

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

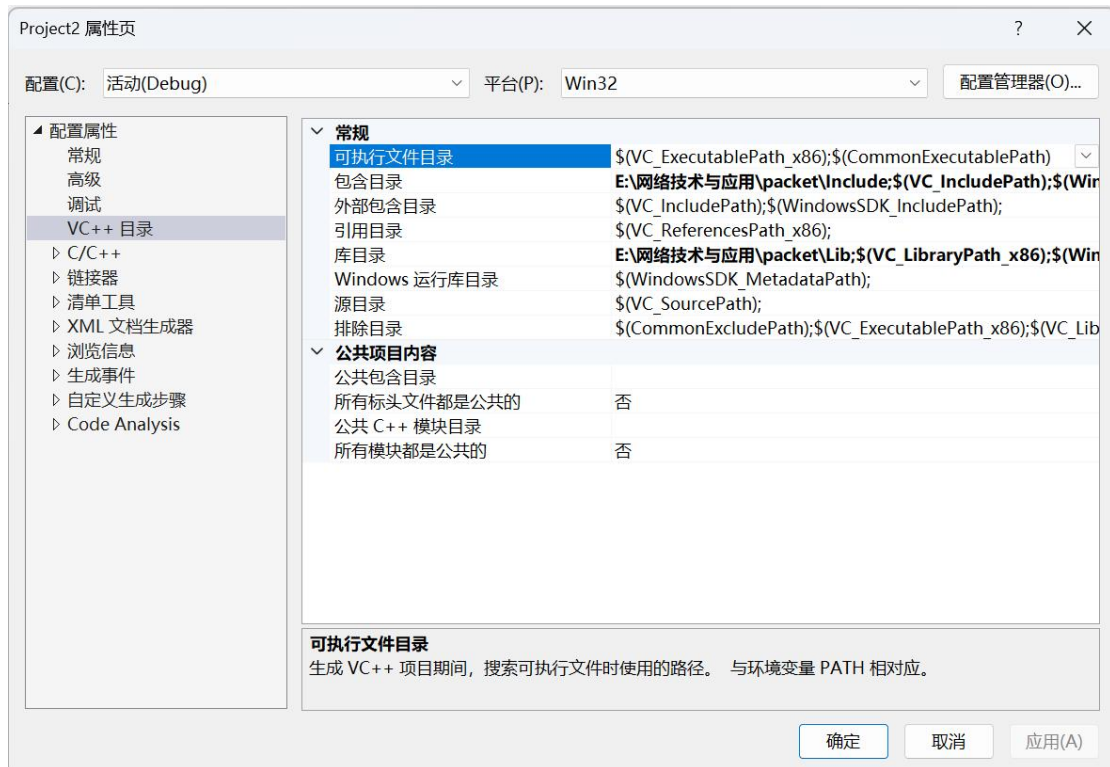
通过编程获取 IP 地址与 MAC 地址的对应关系实验，要求如下：

- 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。
- 通过 Npcap 编程，获取 IP 地址与 MAC 地址的映射关系。
- 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

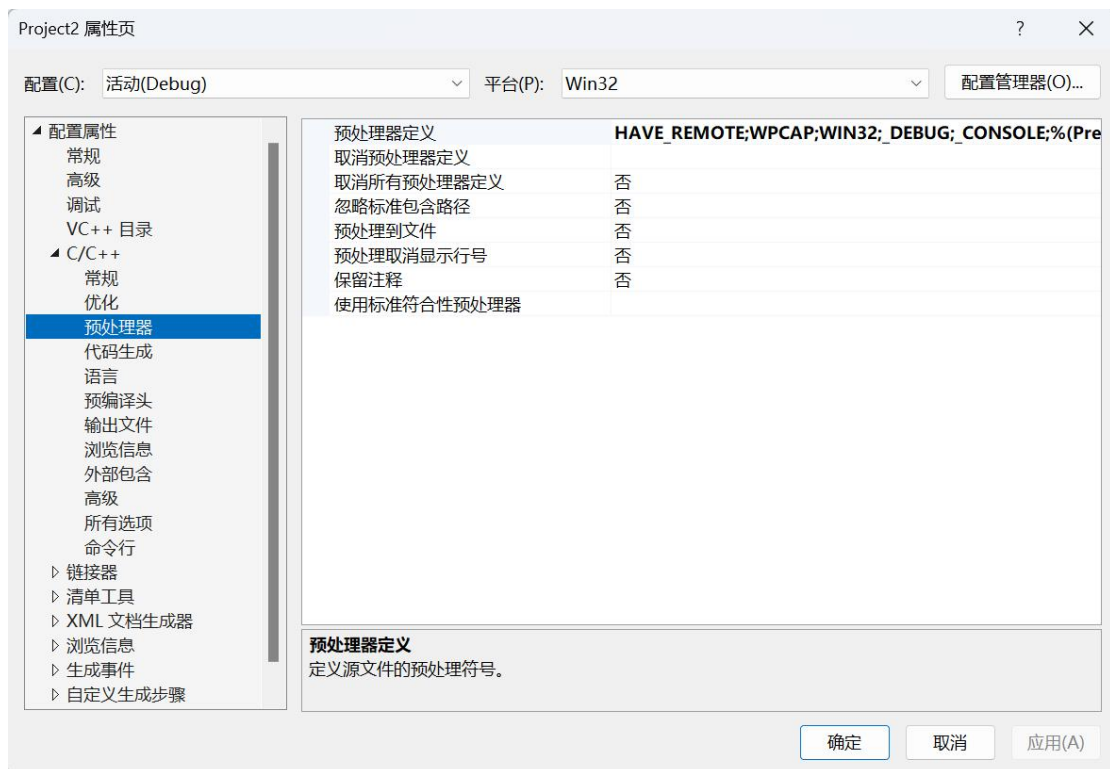
实验准备

1. 配置属性

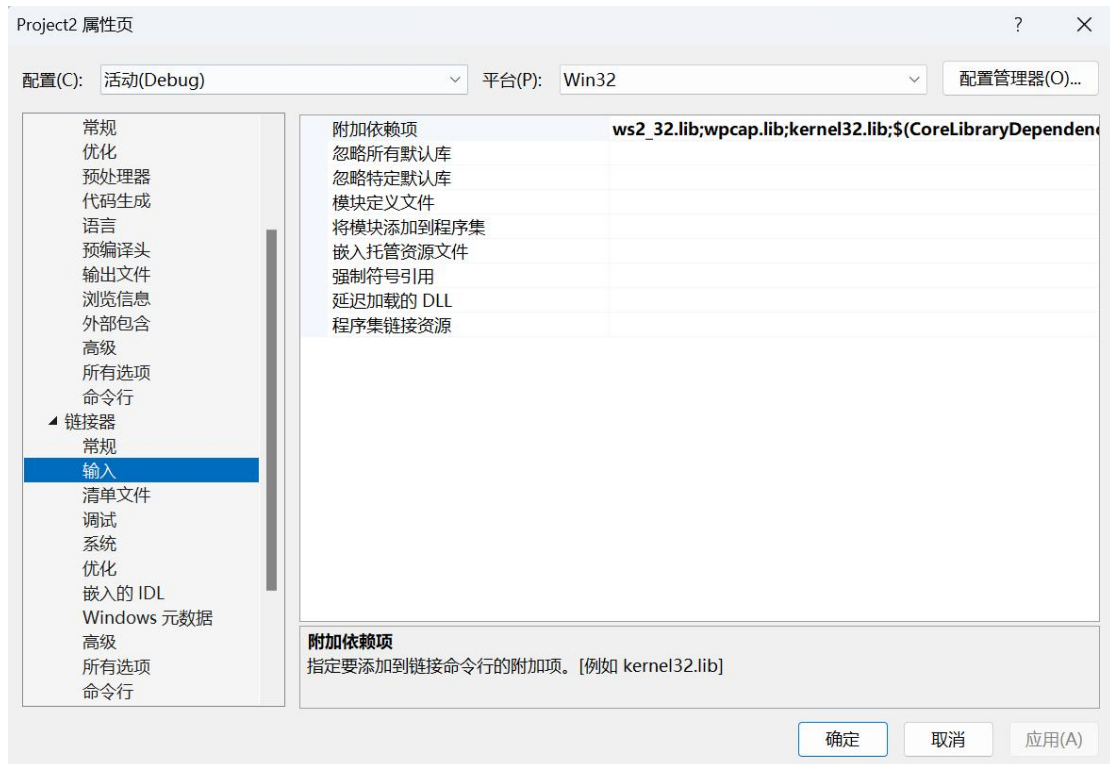
添加 Include 和 Lib 目录：



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

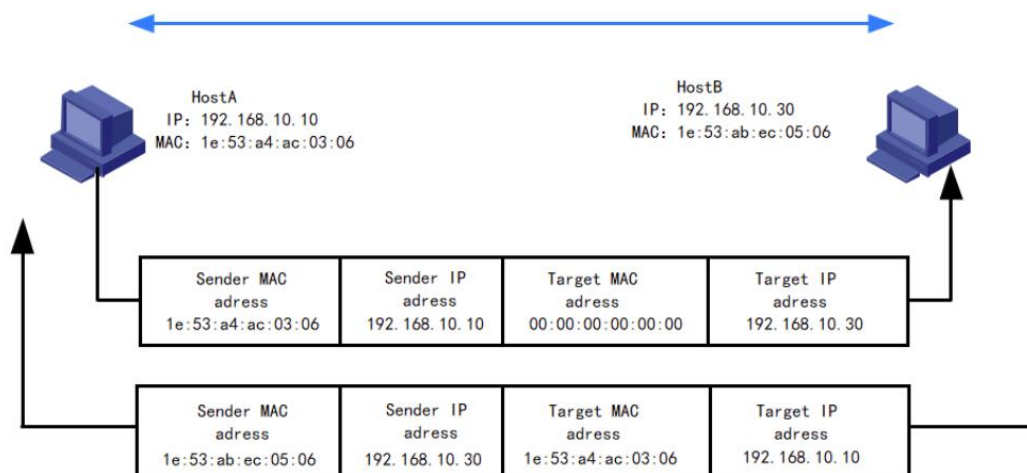
以太网发送数据包使用 WinPcap 提供的 `pcap_sendpacket(pcap_t *p, u_char buf, int size)` 函数，其中的参数：

- `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
8 struct ARPFrame_t //ARP 帧
9 {
10    FrameHeader_t FrameHeader;
11    WORD HardwareType;
12    WORD ProtocolType;
13    BYTE HLen;
14    BYTE PLen;
15    WORD Operation;
16    BYTE SendHa[6];
17    DWORD SendIP;
18    BYTE RecvHa[6];
19    DWORD RecvIP;
20 };
21 #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
13         for (a = ptr->addresses; a != NULL; a = a->next)
14         {
15
16             if (a->addr->sa_family == AF_INET)
17             {
18
19                 cout << "   IP 地址: " << inet_ntoa(((struct
20                 sockaddr_in*)(a->addr))->sin_addr) << endl;
21                 cout << "   子网掩码: " << inet_ntoa(((struct
22                 sockaddr_in*)(a->netmask))->sin_addr) << endl;
23                 cout << "   广播地址: " << inet_ntoa(((struct
24                 sockaddr_in*)(a->broadaddr))->sin_addr) << endl;
25             }
26         }
27
28         index++;
29     }

```

结果展示:

```

网卡1  rpcap://\Device\NPF_{8DA2CB78-1705-4F6A-AE79-AF6467CAC8D1}
描述信息: Network adapter 'WAN Miniport (Network Monitor)' on local host
网卡2  rpcap://\Device\NPF_{B791096F-4DFE-4B55-9F50-0631E8EB1CE7}
描述信息: Network adapter 'WAN Miniport (IPv6)' on local host
网卡3  rpcap://\Device\NPF_{FCCF761B-11CE-4AAC-8BD3-A34EA3FF233F}
描述信息: Network adapter 'WAN Miniport (IP)' on local host
网卡4  rpcap://\Device\NPF_{B83C19DB-DCC0-4C88-8797-A4CF7470C7D0}
描述信息: Network adapter 'Bluetooth Device (Personal Area Network)' on local host
IP地址: 169.254.92.17
子网掩码: 255.255.0.0
广播地址: 169.254.255.255
网卡5  rpcap://\Device\NPF_{D8097B47-1F72-4A66-BBB2-D516FDB42CBB}
描述信息: Network adapter 'Intel(R) Wi-Fi 6E AX211 160MHz' on local host
IP地址: 10.130.50.151
子网掩码: 255.255.128.0
广播地址: 10.130.127.255
网卡6  rpcap://\Device\NPF_{0BA49B7E-0794-4FCB-B0F2-9CFE90CD249E}
描述信息: Network adapter 'VMware Virtual Ethernet Adapter for VMnet8' on local host
IP地址: 192.168.80.1
子网掩码: 255.255.255.0
广播地址: 192.168.80.255
网卡7  rpcap://\Device\NPF_{57BEA21C-59DE-4229-A580-89E0FE114C15}
描述信息: Network adapter 'VMware Virtual Ethernet Adapter for VMnet1' on local host
IP地址: 192.168.190.1
子网掩码: 255.255.255.0
广播地址: 192.168.190.255
网卡8  rpcap://\Device\NPF_{2F7EFA20-DB10-4B61-9952-83202A7283A6}
描述信息: Network adapter 'Microsoft Wi-Fi Direct Virtual Adapter #2' on local host
IP地址: 169.254.104.78
子网掩码: 255.255.0.0

```

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
10     pcap_t* pcap_handle = pcap_open(ptr->name, 1024,
    PCAP_OPENFLAG_PROMISCUOUS, 1000, NULL, errbuf);//打开网卡
11     if (pcap_handle == NULL)
12     {
13         cout << "打开网卡时发生错误: " << errbuf << endl;
14         return 0;
15     }
16     else
17     {
18         cout << "成功打开该网卡" << endl;
19     }
```

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 sizeof (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
18            else
19            {
20                IPPacket = (ARPFrame_t*)pkt_data;
21                if (IPPacket->RecvIP == SendIP && IPPacket->SendIP == RevIP)//判断是
                不是一开始发的包
```



```

22         {
23
24             cout << " 捕获到回复的数据报,请求 IP 与其 MAC 地址对应关系: " <<
endl;
25             printIP(IPPacket->SendIP);
26             cout << "      -----      ";
27             printMAC(IPPacket->SendHa);
28             cout << endl;
29             break;
30         }
31     }
32 }
33 }

```

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

```

1 //向网络发送数据包
2     cout << "\n" << endl;
3     cout << "向网络发送一个数据包" << endl;
4     cout << "输入请求的 IP 地址:";
5     char str[15];
6     cin >> str;
7     RevIP = ARPFrame.RecvIP = inet_addr(str);
8     SendIP = ARPFrame.SendIP = IPPacket->SendIP; //将本机 IP 赋值给数据报的源 IP
9     for (int i = 0; i < 6; i++)
10    {
11        ARPFrame.SendHa[i] = ARPFrame.FrameHeader.SrcMAC[i] =
IPPacket->SendHa[i];
12    }
13
14    if (pcap_sendpacket(pcap_handle, (u_char*)&ARPFrame, sizeof(ARPFrame_t)) != 0)
15    {
16        cout << "ARP 请求发送失败" << endl;
17    }
18    else
19    {
20        cout << "ARP 请求发送成功" << endl;
21    }

```

```

22         while (true)
23         {
24             int n = pcap_next_ex(pcap_handle, &pkt_header, &pkt_data);
25             if (n == -1)
26             {
27                 cout << " 捕获数据包时发生错误: " << errbuf << endl;
28                 return 0;
29             }

```

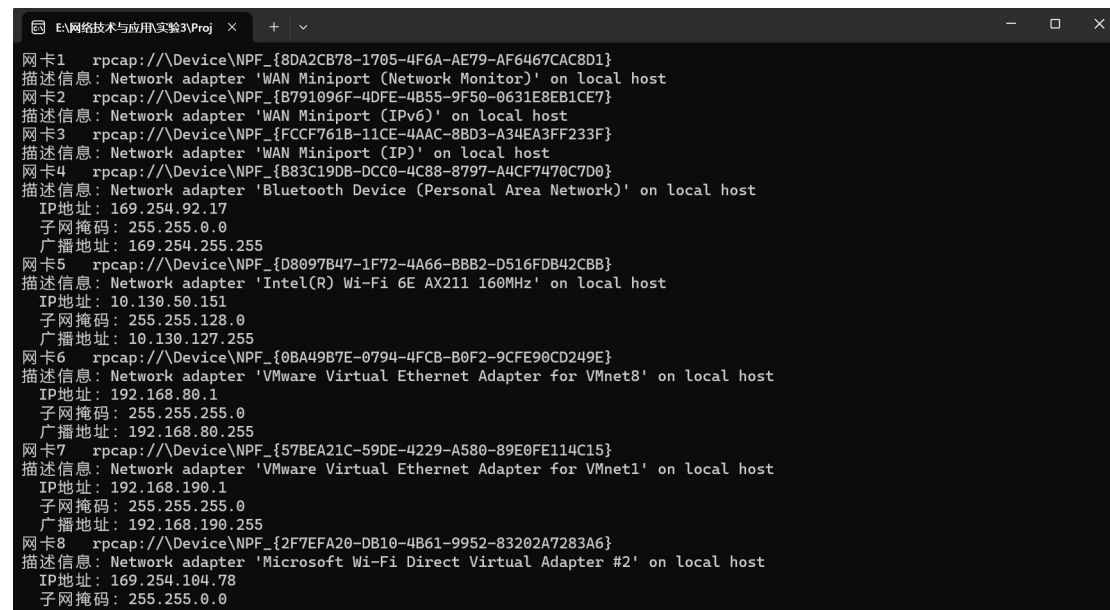
实验结果

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:



```

E:\网络技术与应用实验3\Proj  ×  +  ▾
网卡1  rpcap://\Device\NPF_{8DA2CB78-1705-4F6A-AE79-AF6467CAC8D1}
描述信息: Network adapter 'WAN Miniport (Network Monitor)' on local host
网卡2  rpcap://\Device\NPF_{B791096F-4DFE-4B55-9F50-0631E8EB1CE7}
描述信息: Network adapter 'WAN Miniport (IPv6)' on local host
网卡3  rpcap://\Device\NPF_{FCCF761B-11CE-4AAC-8BD3-A34EA3FF233F}
描述信息: Network adapter 'WAN Miniport (IP)' on local host
网卡4  rpcap://\Device\NPF_{B83C19DB-DCC0-4C88-8797-A4CF7470C7D0}
描述信息: Network adapter 'Bluetooth Device (Personal Area Network)' on local host
IP地址: 169.254.92.17
子网掩码: 255.255.0.0
广播地址: 169.254.255.255
网卡5  rpcap://\Device\NPF_{D8097B47-1F72-4A66-BBB2-D516FDB42CBB}
描述信息: Network adapter 'Intel(R) Wi-Fi 6E AX211 160MHz' on local host
IP地址: 10.130.50.151
子网掩码: 255.255.128.0
广播地址: 10.130.127.255
网卡6  rpcap://\Device\NPF_{0BA49B7E-0794-4FCB-B0F2-9CFE90CD249E}
描述信息: Network adapter 'VMware Virtual Ethernet Adapter for VMnet8' on local host
IP地址: 192.168.80.1
子网掩码: 255.255.255.0
广播地址: 192.168.80.255
网卡7  rpcap://\Device\NPF_{57BEA21C-59DE-4229-A580-89E0FE114C15}
描述信息: Network adapter 'VMware Virtual Ethernet Adapter for VMnet1' on local host
IP地址: 192.168.190.1
子网掩码: 255.255.255.0
广播地址: 192.168.190.255
网卡8  rpcap://\Device\NPF_{2F7EFA20-DB10-4B61-9952-83202A7283A6}
描述信息: Network adapter 'Microsoft Wi-Fi Direct Virtual Adapter #2' on local host
IP地址: 169.254.104.78
子网掩码: 255.255.0.0

```

```
Microsoft Visual Studio 调试  x + -
网卡10  rpcap://\Device\NPF_{Loopback}
描述信息: Network adapter 'Adapter for loopback traffic capture' on local host
网卡11  rpcap://\Device\NPF_{75C91516-7880-45A2-AA01-49CA0E1E170E}
描述信息: Network adapter 'Realtek PCIe GbE Family Controller' on local host
IP地址: 192.168.0.86
子网掩码: 255.255.255.0
广播地址: 192.168.0.255
IP地址: 169.254.42.146
子网掩码: 255.255.0.0
广播地址: 169.254.255.255
网卡12  rpcap://\Device\NPF_{1AEE0FA8-F84D-4A8B-B585-0FB77962E61D}
描述信息: Network adapter 'Sangfor SSL VPN CS Support System VNIC' on local host
IP地址: 169.254.188.59
子网掩码: 255.255.0.0
广播地址: 169.254.255.255
请选择要打开的网卡号: 5
成功打开该网卡
ARP请求发送成功
捕获到回复的数据报, 请求IP与其MAC地址对应关系:
10.130.50.151 ----- d4:d8:53:37:03:ed

向网络发送一个数据包
输入请求的IP地址:10.130.50.156
ARP请求发送成功
捕获到回复的数据报, 请求IP与其MAC地址对应关系如下:
10.130.50.156 ----- 00:00:5e:00:01:0d

E:\网络技术与应用\实验3\Project2\Debug\Project2.exe (进程 21668)已退出, 代码为 0。
按任意键关闭此窗口. . .
```

在终端 ipconfig/all:

```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . : 
   描述. . . . . : Intel(R) Wi-Fi 6E AX211 160MHz
   物理地址. . . . . : D4-D8-53-37-03-ED
   DHCP 已启用 . . . . . : 是
   自动配置已启用. . . . . : 是
   IPv4 地址 . . . . . : 10.130.50.151(首选)
   子网掩码 . . . . . : 255.255.128.0
   获得租约的时间 . . . . . : 2023年11月1日 20:44:00
   租约过期的时间 . . . . . : 2023年11月2日 5:43:55
   默认网关. . . . . : 10.130.0.1
   DHCP 服务器 . . . . . : 10.130.0.1
   DNS 服务器 . . . . . : 222.30.45.41
                           202.113.16.41
   TCP/IP 上的 NetBIOS . . . . . : 已启用
```

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。