实验三 网络数据捕获与分析

目 录

[**1.实验目的** 3](#_Toc118740090)

[**2.实验要求** 3](#_Toc118740091)

[**3.实验环境** 3](#_Toc118740092)

[**4.实验步骤** 3](#_Toc118740093)

[4.1抓取分析MAC帧 3](#_Toc118740094)

[4.1.1抓取 3](#_Toc118740095)

[4.1.2分析 6](#_Toc118740096)

[4.2查看ARP数据 7](#_Toc118740097)

[4.2.1发送ARP请求 7](#_Toc118740098)

[4.2.2查看ARP请求 10](#_Toc118740099)

[4.3分析IP数据报 12](#_Toc118740100)

[4.3.1不分片的IP数据报 12](#_Toc118740101)

[4.3.2分片的IP数据报 13](#_Toc118740102)

[4.4遇到的一个问题及解决方法 16](#_Toc118740103)

[**实验心得体会** 19](#_Toc118740104)

**1.实验目的**

学习并掌握网络数据捕获的方法，并能对网络数据进行分析。

**2.实验要求**

1、学习并掌握网络数据捕获的方法；

2、学习并掌握生成网络测试数据的方法；

3、MAC 帧格式分析，并发现 MAC 帧数据部分的常见类型；

4、ARP 协议分析，包括请求和响应；

5、IP 数据报格式分析（包括分片和不分片两种情况）

**3.实验环境**

硬件：Intel X64兼容机+Intel AX201无线网卡

操作系统：Windows10专业工作站版22623

软件：wireshark4.0.1（稳定版本）

**4.实验步骤**

4.1抓取分析MAC帧

4.1.1抓取

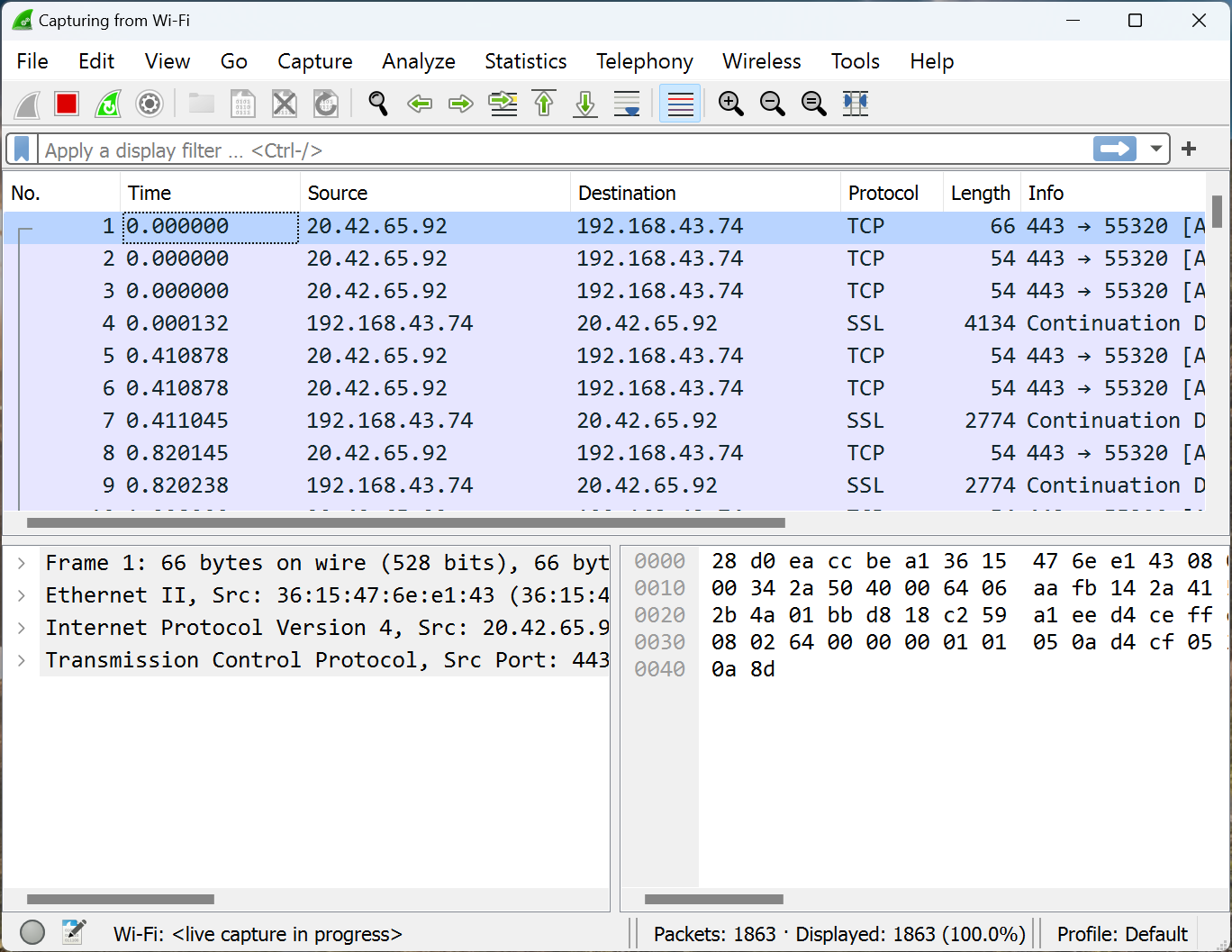
打开wireshark抓包程序，选择当前正在使用的网卡，然后开始自动读取：

注意：如果计算机设置了虚拟网卡，务必在选择时确认是否为当前操作系统正在使用的网卡！

Graphical user interface, text, email

Description automatically generated

选好后就能看到一条一条的网络数据了：



由于现在的计算机都是通过以太网上网的，所以随便选取一条就能看到以太网帧：

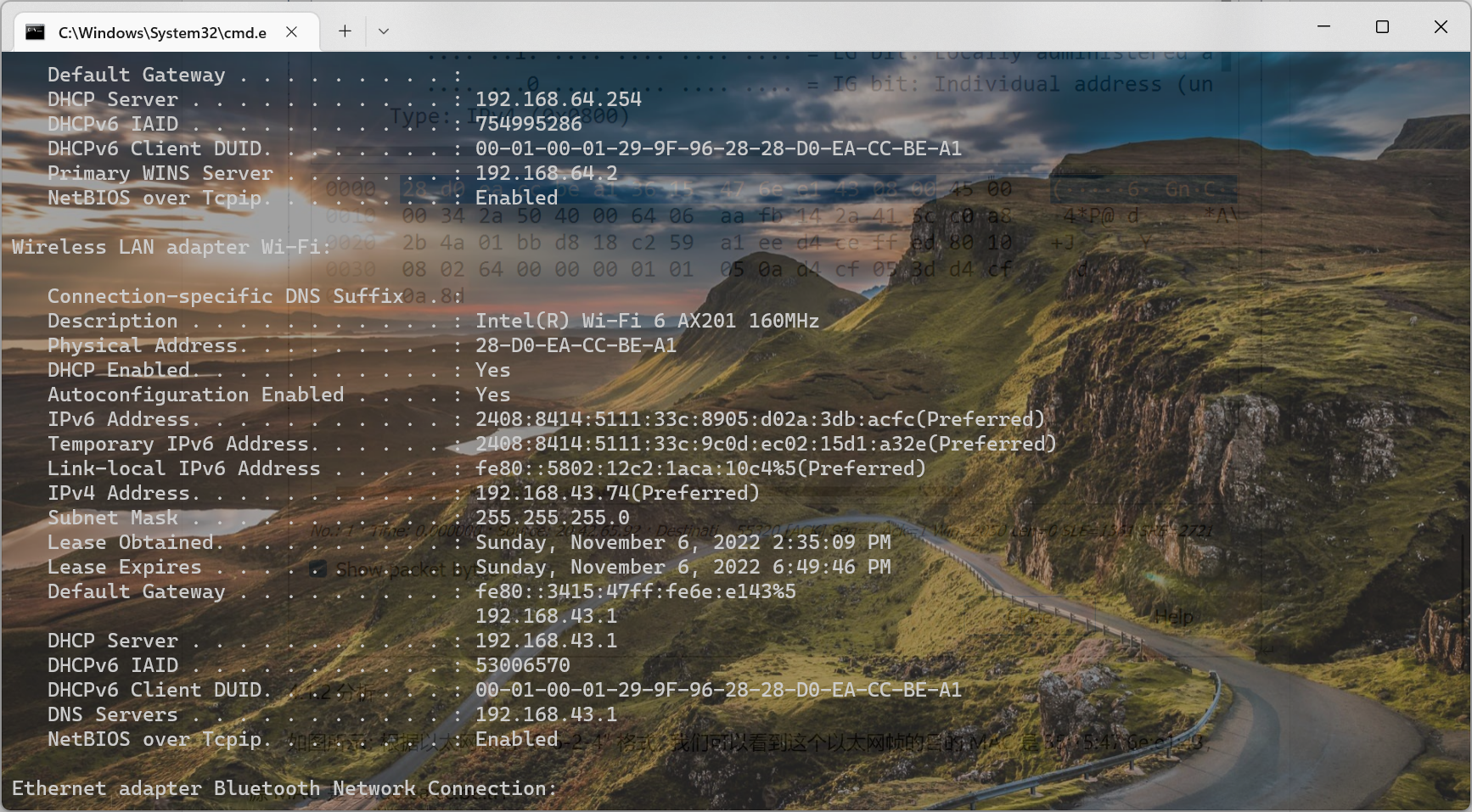
Graphical user interface, text, application

Description automatically generated

4.1.2分析

如图所示：根据以太网帧的“6-6-2-x-4”格式，我们可以看到这个以太网帧的源MAC是36:15:47:6e:e1:43，目的MAC是28:d0:ea:cc:be:a1，类型字段为0x0800，代表这里面封装的是ipv4数据报

打开终端，查看本机网络设置，发现这个目的MAC就是本机网卡的MAC：



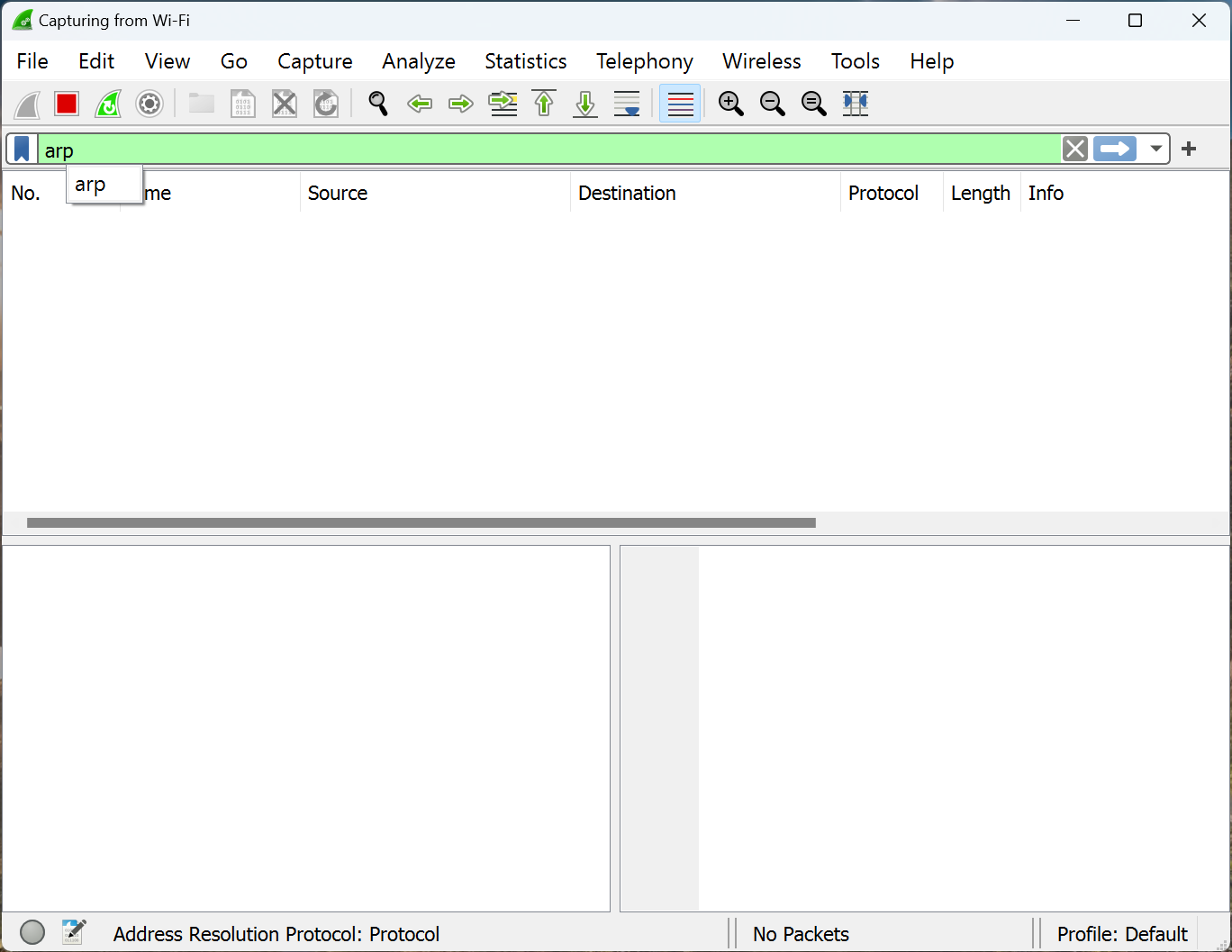
4.2查看ARP数据

4.2.1发送ARP请求

操作系统没有给我们提供直接发送ARP请求的功能，但是我们可以清空本机的ARP缓存，从而强制使计算机重新发送ARP请求

备注：如果计算机连接的是总线型以太网，那么也可以忽略此步骤，因为在总线型以太网上通过混杂模式可以抓取到其他计算机发送的ARP请求和响应、

首先设置一下wireshark只显示arp数据，防止其他的条目干扰观察：



登录到终端：输入arp-d（如果没有权限请以管理员账户登录，如果是Linux请先尝试获取root权限）

如图所示选择“run as administrator”以管理员方式登录终端

Graphical user interface, application

Description automatically generated

输入命令：

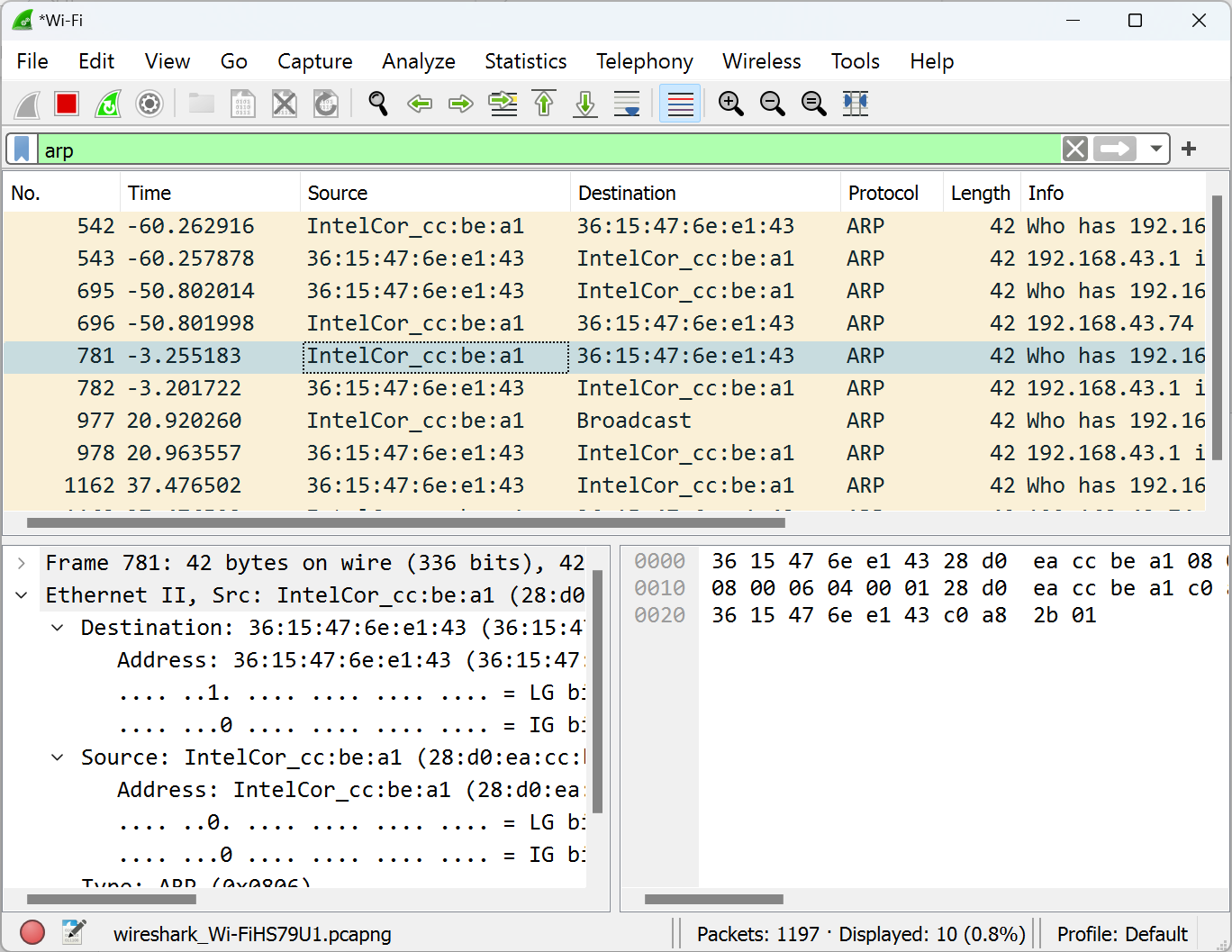
Text

Description automatically generated

这样就完成了请求发送

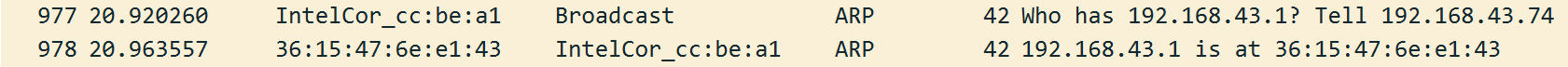
4.2.2查看ARP请求

回到wireshark：



显然，目的地址是广播的代表ARP请求，目的地址为具体MAC的为ARP响应

以这两条为例，请求的目的地址是广播，内容是：“请将192.168.43.1的mac地址告诉ip地址为192.168.43.74，mac地址为28:d0:ea:cc:be:a1的计算机”；响应由36:15:47:6e:e1:43回答，目的地址是28:d0:ea:cc:be:a1，内容是“192.168.43.1的mac地址是36:15:47:6e:e1:43”



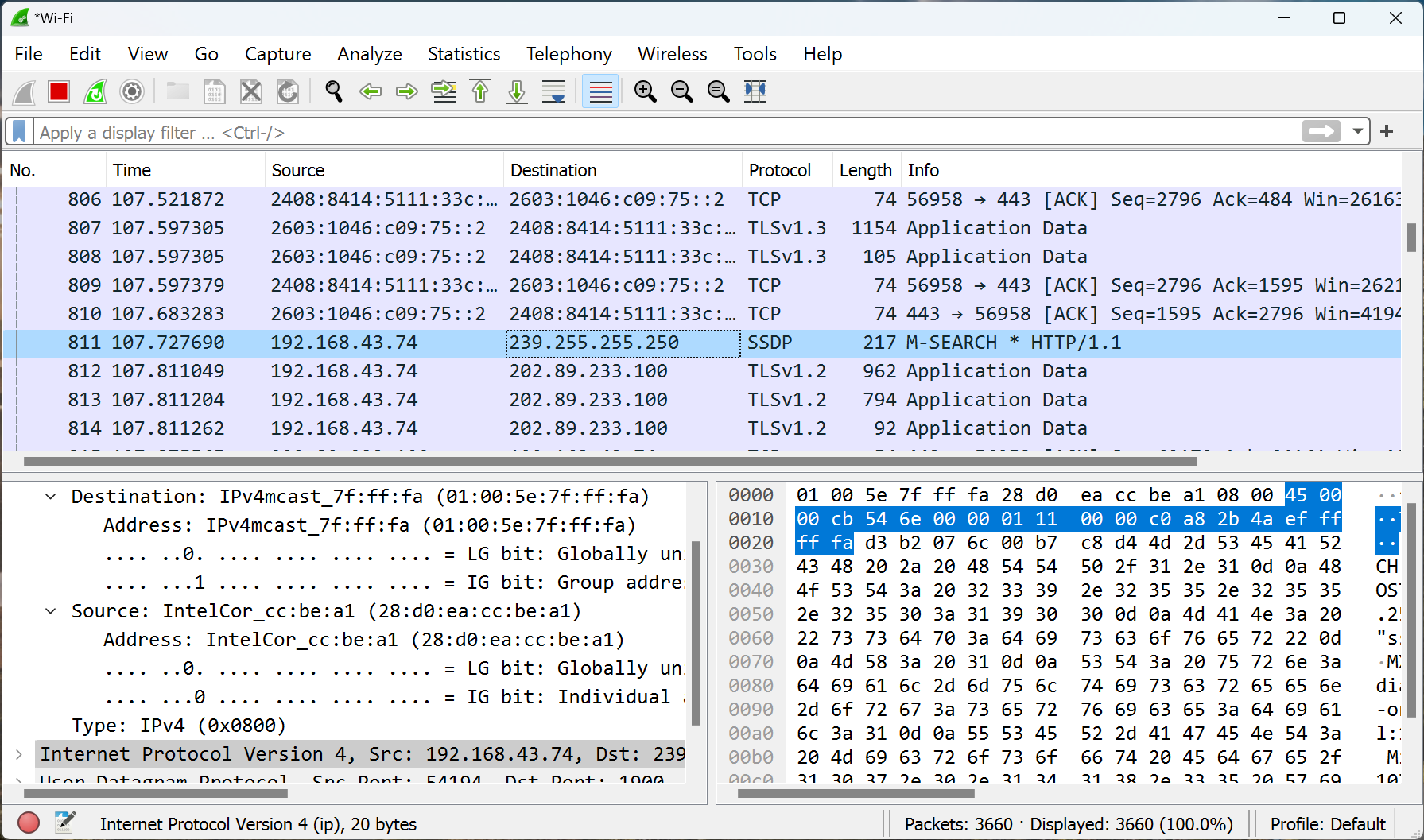
事实上进行此实验时，实验计算机连接的是一个无线路由器，这两个地址，一个是主机，一个是那台路由器，因此这两个地址才反反复复出现。

至于主机的mac地址为什么显示为一串奇怪的字符，这是因为wireshark识别到了这是一块Intel网卡，自动添加了型号信息。

4.3分析IP数据报

4.3.1不分片的IP数据报

由于大部分IP数据报都是不分片的，因此随便找一个看看：



打开这条：

Graphical user interface, text, application

Description automatically generated

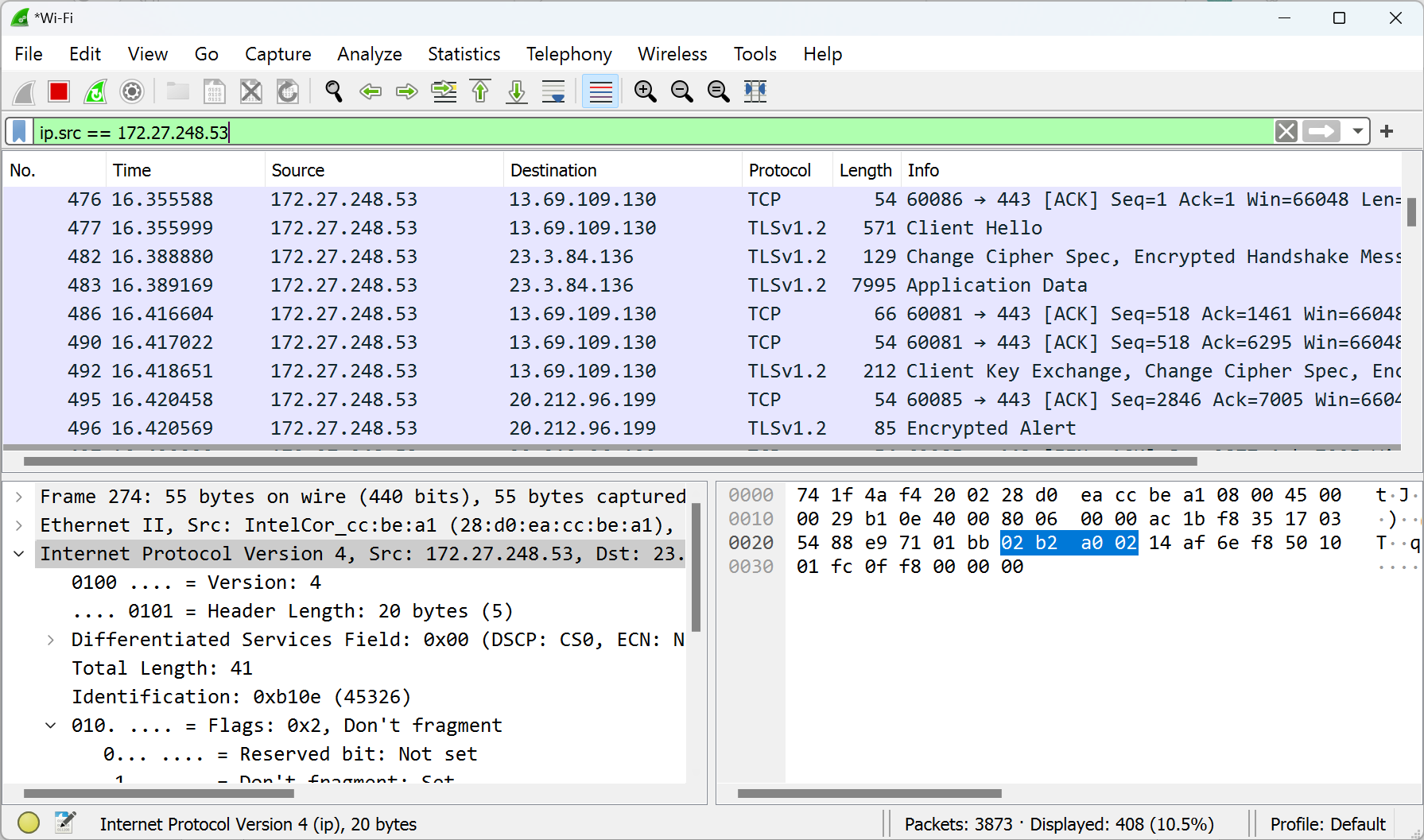
可以很明显的看到：IP版本为4，首部长度5个单位（20字节）源地址192.168.43.74，目的地址239.255.255.250，区分服务字段没有，标识符21614（十进制），没有分片，DF=0，MF=0，总长度203字节

4.3.2分片的IP数据报

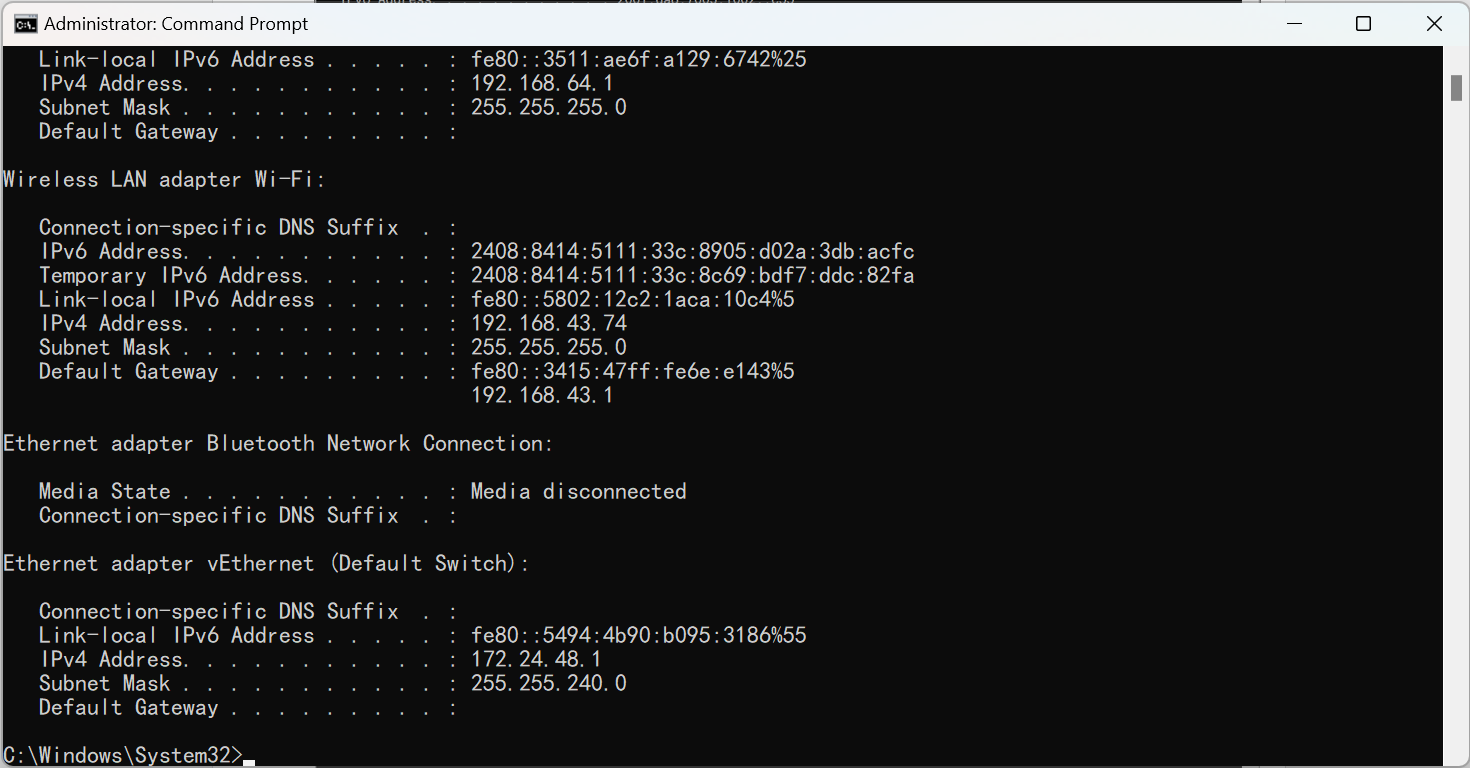
正常情况下使用互联网，是比较难创造一个条件让IP数据报分片的。但我们可以利用ping命令，设置要发送的数据长度，当总长度大于以太网的MTU（1500字节）时，我们就能够观察到效果

设置一下wireshark，让她只显示来自我的的数据

注意：这一段网络不太稳定，切换了好几次，因此我的IP地址有所变化



登录终端，查看一下当前的默认网关：



输入ping命令，回车：

Text

Description automatically generated

6000字节的数据长度，足够产生分片了，此时查看抓包内容：

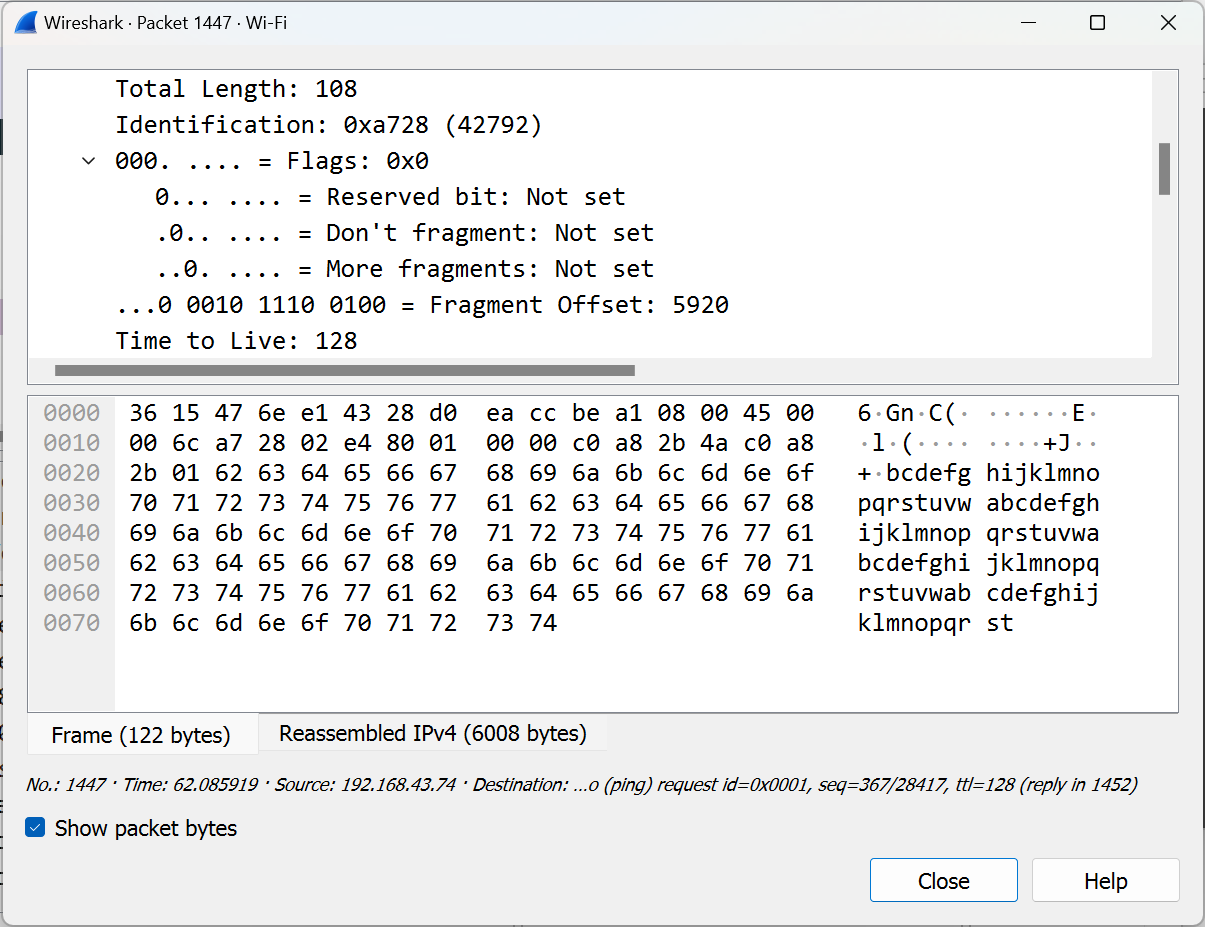
Graphical user interface, application

Description automatically generated

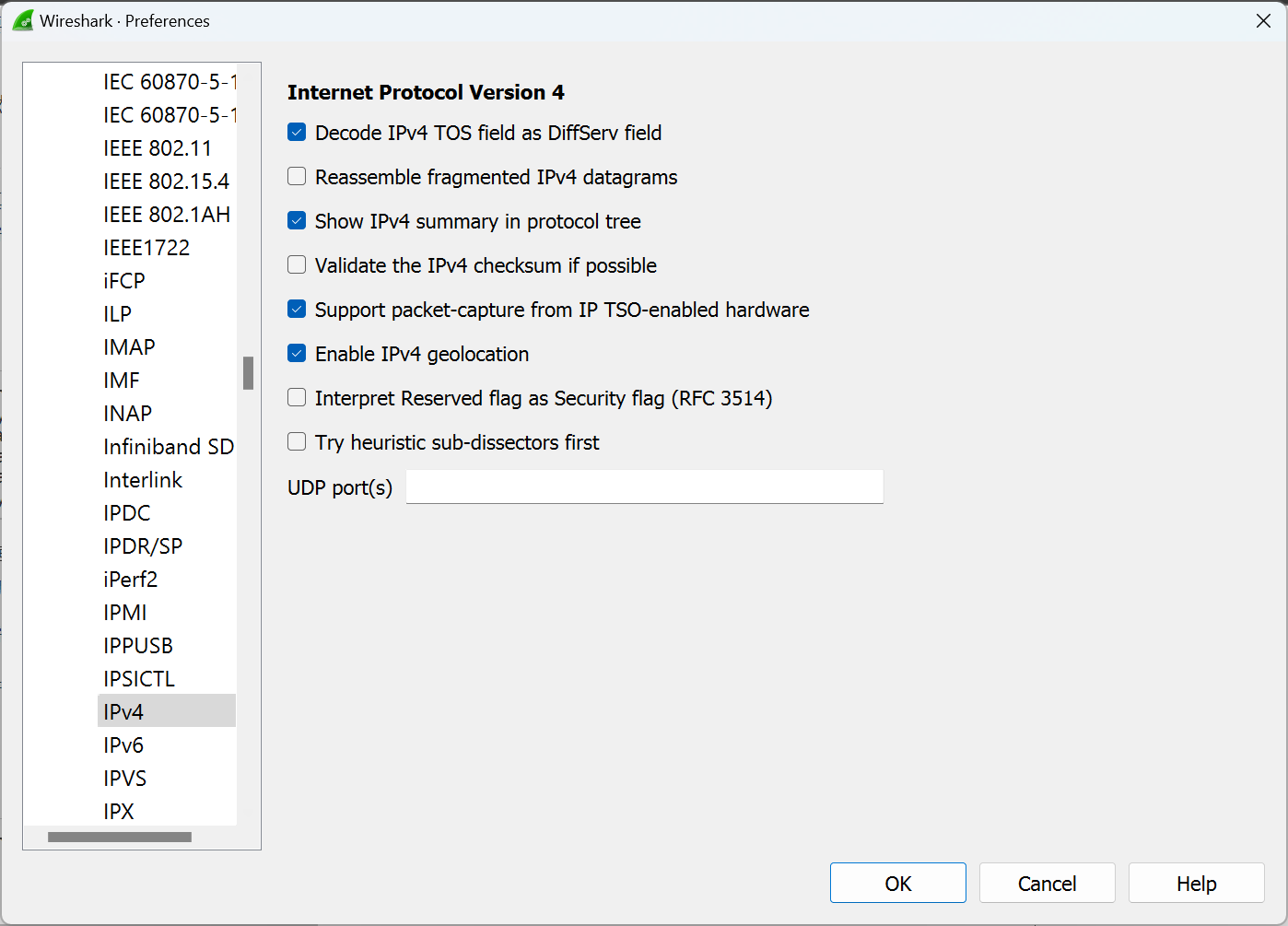
可以看到出现了ICMP请求报文，打开其中一个就可以观察到了

4.4遇到的一个问题及解决方法

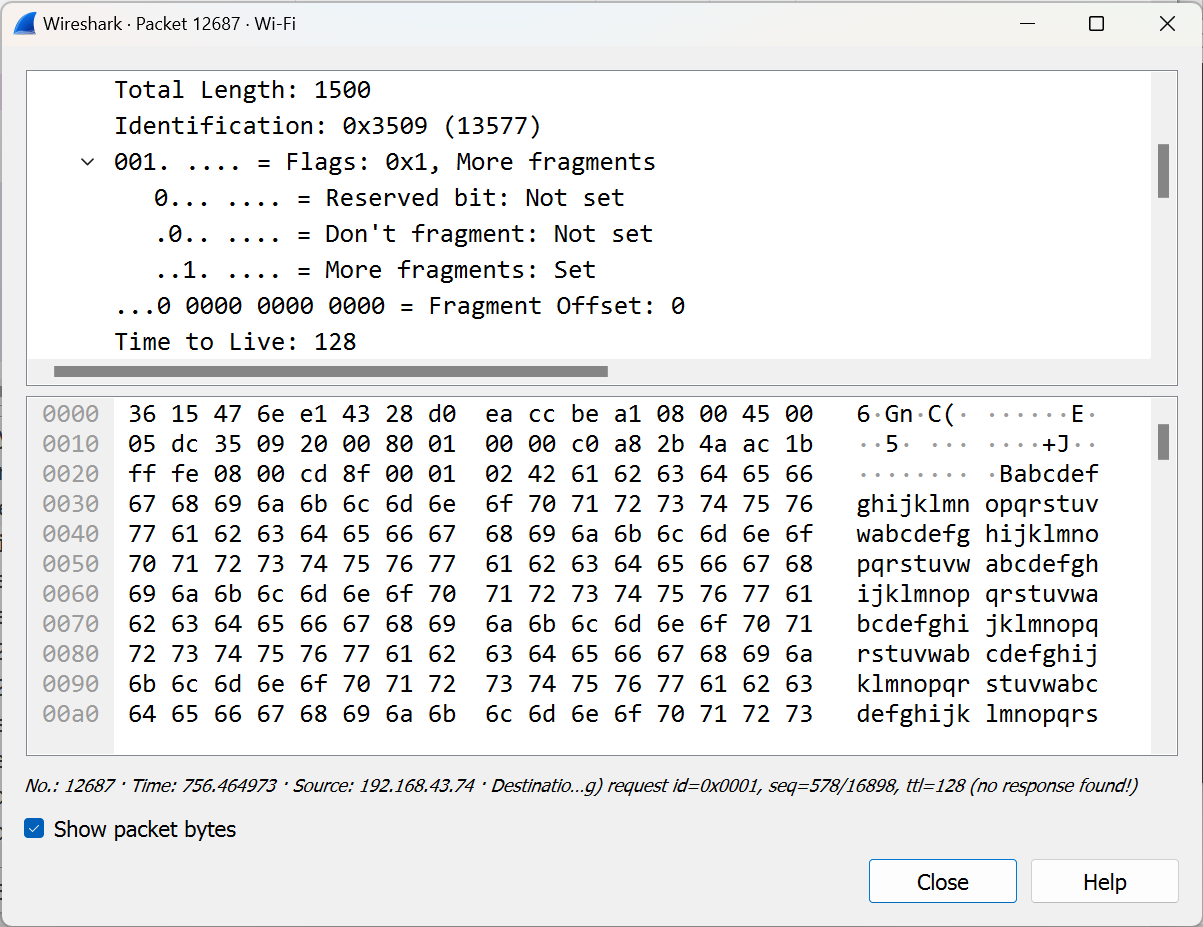
不过，不知道出于何种原因，wireshark总是显示每次发送的最后一段数据，但这也足以让我们观察到片偏移字段了，就是这个fragment offset：



后来发现解决方法其实也很简单，参考自[Wireshark数据包分析实战：IP分片 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/355811066)，应该是软件设置问题，去掉这个Reassemble fragmented IPv4 datagrams选项：



然后再重复上面的实验步骤，就能看到DF，MF字段了：



通过MF字段可以看到，确实发生了IP数据报分片的现象

**实验心得体会**

通过这次实验，加深理解了MAC帧，IP数据报的格式以及以太网MTU，ARP协议，默认网关等知识点，初步学习了网络抓包实用程序的使用方式，锻炼了通过查找资料解决问题的能力。