南 开 大 学

网络空间安全学院学院

网络技术与应用课程报告

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**第3次实验报告**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学号：2011428

姓名：王天行

年级：2020级

专业：密码科学与技术

2022年10月13日

1. **实验内容说明**

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WinPcap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。

1. **实验准备**

### 学习Npcap发送数据包的方法

#### pcap\_sendpacket()

用pcap\_sendpacket()来发送一个数据包，这个函数需要参数：

* 一个装有要发送数据的缓冲区；
* 要发送的长度；
* 一个适配器；

注意：缓冲区中的数据将不被内核协议处理，只是作为最原始的数据流被发送，所以我们必须填充好正确的协议头以便正确的将数据发送

#### pcap\_sendqueue\_alloc()

pcap\_sendpacket()只是提供了一个简单的直接的发送数据的方法，而发送队列提供了一个高级的、强大的、和最优的机制来发送一组数据包，队列实际上是一个装有要发送数据的一个容器，他又一个最大值来表明它能够容纳的最大比特数。

pcap\_sendqueue\_alloc()用来创建爱你一个队列，并指定该队列的大小。

一旦队列被创建就可以调用pcap\_sendqueue\_queue()来将数据存储到队列中，这个函数接受一个带有时间戳和长度的pcap\_pkthdr结构和一个装有数据报的缓冲区。这些参数同样也应用于pcap\_next\_ex()和pcap\_handler()中，所以给要捕获的数据包或要从文件读取的数据包排队就是pcap\_sendqueue\_queue()的事情了。

WinPcap调用pcap\_sendqueue\_transmit()来发送数据包，注意，第三个参数如果非零，那么发送将是同步的，这将站用很大的CPU资源，因为发生在内核驱动的同步发送是通过"brute force"loops的，但是一般情况下能够精确到微秒。

需要指出的是用pcap\_sendqueue\_transmit()来发送比用pcap\_sendpacket()来发送一系列的数据要高效的多，因为他的数据是在内核级上被缓冲。

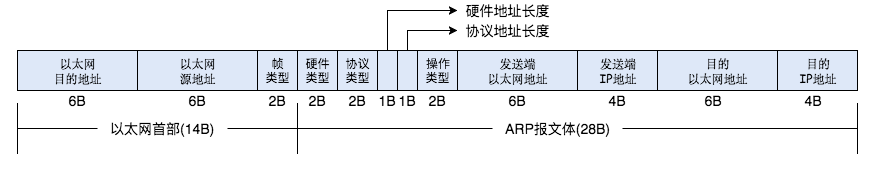
当不再需要队列时可以用pcap\_sendqueue\_destroy()来释放掉所有的队列资源。

### 了解ARP协议与ARP数据包结构

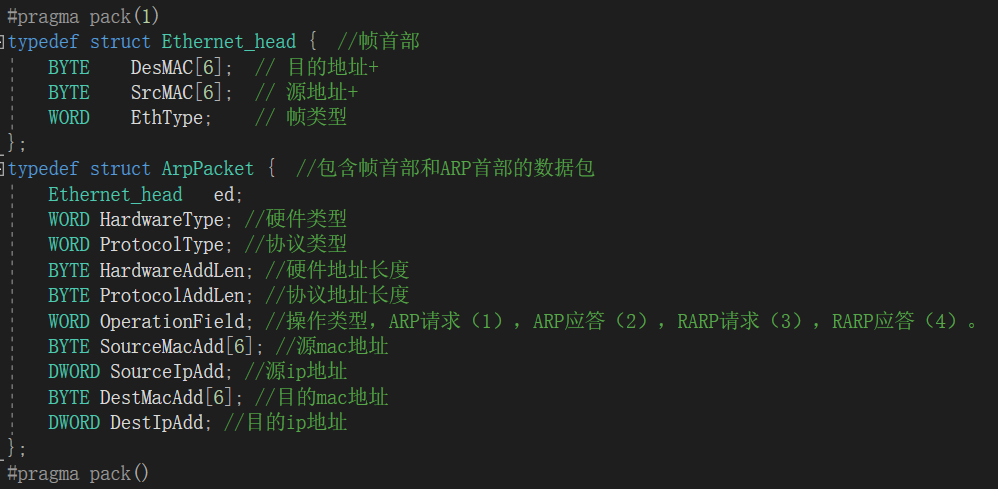
#### ARP请求过程

1. 主机A先在ARP缓存表中查找主机B的MAC地址
2. 如果在A的ARP缓存表中找到了，那么就继续数据封装通信。如果没有找到主机A会先发送ARP的广播包（里面包括了A的IP地址和MAC地址、主机B的IP地址）
3. 既然是广播帧那么内网中的所有存活主机都会收到该ARP的广播包
4. 存活主机会进行检查自身IP地址是否与广播包中的目的IP地址一致，如果不一致的话进行丢弃。如果一致那么就会将主机A的IP和MAC地址添加到自己的ARP缓存表里面，然后再将自己的MAC地址和ARP响应包通过单播方式发送给主机A
5. 然后主机A就可以给主机B发送消息

#### ARP数据包结构



从而得到ARP报文结构：

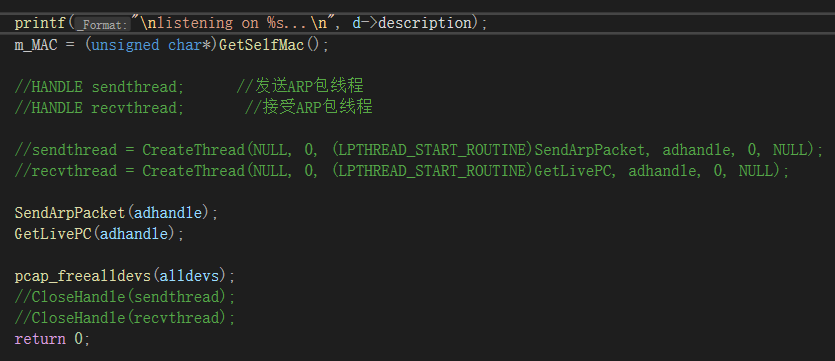


1. **实验过程**

### 整体代码思路

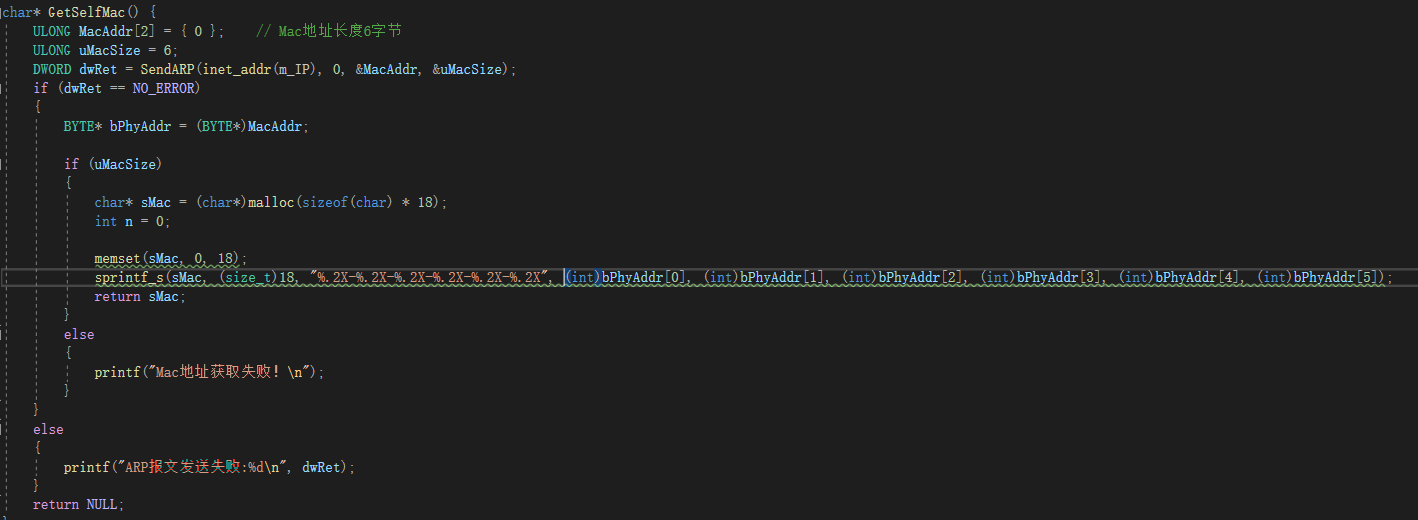
首先与之前的实验相同，获得本机设备列表和接口列表，跳转到指定的网卡并打开网卡。（因为与之前部分代码一致且代码较长，此处就不放代码截图了）

之后需要获取到本机的MAC地址和目的IP地址，并根据已知数据填充ARP数据包，使用sendpacket()函数将数据包发送到网络中去。之后开始读取接收到的ARP数据包，并分析出目的IP地址和MAC地址。

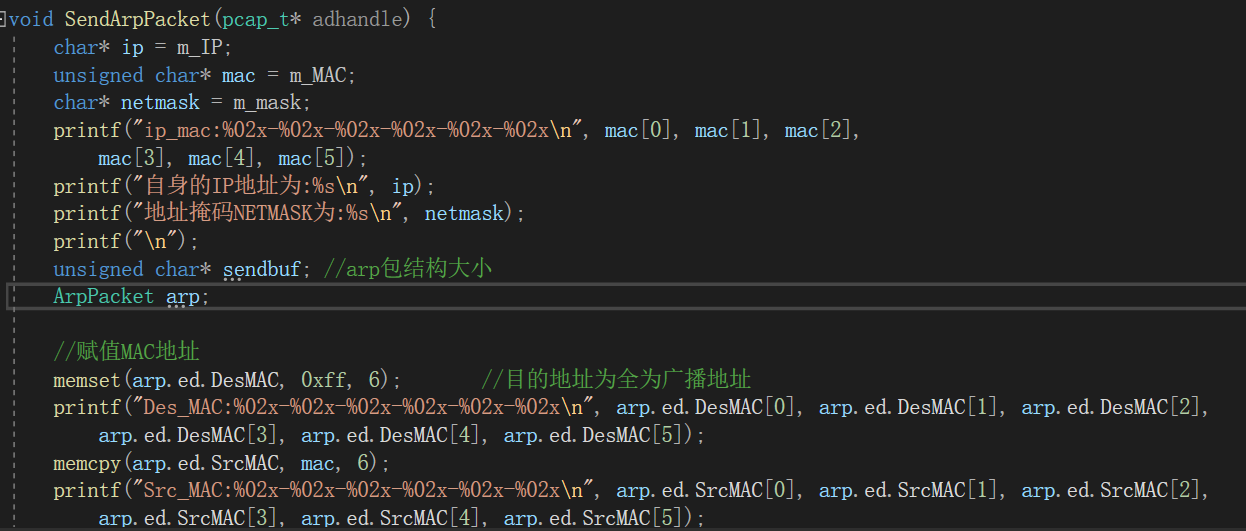


### 涉及函数及其功能

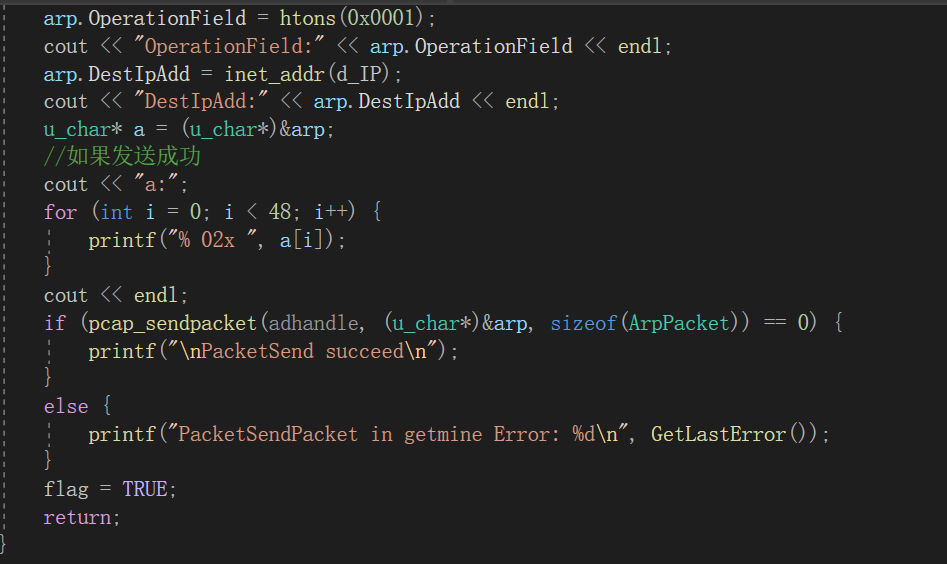
#### 获取本机MAC地址（GetSelfMac()）



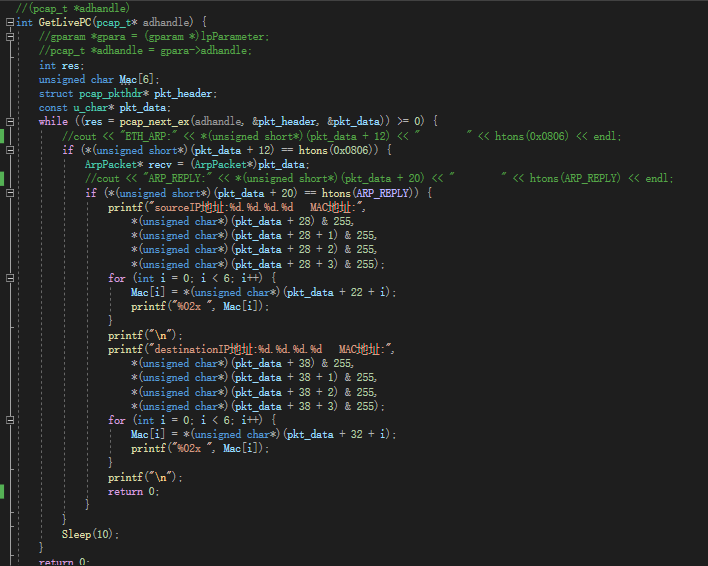
#### 发送数据包（SendArpPacket(adhandle)）





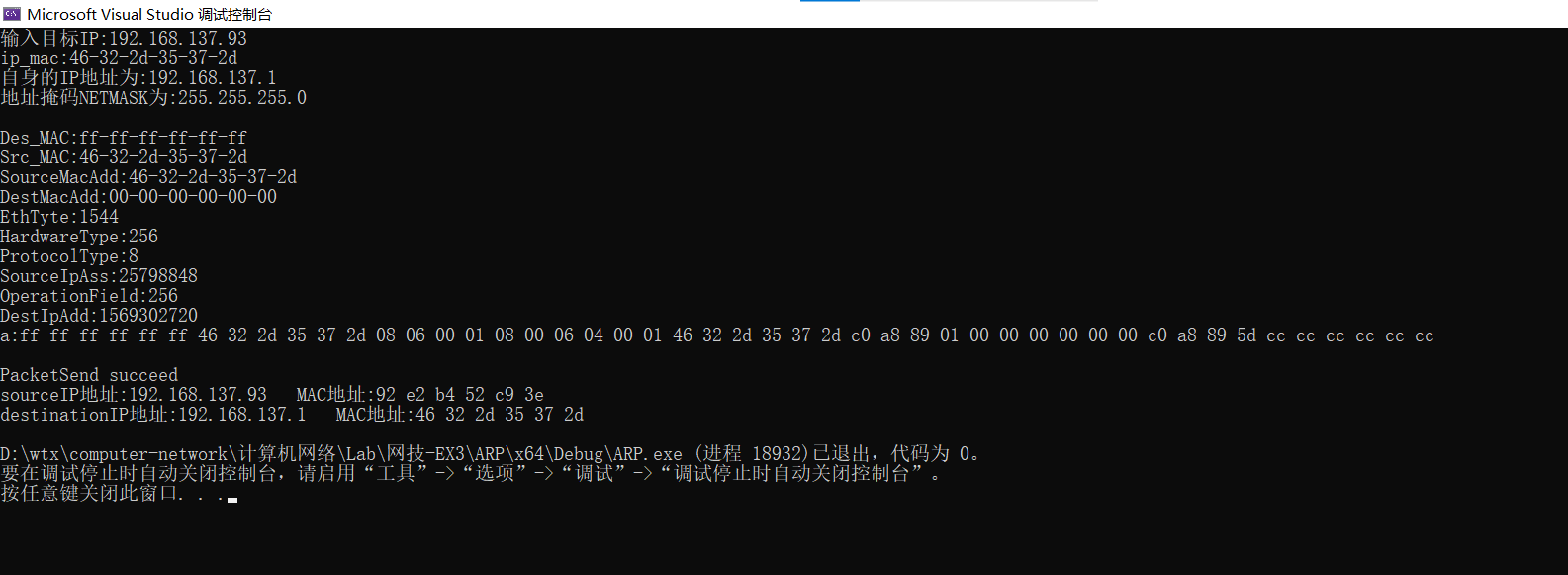


#### 接收数据包（GetLivePC(adhandle)）

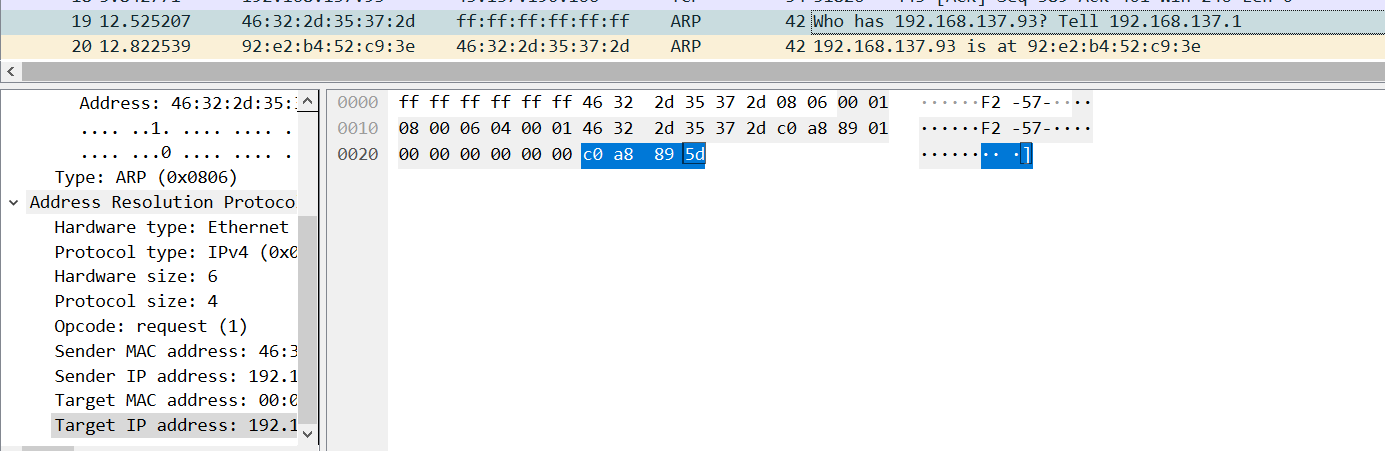


### 实验结果

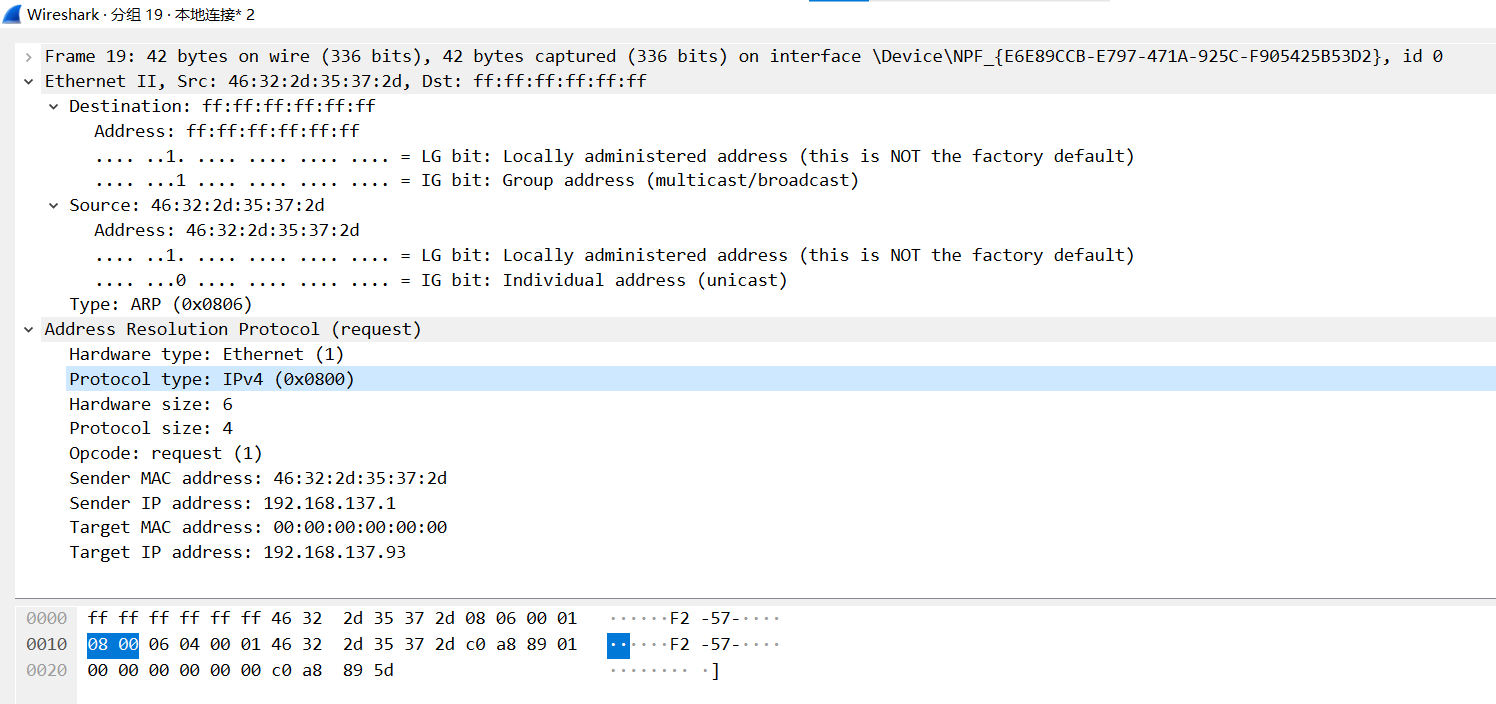
代码输出结果：



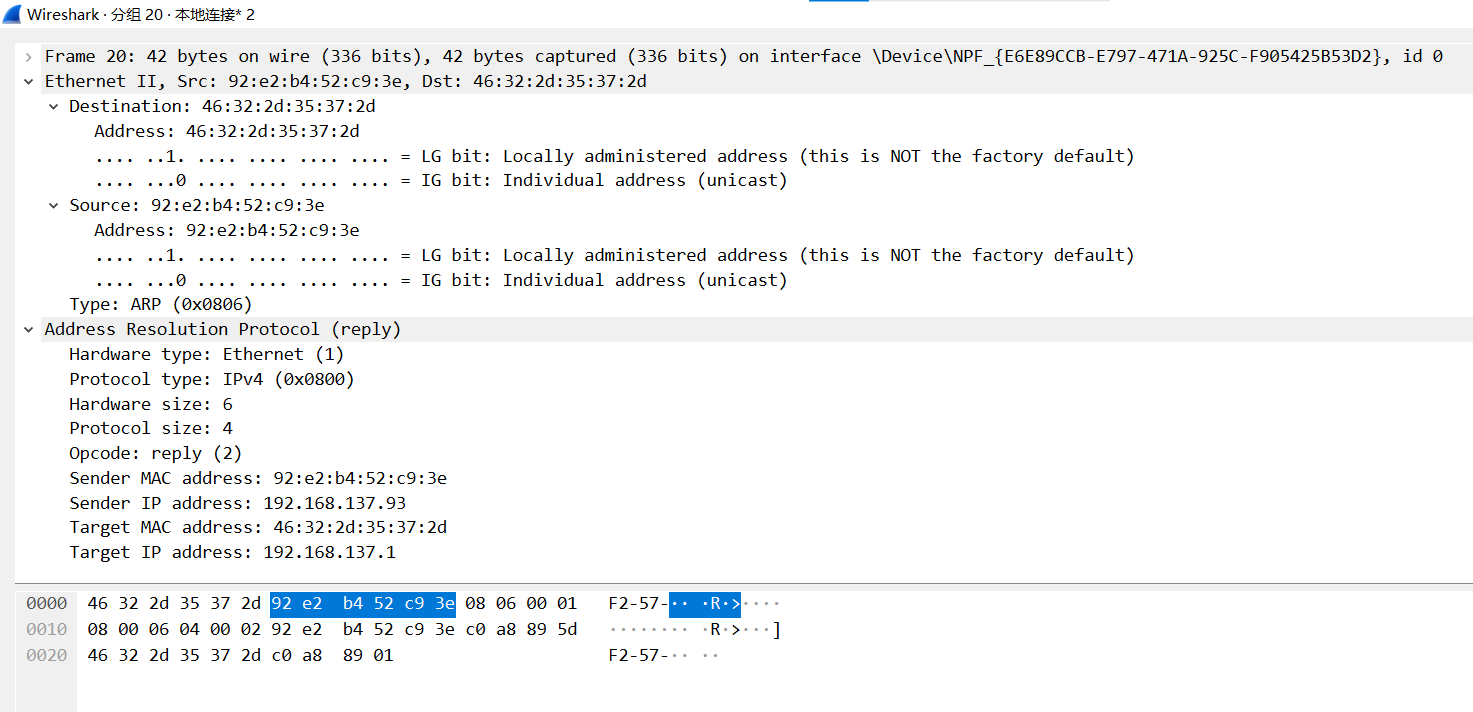
Wireshark抓包结果：



第一个数据包：



第二个数据包：



二者对比：

