《网络技术与应用》课程大作业

实验一：IP 数据报捕获与分析

姓名：王科

学号：1310583

专业：计算机科学与技术

完成日期：2015/11/12

一、 实验要求介绍

本实验要求利用 WinPcap 提供的功能获取网络接口设备列表和各接口的详细信息，同时可以对任意一块网络接口卡进行 IP 数据报进行校验和验证。

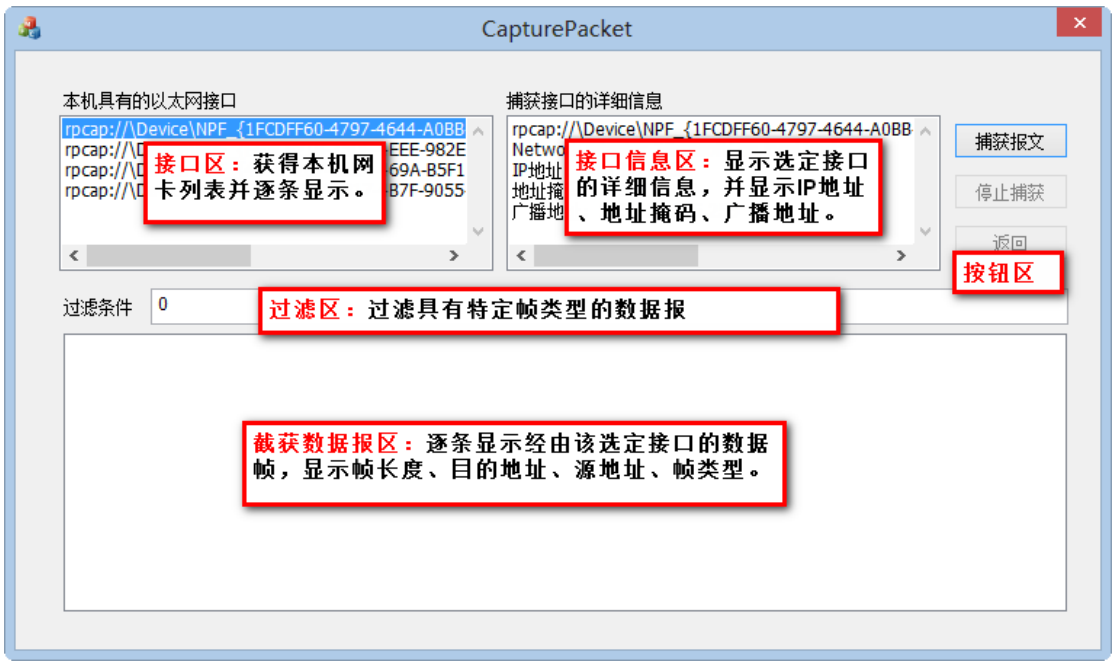
- 实验环境：以太网环境
- 实验方法：利用 WinPcap 或 LibPcap 编写捕获与分析程序
- 实验目的：
  - 学习网络数据包捕获方法；
  - 初步掌握网络监听与分析技术的实现过程；
  - 利用 WinPcap 编写数据报捕获程序，要求能够解析以太网帧的源地址、目的地址和类型/长度域。

二、 实验编译运行环境

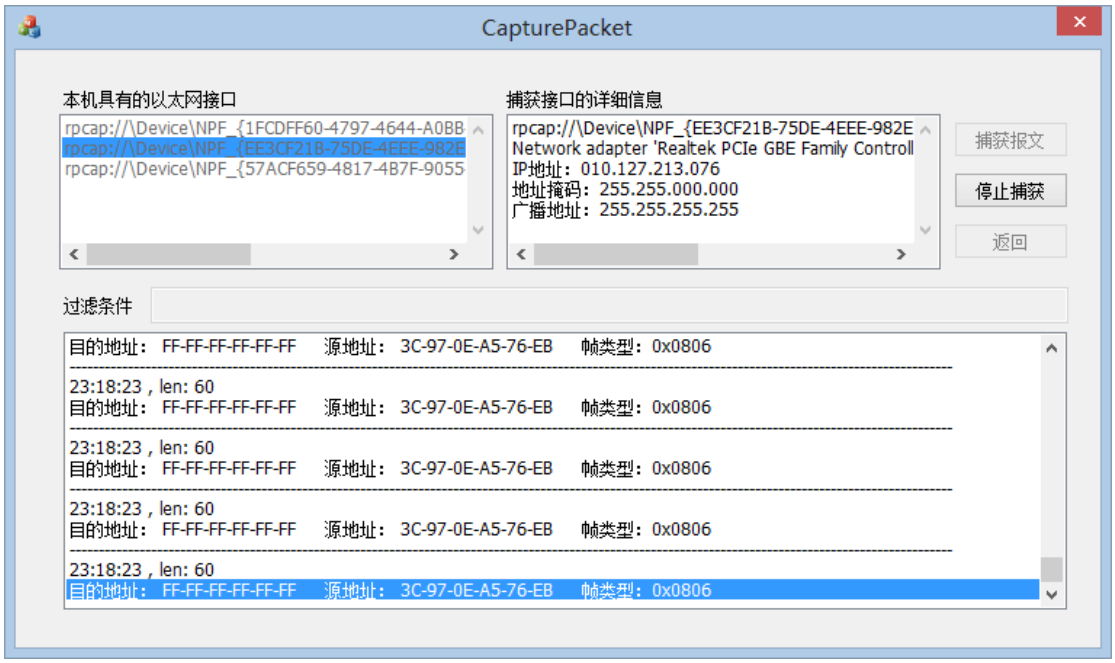
本程序编译环境是：Visual Studio 2012；系统环境是：Windows 8（64 位）；

三、 编写 IP 数据报捕获与分析软件运行效果

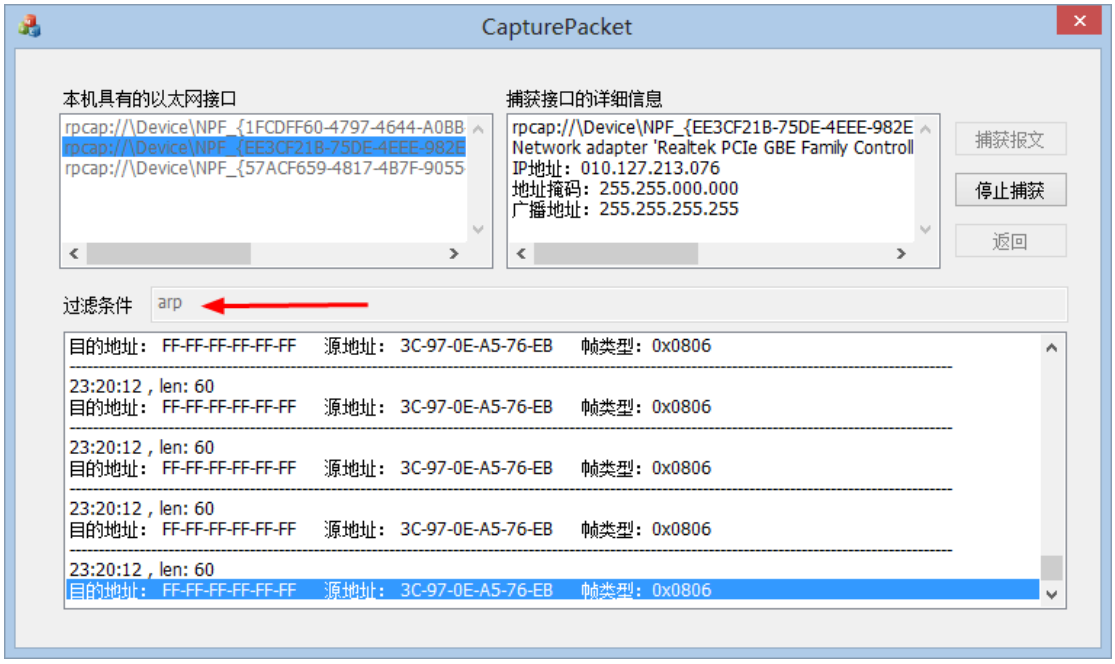
程序最终界面如下：



程序正常捕获数据报界面如下：



使用过滤条件时如下（这里以只筛选出 ARP 报文为例）：



下面就程序各部分详细功能简单介绍：

- **接口区：**

此部分显示当前运行机器上具有的网卡接口列表，并且可以已通过单击选中接口，并且在右侧的“捕获接口的详细信息”区域显示当前选中的接口的详细信息。当点击“捕获报文”按钮进行捕获之后，此部分变成失效状态，即不可更改。

- **接口信息区：**

此部分会显示在“本机具有的以太网接口”区域选中的接口的详细信息，会显示网卡接口的描述信息、IP 地址信息、地址掩码、广播地址。并且会随着接口区选定的接口的改变而改变。

- **按钮区：**

此部分具有“捕获报文”、“停止捕获”、“返回”三个按钮。负责捕获数据报和停止返回的功能。

其中当点击“捕获报文”后就不可以更改网卡接口，并且会将经由该网卡接口的所有数据报逐条捕获显示特定信息在屏幕上。

当点击“停止捕获”按钮后，会停止捕获选定网卡接口的数据报。

当点击“返回”按钮后，会清空在“截获数据报”区域的数据报信息，并且此时可以更改接口列表。

- **过滤区：**

此部分可以输入一个大于等于 0 的整数，默认是 0 的话就会显示所有捕获选定网卡接口的数据报，如果是其他整数的话就会只显示该帧类型的数据报截获信息。

- **截获数据包区：**

此部分会逐条显示捕获数据报，会显示时间、帧长度、目的地址、源地址、帧类型。

四、 程序实现步骤及代码

程序主要使用了 winpcap 编写捕获与分析程序，首先使用 pcap_findalldevs_ex（）函数获取本机的接口设备，然后使用当用户选定一个接口时首先使用 AfxBeginThread()创建一个工作者线程，然后使用自定义消息进行捕获数据报线程与显示对话框数据报线程通信，在工作者线程 UINT Capturer(LPVOID pParm)里面，使用 pcap_open() 函数打开网络接口，然后循环调用 pcap_next_ex 来接受数据报，并且利用窗口的 PostMessage 函数发送消息； 另外在消息处理函数 OnPacket(WPARAM wParam, LPARAM lParam)里面处理捕获到的数据包，并且将其以一定格式显示显示在“截获数据包区域”Listbox 控件中。

程序用到了全局变量有：

- pcap_t* afx_adhandle; //当前打开的网络接口
- struct pcap_pkthdr *afx_header; //截获帧头部
- const u_char *afx_pkt_data; //截获帧数据

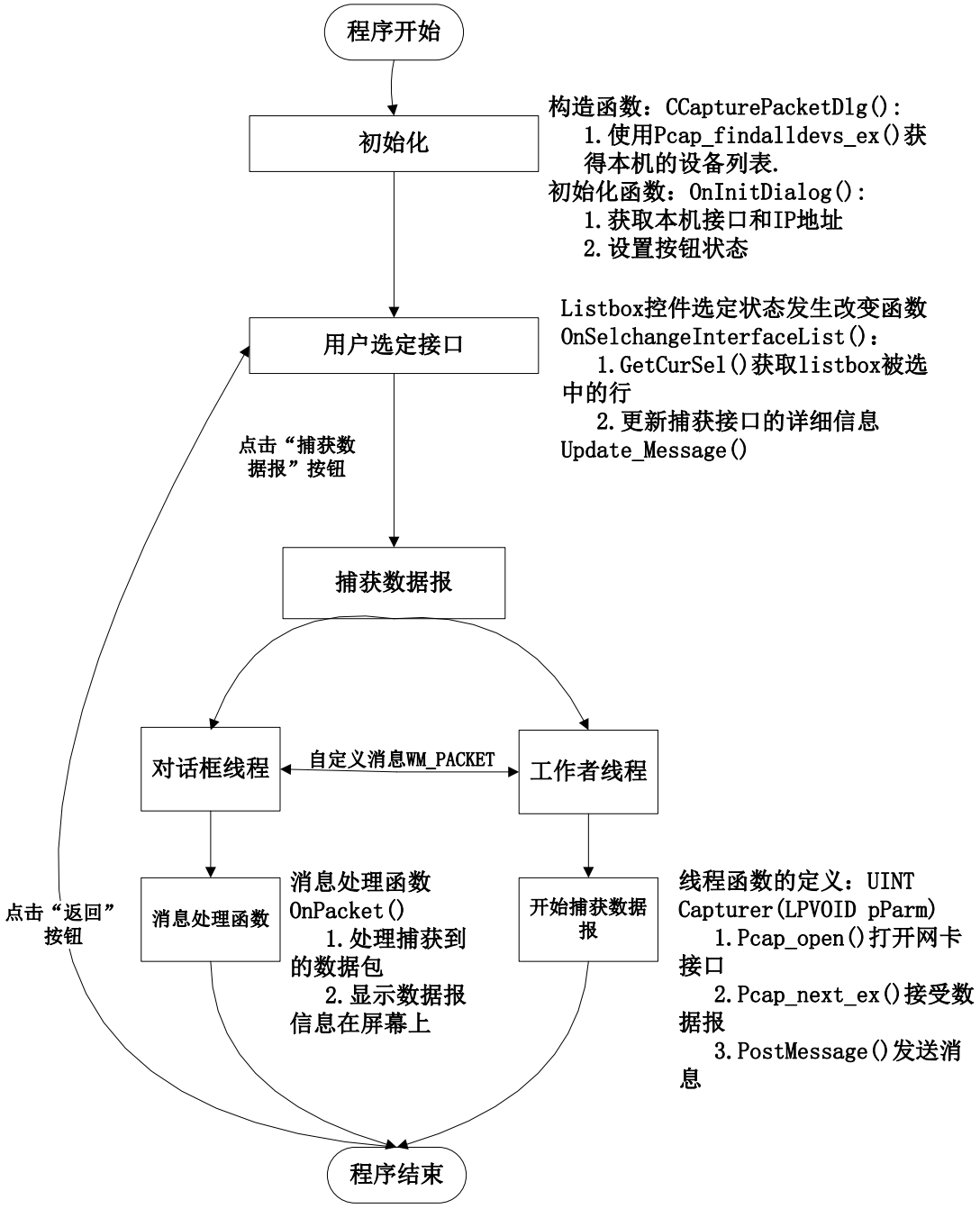
程序创建的数据结构有：

- typedef struct Data_t //包含帧首部和 IP 首部的数据包
- typedef struct FrameHeader_t //帧首部
- typedef struct IPHeader_t //IP 首部

程序创建的全局函数有：

- UINT Capturer(LPVOID pParm); //线程函数的定义

程序的逻辑结构图如下：



程序具体代码如下：

- 构造函数：CCapturePacketDlg(CWnd* pParent)

```
CCapturePacketDlg::CCapturePacketDlg(CWnd* pParent /*=NULL*/)
    : CDialogEx(CCapturePacketDlg::IDD, pParent)
    , m_state(false)
    , m_Capturer(NULL)
    , m_select(0)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);

    //获得本机的设备列表
    char  errbuf[PCAP_ERRBUF_SIZE]; //错误信息缓冲区
    if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING,     //获取本机的接口设备
            NULL,                   //无需认证
            &m_alldevs,                 //指向设备列表首部
            errbuf                  //出错信息保存缓存区
            ) == -1)
    {   MessageBox(L"获取本机设备列表失败："+CString(errbuf),MB_OK);/*错误处理*/}
    m_now=m_alldevs;
}
```

- 初始化函数：OnInitDialog()

```
    // TODO: 在此添加额外的初始化代码
    mc_Interface.SetHorizontalExtent(600);
    mc_Message.SetHorizontalExtent(600);
    m_list.SetHorizontalExtent(2000);

    //获取本机接口和IP地址
    pcap_if_t   *d;     //指向设备链表首部的指针

    for(d= m_alldevs; d != NULL; d= d->next)      //显示接口列表
    {
        mc_Interface.AddString(CString(d->name));   //利用d->name获取该网络接口设备的名字
    }

    Update_Message();//更新信息
    mc_Interface.SetCurSel(0);
    m_stop.EnableWindow(FALSE); //开始使“停止捕获”按钮失效
    m_return.EnableWindow(FALSE);   //开始使“返回”按钮失效
```

- 接口区域选定状态发生改变函数：OnSelchangeInterfaceList()

```
void CCapturePacketDlg::OnSelchangeInterfaceList()
{
    // TODO: 在此添加控件通知处理程序代码
    if(!m_state){
        int N=mc_Interface.GetCurSel();//获取listbox被选中的行的数目
        m_now=m_alldevs;
        while(N--)
        {
            m_now=m_now->next;
        }
        Update_Message();
    }
}
```

- 更新捕获接口的详细信息 Update_Message()

```
//更新捕获接口的详细信息框
void CCapturePacketDlg::Update_Message(void)
{
    //更新捕获接口的详细信息
    mc_Message.ResetContent();//清除原有框的内容
    mc_Message.AddString(CString(m_now->name));          //显示该网络接口设备的名字
    mc_Message.AddString(CString(m_now->description));  //显示该网络接口设备的描述信息

    pcap_addr_t *a;
    a=m_now->addresses;
    for(a=m_now->addresses; a!=NULL; a=a->next){
        if(a->addr->sa_family==AF_INET){  //判断该地址是否IP地址
            CString output;
            struct in_addr temp1;
            ULONG temp2;
            wchar_t pw[20];
            char *p;

            temp2=((struct sockaddr_in *)a->addr)->sin_addr.s_addr;  //显示IP地址
            memcpy(&temp1, &temp2, 4);
            p= inet_ntoa(temp1);
            SHAnsiToUnicode(p, pw, 20);
            output.Format(L"IP地址：%s",pw);
            mc_Message.AddString(output);
```

```
            temp2=((struct sockaddr_in *)a->netmask)->sin_addr.s_addr;  //显示地址掩码
            memcpy(&temp1, &temp2, 4);
            p= inet_ntoa(temp1);
            SHAnsiToUnicode(p, pw, 20);
            output.Format(L"地址掩码：%s",pw);
            mc_Message.AddString(output);

            temp2=((struct sockaddr_in *)a->broadaddr)->sin_addr.s_addr;  //显示广播地址
            memcpy(&temp1, &temp2, 4);
            p= inet_ntoa(temp1);
            SHAnsiToUnicode(p, pw, 20);
            output.Format(L"广播地址：%s",pw);
            mc_Message.AddString(output);

        }
    }

    return void();
}
```

- 点击“开始捕获”按钮 OnClickedCatch()

```
void CCapturePacketDlg::OnClickedCatch()
{
    // TODO: 在此添加控件通知处理程序代码

    m_state=true;   //将是否进入捕获状态标记打开
    mc_Interface.EnableWindow(FALSE);
    //调整按钮状态
    m_catch.EnableWindow(FALSE);
    m_stop.EnableWindow(TRUE);
    m_return.EnableWindow(FALSE);

    m_list.ResetContent();//清除原有框的内容

    //创建工作者线程
    m_Capturer=AfxBeginThread((AFX_THREADPROC)Capturer,NULL,THREAD_PRIORITY_NORMAL);
    if(m_Capturer ==NULL ){
        AfxMessageBox(L"启动捕获数据包线程失败!",MB_OK|MB_ICONERROR);
        return ;
    }
    else   /*打开选择的网卡 */
    {
        m_list.AddString(L"---------------------------------------------------------------
        m_list.AddString(L"监听"+CString(m_now->description)+L" 开始! ");
        m_list.AddString(L"---------------------------------------------------------------
    }

}
```

- 点击“停止捕获”按钮 OnClickedStop()

```
void CCapturePacketDlg::OnClickedStop()
{
    // TODO: 在此添加控件通知处理程序代码

    //调整按钮状态
    m_catch.EnableWindow(TRUE);
    m_stop.EnableWindow(FALSE);
    m_return.EnableWindow(TRUE);
    m_state=false;
}
```

- 点击“返回”按钮 OnClickedReturn()

```
void CCapturePacketDlg::OnClickedReturn()
{
    // TODO: 在此添加控件通知处理程序代码

    //调整按钮状态
    m_state=false;
    mc_Interface.EnableWindow(TRUE);
    m_catch.EnableWindow(TRUE);
    m_stop.EnableWindow(FALSE);
    m_return.EnableWindow(FALSE);
    m_list.ResetContent();//清除原有框的内容
}
```

- 数据包捕获工作者线程 UINT Capturer(LPVOID pParm)

```
//数据包捕获工作者线程
UINT Capturer(LPVOID pParm)
{
    CCapturePacketDlg* dlg = (CCapturePacketDlg*)theApp.m_pMainWnd; //获取对话框句柄

    char errbuff[1000];
    memset(errbuff,0,sizeof(errbuff));

    if ((afx_adhandle= pcap_open(dlg->m_now->name,  // 设备名称
      65536,     // WinPcap获取网络数据包的最大长度
      PCAP_OPENFLAG_PROMISCUOUS,    // 混杂模式
      1000,  // 读超时为1秒
      NULL,
      errbuff   // error buffer
      ) ) == NULL)
    {
        AfxMessageBox(L"打开该设备网卡接口失败!",MB_OK|MB_ICONERROR);
        return -1;
    }
```

```
    //利用pcap_next_ex函数捕获数据包
    /* 此处循环调用 pcap_next_ex来接受数据报*/
    int res;
    while(dlg->m_state&&(res = pcap_next_ex(afx_adhandle,&afx_header,&afx_pkt_data))>=0){
        if(res==0)  //超时情况
            continue;
        //利用窗口的PostMessage函数发送消息
        AfxGetApp()->m_pMainWnd->PostMessage(WM_PACKET,0,0);
        memset(afx_header,0,sizeof(afx_header));
        //memset(afx_pkt_data,0,sizeof(afx_pkt_data));
    }
    if(res==-1)  //获取数据包错误
    {
        AfxGetApp()->m_pMainWnd->PostMessage(WM_PACKET,1,1);
        dlg->m_state=false;
    }

    return 0;
}
```

- 消息处理函数 LRESULT CCapturePacketDlg::OnPacket(WPARAM wParam, LPARAM lParam)

```
//消息处理函数
LRESULT CCapturePacketDlg::OnPacket(WPARAM wParam, LPARAM lParam)
{
    /*……*/    //处理捕获到的数据包
    if(wParam==0&&lParam==0&&m_state==true)
    {

        //显示目的地址，源地址，帧类型

        Data_t  * IPPacket;
        ULONG       SourceIP,DestinationIP;
        IPPacket = (Data_t *) afx_pkt_data;
        SourceIP = ntohl(IPPacket->IPHeader.SrcIP);
        DestinationIP = ntohl(IPPacket->IPHeader.DstIP);


        WORD Kind = IPPacket->FrameHeader.FrameType;
        WORD Len = afx_header->caplen;

        //按帧类型筛选数据包
        UpdateData(true);
        if(m_select&&Kind!=m_select)
            return 0;
```

```
        struct in_addr temp;
        wchar_t pw1[20],pw2[20];
        char *p1,*p2;
        memcpy(&temp, &SourceIP, 4);
        p1= inet_ntoa(temp);
        SHAnsiToUnicode(p1, pw1, 20);
        memcpy(&temp, &DestinationIP, 4);
        p2= inet_ntoa(temp);
        SHAnsiToUnicode(p2, pw2, 20);

        CString output1,output2;
        CString TIME=CTime::GetCurrentTime().Format("%H:%M:%S");
        output1.Format(L"%s , len: %d",TIME,Len);
        output2.Format(L"目的地址： %s     源地址： %s     帧类型：%d",pw2, pw1 ,Kind);

        //将光标设定在最后一行
        m_list.AddString(output1);
        m_list.AddString(output2);
        int num=m_list.GetCount();
        m_list.SetCurSel(num-1);
    }
    else
    {
        m_list.AddString(L"获取数据包结束！");
    }
    m_list.AddString(L"-----------------------------------------------------------------
    return 0;
}
```

五、 实验总结

通过这次“IP 数据报捕获与分析”实验使我对于 IP 数据报和网络传输协议有了更深的理解，初步学会了网络数据包捕获方法，而且初步掌握网

络监听与分析技术的实现过程。

我通过查阅各种资料终于完成了这次的实验，感觉不仅对 IP 数据报等知识理解加深了，更是在实践中应用了理论知识，感觉得到了很大的提升。