南开大学
Nankai University

南　开　大　学

网络空间安全学院

网络技术与应用课程报告

通过编程获取 IP 地址与 MAC 地址的对应关系

学号：2011897

姓名：任薏霖

年级：2020 级

专业：物联网工程

2022 年 10 月 27 日

第 1 节 实验内容说明

1. 实验题目

通过编程获取 IP 地址与 MAC 地址的对应关系

2. 实验内容说明

(1) 通过编程获取 IP 地址与 MAC 地址的对应关系实验，要求如下：

(2) 在 IP 数据报捕获与分析编程实验的基础上，学习 WinPcap 的数据包发送方法。通过 Npcap 编程，获取 IP 地址与 MAC 地址的映射关系；

(3) 程序要具有输入 IP 地址，显示输入 IP 地址与获取的 MAC 地址对应关系界面。界面可以是命令行界面，也可以是图形界面，但应以简单明了的方式在屏幕上显示。编写的程序应结构清晰，具有较好的可读性。

第 2 节 实验准备

（一）配置环境

WinPcap 是一个数据包捕获体系框架，主要功能是进行数据包捕获和网络分析。包括了内核基本的包过滤、低层次的库(packet.lib)、高级别系统无关的函数库(wpcap.dll)。目前，WinPcap 停止维护，故采用 NPcap 进行实验。运行相关程序，需要安装驱动程序。

在 Visual Studio2019 中配置环境过程如下：

1. 下载并安装 WinPcap、Npcap 运行库
2. 下载 WinPcap、Npcap 开发包
3. 创建并设置项目：

(1) 打开项目属性，添加 WPCAP 和 HAVE_REMOTE 两个宏定义。

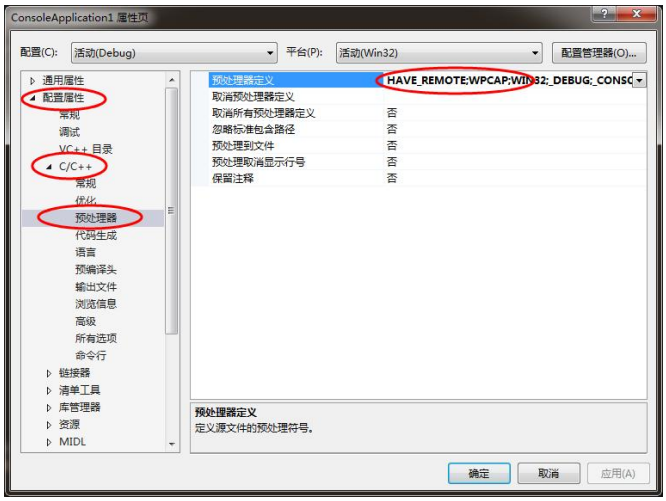


图 1 添加宏定义

(2) 添加 wpcap.lib 和 ws2_32.lib 两个库。

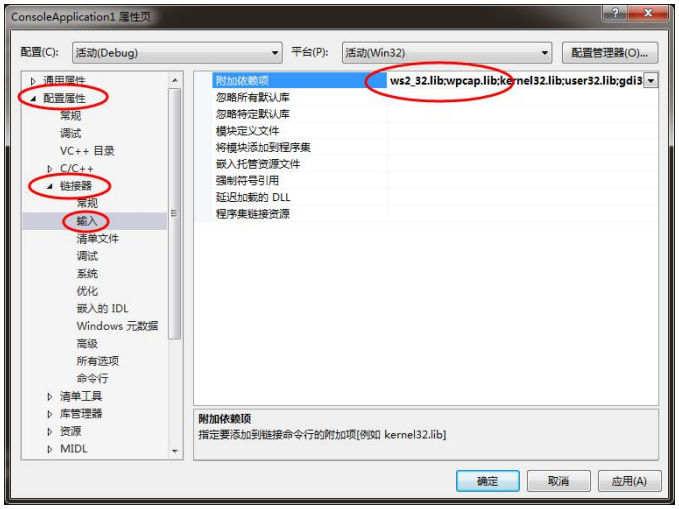


图 2 添加库

(3) 添加包含路径和库路径：

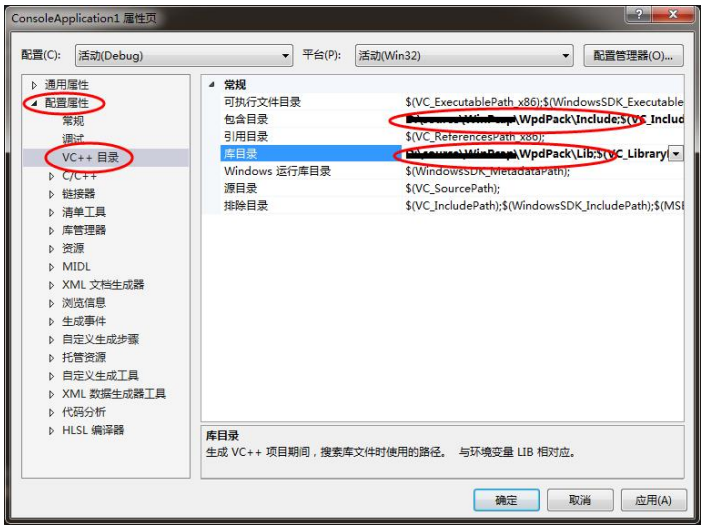


图 3 添加包含路径和库路径

（二）ARP 协议

ARP 协议是“Address Resolution Protocol”（地址解析协议）的缩写。其作用是在以太网环境中，数据的传输所依赖的是 MAC 地址而非 IP 地址，而将已知 IP 地址转换为 MAC 地址的工作是由 ARP 协议来完成的。

在局域网中，网络中实际传输的是“帧”，帧里面是有目标主机的 MAC 地址的。在以太网中，一个主机和另一个主机进行直接通信，必须要知道目标主机的 MAC 地址。而目标 MAC 地址是通过地址解析协议获得的。所谓“地址解析”就是主机在发送帧前将目标 IP 地址转换成目标 MAC 地址的过程。ARP 协议的基本功能就是通过目标设备的 IP 地址，查询目标设备的 MAC 地址，以保证通信的顺利进行。

第 3 节 实验过程

一、实验思路

本次实验需要获取主机网卡中对应 IP 的 MAC 地址，可以利用 ARP 请求方法，过程如下：

1. 获取网络接口卡列表，选择需要捕获 MAC 地址的网卡 A（或选择对应的 IP）；
2. 伪造 ARP 请求报文 S，内容要求如下：
   (1) ARP 请求
   (2) 广播
   (3) 伪造源 MAC 地址和源 IP 地址
   (4) 目的 IP 地址为网卡 A 的 IP 地址
3. 用网卡 A 发送报文 S
4. 对网卡 A 进行流量监听，筛选其中的 ARP 报文（类型为 0x806），捕获网卡 A 的 ARP 响应报文，在响应报文的帧首部源 MAC 地址部分可以看到发送该 ARP 响应的网卡对应的 MAC 地址

二、关键代码分析

（一）定义报文格式

```
/*报文格式定义 */
struct ethernet_header{
    uint8_t mac_dhost[6];      /*目的MAC地址*/
    uint8_t mac_shost[6];      /*源MAC地址*/
    uint16_t frame_type;         /*帧类型*/
};
typedef struct FrameHeader_t {//帧首部
    BYTE DesMAC[6];//目的地址
    BYTE SrcMAC[6];//源地址
    WORD FrameType;//帧类型
}FrameHeader_t;
typedef struct ARPFrame_t {//IP首部
    FrameHeader_t FrameHeader;
    WORD HardwareType;//硬件类型
    WORD ProtocolType;//协议类型
    BYTE HLen;//硬件地址长度
    BYTE PLen;//协议地址长度
    WORD Operation;//操作类型
```

```
    BYTE SendHa[6];//发送方MAC地址
    DWORD SendIP;//发送方IP地址
    BYTE RecvHa[6];//接收方MAC地址
    DWORD RecvIP;//接收方IP地址
}ARPFrame_t;

/*ip地址格式*/
typedef uint32_t in_addr_t;
```

（二）定义报文内容

```
//报文内容
    //将APRFrame.FrameHeader.DesMAC设置为广播地址
    for (int i = 0; i < 6; i++)
        ARPFrame.FrameHeader.DesMAC[i] = 0xff;//表示广播
    //将APRFrame.FrameHeader.SrcMAC设置为本机网卡的MAC地址
    for (int i = 0; i < 6; i++)
        ARPFrame.FrameHeader.SrcMAC[i] = 0x0f;

    ARPFrame.FrameHeader.FrameType = htons(0x806);//帧类型为ARP
    ARPFrame.HardwareType = htons(0x0001);//硬件类型为以太网
    ARPFrame.ProtocolType = htons(0x0800);//协议类型为IP
    ARPFrame.HLen = 6;//硬件地址长度为6
    ARPFrame.PLen = 4;//协议地址长为4
    ARPFrame.Operation = htons(0x0001);//操作为ARP请求
    SerIP = ARPFrame.SendIP = htonl(0x00000000);//设置为任意IP地址
    //本机网卡的MAC地址
    for (int i = 0; i < 6; i++)
        ARPFrame.SendHa[i] = 0x0f;
    //本机网卡上绑定的IP地址
    ARPFrame.SendIP = htonl(0x00000000);
    //设置为0
    for (int i = 0; i < 6; i++)
        ARPFrame.RecvHa[i] = 0;//表示目的地址未知
```

（三）获取设备列表

用 pcap_findalldevs()函数获取网络接口设备列表，将设备列表存储为 alldevs 中，遍历打印列表信息。

```
//获得网卡列表
pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf);
```

（四）打印网卡信息和对应 IP

由用户输入设备信息，void findAllDevice(string& str,string IP[])将该网卡设备

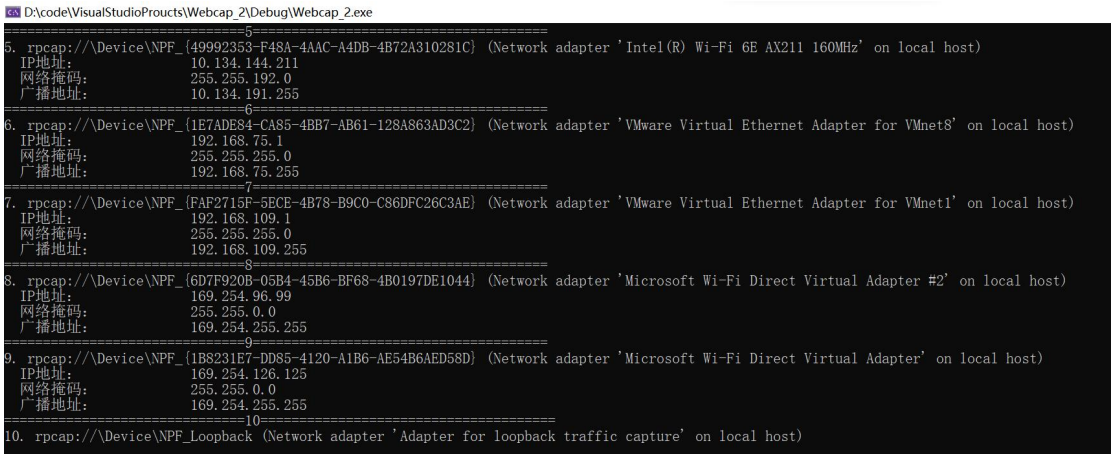
的名称、IP 地址储返回，设备名称 str 用于后面 pcap_open()函数打开该网卡，而 IP 将用于构造 ARP 请求分组

```
    //利用pcap_findalldevs_ex函数获取本机网络接口卡以及网卡绑定的IP地址
    if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf) == -1)//获得网卡列表
    {
        printf("获得网卡列表错误\n");
        exit(1);
    }

    for (d = alldevs; d; d = d->next)
    {
        cout << "==============================="<<++i<<"==================================="<<endl;
        printf("%d. %s", i, d->name);
        if (d->description) {
            printf(" (%s)\n", d->description);

        }
        else
            printf(" (No description available)\n");
        a = d->addresses;
    A:   if (a != NULL) //相对第一次试验，增加输出IP地址，掩码，广播地址的代码
    {
        if (a->addr->sa_family == AF_INET)
        {
            cout << "  IP地址：\t\t" << inet_ntoa(((struct sockaddr_in*)(a->addr))->sin_addr) << endl;
            cout << "  网络掩码：\t\t" << inet_ntoa(((struct sockaddr_in*)(a->netmask))->sin_addr) << endl;
            cout << "  广播地址：\t\t" << inet_ntoa(((struct sockaddr_in*)(a->broadaddr))->sin_addr) << endl;
        }
        a = a->next;
        goto A;
    }
    }
```

打印示例：



（五）打开网络接口

```
    // 找到要选择的网卡结构
    for (d = alldevs, i = 0; i < netcard_id - 1; d = d->next, i++);


    if ((adhandle = pcap_open(d->name, 65536, PCAP_OPENFLAG_PROMISCUOUS, 1000, NULL,
errbuf)) == NULL){
        pcap_freealldevs(alldevs);
        exit(1);
    }
```

（六）发送消息

用 adhandle 网卡发送 ARPFrame 中的内容，报文长度为 sizeof(ARPFrame_t)，如果发送成功，返回 0；

```
pcap_sendpacket(adhandle, (u_char*)&ARPFrame, sizeof(ARPFrame_t))
```

（七）捕获流量

```
    if (pcap_sendpacket(adhandle, (u_char*)&ARPFrame, sizeof(ARPFrame_t)) != 0)
    {
        pcap_freealldevs(alldevs);
        throw - 7;
    }
    else
    {
        inum = 0;
    B:   int jdg_catch_re_arp_p = pcap_next_ex(adhandle, &adhandleheader,
&adhandledata);
        IPPacket = (ARPFrame_t*)adhandledata;
        if (SerIP == IPPacket->SendIP)
            if (ReIP == IPPacket->RecvIP)
            {
```

```
                goto B;
            }
        //根据网卡号寻找 IP 地址，并输出 IP 地址与 MAC 地址映射关系
        if (SerIP == IPPacket->RecvIP)
            if (ReIP == IPPacket->SendIP)
            {
                cout << "IP 地址与 MAC 地址的对应关系如下：" << endl << "IP：";
ip_protocal_addr(IPPacket->SendIP);
                cout << "MAC："; mac_addr(IPPacket->SendHa);
                cout << endl;
            }
            else
                goto B;
        else
            goto B;
    }
```

（八）获取网卡的 MAC 地址

采用的方法为：封装 ARP 请求时使用本机网卡的 IP 和 MAC 地址

1. 判断是否为本机 IP 地址或远程 IP 地址
2. 利用上述方法请求本地网卡的 MAC 地址，将本机 IP 和 MAC 填入报文
3. 重新发送 ARP 请求

```
//根据网卡号寻找IP地址，并输出IP地址与MAC地址映射关系
if (SerIP == IPPacket->RecvIP)
     if (ReIP == IPPacket->SendIP){
            cout << "IP地址与MAC地址的对应关系如下：" << endl << "IP：";
            ip_protocal_addr(IPPacket->SendIP);
            cout << "MAC："; mac_addr(IPPacket->SendHa);
            cout << endl;
     }
//不是本机：寻找远程ip对应的mac地址则重新发送ARP请求
if (!ifIP) {
     SerIP = ARPFrame.SendIP = IPPacket->SendIP;
     for (i = 0; i < 6; i++){
            ARPFrame.SendHa[i] = ARPFrame.FrameHeader.SrcMAC[i] = IPPacket->SendHa[i];
     }
}
```

三、实验结果

由于 ARP 协议广播分组是对相同网段内的主机进行的，不能跨网段直接获取远程主机

的 MAC 地址，因此为实现获取远程主机 Ip 地址与 Mac 的对应关系，采用两个主机连接同一热点进行实验验证；

其中本机 ip 地址为 192.168.43.31；

远程主机 ip 地址为 192.168.43.34；

获取网卡列表

```
=========    获取IP地址与MAC地址映射关系    =========
============================1===================================
1. rpcap://\Device\NPF_{8FD082C6-D341-477D-AF17-A2ADFA40BA0E} (Network adapter 'WAN Miniport (Network Monitor)' on local host)
============================2===================================
2. rpcap://\Device\NPF_{4F554C7F-37FB-4403-95BE-BD2616AB0266} (Network adapter 'WAN Miniport (IPv6)' on local host)
============================3===================================
3. rpcap://\Device\NPF_{DD545DE9-B109-4575-9216-F90C284DDC25} (Network adapter 'WAN Miniport (IP)' on local host)
============================4===================================
4. rpcap://\Device\NPF_{5CEA0E58-6AA1-4A32-B73D-38D9BA73B035} (Network adapter 'Bluetooth Device (Personal Area Network)' on local host)
  IP地址:               169.254.100.223
  网络掩码:             255.255.0.0
  广播地址:             169.254.255.255
============================5===================================
5. rpcap://\Device\NPF_{49992353-F48A-4AAC-A4DB-4B72A310281C} (Network adapter 'Intel(R) Wi-Fi 6E AX211 160MHz' on local host)
  IP地址:               192.168.43.13
  网络掩码:             255.255.255.0
  广播地址:             192.168.43.255
============================6===================================
6. rpcap://\Device\NPF_{1E7ADE84-CA85-4BB7-AB61-128A863AD3C2} (Network adapter 'VMware Virtual Ethernet Adapter for VMnet8' on local host)
  IP地址:               192.168.75.1
  网络掩码:             255.255.255.0
  广播地址:             192.168.75.255
============================7===================================
7. rpcap://\Device\NPF_{FAF2715F-5ECE-4B78-B9C0-C86DFC26C3AE} (Network adapter 'VMware Virtual Ethernet Adapter for VMnet1' on local host)
  IP地址:               192.168.109.1
  网络掩码:             255.255.255.0
  广播地址:             192.168.109.255
============================8===================================
8. rpcap://\Device\NPF_{6D7F920B-05B4-45B6-BF68-4B0197DE1044} (Network adapter 'Microsoft Wi-Fi Direct Virtual Adapter #2' on local host)
  IP地址:               169.254.96.99
  网络掩码:             255.255.0.0
  广播地址:             169.254.255.255
============================9===================================
9. rpcap://\Device\NPF_{1B8231E7-DD85-4120-A1B6-AE54B6AED58D} (Network adapter 'Microsoft Wi-Fi Direct Virtual Adapter' on local host)
  IP地址:               169.254.126.125
  网络掩码:             255.255.0.0
  广播地址:             169.254.255.255
============================10==================================
10. rpcap://\Device\NPF_Loopback (Network adapter 'Adapter for loopback traffic capture' on local host)

==========输入要选择打开的网卡号 (1-10)==========
5
```

查询本机 IP 地址与 MAC 地址的对应关系

```
==========输入要选择打开的网卡号 (1-10)==========
5

接入对应端口 Network adapter 'Intel(R) Wi-Fi 6E AX211 160MHz' on local host...
IP地址与MAC地址的对应关系如下:
IP: 192 - 168 - 43 - 13
MAC: 08 - 8e - 90 - 08 - 3b - 44
```

查询远程 IP 地址与 MAC 地址的对应关系

```
=====================请输入目的IP地址====================
192.168.43.34
=================请输入是否为本机,是:1,否:0=============
0
IP地址与MAC地址的对应关系如下:
IP: 192 - 168 - 43 - 34
MAC: 48 - 89 - e7 - b8 - ee - 85
```

与通过 ipconfig/all 命令获取的 ip 地址与 MAC 地址的对应关系一致