华东师范大学软件工程学院实验报告

实验课程：计算机网络实践	**年级**：2023 级	**实验成绩**：
实验名称：Ethernet	**姓名**：顾翌炜	
实验编号：Lab-2	**学号**：10235101527	**实验日期**：2024/11/29
指导教师：王廷	**组号**：01	**实验时间**：2024/11/29

1 实验目的

1) 学会通过 Wireshark 获取以太网的帧
2) 掌握以太网的结构
3) 分析以太网地址范围
4) 分析以太网的广播帧

2 实验内容和实验步骤

2.1 实验内容

2.1.1 获取以太网的帧

在 cmd 中使用 `ping` 命令发起 `ICMP` 请求，然后使用 `Wireshark` 捕获以太网数据包。

2.1.2 分析以太网的帧

分析**以太网**的帧，画出帧结构。

2.1.3 分析以太网的地址范围

分析**以太网**的地址范围，画出图示关系图。

2.1.4 分析以太网的广播帧

启动 `Wireshark`,在菜单栏的捕获 → 选项中进行设置,选择已连接的以太网,设置捕获过滤器为 `ethermulticast`，捕获以太网的广播帧。

分析以太网的广播帧，回答以下问题:

1) What is the broadcast Ethernet address, written in standard form as Wireshark displays it?

2) Which bit of the Ethernet address is used to determine whether it is unicast or multicast/broadcast?

2.1.5 问题讨论

1) How long are the combined IEEE 802.3 and LLC headers compared to the DIX Ethernet headers? You can use Wireshark to work this out. Note that the Trailer/Padding and Checksum may be shown as part of the header, but they come at the end of the frame.
2) How does the receiving computer know whether the frame is DIX Ethernet or IEEE 802.3? Hint: you may need to both use Wireshark to look at packet examples and read your text near where the Ethernet formats are described.
3) If IEEE 802.3 has no Type field, then how is the next higher layer determined? Use Wireshark to look for the demultiplexing key.

2.2 实验步骤

1) 打开 cmd，使用 `ping www.baidu.com` 命令来发起 ICMP 请求
2) 启动 `Wireshark`，在菜单栏捕获 → 选项重的设置，选择已连接的以太网，设置捕获过滤器为 icmp，将混杂模式设置为关闭，勾选 `enable MAC name resolution`，然后开始捕获
3) 回到 cmd，再次使用 `ping www.baidu.com` 发起 ICMP 请求
4) 回到 `Wireshark`，停止捕获
5) 分析捕获到的以太网的帧，画出帧结构
6) 分析以太网的地址范围，画出图示关系图
7) 启动 `Wireshark`,在菜单栏捕获 → 选项重的设置,选择已连接的以太网,设置捕获过滤器为 `ether multicast`，然后开始捕获以太网的广播帧。
8) 问题讨论

3 实验环境

- 操作系统：`Windows 11 家庭中文版 23H2 22631.4460`
- 网络适配器：`Killer(R)Wi-Fi 6E AX1675i 160MHz Wireless Network Adapter(211NGW)`
- `Wireshark: Version 4.4.1`
- `wget: GNU Wget 1.21.4 built on mingw32`

4 实验过程与分析

4.1 获取以太网的帧

首先，我们在命令行中使用 `ping` 命令发起 `ICMP` 请求。

```
1     C:\User\GHOST> ping www.baidu.com
```

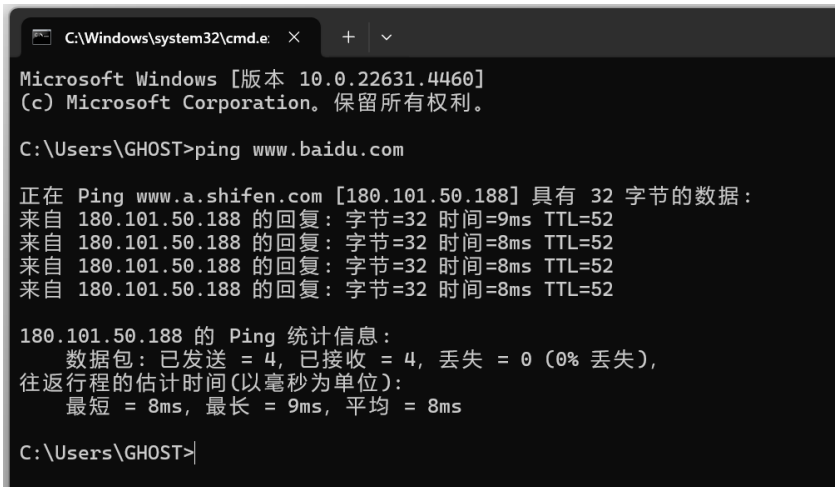

图 1: 使用 ping 命令发起 ICMP 请求

启动 `Wireshark`，在菜单栏捕获 → 选项重的设置，选择已连接的以太网，设置捕获过滤器为 icmp，将混杂模式设置为关闭，勾选 `enable MAC name resolution`，然后开始捕获

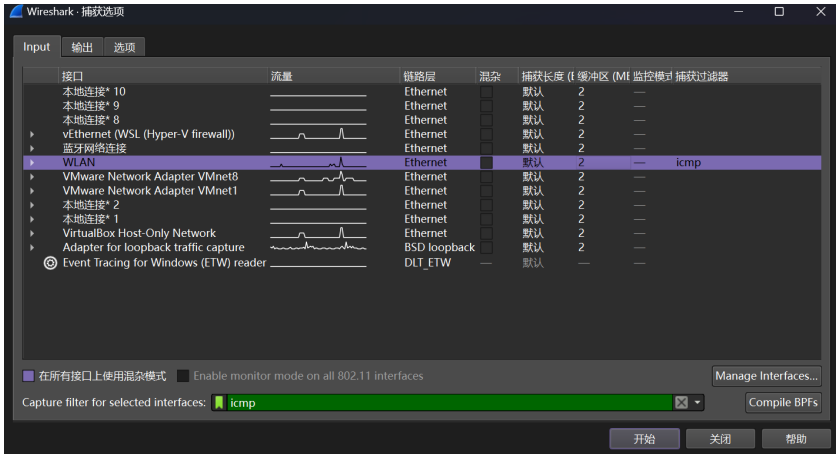

图 2: 设置 Wireshark 捕获过滤器

回到 cmd，再次使用 `ping www.baidu.com` 发起 ICMP 请求

```
C:\Users\GHOST>ping www.baidu.com

正在 Ping www.a.shifen.com [182.61.200.6] 具有 32 字节的数据:
来自 182.61.200.6 的回复: 字节=32 时间=30ms TTL=47
来自 182.61.200.6 的回复: 字节=32 时间=35ms TTL=47
来自 182.61.200.6 的回复: 字节=32 时间=29ms TTL=47
来自 182.61.200.6 的回复: 字节=32 时间=32ms TTL=47

182.61.200.6 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 29ms, 最长 = 35ms, 平均 = 31ms

C:\Users\GHOST>
```

图 3: 再次发起 ICMP 请求

回到 `Wireshark`，停止捕获，得到的结果如下图所示：

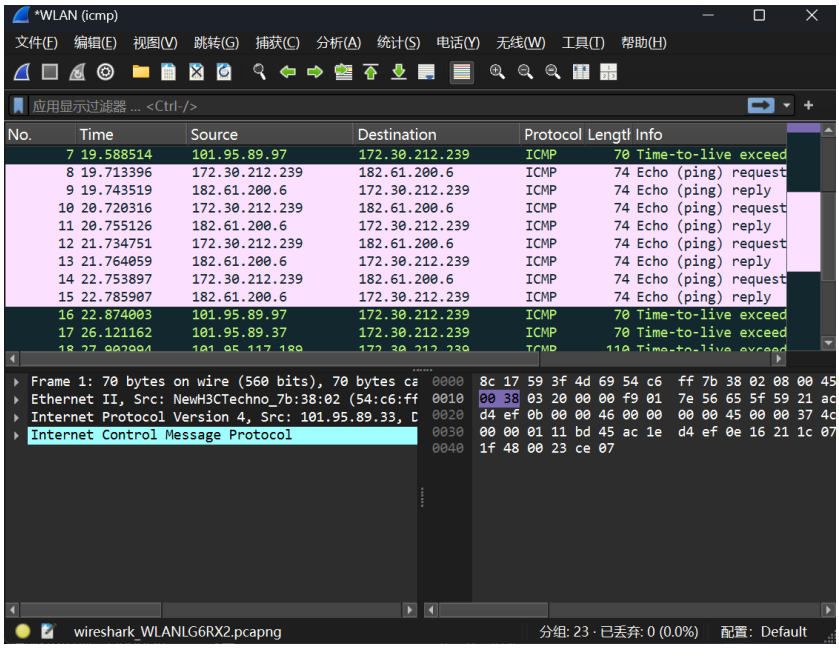

图 4: 捕获结果

4.2 分析以太网的帧

点击捕获到的数据包，选择 **Ethernet II**，可以看到以太网的帧结构如下图所示：

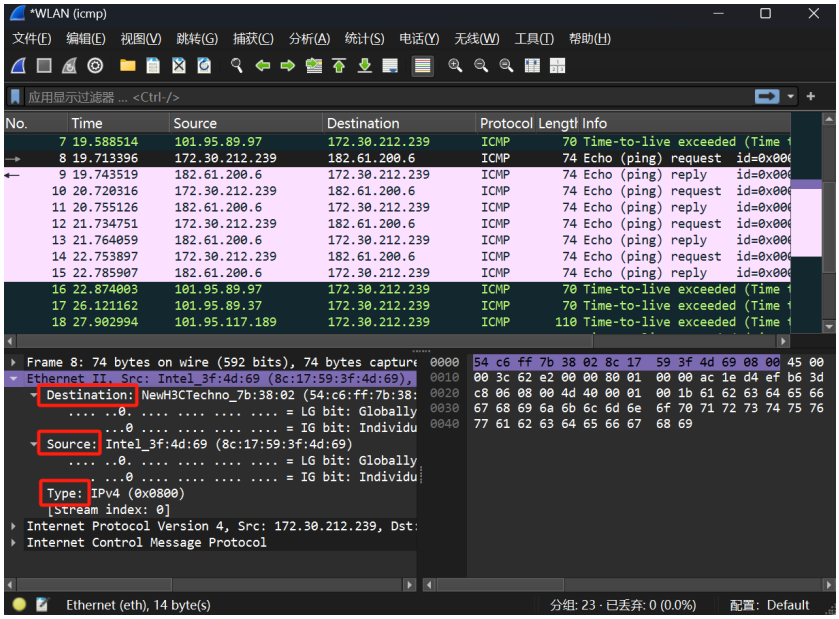

图 5: 以太网的帧结构

以太网的帧结构中包含了：目的地址 **Destination**, 源地址 **Source**, 类型 **Type** 等内容。其中目的地址和源地址都是 **6** 个字节，类型是 **2** 个字节。

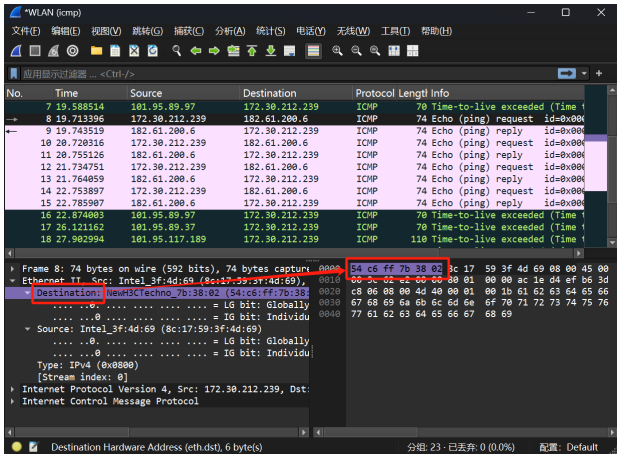

图 6: Destination

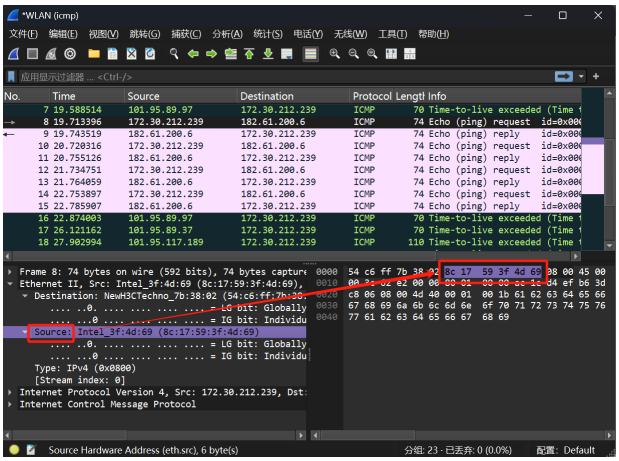

图 7: `Source`

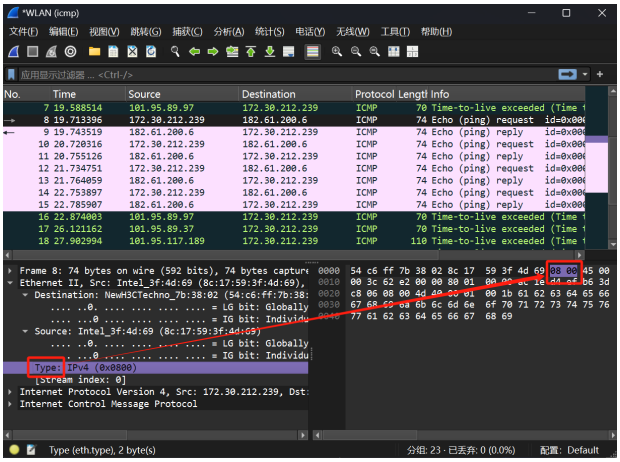

图 8: Type

再通过观察，找到 checksum 内容为 correct：

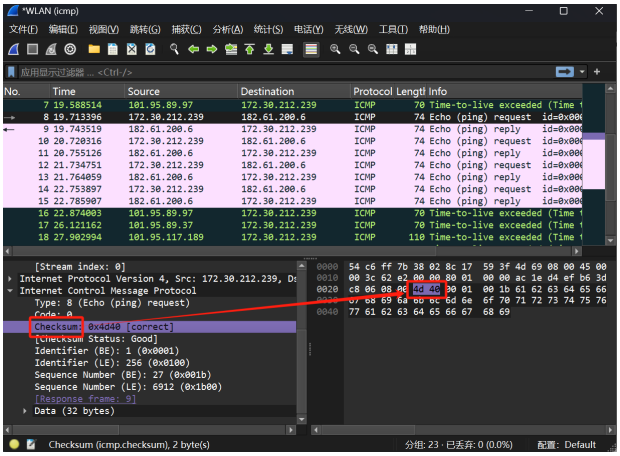

图 9: checksum

由此画出来的帧结构如图所示：

图 10: 以太网帧结构

4.3 分析以太网的地址范围

再结合以下信息：

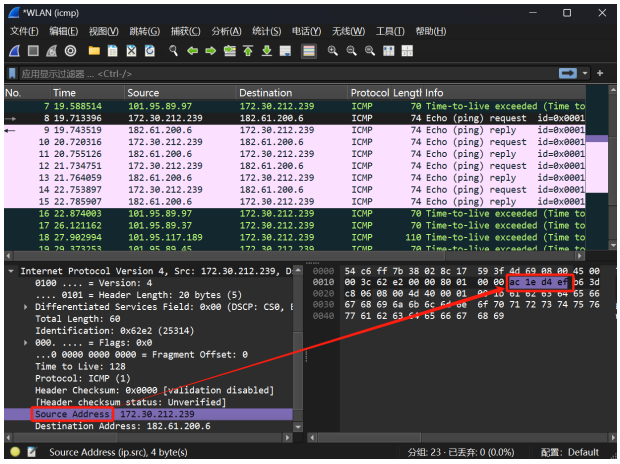

图 11: source 的 IP

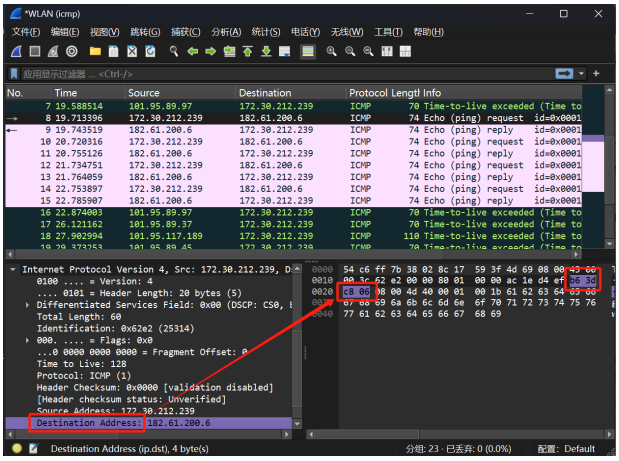

图 12: `destination` 的 IP

根据上面分析得到的以太网帧结构，我们可以得知本机 MAC 地址为 8c:17:59:3f:4d:69，IP 地址为 172.30.212.239，路由器 MAC 地址为 54:c6:ff:7b:38:02，目标 IP 地址为: 182.61.200.6。

可以作出如下的关系图：

图 13: 以太网地址范围关系图

4.4 分析以太网的广播帧

启动 `Wireshark`，在菜单栏捕获→选项重的设置，选择已连接的以太网，设置捕获过滤器为 `ether multicast`，然后开始捕获以太网的广播帧。

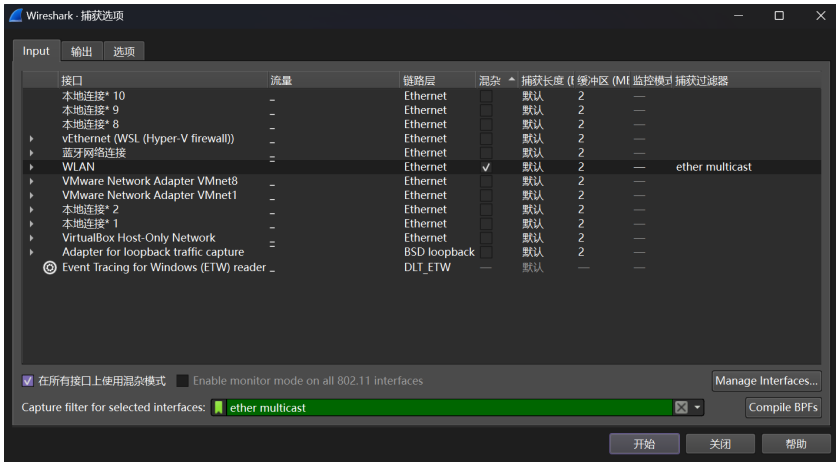

图 14: 设置 Wireshark 捕获过滤器

持续捕获一段时间之后，捕获结果如下图所示：

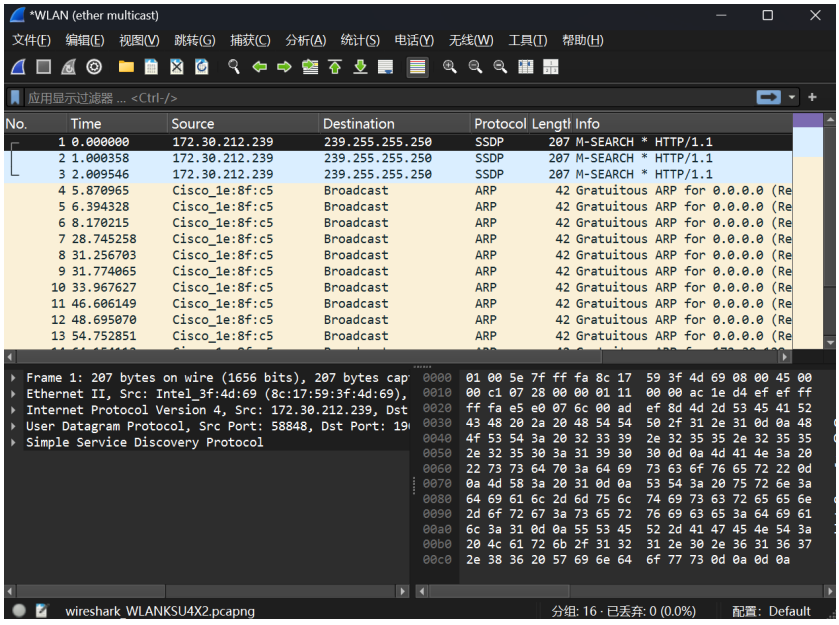

图 15: 捕获结果

选择其中的一个广播帧数据包，如下图所示：

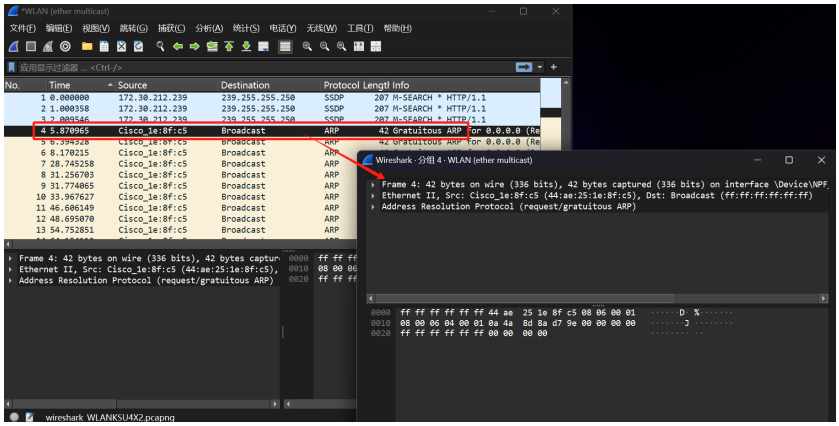

图 16: 广播帧数据包

接下来开始分析问题：

1. 以太网广播帧的地址是什么, 以标准的形式写在 Wireshark 上显示？

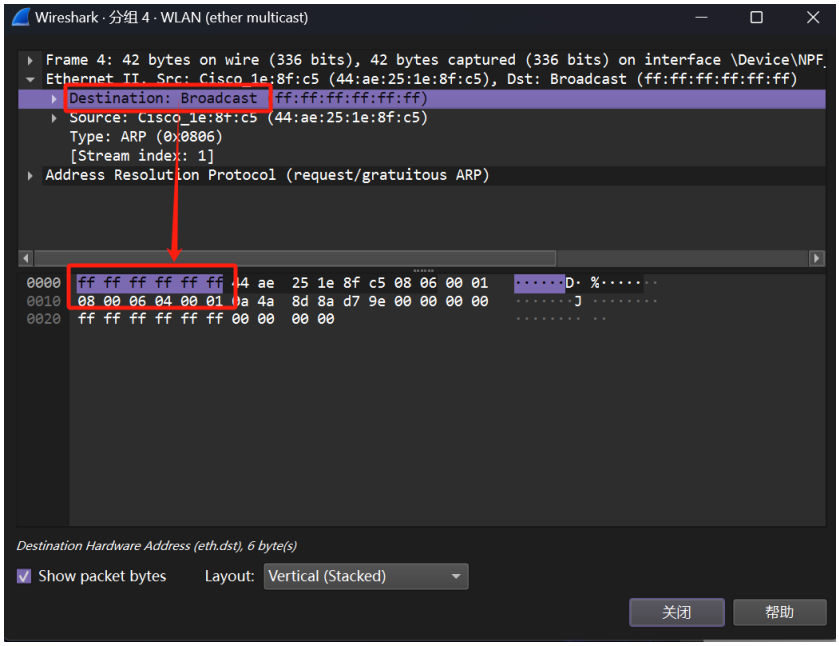

图 17: 广播帧地址

可以看出：广播帧的地址为 ff:ff:ff:ff:ff:ff。

2. 哪几个比特位的以太网地址是用来确定是单播或多播/广播？

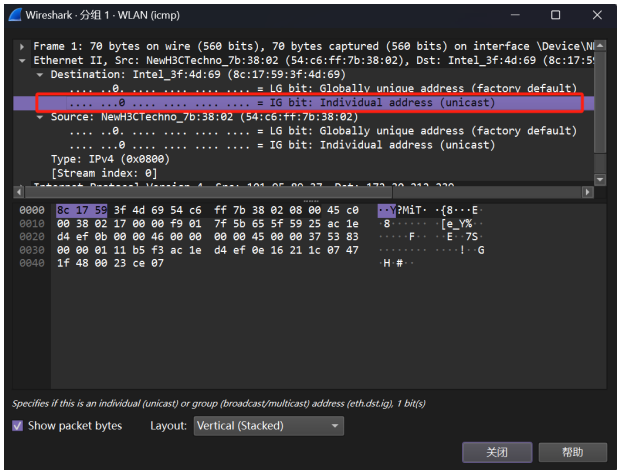

图 18: 单播帧

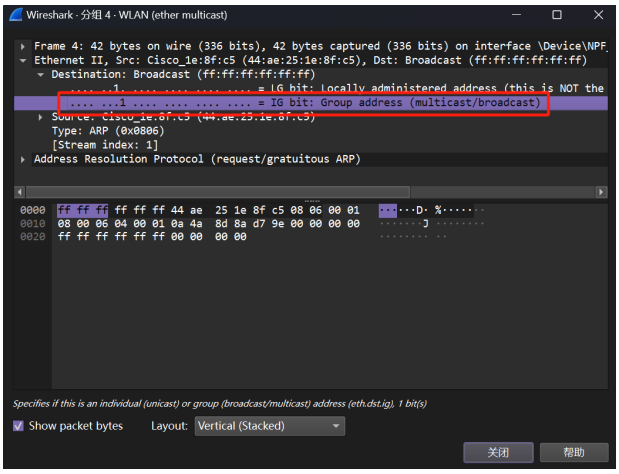

图 19: **广播帧**

对比单播帧和广播帧（如上图所示）可以看出，以太网地址的第一个字节的最后一位（即第八位）为 1，所以可以确定是多播/广播。

4.5 问题讨论

由于抓包了很久一直都没有抓到以太网的帧，所以这里直接使用实验手册里的那份数据来分析。

设置捕获过滤器为 llc，捕获以太网的帧，如下图所示：

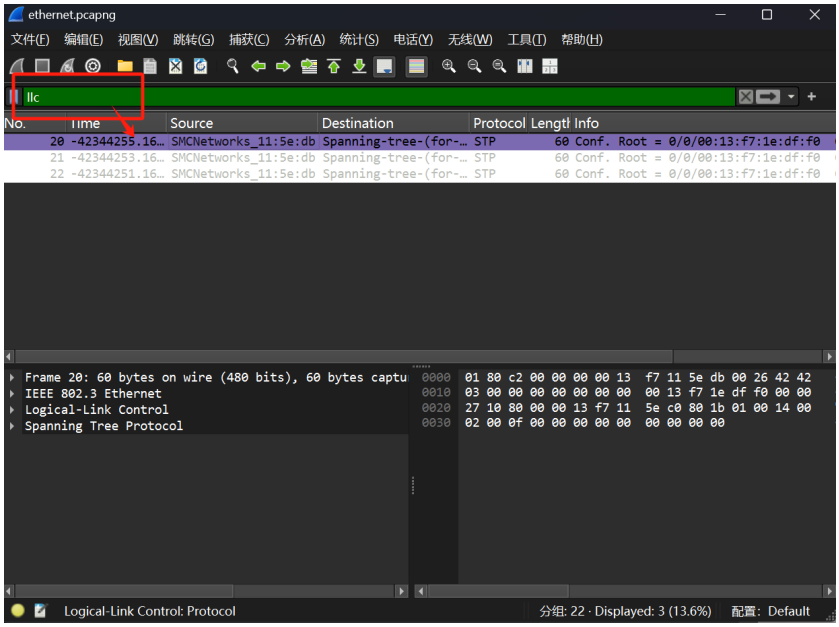

图 20: 捕获 IEEE802.3 以太网的帧

1) How long are the combined IEEE 802.3 and LLC headers compared to the DIX Ethernet headers? You

can use Wireshark to work this out. Note that the Trailer/Padding and Checksum may be shown as part of the header, but they come at the end of the frame.

根据下图所显示：可以看到：**IEEE 802.3** 头部长度为 14 字节，**LLC** 头部长度为 3 字节（**DIX** 以太网头部长度为 14 字节）

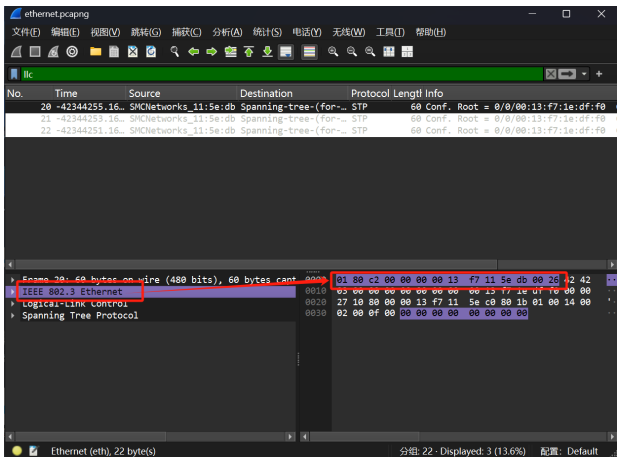

图 21: IEEE802.3 头部

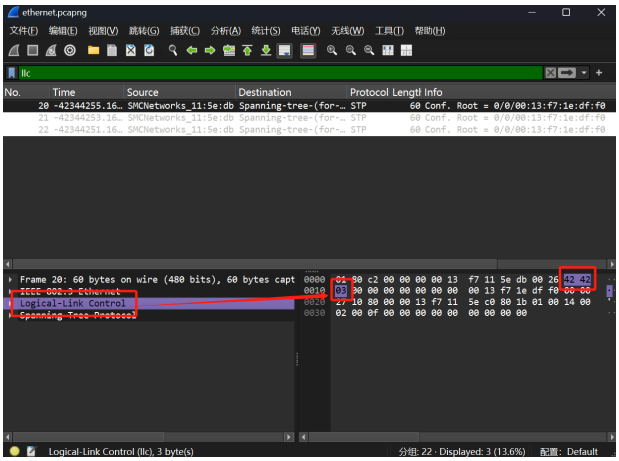

图 22: LLC 头部

2) How does the receiving computer know whether the frame is DIX Ethernet or IEEE 802.3? Hint: you may need to both use Wireshark to look at packet examples and read your text near where the Ethernet formats are described.

根据 **Type/Length** 字段，如果该字段的值小于或等于 1500，则表示 **Length**，为 **IEEE 802.3**，否则表示 **Type**，为 **DIX** 以太网。

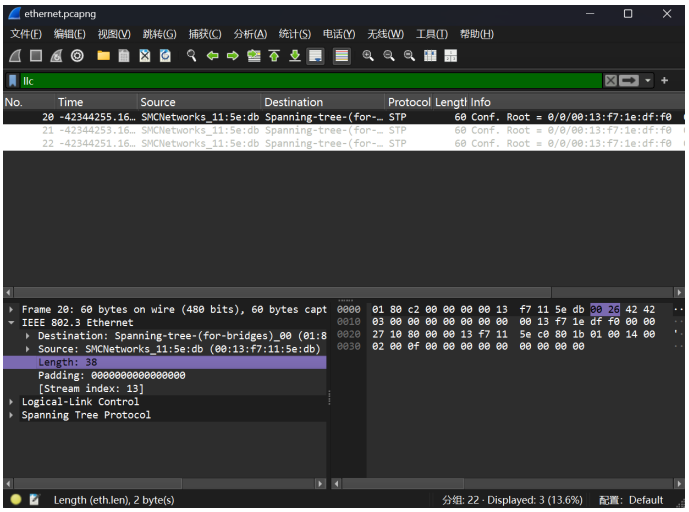

图 23: Length 字段

3) If IEEE 802.3 has no Type field, then how is the next higher layer determined? Use Wireshark to look for the demultiplexing key.

LLC 头中的 **DSAP** 字段可以指示上层协议。例如，此处 **DSAP** 字段为 **0x42**，则表示上层协议为 **STP**。

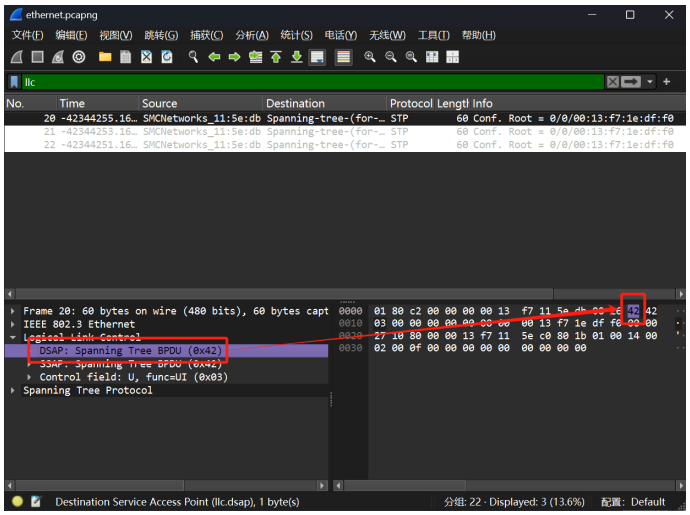

图 24: DSAP 字段

5 实验结果总结

本次实验中，通过使用 **Wireshark** 工具，我不仅学会了如何捕获以太网帧，还深入理解了以太网帧的结构。我详细分析了以太网地址的分配范围，并探究了广播帧的工作机制。此外，我对 **DIX** 以太网标准和 **IEEE 802.3** 标准之间的差异有了更清晰的认识。通过这些学习和探索，我增强了对网络通信协议的理解和应用能力。

6 附录

无