   */
  PingRequest.Type = PING_REQUEST;     // Ping-Request
  PingRequest.Code = CODE_ZERO;        // Always '0'
  PingRequest.ID = htons(RandomID++); // set ping-request's ID to random integer value
// set ping-request's sequence number to random integer value
  PingRequest.SeqNum = htons(RandomSeqNum++);
/* Do checksum of Ping Request */
  PingRequest.CheckSum = 0;
  PingRequest.CheckSum = htons(checksum((uint8*)&PingRequest sizeof PingRequest)));
        :
/* set ICMP Protocol */
  IINCHIP_WRITE(Sn_PROTO(s), IPPROTO_ICMP);
socket(s,Sn_MR_IPRAW,3000,0) ; /* open the SOCKET with IPRAW mode */
/* sendto ping_request to destination */
sendto(s,(uint8 *)&PingRequest sizeof PingRequest),addr,3000);
```

Example 3.1.2 Ping Request

- Ping Reply

Example 3.1.3는 Ping Reply Processing을 보여준다. Ping Reply Processing은 Data의 수신, Ping Reply의 타입판별, 타입에 따른 처리로 구성된다. 수신된 Data의 Type을 확인하여 Ping Reply(0)일 경우 Ping Reply정보를 출력한다.

```
/* ping.c */
/* receive data from a destination */
  rlen = recvfrom(s, (uint8 *)&PingReply,rlen,addr,&port);
/* check the Type */
  if(PingReply.Type == PING_REPLY) {
/* check Checksum of Ping Reply */
     tmp_checksum = ~checksum(&data_buf,len);
   if(tmp_checksum != 0xffff)
          printf("tmp_checksum = %x\r\n",tmp_checksum)
      /* Output the Destination IP and the size of the Ping Reply Message*/
          :
   else{
      printf(" Unknown msg. \n");
}
```

Example 3.1.3 Ping Reply

3.2 Ping Request Demonstration

- test environment
  - MCU : STM32F103C8
  - Wiznet Chip : W5500
  - Used program: Flash Loader Demonstrator, Terminal, WireShark

Ping Request Test는 아래의 사항을 따라 System을 구성한다.

- Serial Cable를 STM32F103C8 + W5500의 보드에 연결한다.
- Ping application의 xxx.bin파일을 Flash Loader Demonstrator를 이용하여 Program한다.

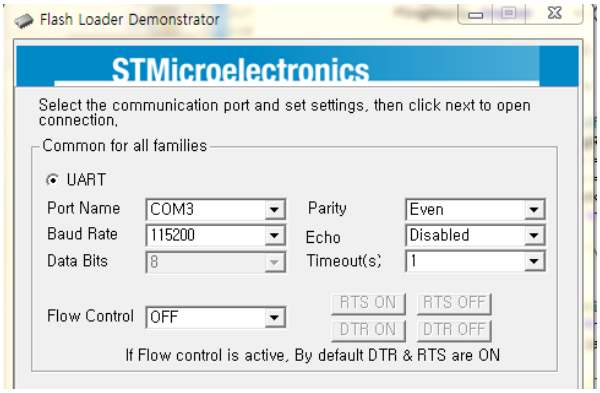

Figure 3.2.1 Flash Loader Demonstrator

- Test PC의 Network Information을 아래와 같이 수정한다.

  Source IP Address : 192.168.0.200

  Gateway IP Address : 192.168.0.1

  Subnet Mask : 255.255.255.0

- Hyper Terminal의 실행/설정 한다.

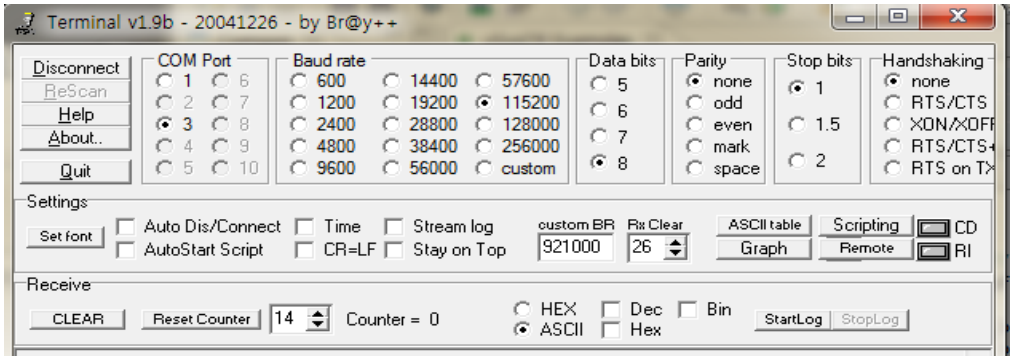

Figure 3.2.2 Terminal setting

- 보드에 전원을 인가한 후 Terminal을 이용하여 실행결과를 확인한다.
  Terminal 창에서 No.0의 Request의 ID/Sequence number와 wireshark의 첫번째 패킷의 ID/Sequence number를 확인하면 같은 패킷 인지 알 수 있다.

```
HCLK = 72MHz

=== W5500 NET CONF ===
MAC: 00:08:DC:00:AB:CD
SIP: 192.168.0.226
GAR: 192.168.0.1
SUB: 255.255.255.0
DNS: 0.0.0.0
======================
------------PING_TEST_START-----------------------
Send Ping Request  to Destination (192.168.0.200 ) ID:1234  SeqNum:4321 CheckSum:726a
Reply from 192.168.0.200  ID:1234 SeqNum:4321  :data size 52 bytes

Send Ping Request  to Destination (192.168.0.200 ) ID:1235  SeqNum:4322 CheckSum:7268
Reply from 192.168.0.200  ID:1235 SeqNum:4322  :data size 52 bytes

Send Ping Request  to Destination (192.168.0.200 ) ID:1236  SeqNum:4323 CheckSum:7266
Reply from 192.168.0.200  ID:1236 SeqNum:4323  :data size 52 bytes

Ping Request = 3, PING_Reply = 3
------------PING TEST OK----------
```

Figure 3.2.3 Execution result of Ping Request

ICMP Ping packet을 전송하기 전에 ARP request packet이 나가야 한다. 만약 ARP response packet 이 수신되지 않는다면 ICMP Ping packet은 전송되지 않는다.

Figure 3.2.4 wireshark 프로그램을 통해 Ping Application의 Packet을 보여준다.

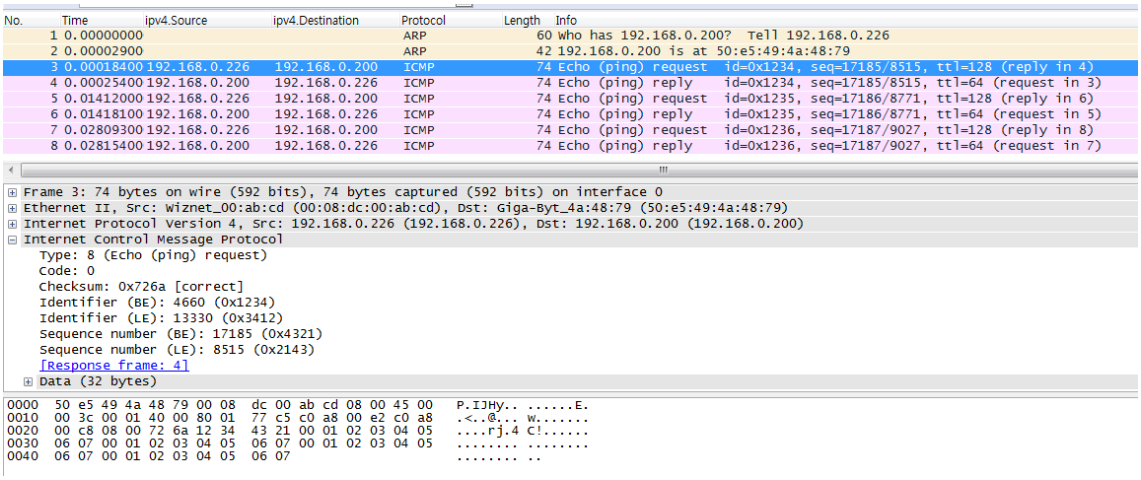

Figure 3.2.4 Execution result of Wireshark

Document History Information

Version	Date	Descriptions
Ver. 1.0	21FEB2014	Release

Copyright Notice

Technical Support: http://wizwiki.net/forum or support@wiznet.co.kr
Sales & Distribution: sales@wiznet.co.kr

For more information, visit our website at http://www.wiznet.co.kr