华东师范大学软件学院实验报告

实验课程：计算机网络实践	年级：2023 级本科	实验成绩：
实验名称：Lab 4 ARP	姓名：张梓卫	
实验编号：（4）	学号：10235101526	实验日期：2024/12/13
指导老师：刘献忠	组号：	实验时间：2 课时

目录

一 实验目的

该实验是课程《计算机网络实践》的第四次实验，全名《ARP》，目标如下：

- 1. 学会通过 Wireshark 获取 ARP 消息
- 2. 掌握 ARP 数据包结构
- 3. 掌握 ARP 数据包各字段的含义
- 4. 了解 ARP 协议适用领域

二 实验内容与实验步骤

- 使用管理员权限打开命令行
- 输入 ipconfig /all，可以获得本地计算机的物理地址
- 输入 netstat -r，可以获得本机路由表
- 输入 arp -a，可以查看 ARP cache
- 输入 arp -d，可以清空 ARP cache

- 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 ARP，将混杂模式设为关闭，点击开始
- 清空 ARP cache

三 实验环境

使用 Wireshark v4.2.5, Windows 11 Pro, Wget Tools 进行实验。
实验报告使用 LaTeX 进行撰写，使用 Vim 编辑器进行文本编辑。

四 实验过程与分析

预备知识

虽然在网络层转发分组使用的是 IP 地址，但是最终我们还是要使用 MAC 地址在实际的网络线路上传输数据。所以光知道目标的 IP 地址是没有用的。ARP 就能够把 IP 地址转换为 MAC 地址。

每一台主机都有一个 ARP 缓存表，里面记录着 IP 地址和 MAC 地址的映射关系，ARP 的职责即动态地维护该表。

查表会有两种情况：1、命中；2、缺失：（1）广播 ARP 请求；（2）更新 ARP 缓存。

ARP 报文结构，在 PPT 上如下所示：

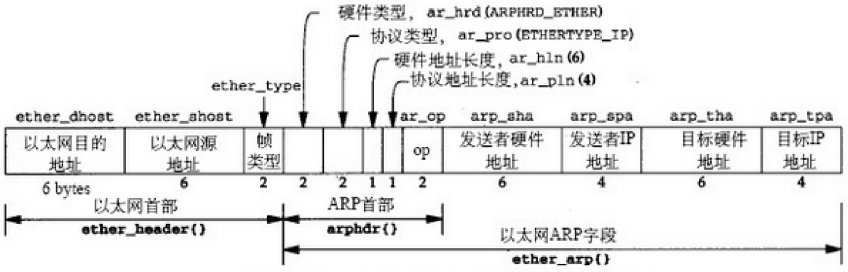


图 1: ARP 报文结构

1 Step 1: Capture a Trace

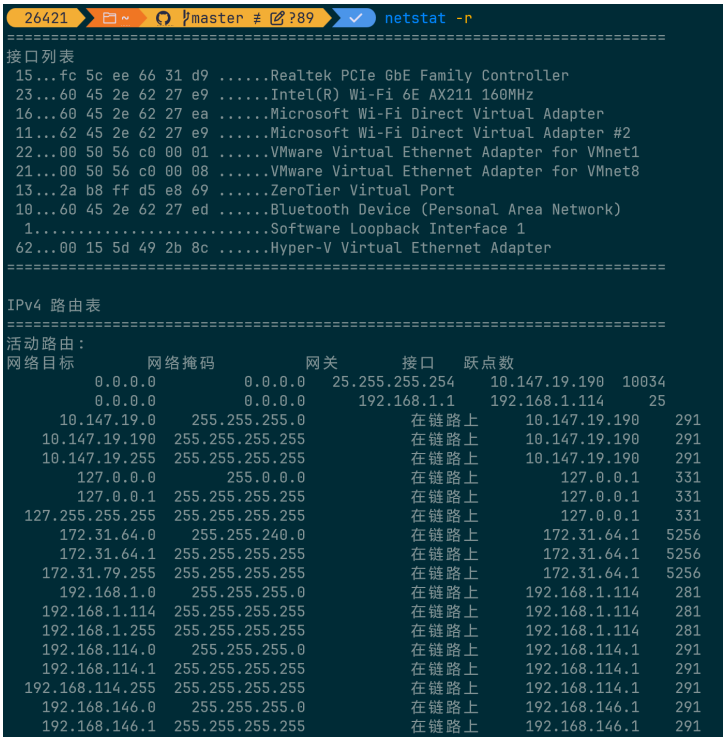
1 .1 使用 ipconfig /all 查找地址

图 2: 使用 ipconfig /all 查找地址

通过图中内容，可以分析得到，本机的 IP 地址为 192.168.1.114，MAC 地址为 FC-5C-EE-66-31-D9。

1 .2 使用 netstat 和 arp 命令

使用 netstat -r 查看路由表。



```
26421 ~ master ≠ ?89 netstat -r
===========================================================================
接口列表
 15...fc 5c ee 66 31 d9 ......Realtek PCIe GbE Family Controller
 23...60 45 2e 62 27 e9 ......Intel(R) Wi-Fi 6E AX211 160MHz
 16...60 45 2e 62 27 ea ......Microsoft Wi-Fi Direct Virtual Adapter
 11...62 45 2e 62 27 e9 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 22...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1
 21...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8
 13...2a b8 ff d5 e8 69 ......ZeroTier Virtual Port
 10...60 45 2e 62 27 ed ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
 62...00 15 5d 49 2b 8c ......Hyper-V Virtual Ethernet Adapter
===========================================================================

IPv4 路由表
===========================================================================
活动路由：
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0   25.255.255.254    10.147.19.190  10034
          0.0.0.0          0.0.0.0      192.168.1.1    192.168.1.114     25
       10.147.19.0    255.255.255.0         在链路上     10.147.19.190    291
     10.147.19.190  255.255.255.255         在链路上     10.147.19.190    291
     10.147.19.255  255.255.255.255         在链路上     10.147.19.190    291
        127.0.0.0        255.0.0.0         在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255         在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255         在链路上         127.0.0.1    331
      172.31.64.0    255.255.240.0         在链路上       172.31.64.1   5256
      172.31.64.1  255.255.255.255         在链路上       172.31.64.1   5256
    172.31.79.255  255.255.255.255         在链路上       172.31.64.1   5256
      192.168.1.0    255.255.255.0         在链路上     192.168.1.114    281
    192.168.1.114  255.255.255.255         在链路上     192.168.1.114    281
    192.168.1.255  255.255.255.255         在链路上     192.168.1.114    281
    192.168.114.0    255.255.255.0         在链路上     192.168.114.1    291
    192.168.114.1  255.255.255.255         在链路上     192.168.114.1    291
  192.168.114.255  255.255.255.255         在链路上     192.168.114.1    291
    192.168.146.0    255.255.255.0         在链路上     192.168.146.1    291
    192.168.146.1  255.255.255.255         在链路上     192.168.146.1    291
```

图 3: 使用 netstat -r 查看路由表

从图中可以看到，默认的网关 IP 为 192.168.1.1。

1 .3 设置 Wireshark 捕获 ARP 数据包

将过滤器设置为 ARP，将混杂模式关闭，点击开始捕获。

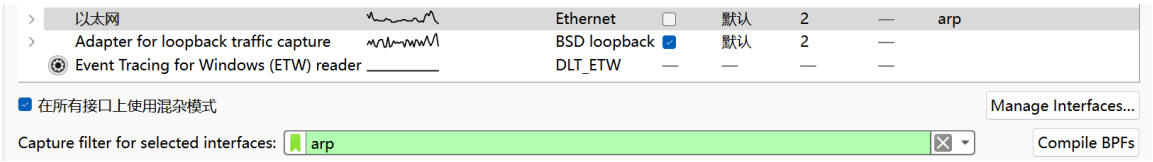


图 4: 设置 Wireshark 的过滤器

设置 rename 选项为开启：

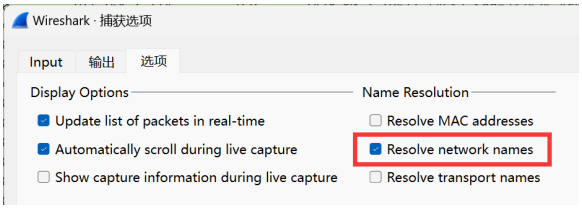
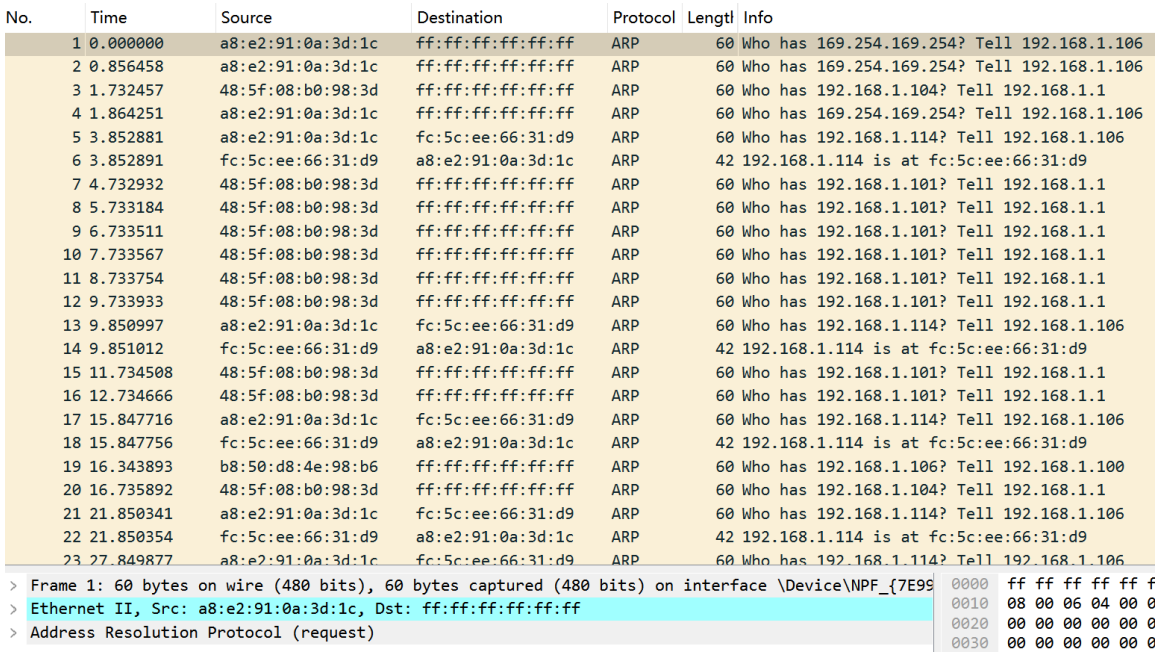


图 5: Wireshark Rename

在 Wireshark 中可以看到很多协议为 ARP 的数据包：



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	a8:e2:91:0a:3d:1c	ff:ff:ff:ff:ff:ff	ARP	60	Who has 169.254.169.254? Tell 192.168.1.106
2	0.856458	a8:e2:91:0a:3d:1c	ff:ff:ff:ff:ff:ff	ARP	60	Who has 169.254.169.254? Tell 192.168.1.106
3	1.732457	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.104? Tell 192.168.1.1
4	1.864251	a8:e2:91:0a:3d:1c	ff:ff:ff:ff:ff:ff	ARP	60	Who has 169.254.169.254? Tell 192.168.1.106
5	3.852881	a8:e2:91:0a:3d:1c	fc:5c:ee:66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
6	3.852891	fc:5c:ee:66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
7	4.732932	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
8	5.733184	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
9	6.733511	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
10	7.733567	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
11	8.733754	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
12	9.733933	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
13	9.850997	a8:e2:91:0a:3d:1c	fc:5c:ee:66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
14	9.851012	fc:5c:ee:66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
15	11.734508	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
16	12.734666	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.101? Tell 192.168.1.1
17	15.847716	a8:e2:91:0a:3d:1c	fc:5c:ee:66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
18	15.847756	fc:5c:ee:66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
19	16.343893	b8:50:d8:4e:98:b6	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.106? Tell 192.168.1.100
20	16.735892	48:5f:08:b0:98:3d	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.1.104? Tell 192.168.1.1
21	21.850341	a8:e2:91:0a:3d:1c	fc:5c:ee:66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
22	21.850354	fc:5c:ee:66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
23	27.849877	a8:e2:91:0a:3d:1c	fc:5c:ee:66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106

```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{7E99
> Ethernet II, Src: a8:e2:91:0a:3d:1c, Dst: ff:ff:ff:ff:ff:ff
> Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff f
0010  08 00 06 04 00 0
0020  00 00 00 00 00 0
0030  00 00 00 00 00 0
```

图 6: Wireshark 捕获的 ARP 数据包

1 .4 促使 ARP 表更新

使用管理员权限打开命令行，打开 Wireshark 开始捕获.

输入命令 **arp -d** 清空 ARP 缓存，再使用 **arp -d 192.168.1.1** 来清空和网关相关的 ARP 缓存。然后使用命令 **arp -a** 检查缓存是否清空成功。

接下来浏览任意的网页，来促使 ARP 表更新，在 Wireshark 中捕获了 ARP 报文以后，点击停止，开始分析。

我打开了 www.baidu.com，使用 Wireshark 抓包完成后，再次使用 **arp -a** 指令，可以发现 ARP 路由表发生了变化：

```
接口: 192.168.1.114 --- 0xf
  Internet 地址         物理地址              类型
  192.168.1.1           48-5f-08-b0-98-3d     动态
  192.168.1.106         a8-e2-91-0a-3d-1c     动态
  192.168.1.112         e8-9c-25-1e-ec-30     动态
  192.168.1.255         ff-ff-ff-ff-ff-ff     静态
  224.0.0.2             01-00-5e-00-00-02     静态
  224.0.0.167           01-00-5e-00-00-a7     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  239.192.152.143       01-00-5e-40-98-8f     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态

接口: 192.168.114.1 --- 0x15
  Internet 地址         物理地址              类型
  192.168.114.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态

接口: 192.168.146.1 --- 0x16
  Internet 地址         物理地址              类型
  192.168.146.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态
```

图 7: ARP 表更新

2 Step 2: Inspect The Trace

2 .1 使用 MAC 地址过滤器

在 Wireshark 中使用 **eth.addr==fc:5c:ee:66:31:d9** 来设置过滤器，以找出与自己的 MAC 地址相关的 ARP 报文。

Notice：ARP 报文包括请求报文和应答报文。

eth.addr==fc:5c:ee:66:31:d9

No.	Time	Source	Destination	Protocol	Lengtl	Info
892	303.846342	a8:e2:91:0a:3d:1c	LCFCElectron_66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
893	303.846356	LCFCElectron_66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
895	305.020306	LCFCElectron_66:31:d9	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.114
896	305.021079	TpLinkTechno_b0:98:3d	LCFCElectron_66:31:d9	ARP	60	192.168.1.1 is at 48:5f:08:b0:98:3d
902	309.846512	a8:e2:91:0a:3d:1c	LCFCElectron_66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
903	309.846522	LCFCElectron_66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
910	315.845004	a8:e2:91:0a:3d:1c	LCFCElectron_66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
911	315.845013	LCFCElectron_66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
916	321.846833	a8:e2:91:0a:3d:1c	LCFCElectron_66:31:d9	ARP	60	Who has 192.168.1.114? Tell 192.168.1.106
917	321.846856	LCFCElectron_66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	192.168.1.114 is at fc:5c:ee:66:31:d9
918	324.398454	LCFCElectron_66:31:d9	a8:e2:91:0a:3d:1c	ARP	42	Who has 192.168.1.106? Tell 192.168.1.114

图 8: 使用 MAC 地址过滤器

2 .2 从本机发送的包

查看从本地发送出去的 request 帧。

895 305.020306 LCFCElectron_66:31:d9 Broadcast ARP 42 Who has 192.168.

```
> Frame 895: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7E
v Ethernet II, Src: LCFCElectron_66:31:d9 (fc:5c:ee:66:31:d9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: LCFCElectron_66:31:d9 (fc:5c:ee:66:31:d9)
    Type: ARP (0x0806)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: LCFCElectron_66:31:d9 (fc:5c:ee:66:31:d9)
    Sender IP address: 192.168.1.114 (192.168.1.114)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1 (192.168.1.1)
```

图 9: 从本机发送的 request 帧

2 .3 绘制 ARP 协议的结构

我们选取刚刚获取到的 request 请求，我把 Wireshark 中显示的结构和对应的字段整合到一起，如下图所示：

```
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: LCFCElectron_66:31:d9 (fc:5c:ee:66:31:d9)
    Sender IP address: 192.168.1.114 (192.168.1.114)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1 (192.168.1.1)
ff ff ff ff ff ff fc 5c  ee 66 31 d9 08 06 00 01   ·····\ ·f1·····
08 00 06 04 00 01 fc 5c  ee 66 31 d9 c0 a8 01 72   ·····\ ·f1····r
00 00 00 00 00 00 c0 a8  01 01                     ···· ··
```

图 10: Wireshark 捕获结果

然后用鼠标依次点击高亮字段，将每个字段用笔标注出来，如下图所示：

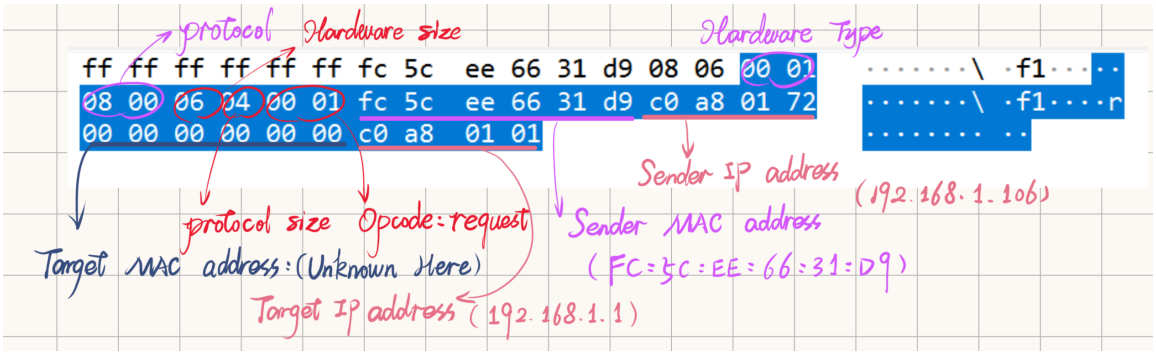


图 11: ARP 协议的结构

1、看以太网的头。

- Destination: Broadcast (FF:FF:FF:FF:FF:FF)，这是一个广播
- Source: FC:5C:EE:66:31:D9
- Type: 上一层协议是 ARP

2、看 ARP 的头

- Hardware type：硬件类型，0x01 表示在以太网上传输
- Protocol type：上一层协议类型
- Hardware size：硬件长度
- Protocol size：协议长度
- Opcode: 操作类型，1 为请求,2 为应答
- Sender MAC address：fc:5c:ee:66:31:d9，发送方 MAC 地址
- Sender lP address：192.168.1.114，发送方 IP
- Target MAC address：00:00:00:00:00:00，表示这是一个广播
- Target IP address：192.168.1.1，目标 IP

其实，这里就是先广播了自己的 MAC 地址和 IP 地址，并询问 IP 地址为 192.168.1.1 的路由器的 MAC 地址是什么。当目标路由器收到 ARP 请求信息，并且自己的 IP 地址和请求信息中的 IP 地址相匹配时，就会返回一个 ARP 回复

2 .4 ARP 的 Reply 帧

接下来观察紧随其后的 Reply 帧的信息：

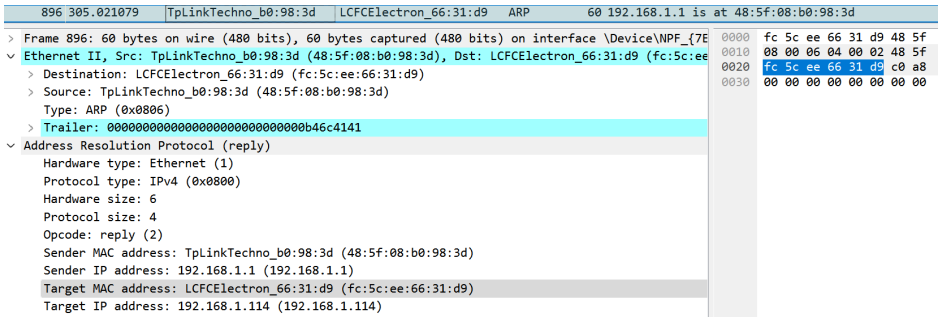


图 12: ARP Reply 帧

Reply 帧的每一个部分的字节长度和 Request 帧的结构相同。但在 Target MAC address 字节段不再全为 0，而是包含了目标 IP 地址路由器的 MAC 地址。除此之外，这里的 Opcode 字段也变成了 2，表示这是一个 Reply 帧。

3 Step 3: ARP Request and Reply

请求数据包应是广播的，而应答数据包是单播的。

注意到 Reply 帧中存在一个 Trailer 字段。它和传统的 Padding 字段有所不同。在以太网帧中，如果数据字段（Payload）长度小于 46 字节，以太网协议要求填充（Padding）字段，将帧的最小长度补充到 64 字节（含头部和 CRC 校验字段）。这是以太网规范的要求，用于确保帧的可靠性。

request 消息是在发送之前就在本地被截获下来，因此没有 padding 字段。但 reply 消息是网卡处收到的字段，因此有 padding 字段。

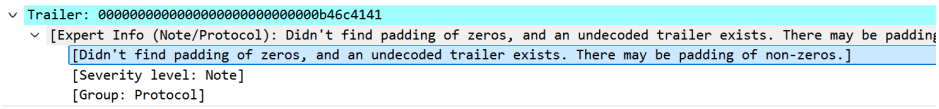


图 13: 非零 Padding 字段

3 .1 Trailer 的显示

截图中提到：

```
Didn't find padding of zeros, and an undecoded trailer exists.
```

它说明发现了一个未解析的 Trailer 字段，而不是传统的 Padding。

3 .2 Trailer 和 Padding 的区别

- **Padding**：是为满足最小帧长的零填充。
- **Trailer**：通常是附加信息（比如某些协议自定义的内容），不属于以太网的固定规范。

3 .3 绘制 Request 和 Reply 帧交互图

根据上述的分析，可以按照要求画出带有 Sender 与 Receiver 的 IP / MAC 的 ARP 帧交流图。

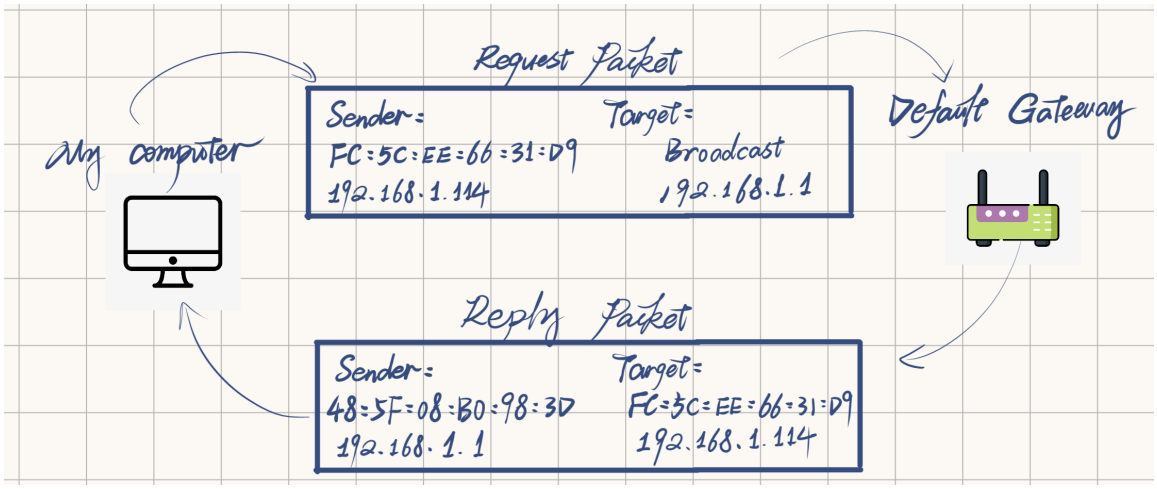


图 14: Request 和 Reply 帧交互图

4 课后思考题

什么样的操作码是用来表示一个请求？应答呢？

根据上面的分析和 Wireshark 抓包的结果，显而易见：请求操作码为 1，回复操作码为 2。

一个请求的 ARP 的报头有多大？应答呢？

都是 28 字节。直接看 Wireshark 的抓包结果，点击 Address Resolution Protocol (request/reply) 查看即可。

对未知目标的 MAC 地址的请求是什么值？

请求的目标 MAC 地址通常为全零，即 00:00:00:00:00:00

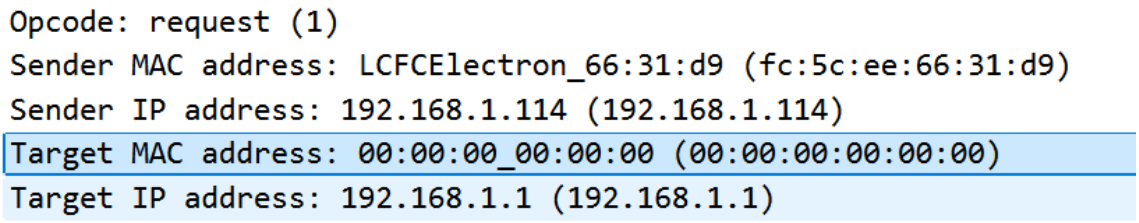


图 15: Question 3

什么以太网类型值说明 ARP 是更高一层的协议？

ARP 的以太网类型值为 0x0806，也是从 Wireshark 的抓包结果中可以看到：

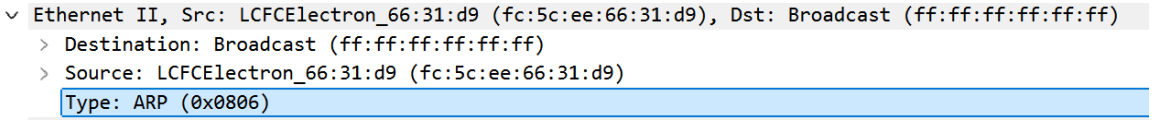


图 16: Question 4

ARP 应答是广播吗？

ARP 应答通常不广播。它使用其以太网广播直接发送给目标。也可以从最低字节的第一位看到这里的值为 0，表示为 unicast。

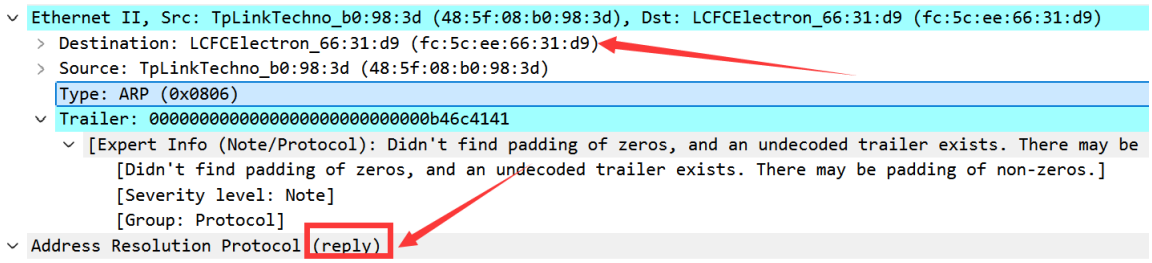


图 17: Question 5

5 更多的 ARP 报文

我们可以打开 lab4stu 中的 arp.pcap，查看更多的 ARP 报文：

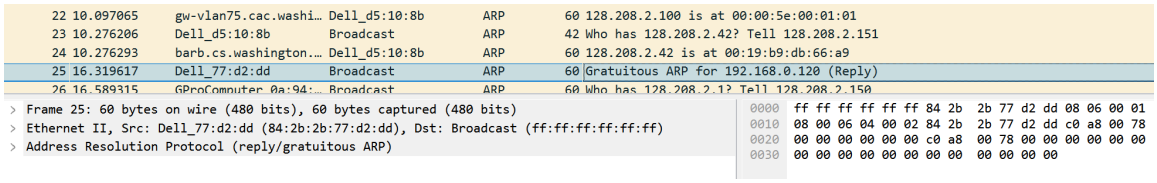


图 18: More

在阅读 Info 字段时，可以看到除了网关和本机计算机之间的 arp 请求通信，还有局域网内许多其他计算机和本机的 ARP 请求（因为是广播形式，所以我们可以收到）。当然，还有计算机发送的 arp 回复，告诉别的计算机我们的物理地址信息。

除此之外我们还可以看到有：

```
25  16.319617 Dell_77:d2:dd Broadcast ARP 60  Gratuitous ARP for 192.168.0.120 (Reply)
```

Gratuitous ARPs, （中文翻译为“无偿 ARP”或“任意 ARP”）是一种特殊的 ARP 报文。其主要特点和目的如下：

这类报文是自己的计算机发送的 arp 请求和答复，目的是为了确保没有其他人正在使用相同的 IP 地址。这类报文具有相同的发送方和目标 IP 地址，目的是检测 IP 冲突、通知其他设备更新 ARP 表或支持高可用性切换。它在网络配置和维护中起到了重要作用，虽然通常不会直接被用户感知，但在后台网络通信中广泛使用。

五 实验结果总结

通过本次 ARP 实验，我学会了在 Wireshark 中获得 ARP 数据包的方法，理解了 ARP(Address Resolution Protocol) 的运作方式，并了解了在 ARP 报文信息的结构及每一部分代表的含义。

其实 ARP 协议只是提供了一种方法，根据目的主机的 IP 地址，获得其 MAC 地址。这便是地址解析的过程。当然，我还通过运用 ipconfig、netstat、arp 等终端常用网络调试命令，进一步丰富了对网络的理解。

六 附录

参考资料

- 什么是 Gratuious ARPs：https://blog.csdn.net/weixin_33754065/article/details/92784934