**计算机网络编程**

**实验报告**

**班级：07111707**

**组长：1120171189 崔程远**

**成员：1120172149 吴沁璇**

**1120172153 张澈**

**1120172163 王晓媛**

**1120172736 张鉴昊**

**1120172765 曾煜瑾**

**1120173326 曾紫飞**

**北京理工大学**

**计算机学院**

**2020年4月**

**第五章 实验9 ARP程序**

**1. 实验目的**

以Winpcap为基础实现ARP地址解析和ARP高速缓冲的记录。借助Winpcap发送帧函数广播ARP请求，捕获ARP响应，并记录对应关系到缓冲。

**2. 实验内容**

程序运行屏幕输出要点：

屏幕显示当前配置的网络适配器，并要求选择捕获适配器编号

显示广播ARP请求包的内容，send函数发送广播

捕获MAC帧中识别ARP响应包，显示包内容

显示登记到缓冲中，显示缓冲区内容

**3. 实验原理**

WinPcap是一个基于Win32平台的，用于捕获网络数据包并进行分析的开源库。它提供了以下功能：捕获原始数据包；在数据包发送给某应用程序前，根据用户指定的规则过滤数据包；将原始数据包通过网络发送出去；收集并统计网络流量信息。

ARP是用于建立IP地址与MAC地址之间对应关系的协议。一台主机先向全网广播发送包含某个IP地址的ARP请求包，该IP地址对应的主机收到之后发送包含自己MAC地址的ARP响应包，本机将响应包对应的MAC地址登记到缓冲中。

**4. 实验环境**

操作系统：Windows 10

编译器：Visual Studio 2017

环境：WinPcap4.1.3 WpdPack

1. **实验步骤**

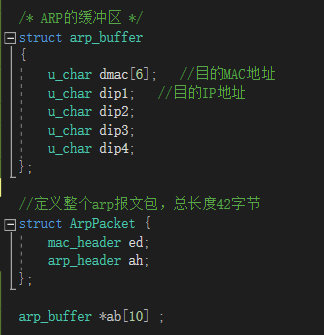
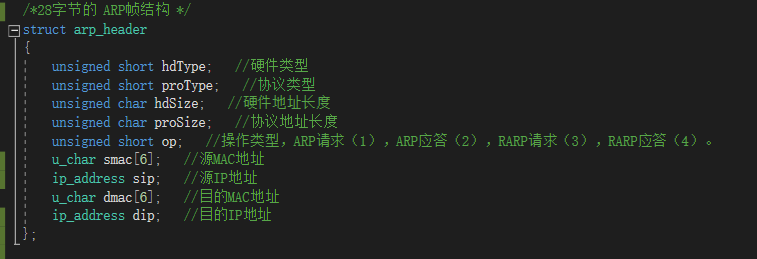
以下是C++代码和具体思路：

1. VS中环境的配置

首先下载WinPcap和WpdPack，然后在vs的项目中添加包含目录和库目录，修改预处理器，添加依赖项。然后就可以进行代码的编写。

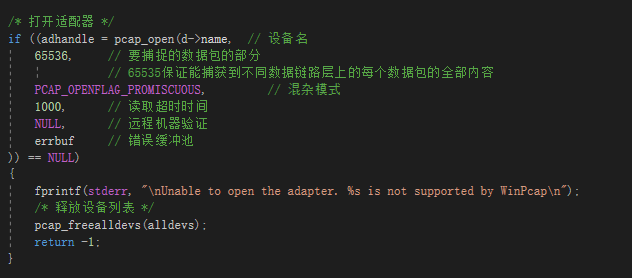
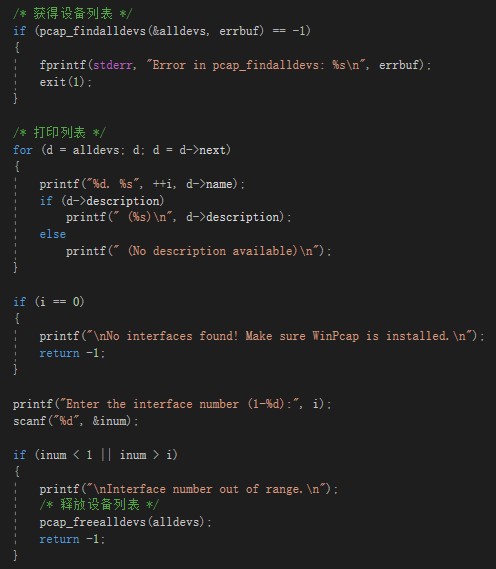
1. 定义数据报头结构

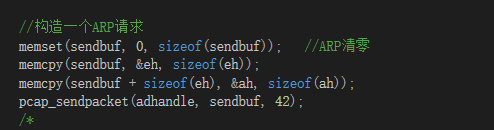
如下图所示，分别定义了以太网帧头、IP地址、ARP数据包首部、ARP缓冲区、ARP报文结构，相应的数据类型进行定义。



1. 获得设备并打印设备列表

如下图，使用了在WinPacp中文文档给出的函数，来获得设备列表和打印列表设备具体信息。还定义了在当前适配器下的send函数。

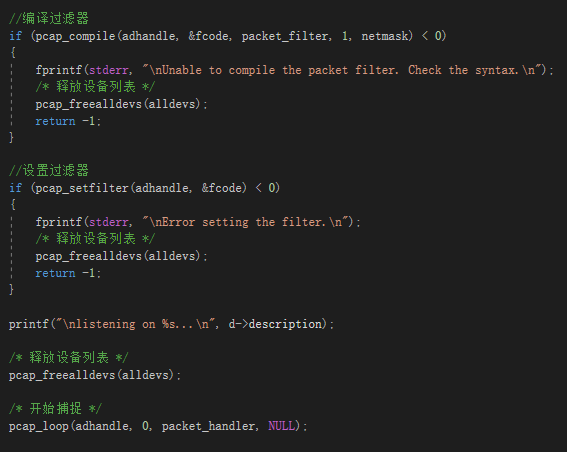




1. 设置过滤器并调用回调函数

由于捕获的报文信息太多，所以需要设置一个过滤器，用于只处理tcp和udp的报文信息，其他信息都过滤。具体过滤器的设置如下。

1



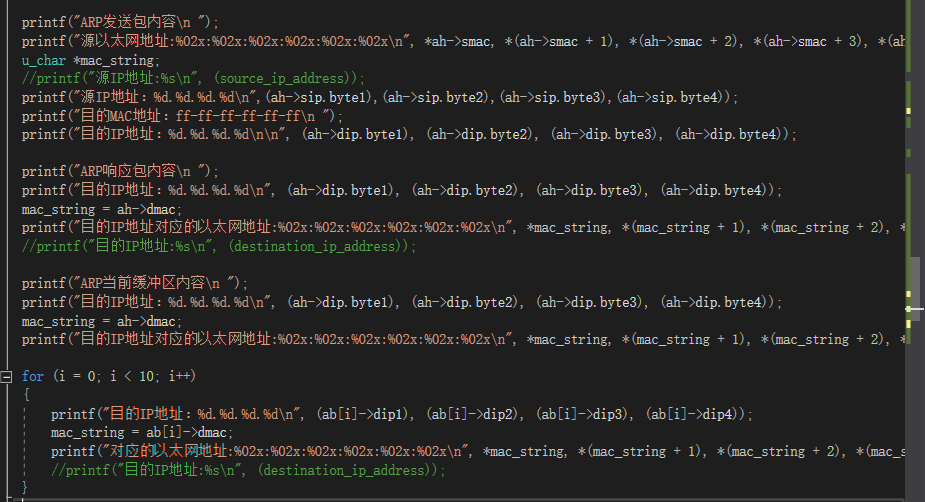
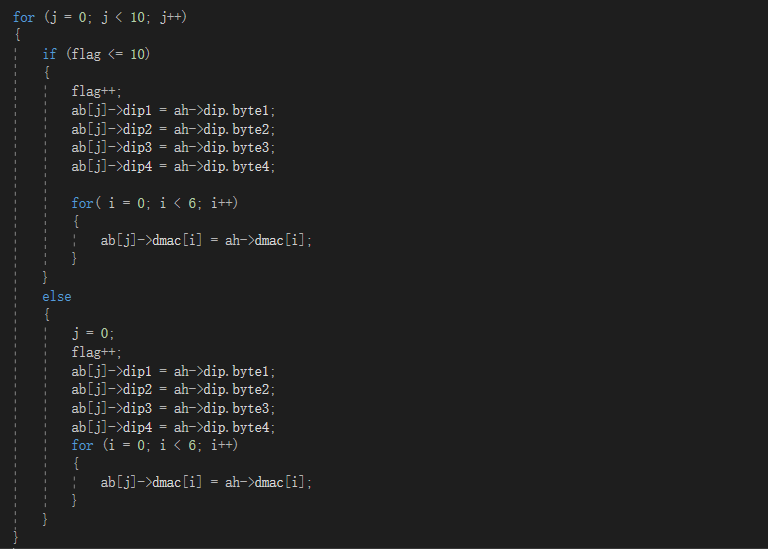
1. 获得各报文具体字段的值和打印各结构的各字段值

首先将时间戳转换为可识别的格式，然后打印数据包的时间戳和长度。ARP包的结构是有14个字节的以太网帧头和28字节的ARP首部加上18字节的数据和4字节的CRC。其中，28字节的ARP帧和14字节以太网帧头单独定义。

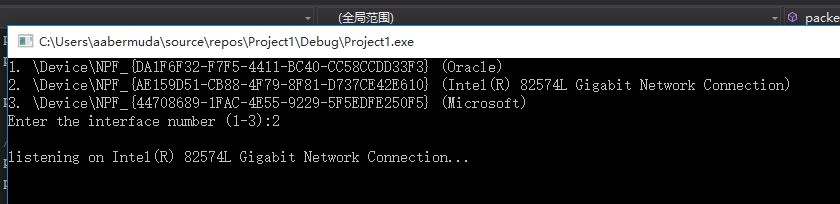
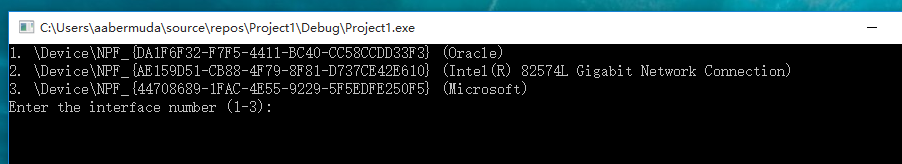
将捕获报文的内容指针所指地址为以太网帧头的地址，再加上14个字节就可得到ARP首部的地址。也就是回调函数的\*pkt\_data量为以太网帧头地址，加上14得到ARP帧的地址。然后进行对应的字段根据字段长度取相应的值。

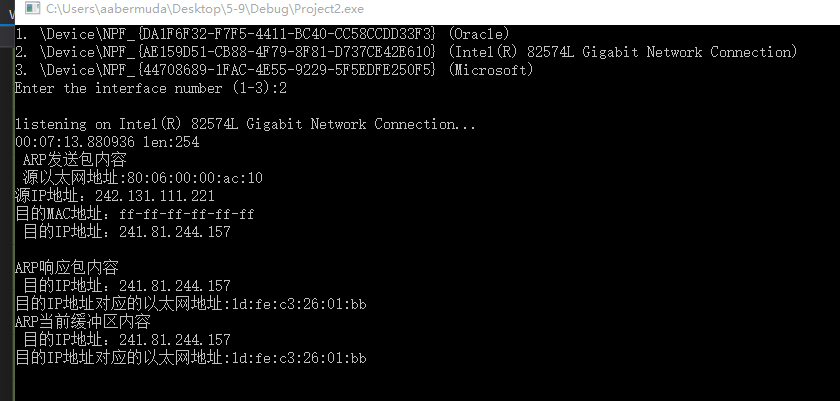
最后是打印ARP请求包、ARP相应包和缓冲区的具体值。由于在代码编译时，程序必须包含的头文件“pcap.h”和string、ostream之间存在问题导致编译失败，大概是里面有一些重复定义，然而这个问题没能得到解决，所以就只能用printf对结构体的每一项一个个打印出来。然后根据各个字段值的数据结构依次输出相应值即可。





运行结果截图：





实验结果如上图所示，选定了2号适配器后，依次打印了ARP请求包、ARP响应包、缓冲区内容的IP地址和对应的MAC地址，所得的结果如上图所示。

**6. 实验总结**

本次实验和第四章的实验差不多，都是利用WinPcap进行包的捕获和分析。本次实验更侧重于对ARP协议的理解和分析。首先得知道输出的请求包和响应包分别是什么，其中存在着什么样的过程。然后用合适的数据结构将ARP包描述出来，捕获的时候将源ip、mac地址和目的ip地址都找到，然后找到与目的ip地址相对于的目的mac地址。

本次实验我也遇到了很多的问题，一开始不知道怎么表示ARP包的各个字段和如何表示缓冲区的存储信息。然后经过不断的试错和调试，一点点的完成了实验。通过本实验，我更加熟悉了ARP包的结构，对其中每个字段具体值都有了更深的理解，还了解到如何利用ARP进行欺骗和监听等有趣的知识。