实验3：通过编程获取IP地址与MAC地址的对应关系

## 网络技术与应用实验3实验报告

学院：网络空间安全学院

专业：密码科学与技术

学号：2112155

姓名：梁婧涵

### 实验要求

通过编程获取IP地址与MAC地址的对应关系实验，要求如下：

（1）在IP数据报捕获与分析编程实验的基础上，学习NPcap的数据包发送方法。

（2）通过NPcap编程，获取IP地址与MAC地址的映射关系。

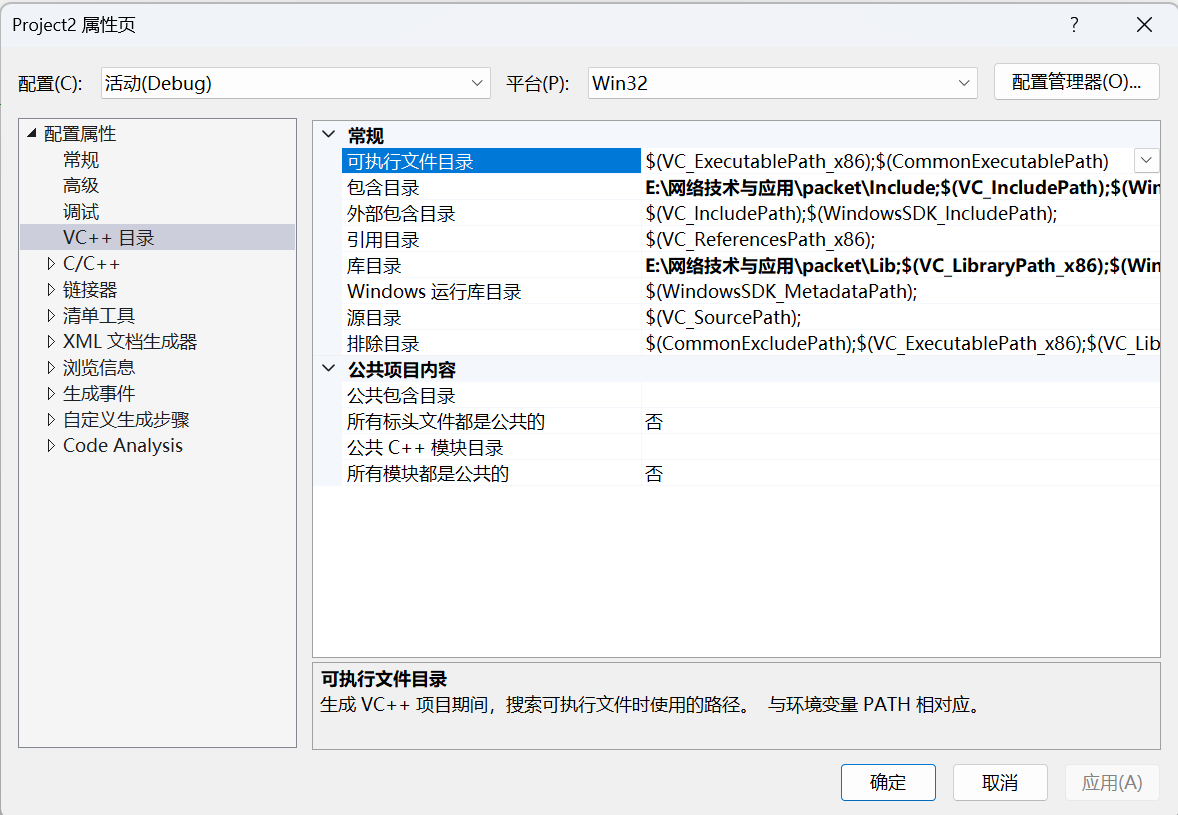
（3）程序要具有输入IP地址，显示输入IP地址与获取的MAC地址对应关系界面。界面可以是命令行界面，也可以是图形界面，但应以简单明了的方式在屏幕上显示。

（4）编写的程序应结构清晰，具有较好的可读性

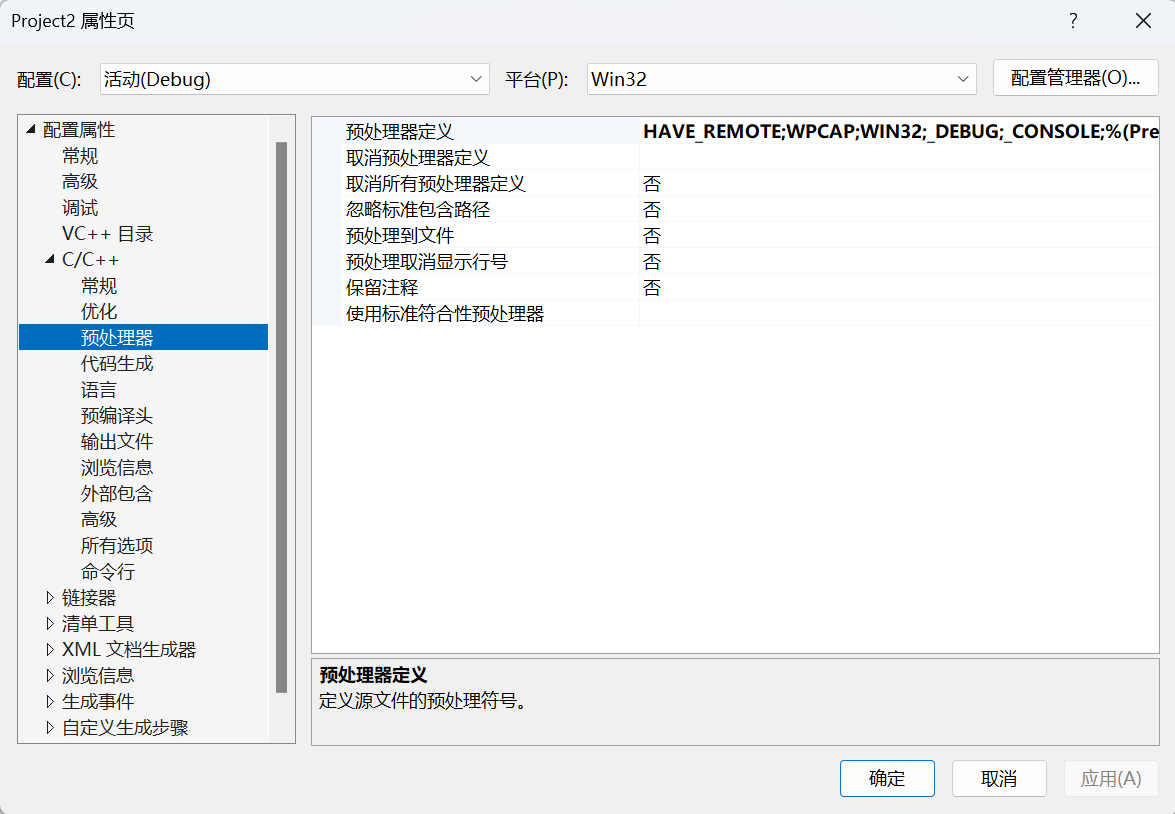
### 实验准备

#### 配置属性

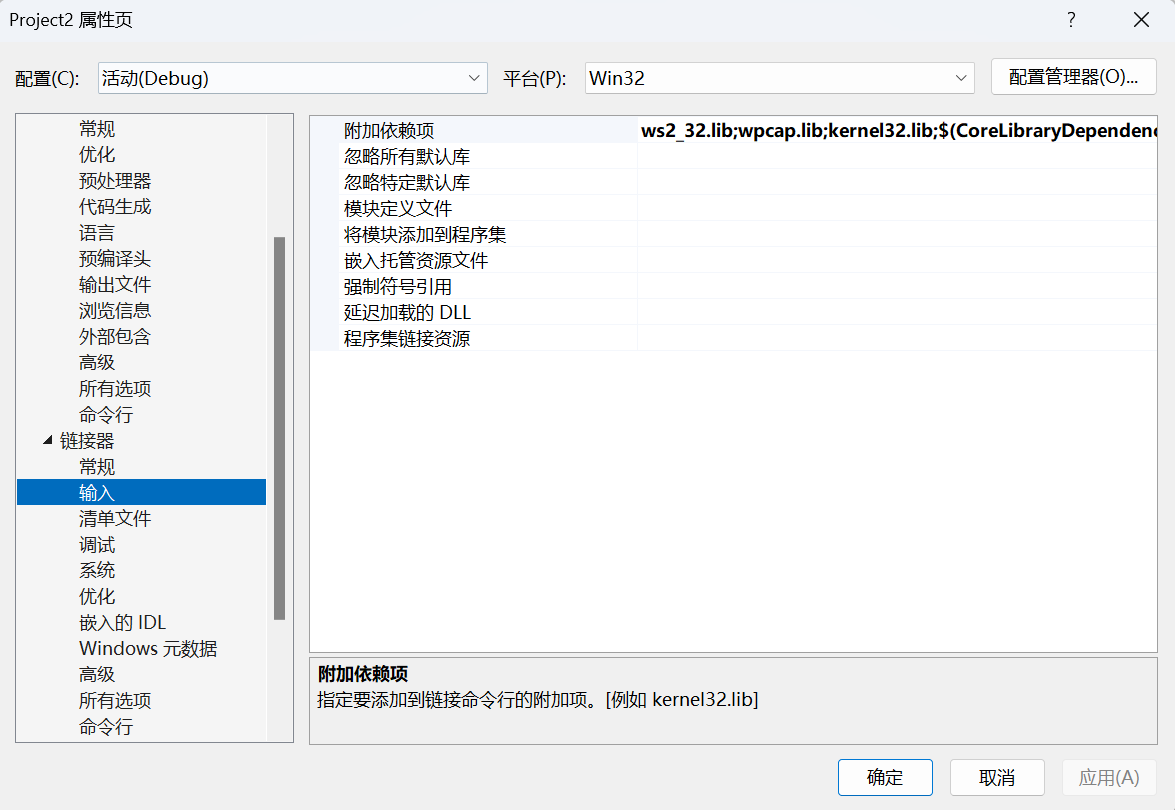
添加Include和Lib目录：



添加预处理器定义：



添加附加依赖项：



#### 2、**WinPcap 的数据包发送方法**

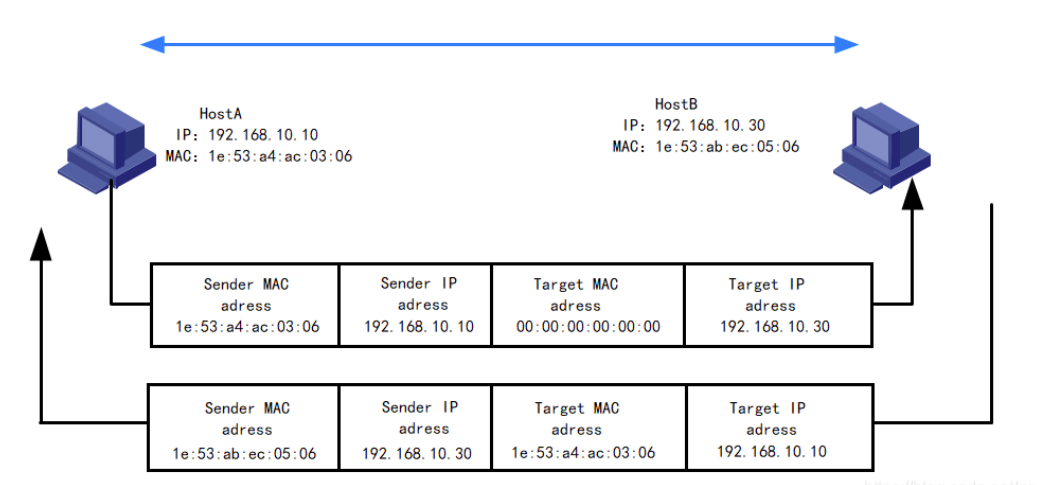
以太网发送数据包使用 WinPcap 提供的 pcap\_sendpacket(pcap\_t \*p, u\_char buf, int size) 函数，其中的参数：

* p 指定函数通过哪块接口网卡发送数据包。
* buf 指向需要发送的数据包。其中不包含以太网帧的 CRC 校验和字段。
* size 指定发送数据包的大小

#### 3、**ARP 的基本思想**

ARP协议是地址解析协议（Address Resolution Protocol）是通过解析IP地址得到MAC地址的，是一个在网络协议包中极其重要的网络传输协议。

在网络访问层中，同一局域网中的一台主机要和另一台主机进行通信，需要通过MAC地址进行定位，然后才能进行数据包的发送。而在网络层和传输层中，计算机之间是通过IP地址定位目标主机，对应的数据报文只包含目标主机的IP地址，而没有MAC地址。因此，在发送之前需要根据IP地址获取MAC地址，然后才能将数据包发送到正确的目标主机，而这个获取过程是通过ARP协议完成的。



假设主机A和B在同一个网段，主机A要向主机B发送信息，具体的地址解析过程如下：

(1) 主机A首先查看自己的ARP表，如果ARP表中含有主机B对应的ARP表项，则主机A直接利用ARP表中的MAC地址，对IP数据包进行帧封装，并将数据包发送给主机B。

(2) 如果主机A在ARP表中找不到对应的MAC地址，则将缓存该数据报文，然后以广播方式发送一个ARP请求报文。ARP请求报文中的发送端IP地址和发送端MAC地址为主机A的IP地址和MAC地址，目标IP地址和目标MAC地址为主机B的IP地址和全0的MAC地址。由于ARP请求报文以广播方式发送，该网段上的所有主机都可以接收到该请求，但只有被请求的主机（即主机B）会对该请求进行处理。

(3) 主机B比较自己的IP地址和ARP请求报文中的目标IP地址，当两者相同时进行如下处理：将ARP请求报文中的发送端（即主机A）的IP地址和MAC地址存入自己的ARP表中。之后以单播方式发送ARP响应报文给主机A，其中包含了自己的MAC地址。

(4) 主机A收到ARP响应报文后，将主机B的MAC地址加入到自己的ARP表中以用于后续报文的转发，同时将IP数据包进行封装后发送出去。

### 实验过程

#### 1、设计思路

1. 获取网络接口卡列表，选择需要捕获MAC地址的网卡
2. 构造ARP请求报文：ARP请求、广播、虚拟源MAC地址和源IP地址、目的IP地址为所选网卡的IP地址
3. 用所选网卡发送报文
4. 对所选网卡进行流量监听，筛选其中的ARP报文（类型为0x806），捕获该网卡的 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

#### 2、关键代码展示

##### 2.1 ARP报文格式（1bit对齐）

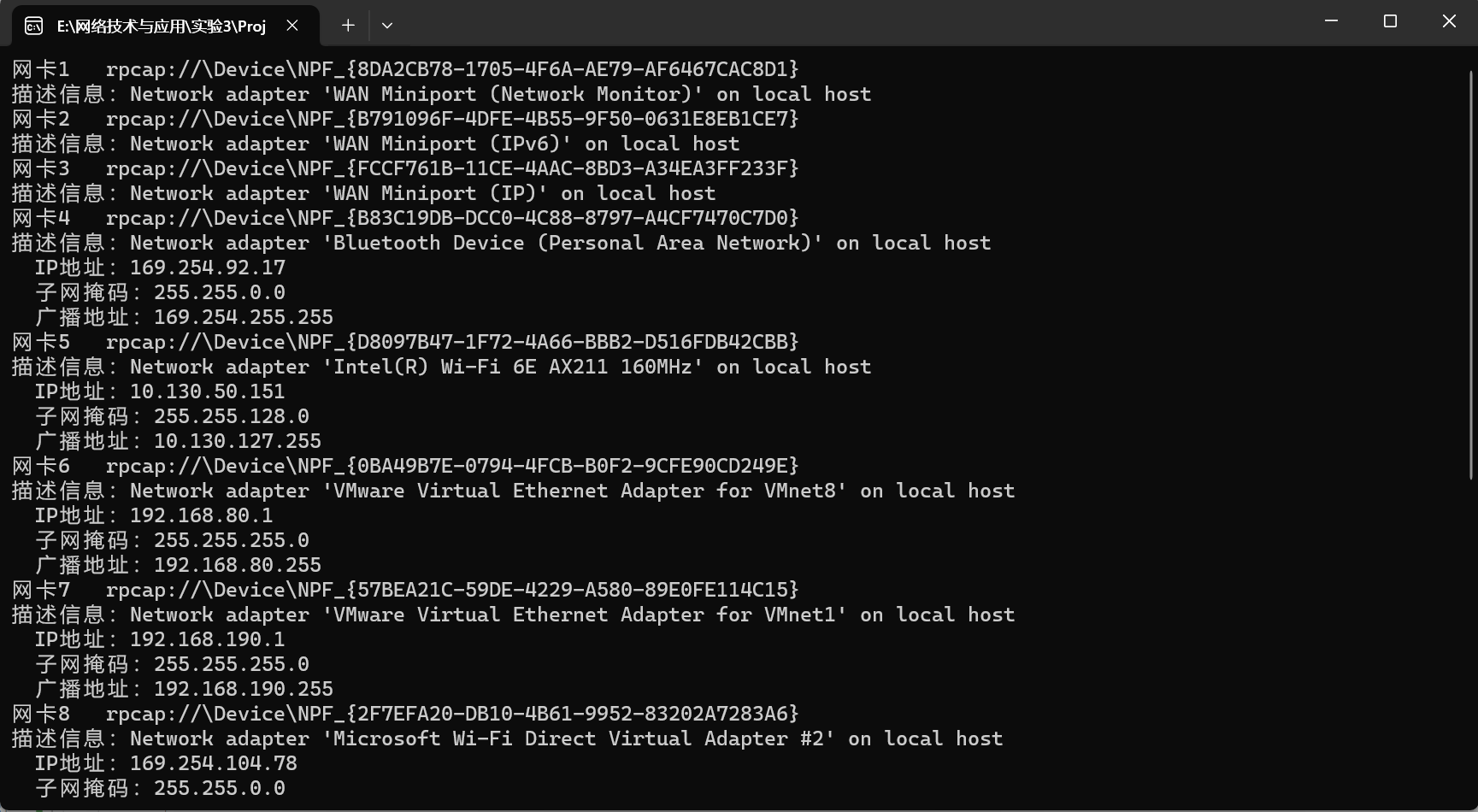
1. struct FrameHeader\_t //帧首部
2. {
3. BYTE DesMAC[6]; //目的地址
4. BYTE SrcMAC[6]; //源地址
5. WORD FrameType; //帧类型
6. };
7. struct ARPFrame\_t //ARP帧
8. {
9. FrameHeader\_t FrameHeader;
10. WORD HardwareType;
11. WORD ProtocolType;
12. BYTE HLen;
13. BYTE PLen;
14. WORD Operation;
15. BYTE SendHa[6];
16. DWORD SendIP;
17. BYTE RecvHa[6];
18. DWORD RecvIP;
19. };
20. #pragma pack() //恢复缺省对齐方式

##### 2.2 获取设备列表，打印网卡信息,IP,子网掩码，广播地址信息

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。

1. //获得本机的设备列表
2. if (pcap\_findalldevs\_ex(PCAP\_SRC\_IF\_STRING, NULL, &alldevs, errbuf) == -1)
3. {
4. cout << "获取网络接口时发生错误:" << errbuf << endl;
5. return 0;
6. }
7. //显示接口列表
8. for (ptr = alldevs; ptr != NULL; ptr = ptr->next)
9. {
10. cout << "网卡" << index + 1 << "\t" << ptr->name << endl;
11. cout << "描述信息：" << ptr->description << endl;
12. for (a = ptr->addresses; a != NULL; a = a->next)
13. {
14. if (a->addr->sa\_family == AF\_INET)
15. {
16. cout << " IP地址：" << inet\_ntoa(((struct sockaddr\_in\*)(a->addr))->sin\_addr) << endl;
17. cout << " 子网掩码：" << inet\_ntoa(((struct sockaddr\_in\*)(a->netmask))->sin\_addr) << endl;
18. cout << " 广播地址：" << inet\_ntoa(((struct sockaddr\_in\*)(a->broadaddr))->sin\_addr) << endl;
19. }
20. }
21. index++;
22. }

结果展示：



##### 2.3 选择网卡

1. int num;
2. cout << "请选要打开的网卡号：";
3. cin >> num;
4. ptr = alldevs;
5. for (int i = 1; i < num; i++)
6. {
7. ptr = ptr->next;
8. }
9. pcap\_t\* pcap\_handle = pcap\_open(ptr->name, 1024, PCAP\_OPENFLAG\_PROMISCUOUS, 1000, NULL, errbuf);//打开网卡
10. if (pcap\_handle == NULL)
11. {
12. cout << "打开网卡时发生错误：" << errbuf << endl;
13. return 0;
14. }
15. else
16. {
17. cout << "成功打开该网卡" << endl;
18. }

##### 2.4 设置过滤器（ARP包）

1. //编译过滤器，只捕获ARP包
2. u\_int netmask;
3. netmask = ((sockaddr\_in\*)(ptr->addresses->netmask))->sin\_addr.S\_un.S\_addr;
4. bpf\_program fcode;
5. char packet\_filter[] = "ether proto \\arp";
6. if (pcap\_compile(pcap\_handle, &fcode, packet\_filter, 1, netmask) < 0)
7. {
8. cout << "无法编译数据包过滤器。检查语法";
9. pcap\_freealldevs(alldevs);
10. return 0;
11. }
12. //设置过滤器
13. if (pcap\_setfilter(pcap\_handle, &fcode) < 0)
14. {
15. cout << "过滤器设置错误";
16. pcap\_freealldevs(alldevs);
17. return 0;
18. }

##### 2.5 定义报文内容

1. //定义报文格式
2. for (int i = 0; i < 6; i++)
3. {
4. ARPFrame.FrameHeader.DesMAC[i] = 0xff;
5. ARPFrame.FrameHeader.SrcMAC[i] = 0x0f;
6. ARPFrame.RecvHa[i] = 0;//设置为0
7. ARPFrame.SendHa[i] = 0x0f;
8. }
9. ARPFrame.FrameHeader.FrameType = htons(0x806);//帧类型为ARP
10. ARPFrame.HardwareType = htons(0x0001);//硬件类型为以太网
11. ARPFrame.ProtocolType = htons(0x0800);//协议类型为IP
12. ARPFrame.HLen = 6;//硬件地址长度为6
13. ARPFrame.PLen = 4; // 协议地址长为4
14. ARPFrame.Operation = htons(0x0001);//操作为ARP请求
15. SendIP = ARPFrame.SendIP = htonl(0x00000000);//源IP地址

##### 2.6 发送信息并捕获返回的数据包

用pcap\_handle网卡发送ARPFrame中的内容，报文长度为sizzeof（ARPFrame\_t）

用选中网卡发送报文，发送成功返回0，循环捕获

1. pcap\_sendpacket(pcap\_handle, (u\_char\*)&ARPFrame, sizeof(ARPFrame\_t));
2. cout << "ARP请求发送成功" << endl;
3. while (true)
4. {
5. int rtn = pcap\_next\_ex(pcap\_handle, &pkt\_header, &pkt\_data);
6. if (rtn == -1)
7. {
8. cout << " 捕获数据包时发生错误：" << errbuf << endl;
9. return 0;
10. }
11. else
12. {
13. if (rtn == 0)
14. {
15. cout << " 没有捕获到数据报" << endl;
16. }
17. else
18. {
19. IPPacket = (ARPFrame\_t\*)pkt\_data;
20. if (IPPacket->RecvIP == SendIP && IPPacket->SendIP == RevIP)//判断是不是一开始发的包
21. {
22. cout << " 捕获到回复的数据报,请求IP与其MAC地址对应关系：" << endl;
23. printIP(IPPacket->SendIP);
24. cout << " --------------------- ";
25. printMAC(IPPacket->SendHa);
26. cout << endl;
27. break;
28. }
29. }
30. }
31. }

##### 2.7 获取远程网卡MAC地址

思路：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机网卡的MAC地址，将本机IP和MAC填入报文
3. 重新发送ARP请求
4. //向网络发送数据包
5. cout << "\n" << endl;
6. cout << "向网络发送一个数据包" << endl;
7. cout << "输入请求的IP地址:";
8. char str[15];
9. cin >> str;
10. RevIP = ARPFrame.RecvIP = inet\_addr(str);
11. SendIP = ARPFrame.SendIP = IPPacket->SendIP;//将本机IP赋值给数据报的源IP
12. for (int i = 0; i < 6; i++)
13. {
14. ARPFrame.SendHa[i] = ARPFrame.FrameHeader.SrcMAC[i] = IPPacket->SendHa[i];
15. }
16. if (pcap\_sendpacket(pcap\_handle, (u\_char\*)&ARPFrame, sizeof(ARPFrame\_t)) != 0)
17. {
18. cout << "ARP请求发送失败" << endl;
19. }
20. else
21. {
22. cout << "ARP请求发送成功" << endl;
23. while (true)
24. {
25. int n = pcap\_next\_ex(pcap\_handle, &pkt\_header, &pkt\_data);
26. if (n == -1)
27. {
28. cout << " 捕获数据包时发生错误：" << errbuf << endl;
29. return 0;
30. }

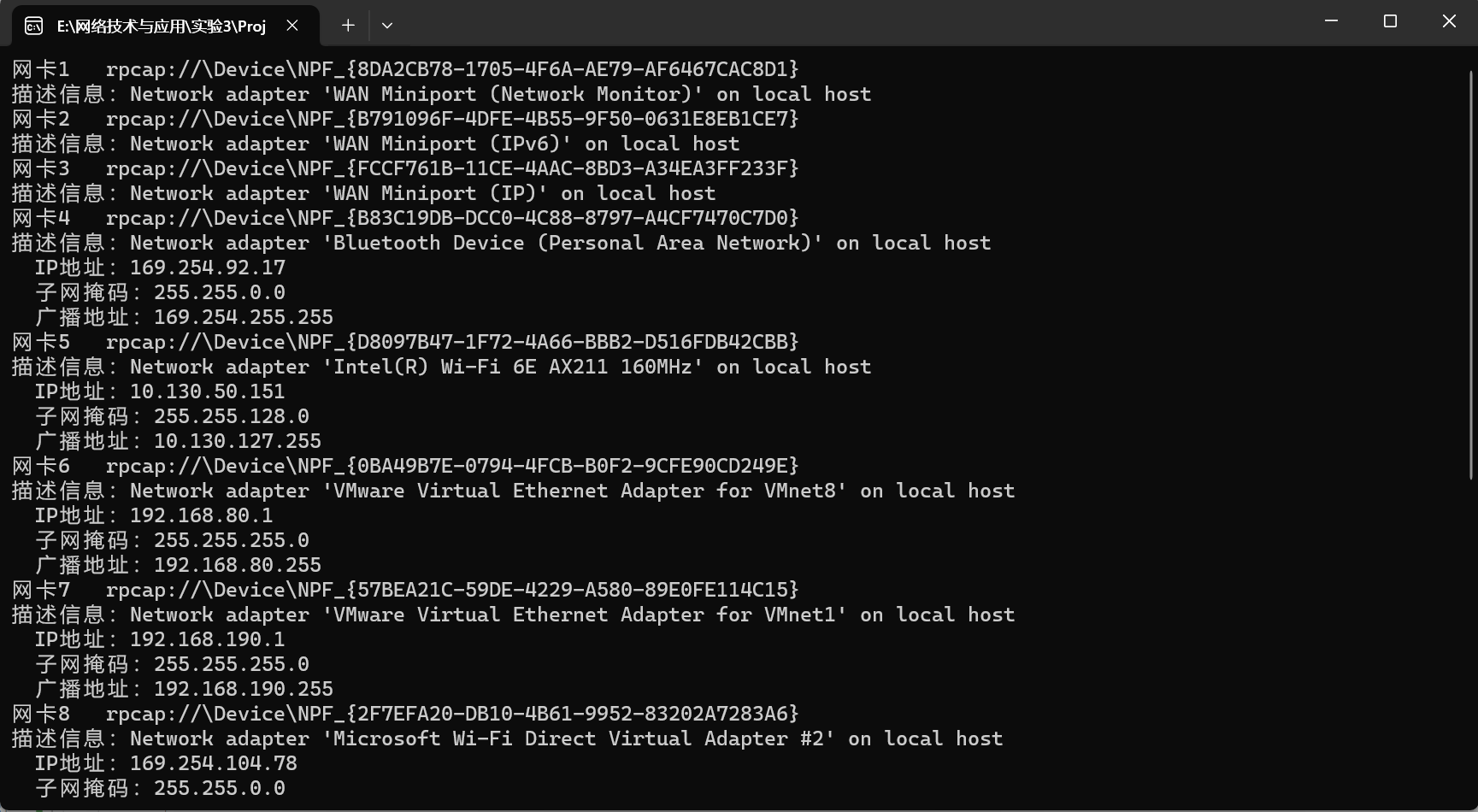
### 实验结果

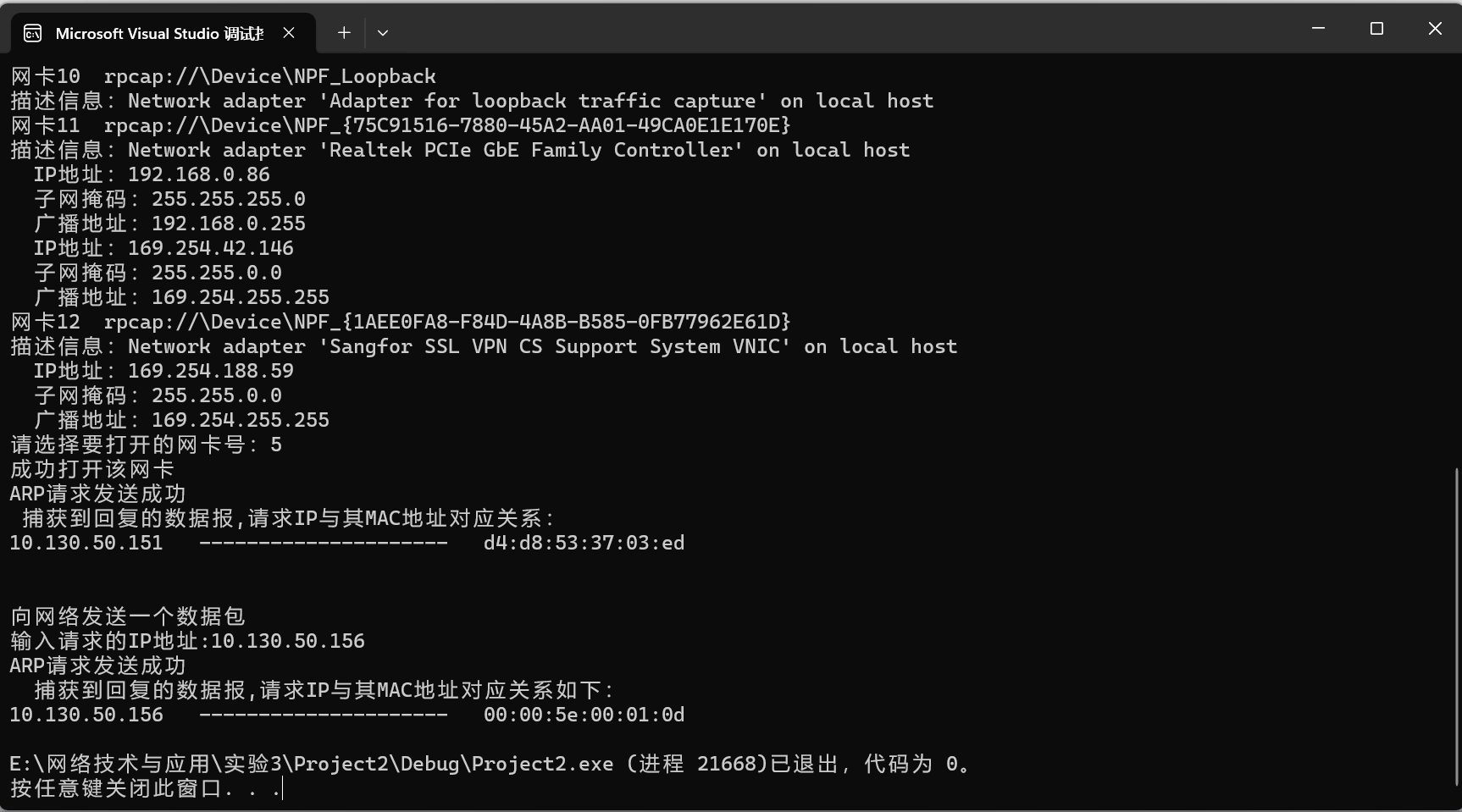
ARP协议广播分组是对相同网段内的主机进行的，不能跨网段直接获取远程主机的MAC地址。实现获取远程主机IP地址与MAC的对应关系要采用两个主机连接同一热点进行实验验证

本机IP：10.130.50.151

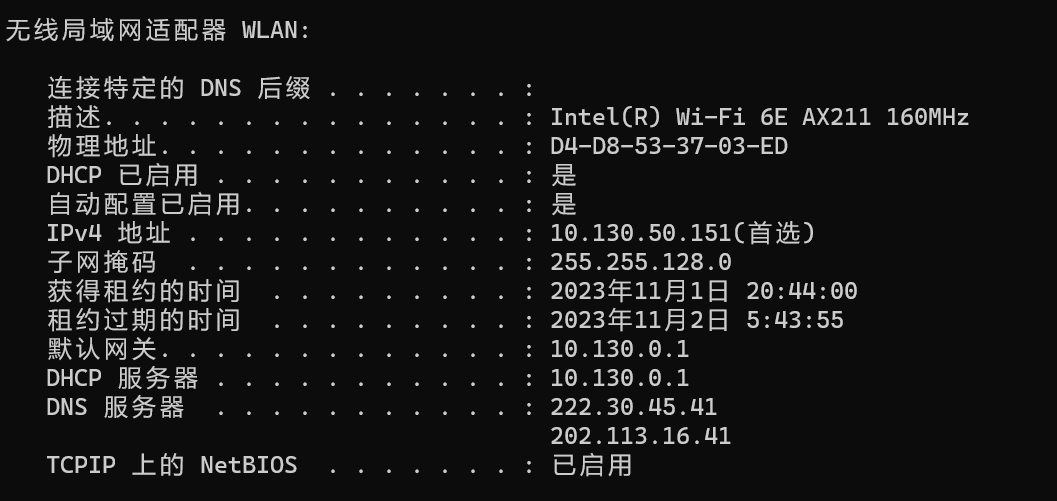
远程主机IP：10.130.50.156

设备列表信息：





在终端ipconfig/all：



与捕获结果相同

### 总结

通过获取 IP 地址与 MAC 地址对应关系实验更深刻的理解了 ARP 协议和以太网各终端主机设备之间的通信过程和通信方式。同时学习 了ARP 报文的格式与 ARP 协议的相关设定，对网络通信中 IP 地址与 MAC 地址的作用和意义有了更深刻的理解。