《计算机网络》实验报告

(2022~2023 **学年第一学期**)

实验名称：Wireshark 软件使用与 ARP 协议分析

学　　院：软件学院

姓　　名：李鹏

学　　号：2113850

指导老师：张圣林

2023 年 11 月 6 日

目录

实验一：Wireshark 软件使用与 ARP 协议分析

1 实验目的

本实验的目的是学习使用 Wireshark 软件，了解以太网 MAC 帧的基本结构，以及掌握 ARP 协议的特点和工作过程。具体目标如下：

(1) 学习 Wireshark 软件的基本使用，包括如何安装、启动、配置捕获网络数据包。

(2) 理解 ARP 协议的基本原理，包括如何将 IP 地址解析为 MAC 地址，以便在局域网内进行通信。

(3) 观察和分析 ARP 请求和响应数据包之间的交互过程，了解 MAC 地址的解析和映射过程。

(4) 掌握通过 Wireshark 软件提取和解析 ARP 数据包的方法，包括如何识别数据包中的源 MAC 地址、目标 MAC 地址等关键信息。

2 实验条件

本实验使用的硬件和软件环境如下：

硬件：一台连接到互联网的 PC 机，位于局域网中。

软件： Wireshark 4.0.10（最新版本，从 Wireshark 官网下载并安装）。

3 实验原理

3.1 Wireshark 简介

Wireshark 是一款开源的网络数据包分析工具，旨在捕获、解码和分析网络数据包，以帮助用户了解网络通信的细节和问题。它具有图形用户界面，支持多种操作系统，包括 Windows、Linux 和 macOS。Wireshark 能够抓取各种网络协议的数据包，并以可视化的方式呈现这些数据包的内容。

Wireshark 的基本使用包括安装和启动软件，选择网络接口以开始数据包捕获，使用过滤器筛选感兴趣的数据包，以及查看捕获的数据包的详细信息。此外，Wireshark 还提供了统计功能，用于分析网络活动的性能和模式。

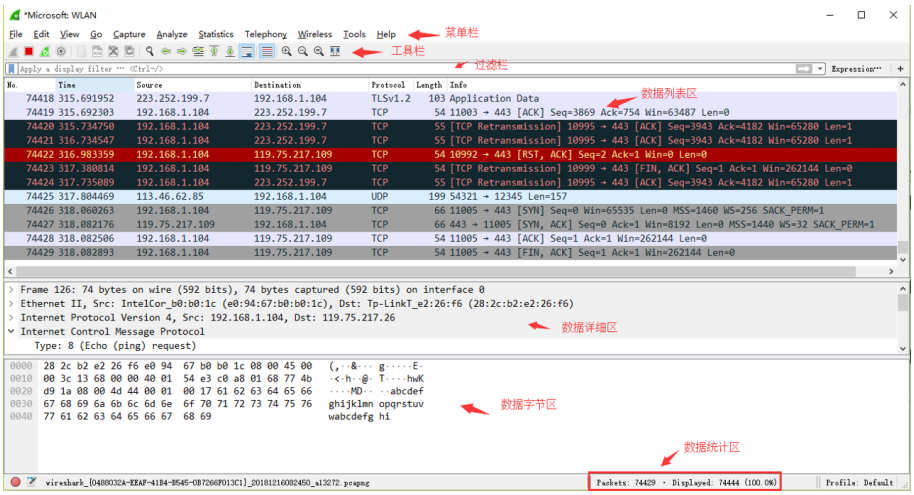

图 1: Wireshark

3.2 以太网 MAC 格式

本实验基于使用最广泛的有线局域网（以太网 Ethernet II），以太网的帧结构如 2所示。其中，MAC 地址（Media Access Control Address，媒体访问控制地址）或称物理地址（Physical Address），用于在网络中标识网卡。MAC 地址的长度为 48 位（6 个字节），通常表示为 12 个 16 进制数，例如：00-16-EA-AE-3C-40。

以太网（Ethernet）是一种广泛应用于局域网的有线网络技术。以太网数据帧的格式如下：

- 目标 MAC 地址：占 6 个字节，表示数据帧的接收者的 MAC 地址。
- 源 MAC 地址：占 6 个字节，表示数据帧的发送者的 MAC 地址。
- 类型字段：占 2 个字节，表示数据帧的类型，例如 IPv4 或 IPv6。
- 数据字段：包含数据的实际内容，长度可变。
- FCS 字段：帧校验序列，用于校验数据帧的完整性。

MAC 地址是数据链路层地址，通常由网络适配器（网卡）厂商分配，前 3 个字节是组织唯一标识符（OUI），后 3 个字节是设备的唯一标识符。MAC 地址的长度为 48 位。

前导字符	目的 MAC 地址	源 MAC 地址	类型	IP 数据报	帧校验
8 字节	6 字节	6 字节	2 字节	46-1500 字节	4 字节

图 2: 以太网帧格式

3.3 ARP 协议及数据包格式

地址解析协议 (Address Resolution Protocol, ARP) 的主要作用是将 IP 地址解析为 MAC 地址。当一个主机或网络设备要发送数据给目标主机时，必须知道目标主机的网络层地址（即 IP 地址），并且在数据链路层进行封装时，还需要知道目标主机（或下一跳路由器）的 MAC 地址。ARP 的工作原理如下：

(1) 主机 A 查找 ARP 缓存表，如果找到主机 B 的 MAC 地址，将数据包发送给主机 B。

(2) 如果主机 A 在 ARP 表中找不到主机 B 的 MAC 地址，它会广播 ARP 请求，请求其他主机帮助解析主机 B 的 MAC 地址。

(3) 主机 B 收到 ARP 请求后，将自己的 MAC 地址和 IP 地址发送给主机 A，主机 A 更新 ARP 表。

(4) 主机 A 收到 ARP 响应后，将主机 B 的 MAC 地址存储在 ARP 表中，以备将来的通信。

ARP 数据包的格式如下：

- 硬件类型：2 字节，表示网络接口类型，以太网为 1。
- 协议类型：2 字节，表示网络协议类型，IPv4 为 0x0800。
- 硬件地址长度：1 字节，表示 MAC 地址的长度，以太网为 6。
- 协议地址长度：1 字节，表示 IP 地址的长度，IPv4 为 4。
- 操作码：2 字节，表示 ARP 请求或 ARP 响应。
- 发送者 MAC 地址：6 字节，发送 ARP 请求或 ARP 响应的主机的 MAC 地址。
- 发送者 IP 地址：4 字节，发送 ARP 请求或 ARP 响应的主机的 IP 地址。

- 目标 MAC 地址：6 字节，ARP 响应时为主机 A 的 MAC 地址，ARP 请求时为全 0。
- 目标 IP 地址：4 字节，ARP 响应时为主机 A 的 IP 地址，ARP 请求时为主机 B 的 IP 地址。

4 实验内容

以下是本次实验的具体步骤：

4.1 Wireshark 软件的基本使用

(1) 安装 Wireshark 软件。

(2) 启动 Wireshark 并选择网络接口，开始捕获数据包。

(3) 在 Wireshark 中观察捕获的数据包，可自定义显示过滤器来筛选感兴趣的数据包。

(4) 导出特定的数据包，以便后续分析。

4.2 观察 MAC 地址

(1) 启动 Wireshark 并开始捕获数据包。

(2) 打开命令行窗口，分别使用 ping 命令 ping 网关和同一子网内的一台主机。

(3) 在 Wireshark 中观察捕获的数据包，重点关注以太网帧的目标 MAC 地址和源 MAC 地址。

(4) 确认 MAC 地址类型、OUI 信息以及 I/G 和 G/L 位的含义。

4.3 分析以太网帧结构

(1) 选择一个捕获的数据包，点击以太网 II 部分以展开其详细信息。

(2) 查看以太网帧的各个字段，包括目标 MAC 地址、源 MAC 地址、类型字段、数据字段和 FCS 字段。

(3) 分析数据包的各个字段的含义和作用。

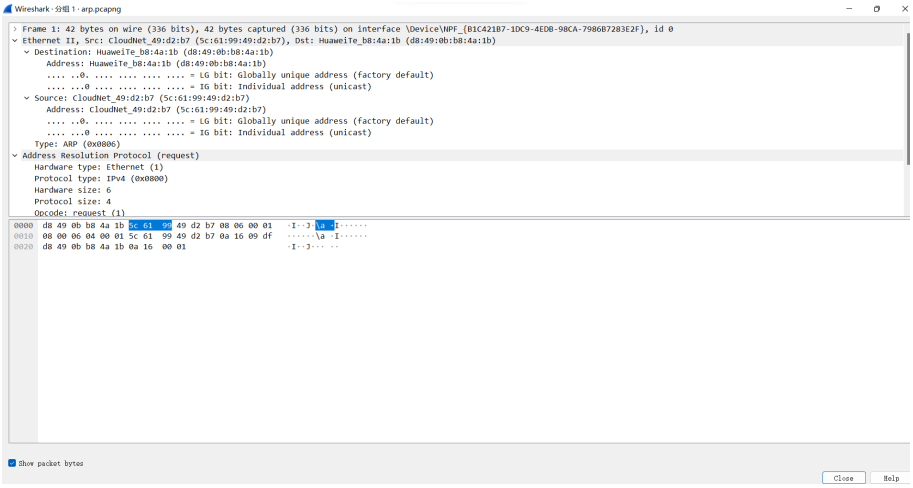

图 3: 以太网帧分析

以太网帧结构分析: 如图 3是一个以太网帧数据包，对其进行分析如下：

```
1 0000    d8 49 0b b8 4a 1b 5c 61 99 49 d2 b7 08 06 00 01
2 0010    08 00 06 04 00 01 5c 61 99 49 d2 b7 0a 16 09 df
3 0020    d8 49 0b b8 4a 1b 0a 16 00 01
```

前导码 (Preamble): 前导码是一个 7 字节的字段（d8 49 0b b8 4a 1b 5c），用于同步接收端的时钟。在以太网中，前导码指示数据包的开始。

目标 MAC **地址** (Destination MAC): 目标 MAC 地址是 6 字节的字段（61 99 49 d2 b7 08），表示数据包应该传送到的目标设备的 MAC 地址。在这个例子中，目标 MAC 地址为 ‘61:99:49:d2:b7:08‘。

源 MAC **地址** (Source MAC): 源 MAC 地址也是 6 字节的字段（06 04 00 01 5c 61），表示数据包的发送者的 MAC 地址。在这个例子中，源 MAC 地址为 ‘06:04:00:01:5c:61‘。

帧类型 (Frame Type): 帧类型字段是 2 字节（99 49），指示了以太网帧中的上层协议。在这个例子中，帧类型为 ‘9949‘。

数据: 接下来的部分是实际数据部分，其中包括 ARP 请求的内容。

CRC **校验码**: CRC 校验码是 4 字节的字段（0a 16 09 df），用于检查数据包在传输过程中是否损坏。

FCS (Frame Check Sequence): FCS 是 4 字节的字段（d8 49 0b b8），与 CRC 校验码一起用于数据包的完整性检查。

ARP **帧分析**:

目标 MAC 地址: d8:49:0b:b8:4a:1b
源 MAC 地址: 5c:61:99:49:d2:b7
帧类型: 0806 (ARP 帧)

ARP 帧: 硬件类型: 0800 (以太网) 协议类型: 0806 (IPv4) 硬件地址长度: 06 (MAC 地址长度为 6 字节) 协议地址长度: 04 (IPv4 地址长度为 4 字节) 操作码: 0001 (ARP 请求) 发送者 MAC 地址: 5c:61:99:49:d2:b7 发送者 IP 地址: 0a.16.09.df 目标 MAC 地址: 00:00:00:00:00:00 目标 IP 地址: 0a:16:09:df

这个以太网帧包含一个 ARP 请求,发送者的 MAC 地址和 IP 地址是 ‘5c:61:99:49:d2:b7‘和 ‘0a.16.09.df‘。目标 MAC 地址通常为 0，因为 ARP 请求不包括目标 MAC 地址，而是在 ARP 响应中被填充。

4.4 ARP 协议分析

(1) 使用命令行窗口执行以下命令：arp -d（清空本机 ARP 缓存），然后启动 Wireshark。

(2) 使用 ping 命令 ping 同一子网内的另一台主机，观察 Wireshark 中捕获的 ARP 请求和响应数据包。

(3) 分析 ARP 请求和响应的过程，包括各个字段的含义和作用。

(4) 使用命令行窗口执行 arp -d（再次清空本机 ARP 缓存），然后 ping 与本机不在同一子网内的 IP 地址或域名，观察 Wireshark 中捕获的 ARP 数据包。

(5) 分析不在同一子网内的 ARP 请求和响应的过程，包括各个字段的含义和作用。

5 实验结论及心得体会

通过本次实验，我学习了 Wireshark 软件的基本使用方法，包括软件的安装、配置和捕获数据包的操作；了解了以太网 MAC 帧的结构，MAC 地址的类型，各字段的含义；明白了 ARP 协议的工作原理和数据包格式，通过清空 ARP 缓存来触发 ARP 请求。

在本次实验中，我遇到了一些问题，通过查阅资料和同学讨论，最后解决了。这次实验让我更深入地了解了以太网帧格式，数据包的捕获，对课上讲述的知识有了体会。

6 代码截图

以下是实验中的部分截图：

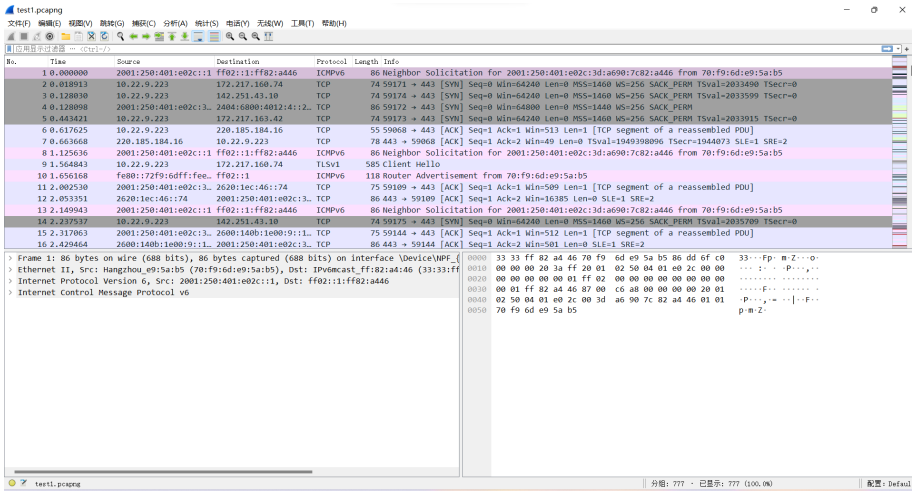

图 4: 使用 Wireshark 捕获数据包

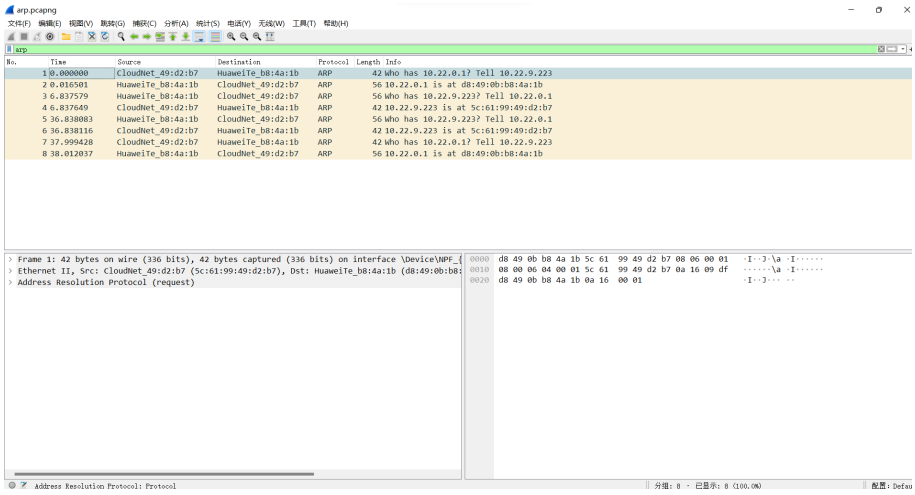

图 5: 过滤筛选 arp 协议

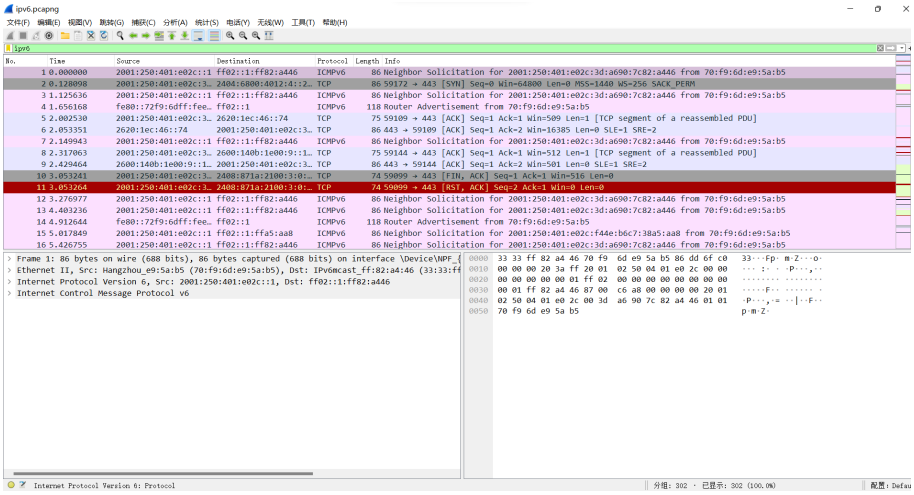

图 6: 过滤筛选 ipv6 协议

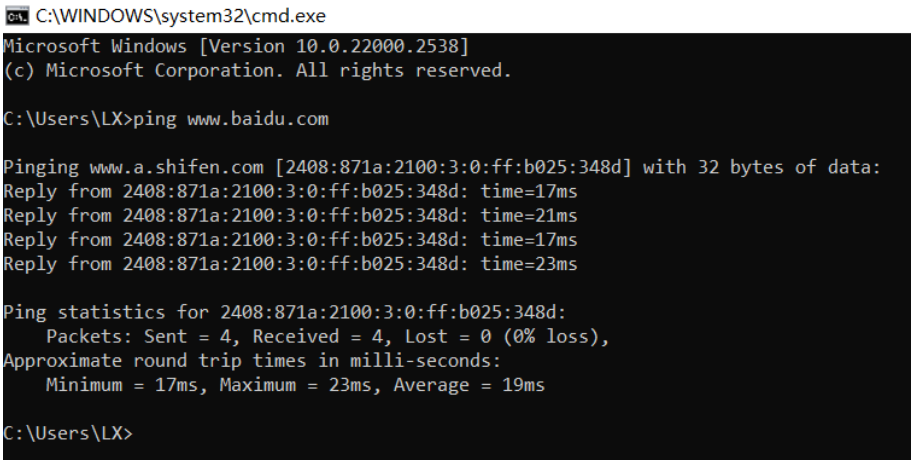

图 7: ping 命令的使用

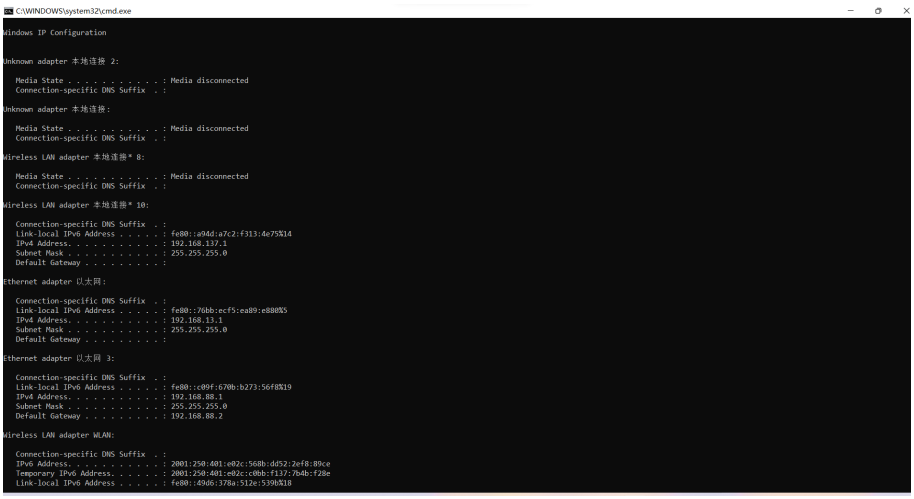

图 8: 查看本机 IP

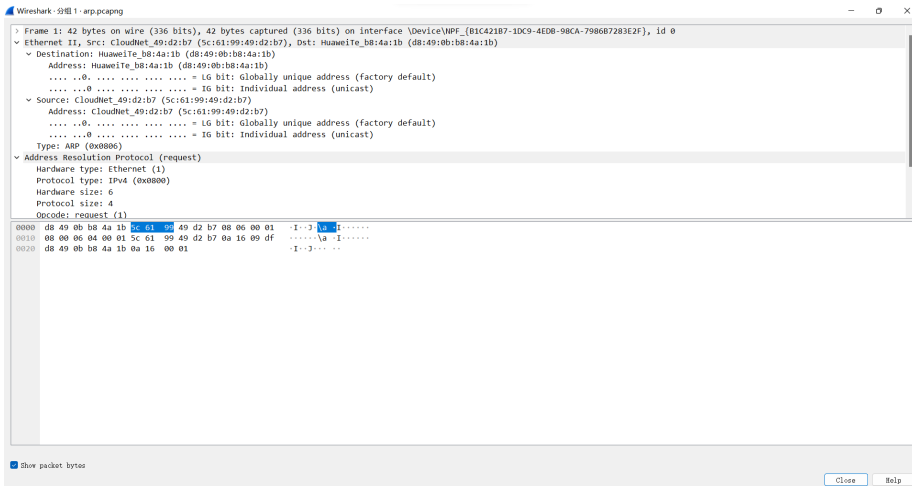

图 9: 分析以太网帧

7 核心代码/核心过滤语法

本次实验的核心代码和过滤语法如下:

- 捕获特定 IP 的数据包:

```
ip.addr == 10.22.34.130
```

- 捕获特定源或目标 MAC 地址的数据包:

```
eth.src == 00-50-56-C0-00-08
```

- 捕获 ARP 数据包:

```
arp
```

- 捕获 HTTP 数据包:

```
http
```

- 捕获 TCP 数据包：

```
tcp
```

- 捕获 UDP 数据包：

```
udp
```

- 使用逻辑运算符 AND 和 OR 组合过滤条件，例如捕获源 IP 为 192.168.1.1 或目标 IP 为 192.168.1.2 的 HTTP 数据包：

```
(ip.src == 192.168.1.1 and http) or (ip.dst == 192.168.1.2 and http)
```

- 捕获特定端口号的数据包（例如，HTTP 端口 80）：

```
port == 80
```

- 捕获特定源 IP 和目标 IP 之间的数据包：

```
ip.src == 192.168.1.1 and ip.dst == 192.168.1.2
```