

# 데이터 통신

- ARP -

제출일	17.05.16
분 반	01
조	4조
조장	김규태
조원	신종욱
	윤동현
	이상인
	정석현
	허원철

## **-목 차-**

### **1 실습 개요**

- 실습 일시 및 장소
- 실습 목적
- 실습 시나리오

### **2 프로토콜 스택**

- 구조 설명

### **3 구현 설명**

### **4 결과 분석**

### **5 느낀 점**

# 1 실습 개요

## -실습 일시 및 장소

실습 일시: 4월19일~5월16일

실습 장소: 학생생활관,카페 및 공대5호관

## -실습 목적

ARP프로토콜의 기능을 구현한다. 패킷을 전달하기 위해서는 IP주소와 MAC주소를 알고 있어야 한다. ARP란 주소를 알아내는 방법으로 전송하고자 하는 상대방의 IP주소를 이용하여 메시지를 보내 상대방의 MAC주소를 알아내는 프로토콜이다. 목적지의 IP에 상대방의 IP를 삽입하고 MAC주소를 브로드캐스트로 설정한다. 메시지를 받는 쪽에서 자신의 IP주소와 비교하여 만약 일치할 경우 응답 메시지를 보내는데 이때 응답하는 쪽의 MAC주소를 같이 보낸다. 응답하는 쪽의 메시지를 보내는 쪽에서 받아서 해석하여 MAC주소를 알아낸다. 이러한 방법으로 IP를 이용하여 MAC주소를 받아오며 이러한 ARP는 Proxy와 Gratuitous 가 있다. Proxy의 경우 메시지를 받는 쪽에서 자신의 IP와 일치하지 않을 경우 라 하더라도 자신의 proxy table에 저장되어 있을 경우 대신해서 응답 메시지를 보내준다. 이러한 기능은 라우터를 구현할 때 이용되므로 이번 주소에서는 수동으로 IP를 입력하도록 한다. Gratuitous의 경우 MAC주소가 변경될 경우 자신의 MAC주소를 알리기 위하여 사용되는 것으로 자신의 IP주소와 변경된 MAC주소를 다시 한번 보내고 응답하는 쪽에서 자신의 Cache Table에 저장되어 있는 IP일 경우 MAC주소를 변경한다.

## -실습 시나리오

PC를 랜선으로 연결한 후 자신의 IP주소를 설정한다.

보내는 곳의 IP주소를 설정하여 ARP메시지를 전송한다.

목적지의 ip주소를 cache table에 저장하며 incomplete상태로 설정한다.

만약 응답을 받았을 경우 incomplete상태에서 complete상태로 변경한다.

받지 못했을 경우 일정시간이 지나면 cache table에서 해당 ip를 삭제한다.

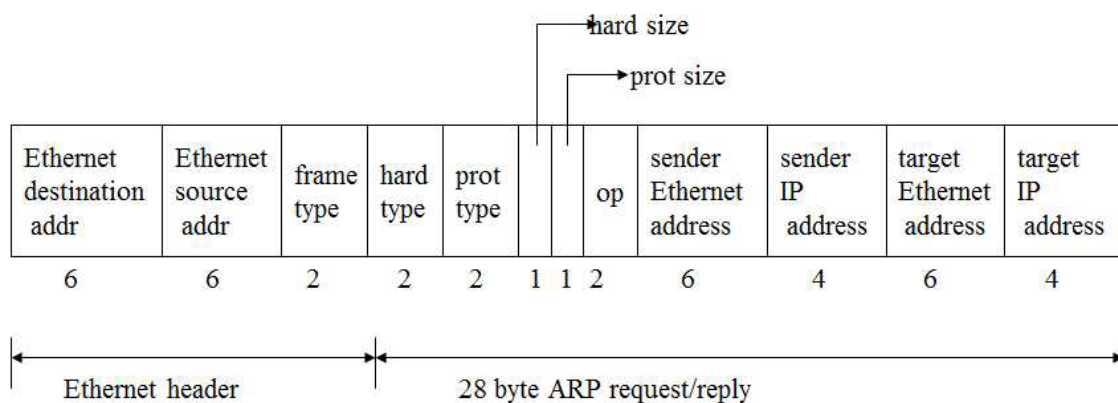
Gratuitous ARP의 경우 자신의 MAC주소가 변경될 경우에 사용하며 자신의 ip주소와 변경된 MAC주소를 브로드캐스트를 이용하여 전송하며 메시지를 받는 입장에서 해당IP가 자신의 Cache table에 있을 경우 해당하는 IP의 MAC주소를 변경한다.

Proxy ARP entry에 수동으로 Ip와 MAC을 입력하여 추가/삭제 할 수 있게 한다.

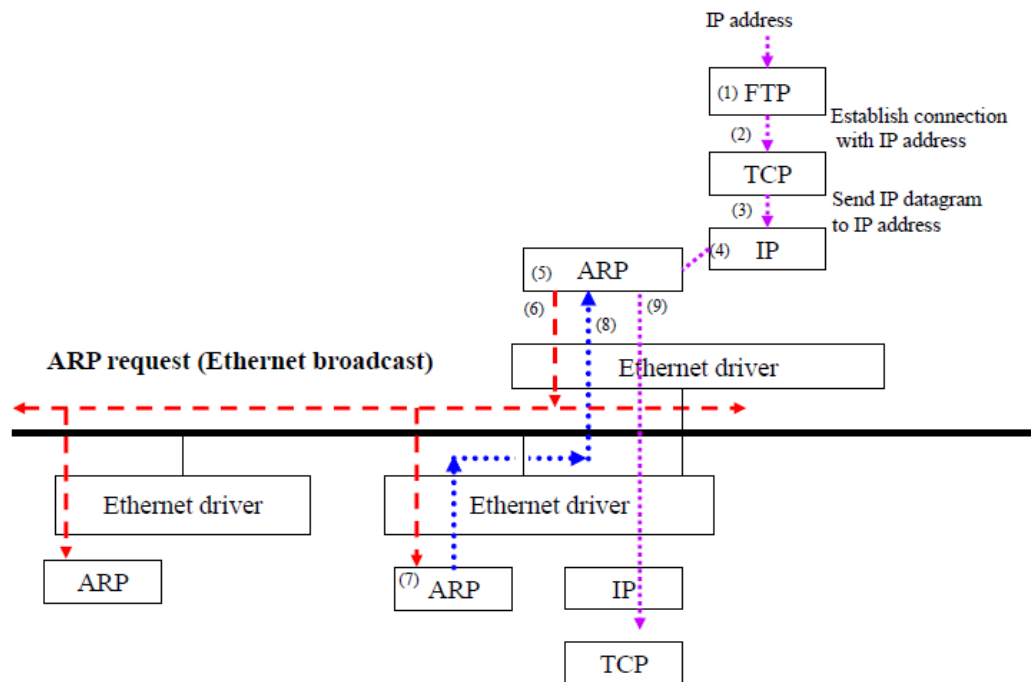
만약 받은 메시지의 IP가 자신의 Proxy ARP Entry에 존재할 경우 응답메시지를 보낸다.

## 2 프로토콜 스택

### -구조 설명

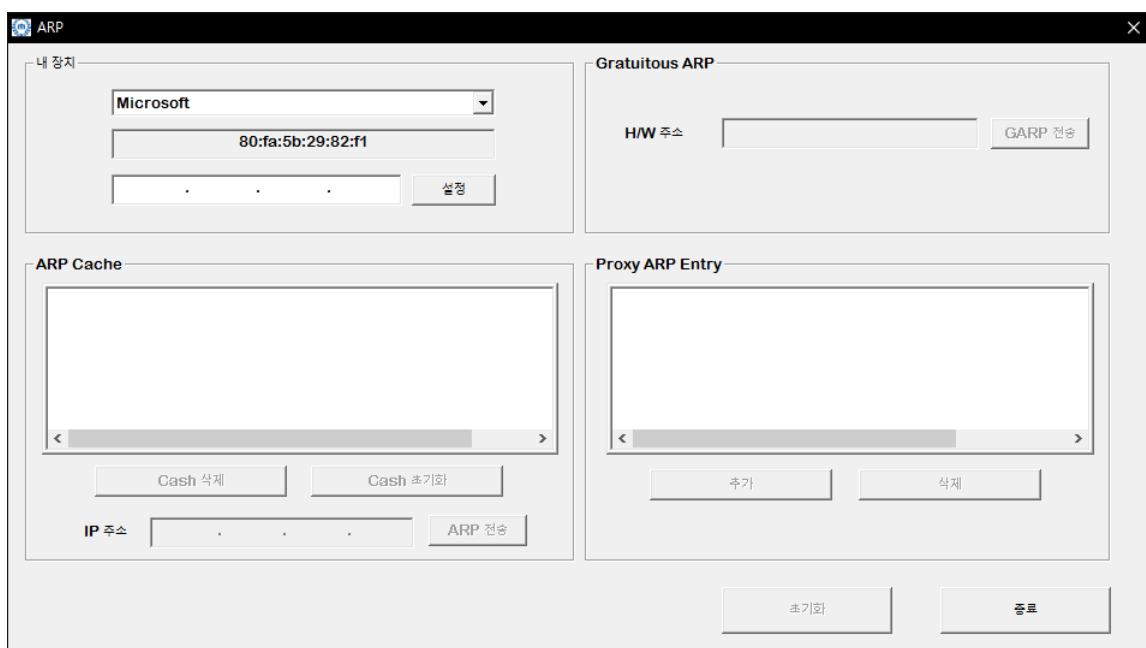


다음과 같은 frame을 이용하여 IP주소로 MAC주소를 알아오며 요청 시에는 target Ethernet address를 빈칸으로 하여 전송한다. 응답을 받은 쪽에서 IP주소를 비교하여 자신의 것과 일치하면 target Ethernet address를 채워준다. 이후 응답 시 target과 sender의 정보를 바꿔 요청자에게 다시 전송한다.



위와 같은 형태로 통신이 이루어지며 주소가 입력될 경우 FTP > TCP > IP > ARP 를 따라 layer가 이동된다. 이후 Ethernet broadcast를 이용하여 메시지를 전송하며 받는 쪽에서 ARPPlayer에서 자신의 IP와 일치하면 자신의 MAC주소를 추가하여 메시지를 돌려보낸다. 요청한 쪽은 받은 메시지를 이용하여 MAC주소를 알아낼 수 있게 된다.

### 3 구현 설명



## ARPDlg.cpp

프로그램 실행 시 버튼과 화면을 초기화 하여 프로그램을 사용 가능하게 하며 layout manager를 사용하여 layer들의 상위/하위를 결정해준다.

```
#define TIMER_REPLYTIME      180000
#define TIMER_VALIDTIME     1200000
```

Cache table을 초기화 할 때 사용하는 변수로서 위의 시간만큼 응답을 기다리며 정해진 시간(3분)이 지날 경우 Cache table에서 삭제한다. 또한 complete의 경우에도 정해진 사용시간(20분)동안 사용하지 않을 경우 역시 Cache table에서 제거한다.

```
// ARP Header 설정
m_ARP->SetHardType(ntohs(ARP_HARDTYPE)); // 1
m_ARP->SetProtType(ntohs(ARP_PROTOTYPE_IP)); // 0x0800
m_ARP->SetOpcode(ntohs(ARP_OPCODE_REQUEST)); // 1

// Byte Ordering 하지 않는 것.
m_ARP->SetSenderEtherAddress(arp_SenderEther); // Sender의 MAC 주소
m_ARP->SetSenderIPAddress(arp_SenderIP); // Sender의 IP 주소
m_ARP->SetTargetEtherAddress(arp_TargetEther); // Target의 MAC 주소
m_ARP->SetTargetIPAddress(arp_TargetIP); // Target의 IP 주소

// Ethernet Header 설정
memset(ether_DestEther, 0xff, 6); // Broadcast
m_Ether->SetEnetDstAddress(ether_DestEther); // Broadcast
m_Ether->SetEnetSrcAddress(arp_SenderEther); // Sender MAC 주소
```

```
inline void CARPDlg::SettingARPMsg()
```

ARP message를 설정하는 곳으로 각 frame에 해당하는 값을 넣어주며 Ethernet Hedaer의 dest는 broadcast로 설정한다.

```

m_ARP->SetHardType(ntohs(ARP_HARDTYPE));
m_ARP->SetProtType(ntohs(ARP_PROTOTYPE_IP));
m_ARP->SetOpcode(ntohs(ARP_OPCODE_REQUEST));
//sender 와 target의 IP가 동일하게 설정된다.
m_ARP->SetSenderEtherAddress(arp_SenderEther);
m_ARP->SetSenderIPAddress(arp_SenderIP);
m_ARP->SetTargetEtherAddress(arp_TargetEther);
m_ARP->SetTargetIPAddress(arp_SenderIP);

memset(broadcast,0xff,6);
m_Ether->SetEnetDstAddress(broadcast);
m_Ether->SetEnetSrcAddress(arp_SenderEther);

```

inline void CARPDlg::SettingGARPMsg()

ARP message를 와 거이 동일하며 sender와 target의 IP가 동일하게 설정된다.

## ARP.CPP

전체적인 다이얼로그를 컨트롤하는 코드입니다.

```

CARPApp::CARPApp()
{
    // 여기에 생성 코드가 추가된다.
}

CARPApp theApp;
// CARPApp 선언

BOOL CARPApp::InitInstance()
{
    CWinApp::InitInstance();
    // 셸 목록 뷰 컨트롤이 포함되어 있는 경우 셸 관리자를 만듭니다.
    CShellManager *pShellManager = new CShellManager;
    // 표준 초기화
    CARPDlg dlg;
    dlg.DoModal();
    // 다이얼로그 생성 및 함수 실행
    if (pShellManager != NULL)
    {
        delete pShellManager;
    }
    // 대화 상자가 끝났으므로 응용 프로그램의 메시지 펌프를 시작하지 않고 응용 프로그램을 끝낼 수 있도록 FALSE를 return한다
    return FALSE;
}

```

프로그램이 시작될 때 다이얼을 시작하고 셸을 생성하고 초기화를 하는 과정이다.

상자가 꺼지면 프로그램을 종료하도록 false를 리턴한다.

# ARPLAYER.cpp

ARP GARP PROXY ARP 기능을 구현한 레이어로

```

// 하위 Layer에서 받은 ppayload를 ARP Layer Header구조에 캐스팅해서 읽음
PARP_HEADER pFrame = (PARP_HEADER) ppayload ;
BOOL bSuccess = FALSE ;

// 하위 Layer인 Ethernet Layer 객체 생성
m_Ether = (CEthernetLayer *)GetUnderLayer( );

int index;

// 아래의 4가지 조건을 만족하는 경우는 "ARP"
// ARP 메시지의 Destination IP와 내 IP가 같다
// ARP 메시지의 Source IP와 자신 IP가 다르다
// ARP 메시지의 Destination IP와 source IP가 다르다
// Proxy Cache Entry에 없다
if( ( memcmp((char *)pFrame->arp_dstip, (char*)m_sHeader.arp_srcip, 4) == 0 ) &&
    ( memcmp((char *)pFrame->arp_srcip, (char*)m_sHeader.arp_srcip, 4) ) &&
    ( memcmp((char *)pFrame->arp_srcip, (char*)pFrame->arp_dstip, 4) ) &&
    ( ! ProxySearchExist(pFrame->arp_dstip) ) )
{
    if(ntohs(pFrame->arp_op) == ARP_OPCODE_REQUEST) // 상대방에게 요청 받은 경우
    {
        m_Ether->SetEnetDstAddress(pFrame->arp_srcnet, addr_e);
        m_Ether->SetEnetSrcAddress(m_sHeader.arp_srcnet, addr_e);
        SetHardType(ntohs(ARP_HARDTYPE));
        SetProtType(ntohs(ARP_PROTOTYPE_IP));
        SetOpcode(ntohs(ARP_OPCODE_REPLY));
        SetSenderEtherAddress(m_sHeader.arp_srcnet, addr_e);
        SetSenderIPAddress(m_sHeader.arp_srcip, addr_i);
        SetTargetEtherAddress(pFrame->arp_srcnet, addr_e);
        SetTargetIPAddress(pFrame->arp_srcip, addr_i);

        index = CacheAddItem(ppayload);
        // Cache Table에 추가 하고, 상위 Layer로 전달하여 Receive
        bSuccess = mp_aUpperLayer[0]->Receive((unsigned char*)index);
        mp_UnderLayer->Send((unsigned char*) &m_sHeader, ARP_HEADER_SIZE);
    }
    // 헤더를 하위 레이어로 전달
    else if(ntohs(pFrame->arp_op) == ARP_OPCODE_REPLY) // 상대방에게 응답 받은 경우
    {
        index = CacheUpdate(ppayload);
        bSuccess = mp_aUpperLayer[0]->Receive((unsigned char*)index);
        // Cache Table을 Update 해주고, 상위 Layer로 전달하여 Receive
    }
}
}

```

ARP 메시지를 보낼 때 4가지조건 이 만족하면 ARP 기능을 한다 이때 요청을 받은 경우에는 캐시 테이블에 추가하고 상위 Layer 로 전달하고 헤더를 하위Layer로 전달한다.

반대로 응답을 받은 경우 캐시 테이블을 update 하고 상위 Layer 로 전달한다.



```

// "Gratituous ARP"
// Entry에 등록된 IP주소에 맥주소를 변경하는것이다.
// ARP 메시지의 Source IP와 Destination과 같음.
else if(memcmp((char *)pFrame->arp_srcip.addr_i,(char*)pFrame->arp_dstip.addr_i,4) == 0)
{
    // ARP 메시지의 opcode가 Request(2) 이고, Source IP와 내 IP가 같으면
    // opcode를 Reply(1)로 바꿔서, MAC주소를 입력한 MAC주소로 변경하여 하위 Layer에 전달하여 전송한다
    if( ( ntohs(pFrame->arp_op) == ARP_OPCODE_REQUEST ) &&
        ( memcmp((char *)pFrame->arp_srcip.addr_i,(char*)m_sHeader.arp_srcip.addr_i,4) == 0 ) )
    {
        SetOpcode(ntohs(ARP_OPCODE_REPLY));
        SetSenderEtherAddress(m_sHeader.arp_srcenet.addr_e);
        mp_UnderLayer->Send((unsigned char*) &m_sHeader, ARP_HEADER_SIZE);
    }
    // ARP 메시지의 opcode가 Request(2) 이고, 이미 ARP Cache Table에 존재한다면
    // Cache Table을 Update 해주고, 상위 Layer로 전달하여 Receive
    else if( ( ntohs(pFrame->arp_op) == ARP_OPCODE_REQUEST ) &&
        ( CacheSearchExist(pFrame->arp_dstip) ) )
    {
        index = CacheUpdate(ppayload);
        bSuccess = mp_aUpperLayer[0]->Receive((unsigned char*) index);
    }
}
// 위의 두 조건이 모두 만족하지 않는다면 "Proxy ARP"
else
{
    // ARP 메시지의 opcode가 Request(2) 이고, Proxy ARP Entry에 존재한다면
    if( ( ntohs(pFrame->arp_op) == ARP_OPCODE_REQUEST ) &&
        ( ProxySearchExist(pFrame->arp_dstip) ) )
    {
        // 상매가 요청한 IP 주소와 MAC 주소를 등록후 보낸다.
        m_Ether->SetEnetDstAddress(pFrame->arp_srcenet.addr_e);
        m_Ether->SetEnetSrcAddress(m_sHeader.arp_srcenet.addr_e);
        SetHardType(ntohs(ARP_HARDTYPE));
        SetProtType(ntohs(ARP_PROTOTYPE_IP));
        SetOpcode(ntohs(ARP_OPCODE_REPLY));
        // Sender의 MAC 주소와 IP 주소는 Proxy ARP에서 지정
        SetTargetEtherAddress(pFrame->arp_srcenet.addr_e);
        SetTargetIPAddress(pFrame->arp_srcip.addr_i);

        // ARP Cache Table에 Item 추가
        CacheAddItem(ppayload);

        // 상위 Layer에 전달하여 Receive (256은 Byte Ordering 해서 0x100 -> 0x001)
        bSuccess = mp_aUpperLayer[0]->Receive((unsigned char*) 256);
    }
}

```

다음은 Gratuitous ARP 기능으로 entry에 등록된 IP주소의 mac 주소를 변경하고 IP주소 중복사용여부를 체크한다. ARP 메시지의 opcode가 Request(2) 이고, Source IP와 내 IP가 같으면 opcode를 Reply(1)로 바꿔서, MAC주소를 입력한 MAC주소로 변경하여 하위 Layer에 전달하여 전송한다

ARP 메시지의 opcode가 Request(2) 이고, 이미 ARP Cache Table에 존재한다면

Cache Table을 Update 해주고, 상위 Layer로 전달한다.

```

// 위의 두 조건이 모두 만족하지 않는다면 "Proxy ARP"
else
{
    // ARP 메시지의 opcode가 Request(2) 이고, Proxy ARP Entry에 존재한다면
    if( ( ntohs(pFrame->arp_op) == ARP_OPCODE_REQUEST ) &&
        ( ProxySearchExist(pFrame->arp_dstip) ) )
    {
        // 상대가 요청한 IP 주소와 MAC 주소를 등록후 보낸다.
        m_Ether->SetEnetDstAddress(pFrame->arp_srcenet.addr_e);
        m_Ether->SetEnetSrcAddress(m_sHeader.arp_srcenet.addr_e);
        SetHardType(ntohs(ARP_HARDTYPE));
        SetProtType(htohs(ARP_PROTOTYPE_IP));
        SetOpcode(ntohs(ARP_OPCODE_REPLY));
        // Sender의 MAC 주소와 IP 주소는 Proxy ARP에서 지정
        SetTargetEtherAddress(pFrame->arp_srcenet.addr_e);
        SetTargetIPAddress(pFrame->arp_srcip.addr_i);

        // ARP Cache Table에 Item 추가
        CacheAddItem(ppayload);

        // 상위 Layer에 전달하여 Receive (256은 Byte Ordering 해서 0x100 -> 0x001)
        bSuccess = mp_aUpperLayer[0]->Receive((unsigned char*) 256);

        mp_UnderLayer->Send((unsigned char*) &m_sHeader, ARP_HEADER_SIZE);
    }
}

return bSuccess;
}

```

PROXY ARP 기능은 ARP 메시지의 opcode가 Request(2) 이고, Proxy ARP Entry에 존재한다면

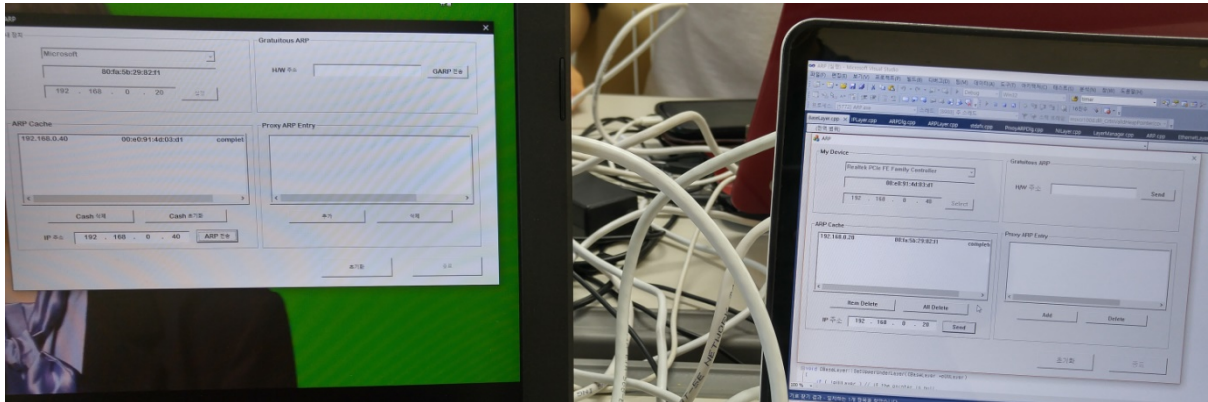
상대가 요청한 IP 주소와 MAC 주소를 등록 후 보내고 Sender의 MAC 주소와 IP 주소는 Proxy ARP에서 지정 한 다음 ARP Cache Table에 Item 추가하고 상위 Layer에 전달한다.

## ProxyARPDlg.cpp

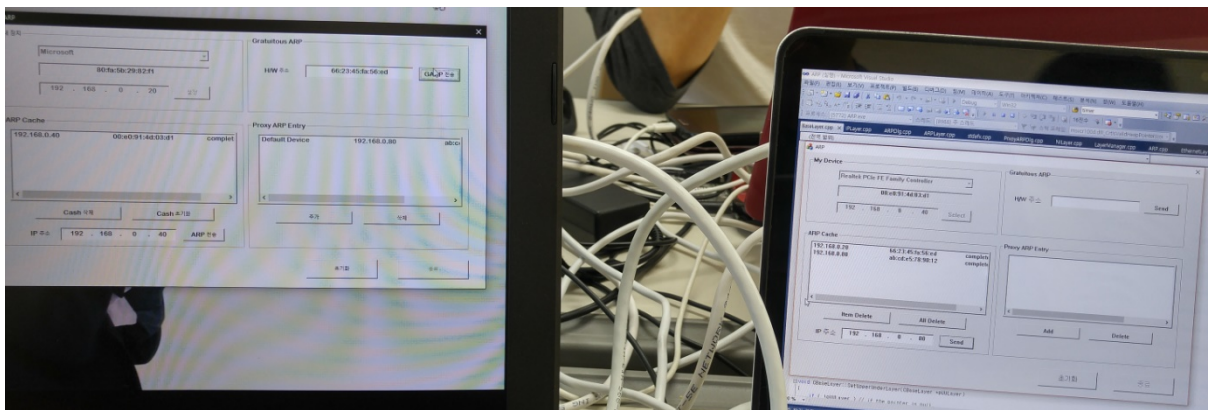
Proxy ARP Table에는 Device Name과 IP Address와 Ethernet Address필드가 있다. 라우터를 거치는 통신에서 사실 ARP의 target address는 sender가 아니기 때문에 proxy ARP는 ARP와 HOST사이의 연결고리라고 할 수 있다. ARP Packet이 발생하면 proxy ARP역할을 하는 PC또는 라우터가 기존의 ARP를 대신 ARP reply하여 proxy table에서 일치하는 receiver에게 Packet을 전달한다. 처음 ARP는 HOST IP와 sender의 MAC주소를 broadcast한다. 그런 뒤에 라우터가 만약 receiver의 IP주소를 갖고 있다면 라우터는 자신의 MAC주소를 sender에게 보내고 받은 PACKET을 receiver에게 보낸다.

Proxy ARP는 ARP 요청 메시지를 받았을 때, 내 호스트 주소가 ARPRequest패킷의 대상 프로토콜 주소가 아니라도, 임의의 host를 생성해서 IP와 MAC주소를 할당하여 해당 목적지 host에 대한 ARP 응답 메시지를 대신 보내주는 기능을 한다

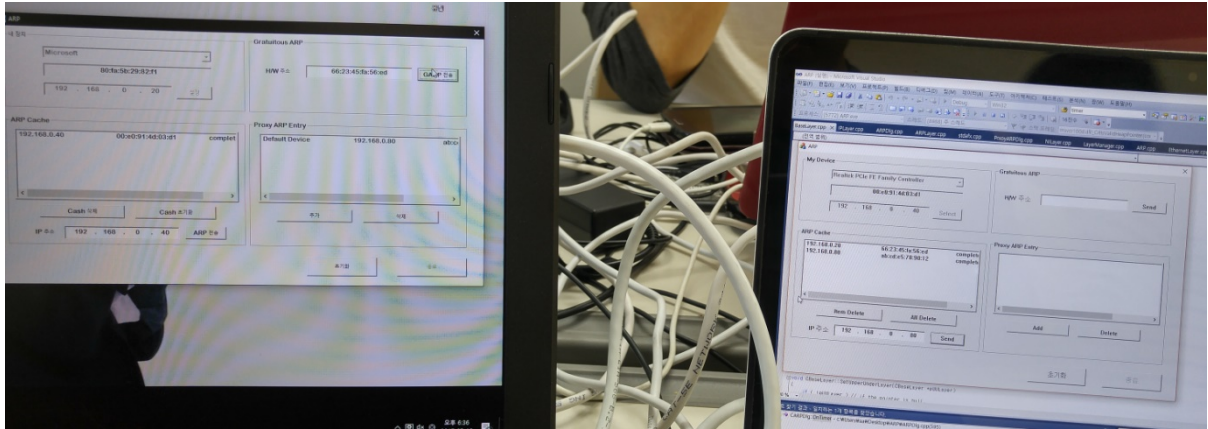
## 4. 결과 분석



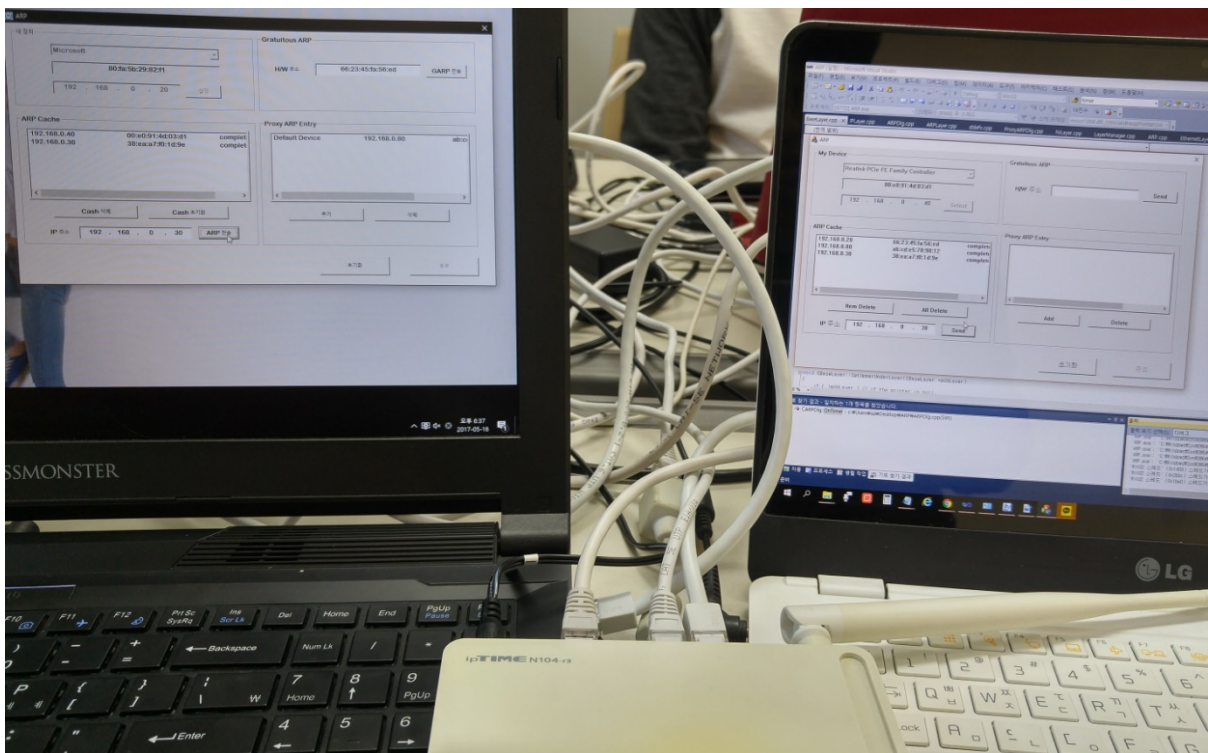
-ARP Message를 전송하여 MAC주소를 받아온다



-ARP Message를 전송하여 proxy entry의 MAC주소를 받아온다



-GARP Message를 전송하여 MAC주소를 받아온다



-공유기를 사용하여 3대의 컴퓨터 테스트

## 5. 느낀 점

- 뿌듯한 과제였다.
- 저번의 과제와 비슷한 덕분에 저번보다 할만했다.
- 통신에 대하여 점점 넓은 시야가 생기는 것 같아서 아주 좋은 실습이었다고 생각합니다.
- 우리가 쓰는 통신 시스템을 좀 더 알게 되었다.
- 다음에 할 라우터 과제가 기대된다.