**计算机网络实验报告（实验3）**

  班级 22计算机科学与技术3班

 学号 2022334323029 姓名 张雅瑞

实验时间：

|  |  |  |
| --- | --- | --- |
| **课程目标2** | **课程目标3** | **报告成绩** |
|  |  |  |

1. 实验名称：

**Wireshark 软件使用与 ARP 协议分析，IP 与 ICMP 分析**

1. 实验目的：
   * 1. 学习 Wireshark 基本操作：重点掌握捕获过滤器和显示过滤器。
     2. 观察 MAC 地址：了解 MAC 地址的组成，辨识 MAC 地址类型。
     3. 分析以太网帧结构：观察以太网帧的首部和尾部，了解数据封装成帧的原理。
     4. 分析 ARP 协议：抓取 ARP 请求和应答报文，分析其工作过程。
     5. 使用 Wireshark 软件，观察 IP数据报的基本结构，分析数据报的分片；掌握基于 ICMP 协议的 ping 和 traceroute 命令及其工作原理。
2. 实验方案：

*【说明：根据实验目的，给出实验方案（能够确定研究的问题并形成假设，能够选择研究路线，设置可观察，可测量的观察点及需要采集的数据，形成合理的实验方案。）本部分仅用来说明，请在提交报告中删除，这部分我们观察学生会不会写实验方案，同时，应含以下内容：  
1.描述在实验中如何选择使用适当的工具（Wireshark 软件,ping工具和tracert工具）。  
2. 如何实现网络协议分析软件在网络环境中进行数据包捕获。*

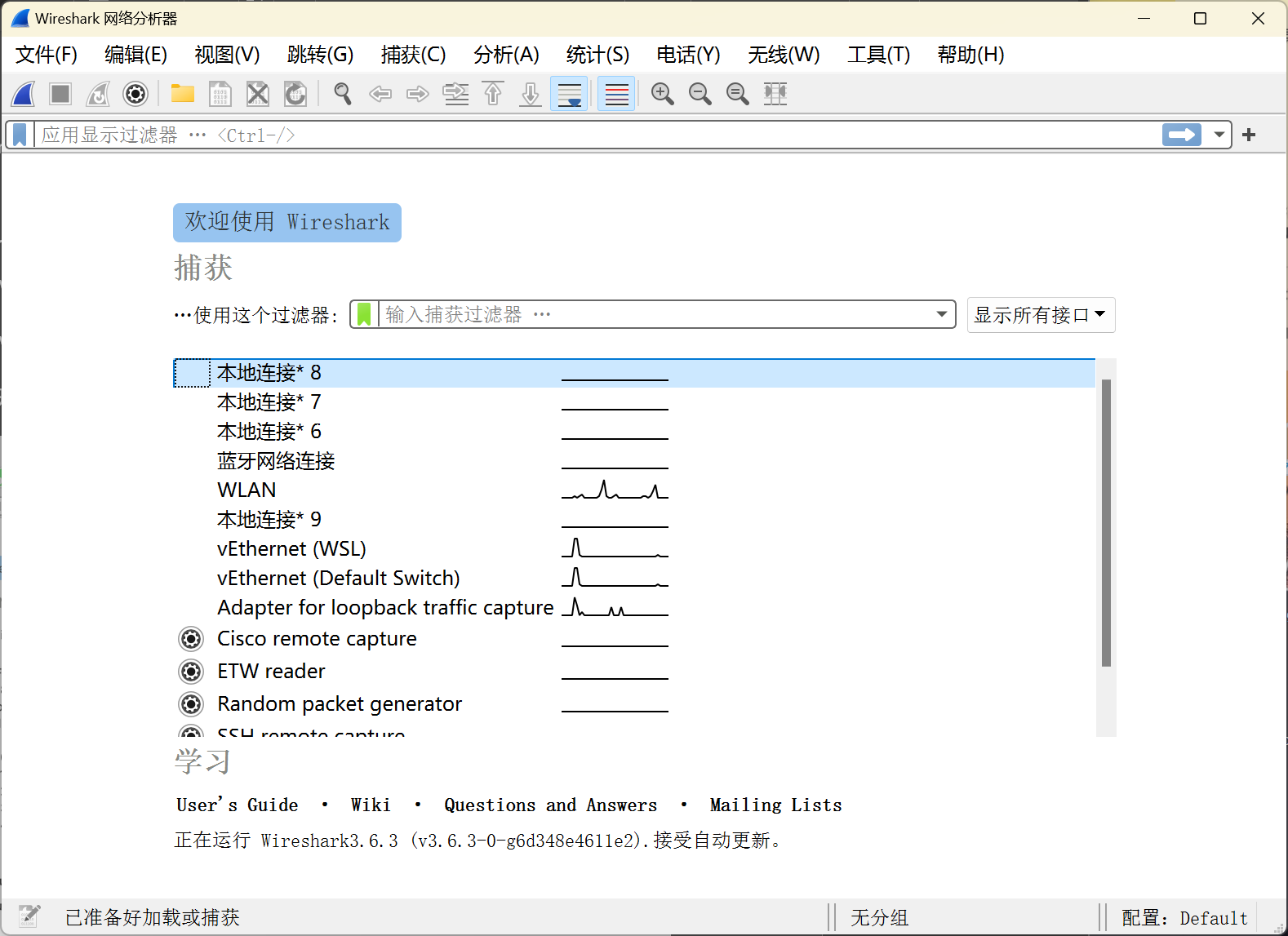
*语言组织很重要，即不要写成具体的操作步骤，又要描述出观察点，重点。】*

**实验3.1 Wireshark 软件使用与ARP 协议分析**

**3.1.1 WireShark** 基本使用

1. 通过 Wireshark 官网下载最新版软件，按默认选项安装。

2. 运行 Wireshark 软件，程序界面会显示当前的网络接口列表，点击要观察的网络接口，开始捕捉数据包，Wireshark 软件选择网络接口的界面如图所示。

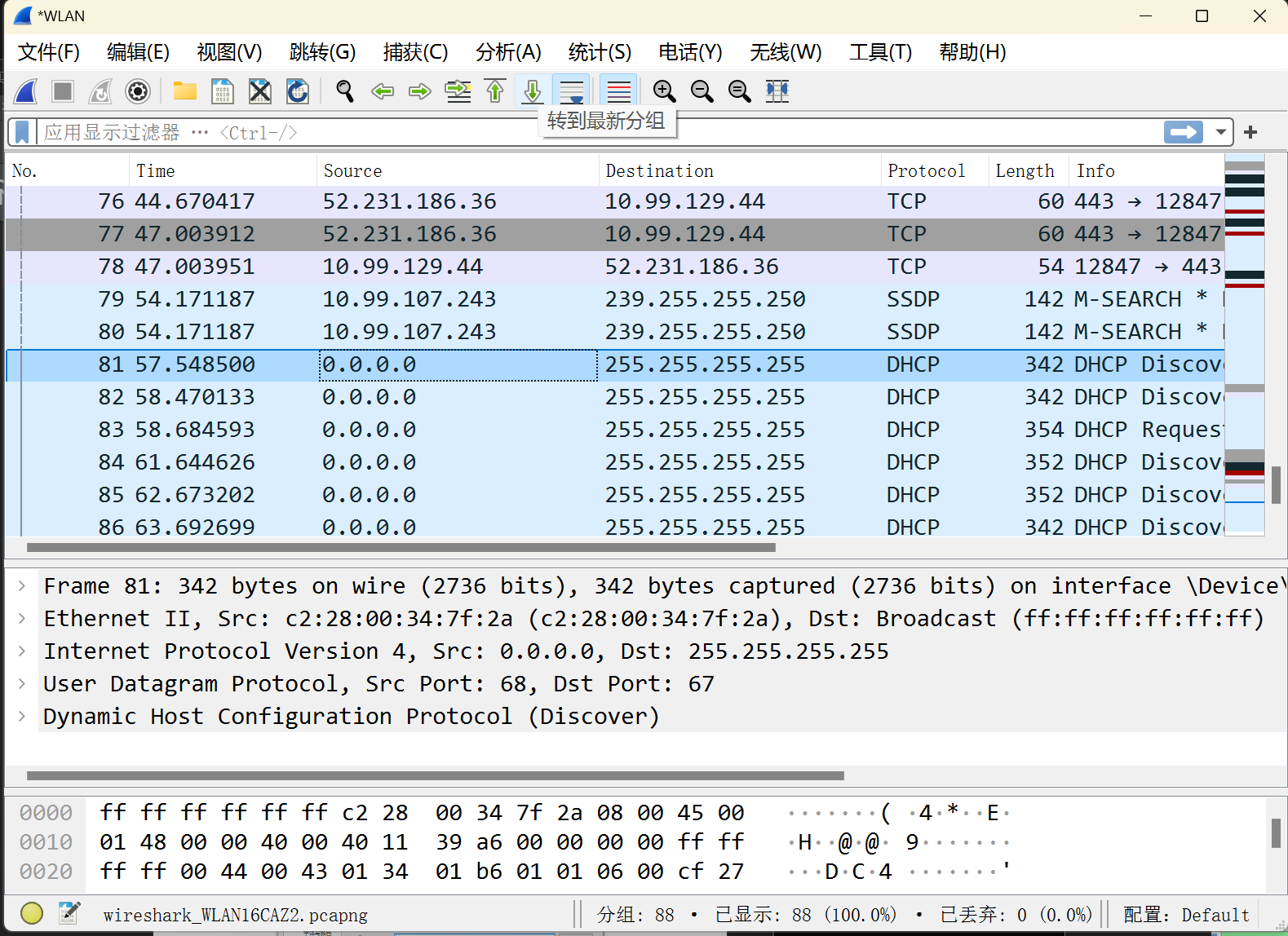


3. 点击工具栏上的红色方块按钮停止捕捉。

4. 菜单、工具栏、状态栏和主窗口如图所示。

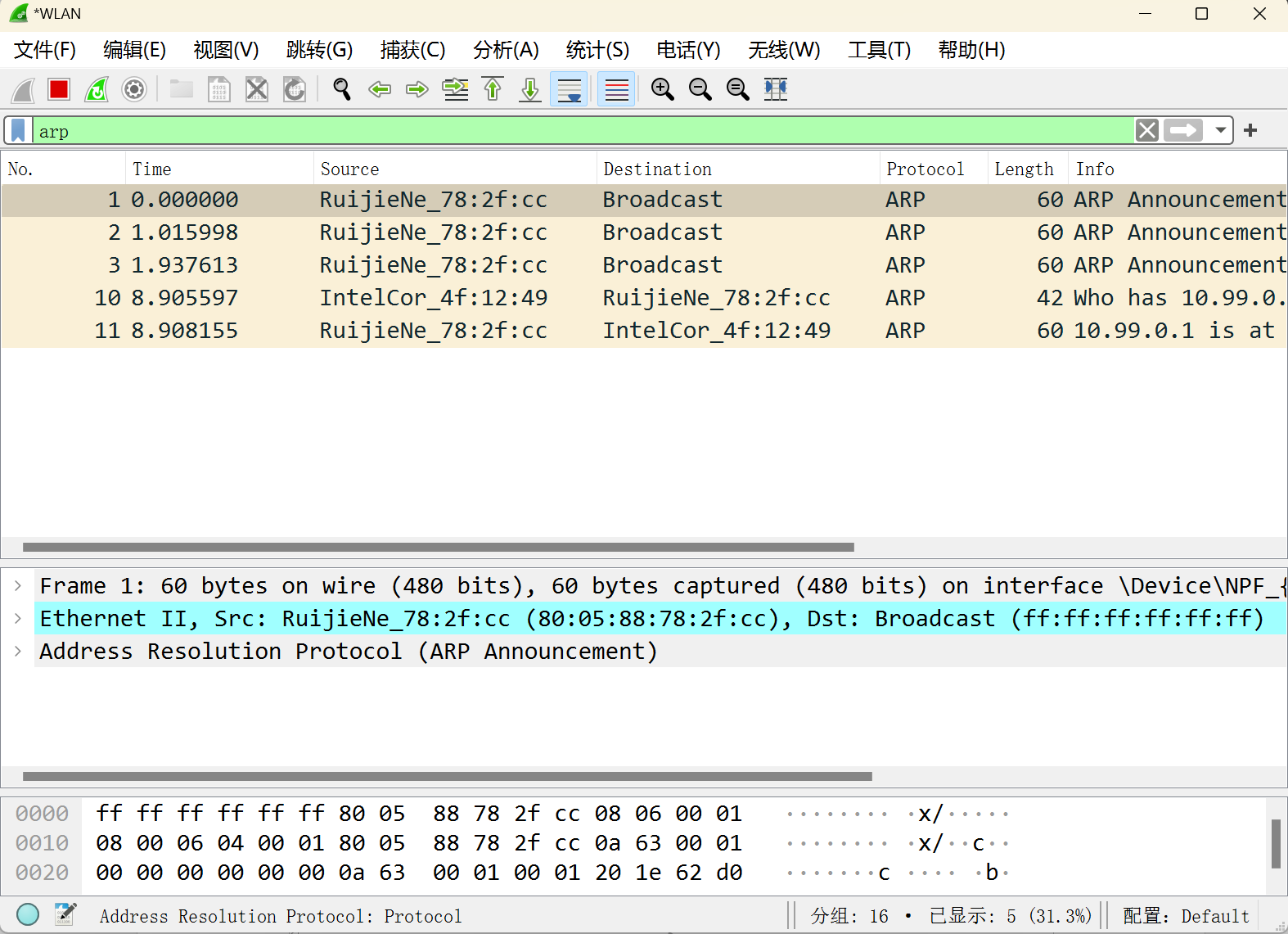
5. 使用“显示过滤器”可以方便地从捕获的数据包中筛选出要观察的数据包。显示过滤器支持若干的过滤选项：源 MAC、目的 MAC、源 IP、目的 IP、TCP/UDP

传输协议、应用层协议（HTTP, DHCP）、源端口 Port、目的端口 Port 等。



6. 通过主菜单“文件”/“导出特定分组”，可以保存捕获的网络数据

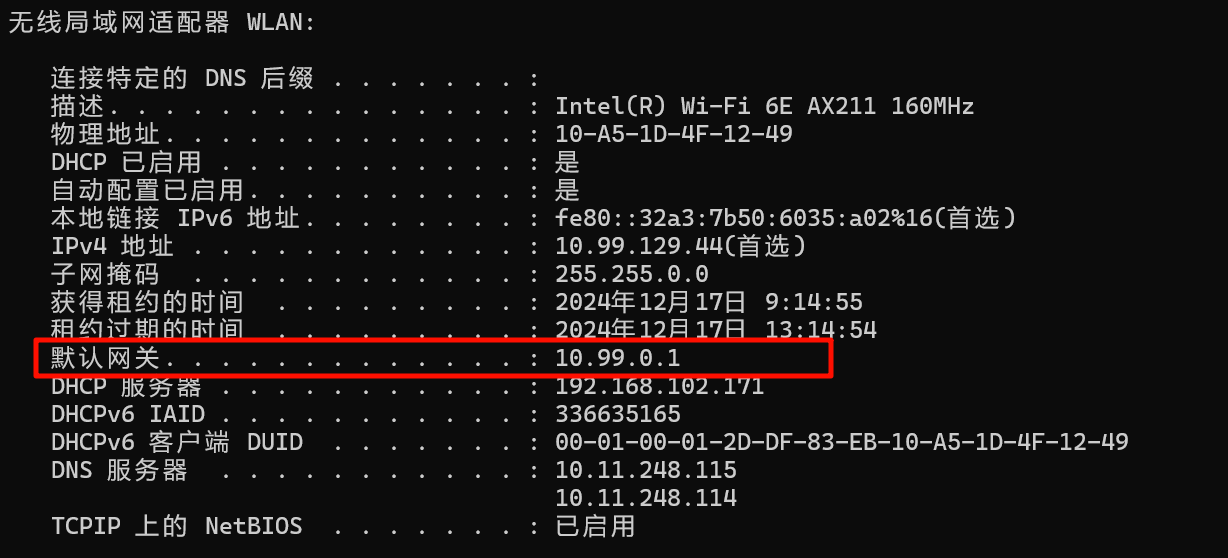
7. 如果只想捕捉特定的数据包，可以使用菜单“捕获”/“捕获过滤器”选定想要 的类型。例如，选择“IPv4 only”，Wireshark 只抓取 ipv4 类型的数据包。



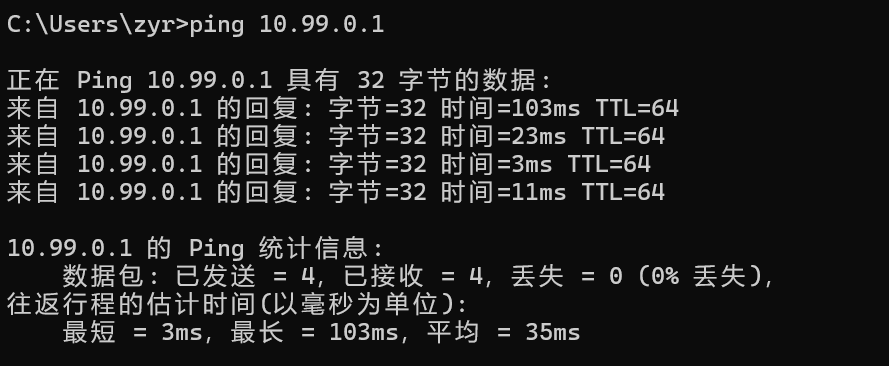
**3.1.2** 观察 **MAC** 地址

启动 Wireshark 捕捉数据包，在命令行窗口分别 ping 网关和 ping 同网段的一台主机，分析本机发出的数据包。重点观察以太网帧的 Destination 和 Source 的 MAC 地址， 辨识 MAC 地址类型，解读 OUI 信息、I/G 和 G/L 位。

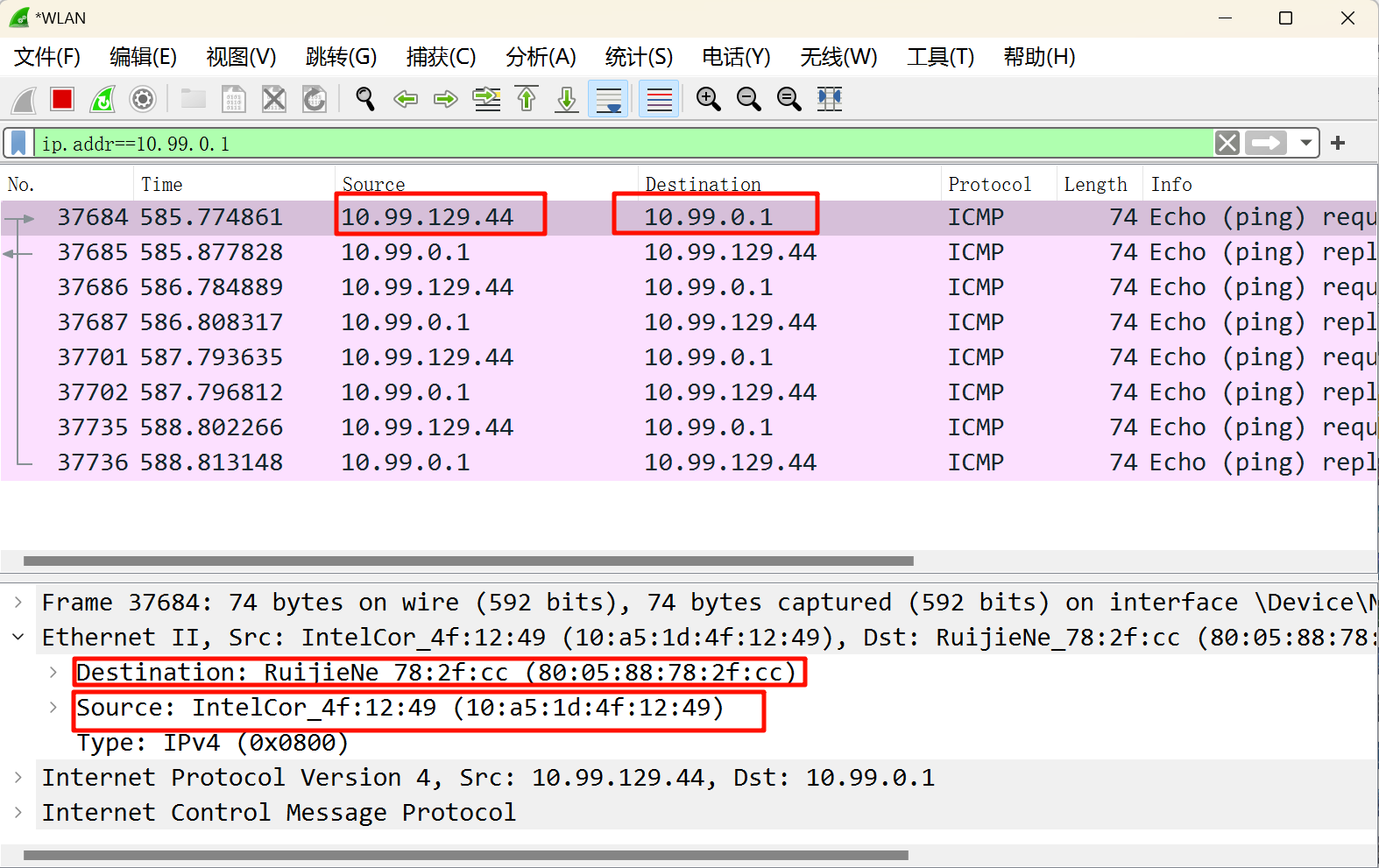
**3.1.2.1 ping网关**

ipconfig/all命令查看无线网WALN的默认网关10.99.0.1

ping 10.99.0.1（网关）



用ip.addr==10.99.0.1获取本机发出的数据包



观察以太网帧的Destination和Source的MAC地址，辨识MAC地址类型，解读OUI信息、I/G和G/L位。

（1）Source的MAC地址：(10:a5:1d:4f:12:49) 本地MAC地址。

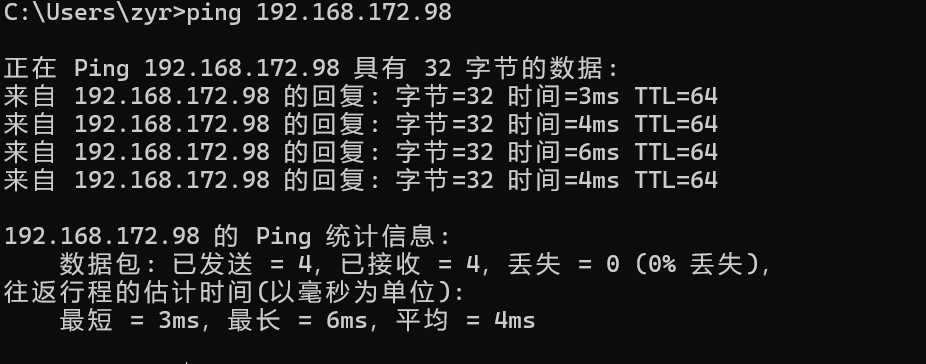
（2）Destination的MAC地址：(80:05:88:78:2f:cc) 默认网关地址。

1. **MAC 地址类型**
   * **单播地址（Unicast Address）**：
     + 对于源 MAC 地址（10:a5:1d:4f:12:49），这是一个单播地址。单播 MAC 地址用于标识网络上的一个特定设备，在以太网帧的源地址字段中，通常使用单播地址来指示发送该帧的设备。
     + 对于目的 MAC 地址（80:05:88:78:2f:cc），如果它是默认网关地址，也是一个单播地址。因为默认网关是网络中的一个特定设备，帧被发送到这个特定的网关设备进行转发等操作。
2. **OUI（Organizationally Unique Identifier）信息**
   * MAC 地址的前 24 位（对于 IEEE 802 标准的 MAC 地址）代表 OUI。
   * 对于源 MAC 地址，OUI 是 “10:a5:1d”，对于目的 MAC 地址，OUI 是 “80:05:88”。OUI 由 IEEE 分配给不同的设备制造商，通过查询 IEEE 的 OUI 数据库，可以确定这些 MAC 地址所属设备的制造商。不过，实际查询需要使用专门的工具或访问相关的数据库网站。
3. **I/G（Individual/Group）和 G/L（Global/Local）位解读**
   * 在 MAC 地址中，第一个字节的最低位是 I/G 位（第 8 位，从左往右数）。如果 I/G 位为 0，表示这是一个单播地址（用于标识单个设备）；如果 I/G 位为 1，表示这是一个组播地址（用于将帧发送到一组设备）。
   * 对于源 MAC 地址（10:a5:1d:4f:12:49）和目的 MAC 地址（80:05:88:78:2f:cc），因为它们是单播地址，所以 I/G 位为 0。
   * 第一个字节的第二位是 G/L 位（第 7 位，从左往右数）。如果 G/L 位为 0，表示这是一个全球唯一（由 IEEE 分配）的 MAC 地址；如果 G/L 位为 1，表示这是一个本地管理的 MAC 地址。通常情况下，大多数 MAC 地址的 G/L 位为 0，表明是全球唯一的 MAC 地址，这里给定的源地址和目的地址未提及是本地管理，推测其 G/L 位为 0，是全球统一分配的 MAC 地址。

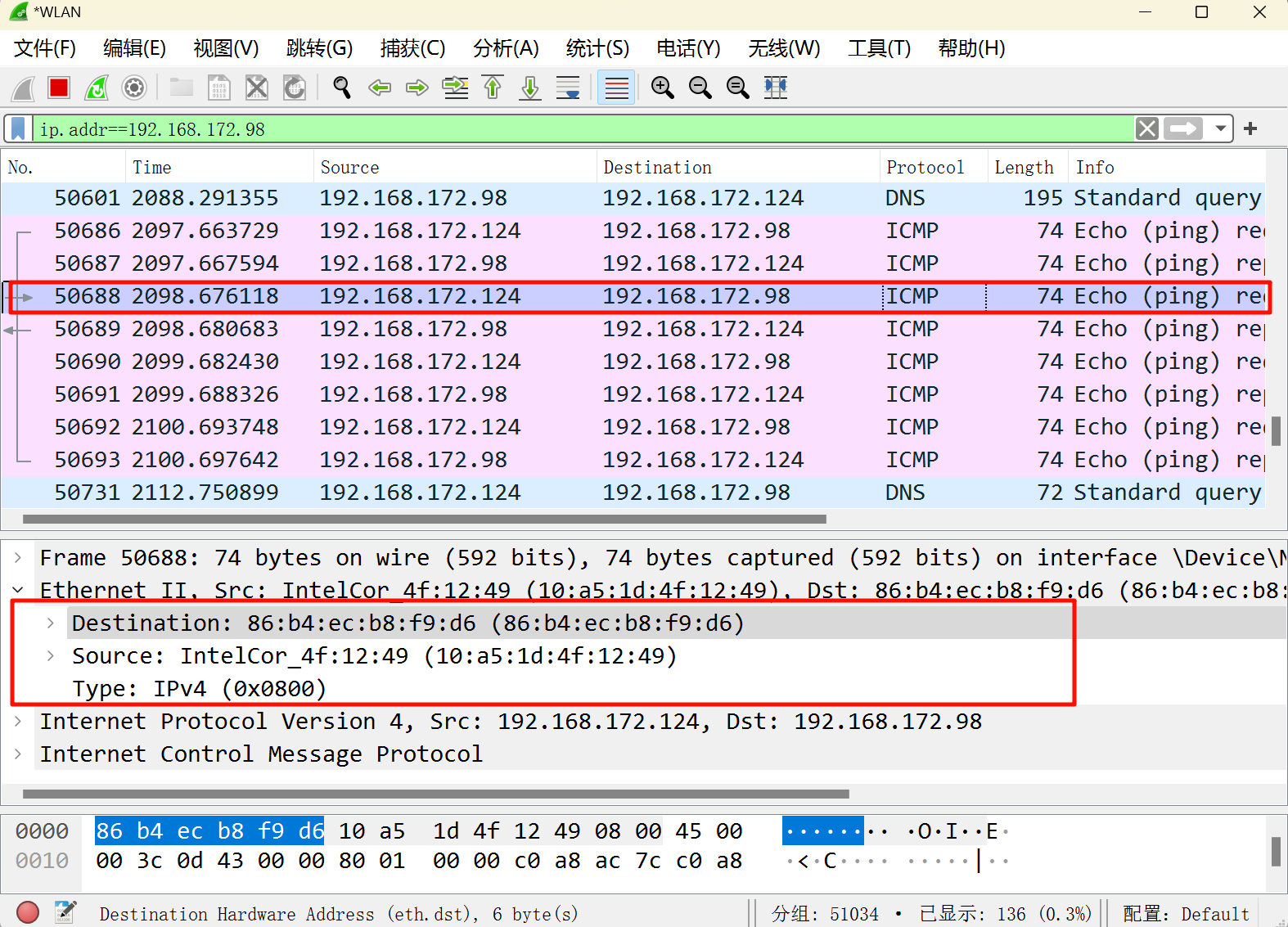
**3.1.2.2 ping同网段的一台主机**

切换到另一网络，

192.168.172.98 为同网段的一台主机

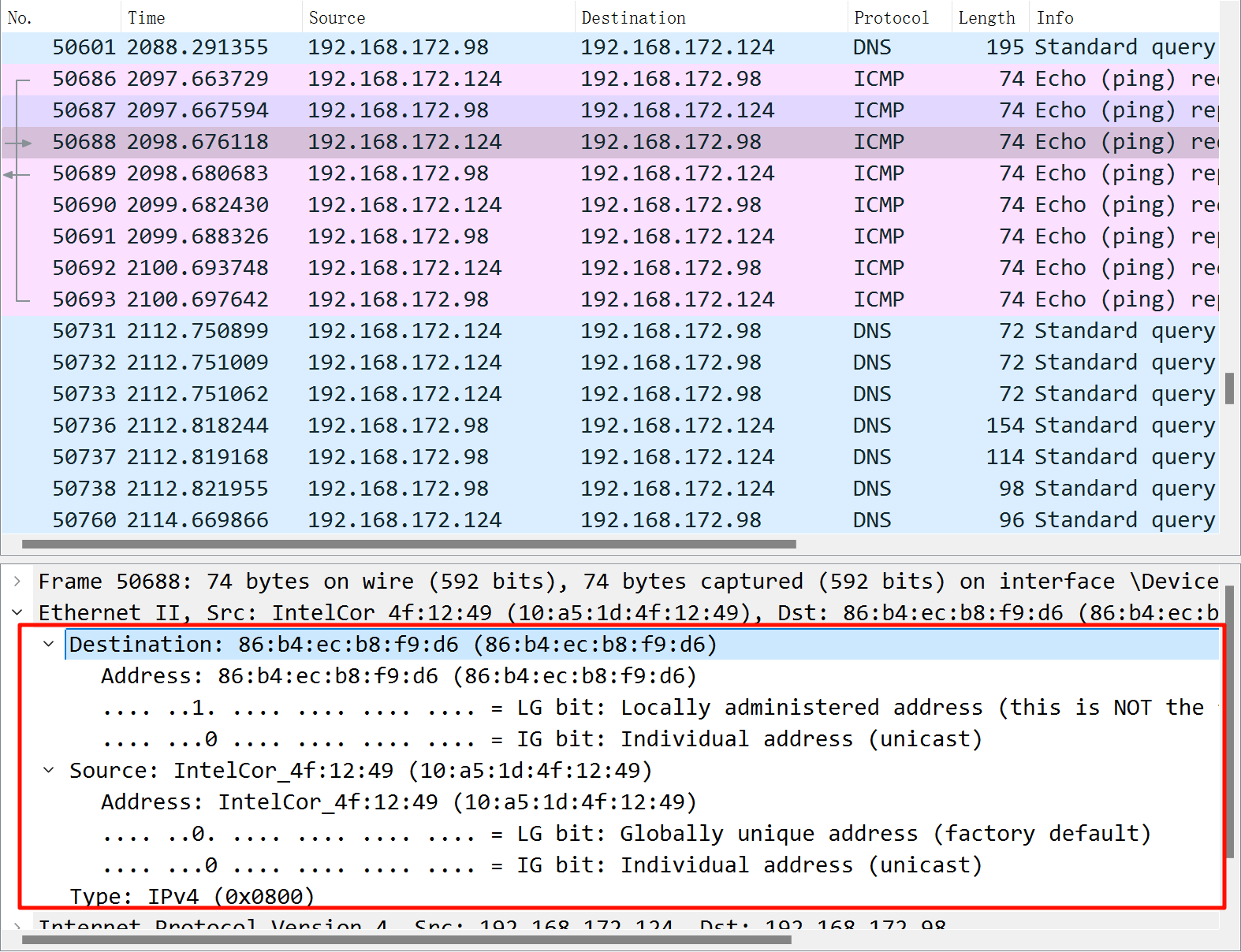


ip.addr==192.168.172.98



**3.1.3** 分析以太网的帧结构

选择其中一个数据包，点击 Ethernet II 展开，查看 MAC 帧的各个字段。

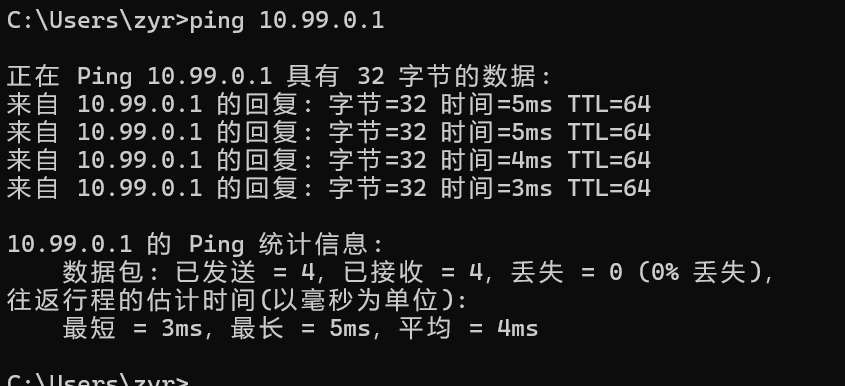


|  |
| --- |
| Ethernet II, Src: IntelCor\_4f:12:49 (10:a5:1d:4f:12:49), Dst: 86:b4:ec:b8:f9:d6 (86:b4:ec:b8:f9:d6)  Destination: 86:b4:ec:b8:f9:d6 (86:b4:ec:b8:f9:d6)  Address: 86:b4:ec:b8:f9:d6 (86:b4:ec:b8:f9:d6)  .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)  .... ...0 .... .... .... .... = IG bit: Individual address (unicast)  Source: IntelCor\_4f:12:49 (10:a5:1d:4f:12:49)  Address: IntelCor\_4f:12:49 (10:a5:1d:4f:12:49)  .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)  .... ...0 .... .... .... .... = IG bit: Individual address (unicast)  Type: IPv4 (0x0800) |

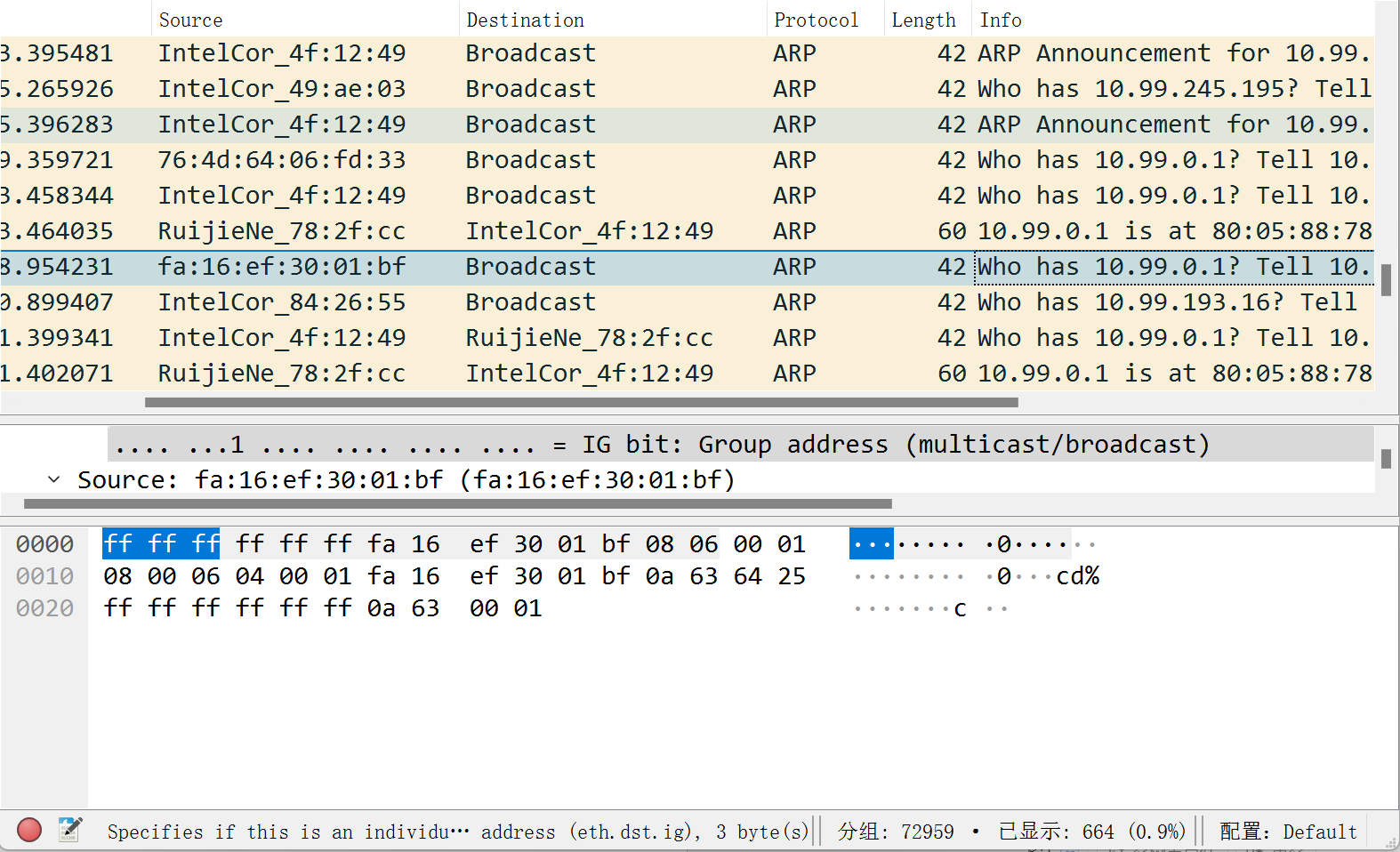
**3.1.4 ARP** 协议分析

1. 使用 arp –F 命令，清空本机的 ARP 缓存，开启 Wireshark， ping 本机的同网段地址，在显示过滤器条框中输入“arp ”，观察捕获的 ARP 报文的各个字段，分析请求/响应的过程。

ping 本机的同网段地址



在显示过滤器条框中输入“arp ”，定位到10.99.0.1

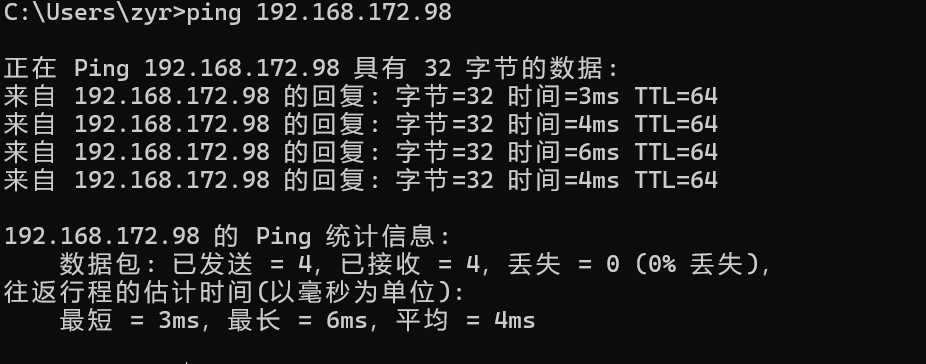


观察捕获的 ARP 报文的各个字段，分析请求/响应的过程

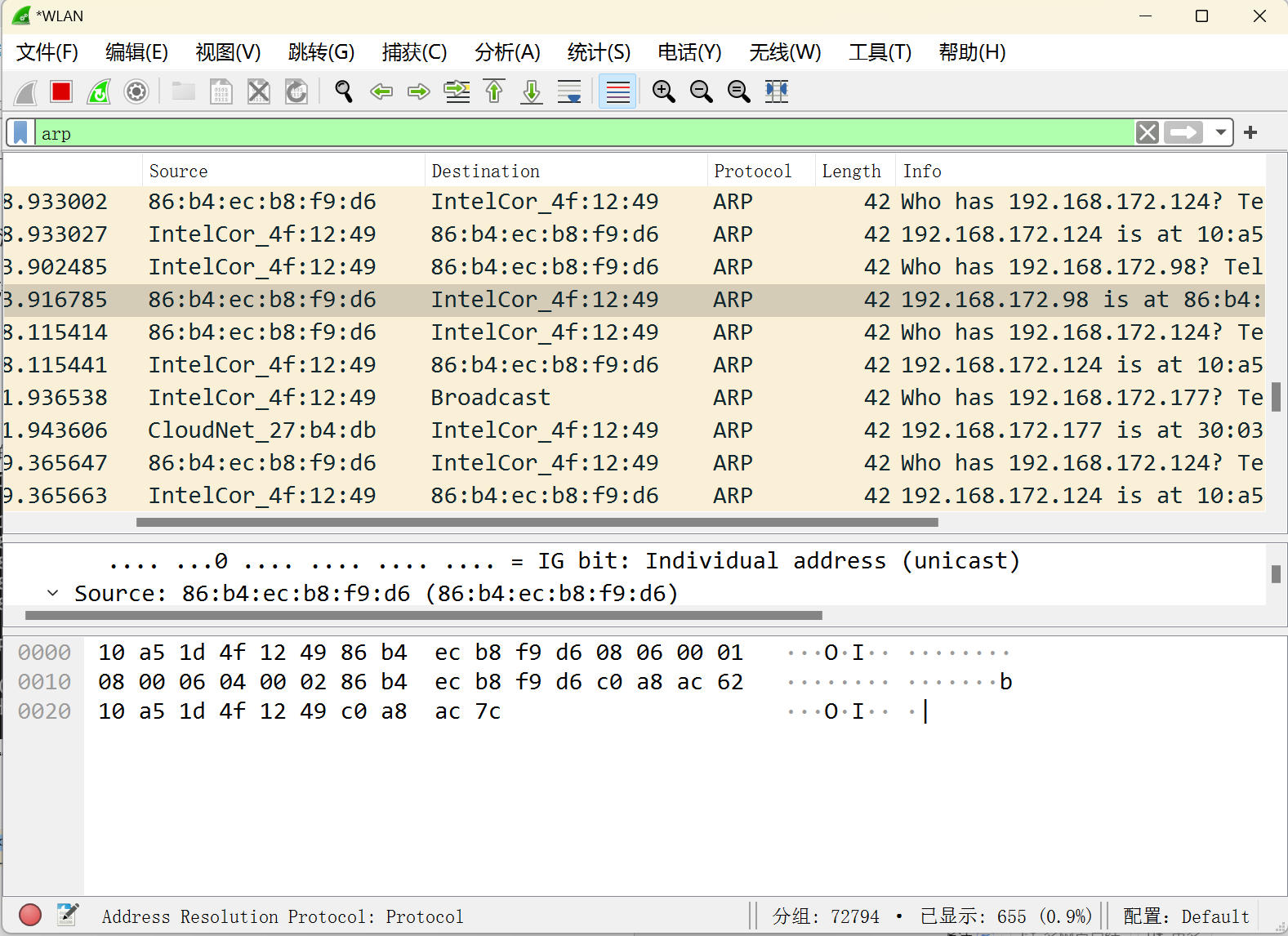
|  |
| --- |
| **以太网首部（前 14 字节）**     * **目的 MAC 地址**：ff ff ff ff ff ff，这是广播地址，意味着这个 ARP 报文是要在当前网段内进行广播，让同一网段中的所有设备都能接收到此报文，通常 ARP 请求会采用广播的方式发送，目的是询问目标 IP 地址对应的 MAC 地址，所以初步判断此处可能是 ARP 请求报文的一个特征体现。 * **源 MAC 地址**：fa 16 ef 30 01 bf，这代表发送该 ARP 报文的设备自身的 MAC 地址，通过这个地址可以识别出报文是从哪个具体设备发出的。 * **类型字段**：08 06，表明后面的数据载荷为 ARP 协议报文，在以太网的帧格式中，通过这个类型值来区分不同的上层协议，08 06 专门对应 ARP 协议。   **ARP 报文主体部分（从第 15 字节开始）**     * **硬件类型（2 字节）**：00 01，表示硬件类型为以太网，这是常见网络环境（如大多数局域网）中所使用的硬件类型设定，对应的值就是 00 01 。 * **协议类型（2 字节）**：08 00，说明上层协议是 IP 协议，也就是该 ARP 请求 / 响应是围绕 IP 地址和 MAC 地址之间的映射关系展开，且与 IP 协议相关联。 * **硬件地址长度（1 字节）**：04，意味着硬件地址（即 MAC 地址）的长度是 4 字节，不过这里存在错误，正常以太网中 MAC 地址长度应为 6 字节，此处数据可能有误或者在特殊自定义的环境下有不同解释，按常规以太网理解应该是 6 字节。 * **协议地址长度（1 字节）**：04，表示协议地址（也就是 IP 地址）的长度是 4 字节，符合 IPv4 地址的实际长度情况。 * **操作码（2 字节）**：00 01，这明确表示该 ARP 报文是一个 ARP 请求报文。因为按照 ARP 协议规定，操作码为 00 01 时，发送方在向网络中的其他设备发起询问，想获取特定 IP 地址对应的 MAC 地址，自身并不知道目标的 MAC 地址情况，所以通过广播这个请求报文来进行询问。 * **发送方硬件地址（6 字节）**：fa 16 ef 30 01 bf，再次表明发送该 ARP 请求的设备自身的 MAC 地址，和以太网首部的源 MAC 地址相呼应，确认了报文的来源。 * **发送方协议地址（4 字节）**：0a 63 64 25，这是发送 ARP 请求的设备自身的 IP 地址，在请求报文中，发送方通过这个 IP 地址与 MAC 地址的组合告知其他设备自己是谁，同时希望获取目标 IP 地址对应的 MAC 地址。 * **目标硬件地址（6 字节）**：ff ff ff ff ff ff，在 ARP 请求中，由于发送方不知道目标设备的 MAC 地址，所以这里会填充全 0 或者广播地址（像此处用广播地址的情况也是有的，不同实现方式可能存在差异），其目的是让目标设备接收到请求后能识别出是在询问自己的 MAC 地址（基于 IP 地址匹配等情况）。 * **目标协议地址（4 字节）**：0a 63 00 01，这代表目标设备的 IP 地址，也就是发送方想要知道其 MAC 地址对应的那个 IP 地址，发送方通过广播的 ARP 请求向整个网段询问该 IP 地址对应的 MAC 地址是谁的。   **请求 / 响应过程整体分析**  综合上述对各个字段的解读，可以梳理出如下的 ARP 请求 / 响应过程情况：  发送方设备（MAC 地址为 fa 16 ef 30 01 bf ，IP 地址为 0a 63 64 25 ）想要与 IP 地址为 0a 63 00 01 的目标设备进行通信，但它不知道目标设备的 MAC 地址。于是，发送方构造了这个 ARP 请求报文，将目的 MAC 地址设置为广播地址（ff ff ff ff ff ff ）进行全网段广播，告知其他设备自己在寻找 IP 地址为 0a 63 00 01 对应的 MAC 地址。网络中的其他设备接收到这个广播的 ARP 请求报文后，会检查目标协议地址是否与自己的 IP 地址匹配，如果匹配（也就是目标 IP 地址对应的设备自身），那么相应的设备就会构造一个 ARP 响应报文来回复发送方，告知自己的 MAC 地址与 IP 地址的对应关系。不过，从当前给出的这个报文来看，它只是 ARP 请求报文，还未体现出对应的响应报文内容。 |

1. 使用 arp –F 命令，清空本机的 ARP 缓存。开启 Wireshark，ping 与本机网段不同的 IP 地址或域名，观察捕获的 ARP 报文的各个字段，分析请求/响应的过程。

192.168.172.98 为同网段的一台主机



在显示过滤器条框中输入“arp ”，定位到192.168.172.98



观察捕获的 ARP 报文的各个字段，分析请求/响应的过程

|  |
| --- |
| **以太网首部（前 14 字节）**     * **目的 MAC 地址**：10 a5 1d 4f 12 49，这是接收该 ARP 报文的目标设备的 MAC 地址。在 ARP 请求时，通常若目标 MAC 地址未知，此处应为广播地址（全 FF:FF:FF:FF:FF:FF ）；而这里出现了具体的 MAC 地址，很可能这是一个 ARP 响应报文，该地址指向之前发出 ARP 请求的设备，也就是要将响应内容回复给这个 MAC 地址对应的设备。 * **源 MAC 地址**：86 b4 ec b8 f9 d6，代表发送该 ARP 报文的设备自身的 MAC 地址，通过这个地址能够明确报文是从哪个具体设备发出的。 * **类型字段**：08 06，此值表明后面承载的数据是 ARP 协议报文，在以太网帧中，通过该类型值来区分上层不同的协议类型，08 06 专门用于标识 ARP 协议。   **ARP 报文主体部分（从第 15 字节开始）**     * **硬件类型（2 字节）**：00 01，意味着硬件类型为以太网，这是常见局域网环境中所对应的硬件类型设定，一般用 00 01 来表示以太网的硬件类型。 * **硬件地址长度（1 字节）**：06，表示硬件地址（即 MAC 地址）的长度是 6 字节，这与以太网中 MAC 地址实际的字节长度相符。 * **协议类型（2 字节）**：08 00，说明上层协议是 IP 协议，也就表明该 ARP 报文的请求 / 响应操作是围绕 IP 地址与 MAC 地址之间的映射关系开展，并且和 IP 协议相关联。 * **协议地址长度（1 字节）**：04，代表协议地址（也就是 IP 地址）的长度是 4 字节，符合 IPv4 地址的长度标准。 * **操作码（2 字节）**：00 02，根据 ARP 协议规范，操作码为 00 02 时，表明这是一个 ARP 响应报文。意味着发送该报文的设备是在回复之前收到的 ARP 请求，告知对方自己的 MAC 地址与相关 IP 地址的对应关系。 * **发送方硬件地址（6 字节）**：86 b4 ec b8 f9 d6，再次强调了发送 ARP 报文的设备自身的 MAC 地址，与以太网首部中的源 MAC 地址一致，确定了报文的发送来源。 * **发送方协议地址（4 字节）**：c0 a8 ac 62，这是发送该 ARP 响应报文的设备的 IP 地址，通过这个 IP 地址与 MAC 地址的组合，接收方能够知晓是哪个 IP 地址对应的设备在回复自己的请求。 * **目标硬件地址（6 字节）**：10 a5 1d 4f 12 49，在 ARP 响应报文中，这里填写的是之前发送 ARP 请求的设备的 MAC 地址，即要把这个响应信息准确传达给请求方，与以太网首部中的目的 MAC 地址相呼应，明确了响应的目标。 * **目标协议地址（4 字节）**：c0 a8 ac 7c，代表之前发起 ARP 请求的设备的 IP 地址，发送方通过响应报文告知这个 IP 地址对应的自己的 MAC 地址信息，完成地址解析的反馈过程。   **请求 / 响应过程整体分析**  综合对上述各字段的分析，可以梳理出以下 ARP 请求 / 响应过程：  起初，设备（MAC 地址为 10 a5 1d 4f 12 49 ，IP 地址为 c0 a8 ac 7c ）想要与 IP 地址为 c0 a8 ac 62 的另一台设备通信，但不知道对方的 MAC 地址，于是它发出了一个 ARP 请求报文（操作码为 00 01 ，且目的 MAC 地址可能是广播地址等方式向网段内广播询问）。之后，IP 地址为 c0 a8 ac 62 、MAC 地址为 86 b4 ec b8 f9 d6 的设备收到了这个 ARP 请求，便构造并发送了当前我们所观察到的这个 ARP 响应报文（操作码为 00 02 ），将自己的 MAC 地址信息反馈给发起请求的设备，使得发起请求的设备知晓了目标 IP 地址对应的 MAC 地址，从而可以基于此进行后续的网络通信操作。 |

思考题

1. 使用了显示过滤器后，Wireshark 的抓包工作量会减少吗？

|  |
| --- |
| 使用显示过滤器后，Wireshark 的抓包工作量不会减少。  Wireshark 的抓包是由捕获过滤器（Capture Filter）控制的，它决定了哪些数据包被抓取并存储到缓冲区中。而显示过滤器（Display Filter）是在已经捕获的数据包基础上进行筛选，用于决定哪些已捕获的数据包显示在界面上。  所以，即使使用了显示过滤器，Wireshark 仍然会抓取所有符合捕获条件（或者没有设置捕获过滤器时抓取所有经过网卡的数据包）的数据包，只是在显示环节按照显示过滤器的规则来展示部分数据包。 |

1. MAC 帧的长度和 IP 数据报的长度有怎样的关系？请用你的数据记录进行验证。

|  |
| --- |
| **理论关系**   * MAC 帧是数据链路层的协议数据单元，它包含了 IP 数据报。MAC 帧的长度范围是 64 - 1518 字节（不包括前导码和帧开始定界符）。IP 数据报长度理论上最小是 20 字节（仅包含首部，没有数据部分），最大是 65535 字节，但在实际传输中，由于链路层的 MTU（最大传输单元）限制，IP 数据报长度要适应链路层的要求。 * 一般情况下，MAC 帧的长度大于等于 IP 数据报长度加上 MAC 帧首部和尾部的长度。即   **示例：**  Ip地址大小    Mac帧大小 |

1. ping 同一局域网内的主机和局域网外的主机，都会产生 ARP 报文么？所产生的ARP 报文有何不同，为什么？

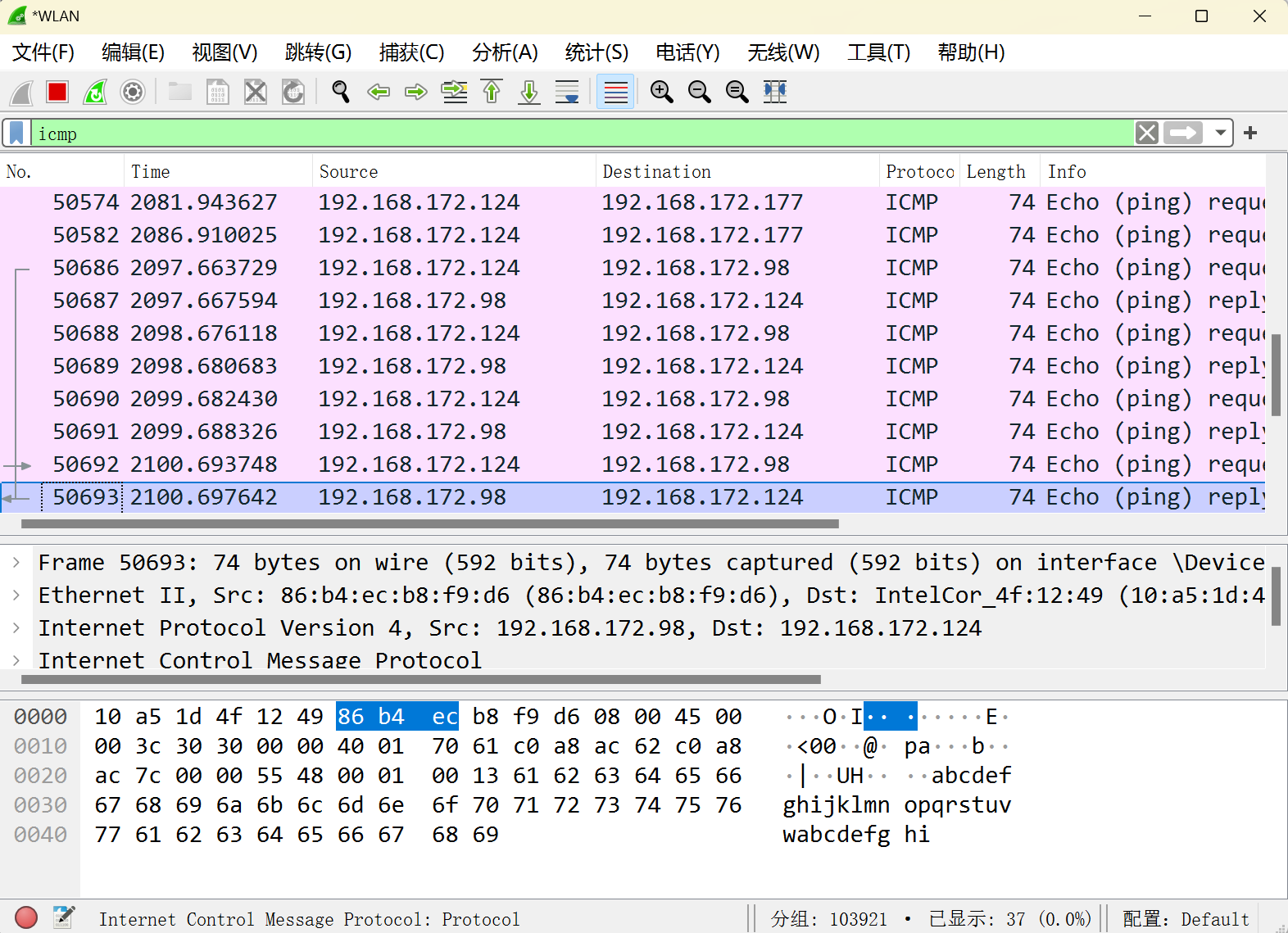
|  |
| --- |
| 1. **是否都会产生 ARP 报文**    * ping 同一局域网内的主机会产生 ARP 报文。因为在局域网内通信需要通过 MAC 地址来传输数据，当发送主机不知道目标主机 MAC 地址时，会发送 ARP 请求报文来获取。    * ping 局域网外的主机也会产生 ARP 报文。不过情况更复杂，首先会产生 ARP 报文用于获取默认网关的 MAC 地址，因为数据要先发送到网关才能转发到外网。 2. **产生的 ARP 报文的不同之处及原因**    * **目标不同**：      + 对于局域网内 ping，ARP 请求报文的目标 IP 是局域网内被 ping 的主机的 IP 地址，目标是获取该主机的 MAC 地址，以便在局域网内直接发送 ICMP 回显请求报文。      + 对于局域网外 ping，ARP 请求报文的目标 IP 是默认网关的 IP 地址，目的是获取网关的 MAC 地址，这样 ICMP 回显请求报文才能先发送到网关进行后续的转发。    * **作用范围不同**：      + 局域网内的 ARP 报文只在局域网内部传播，用于解析局域网内主机的 MAC - IP 映射关系。      + 局域网外 ping 产生的 ARP 报文用于本地局域网内的网关地址解析，之后数据通过网关转发到外网，其作用主要是完成本地网络与外部网络连接的第一步。 |

**3.2-----IP与ICMP分析**

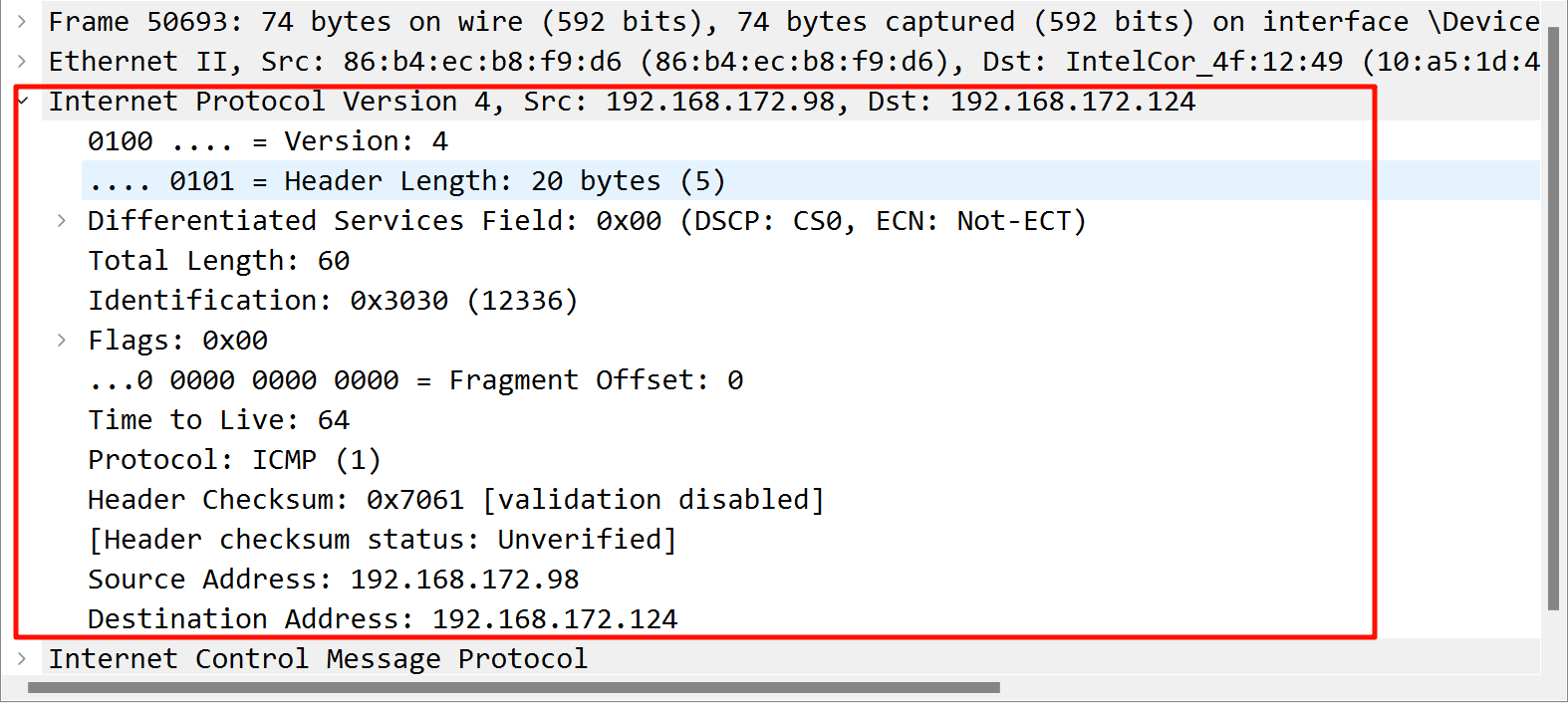
**5.1 ping** 命令

本机启动 Wireshark 软件，选择要监听的网络接口（如eth0、wlan0）；然后在终端发起网络命令：ping IP 地址/域名。

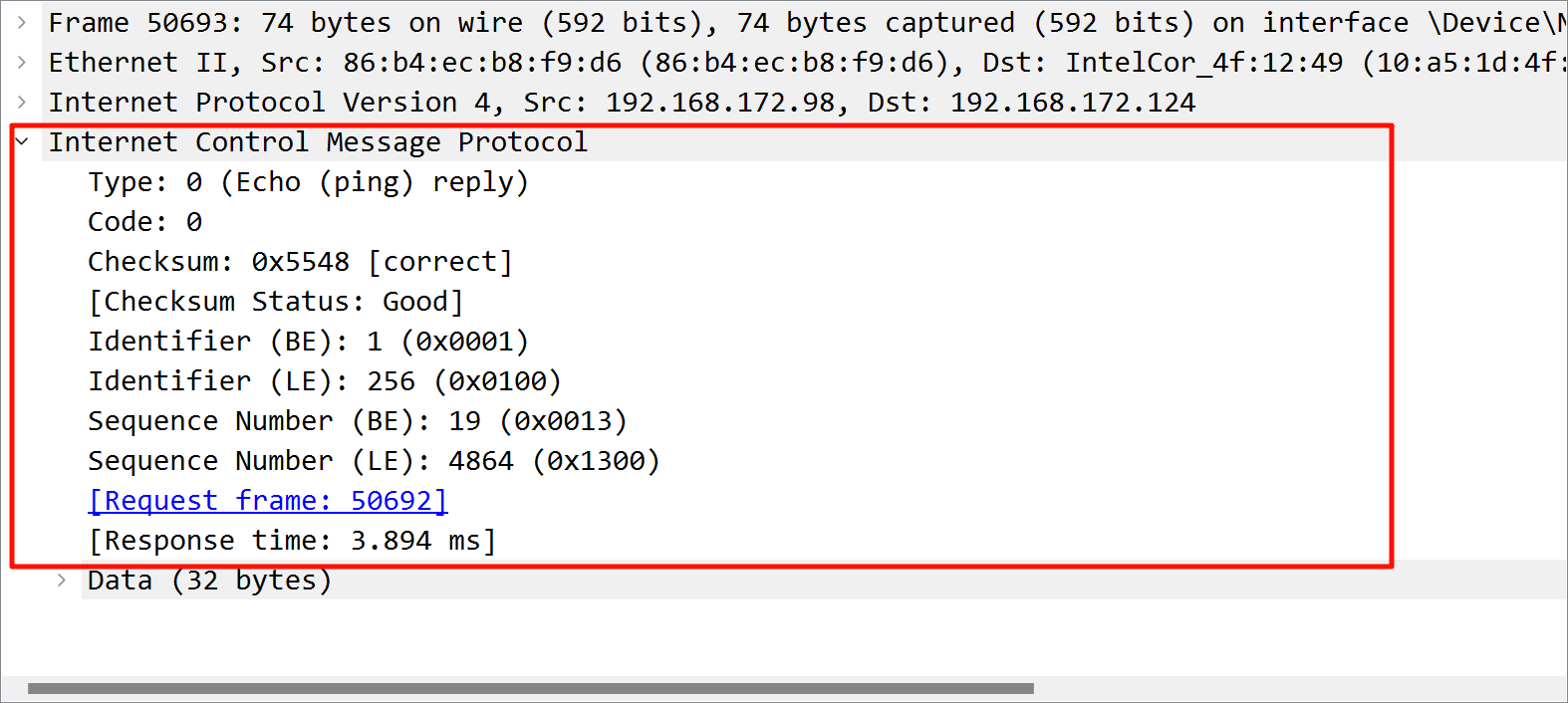
1. 在 Wireshark 监视器中设置过滤条件。例如设置过滤条件为 icmp，则显示出所捕获的 ICMP 数据包。



1. 点击 Internet Protocol Version 4 展开，查看 IP 数据报，特别观察 IP数据报的首部字段内容。



1. 点击 Internet Control Message Protocol 展开，查看 ICMP 报文，并解释回显（Echo Request 和 Echo Reply）报文的首部字段。



Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

|  |
| --- |
| **ICMP（Internet Control Message Protocol）回显报文首部字段解释**   * + **Type（类型）字段**：     - 在 ICMP 报文中，Type 字段用于标识 ICMP 消息的类型。对于回显（ping）相关的报文，当 Type 值为 0 时，表示这是一个 Echo Reply（回显应答）报文。相对应的，当 Type 值为 8 时，表示 Echo Request（回显请求）报文。这些类型值定义了 ICMP 消息的基本功能，帮助接收方设备理解这个 ICMP 报文是用于请求还是应答 ping 操作。   + **Code（代码）字段**：     - Code 字段用于进一步细分 Type 字段定义的消息类型。当 Type 为 0（Echo Reply）时，Code 值为 0 通常表示正常的回显应答，没有附带特殊的错误或状态信息。它是在对 Echo Request 进行常规响应时使用的标准代码，表明该回显应答是正常的、符合预期的 ping 应答操作。 |

1. 清空 Wireshark 监控器，重新发起网络命令：ping IP 地址/域名–l#length，并解释对比前后两次执行 ping 命令的结果。其中，-l #length 确定 echo数据报的长度为 #length，其默认值为 32 字节，且小于 65,527 字节。

|  |
| --- |
| 第一次ICMP数据负载为2000字节，第二次为32字节。 |

1. 可以多次改变 #length 的大小（例如 1000 字节、2000 字节和 4000 字节），观察IP 数据报何时会分片？请解释 IP 数据报分片的原因和具体情况。提示：请先确认该网络的 MTU，可在 Wireshark 记录中查找“IPv4 fragments”项目。

|  |
| --- |
| 确认该网络的 MTU    在ICMP数据负载为1008时，不分片；    在ICMP数据负载为2008字节时，分片情况：    在ICMP数据负载为4008字节时，分片情况：      解释：MTU值是1500字节。包括IP头（20字节）和ICMP头（8字节）在内，数据部分最大不能超过1472字节，否则就会发生分片。 |

**5.2 traceroute** 命令

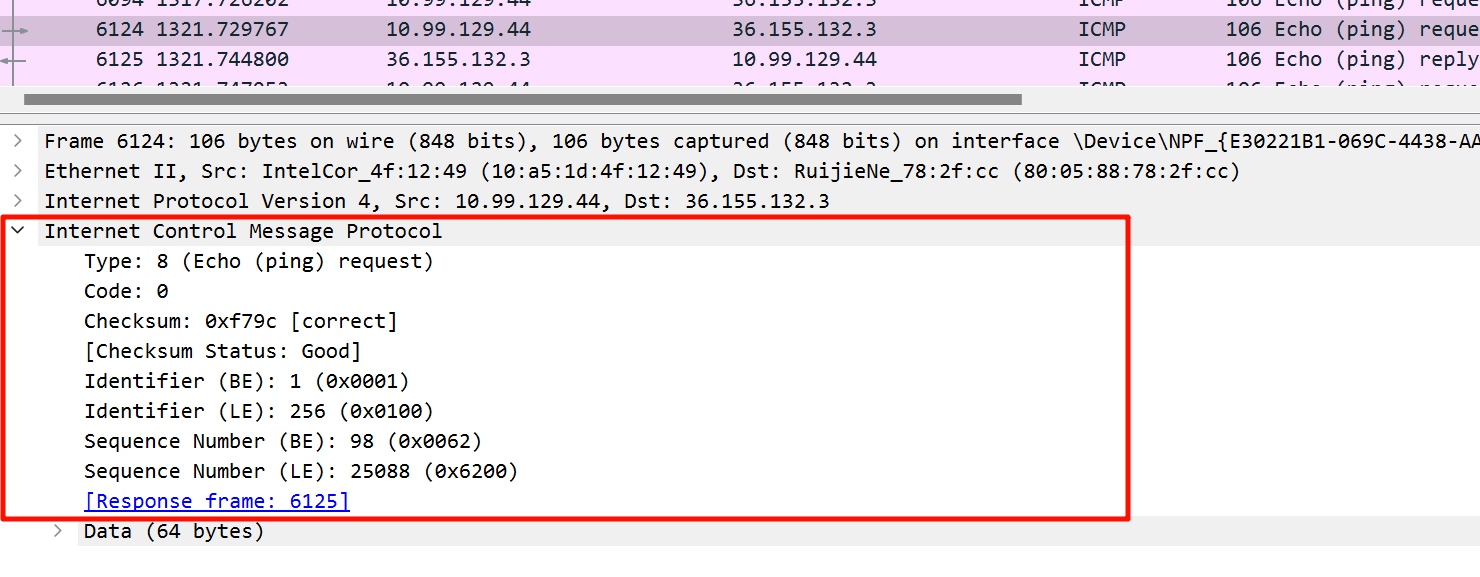
本机启动 Wireshark 软件，选择要监听的网络接口（如eth0、wlan0）；然后在终端发起网络命令：traceroute IP 地址/域名。

1. 启动 Wireshark 软件，选择要监听的网络接口，设置过滤条件 icmp。

2. 在终端中使用 traceroute 命令，目的主机是外网的一台设备。

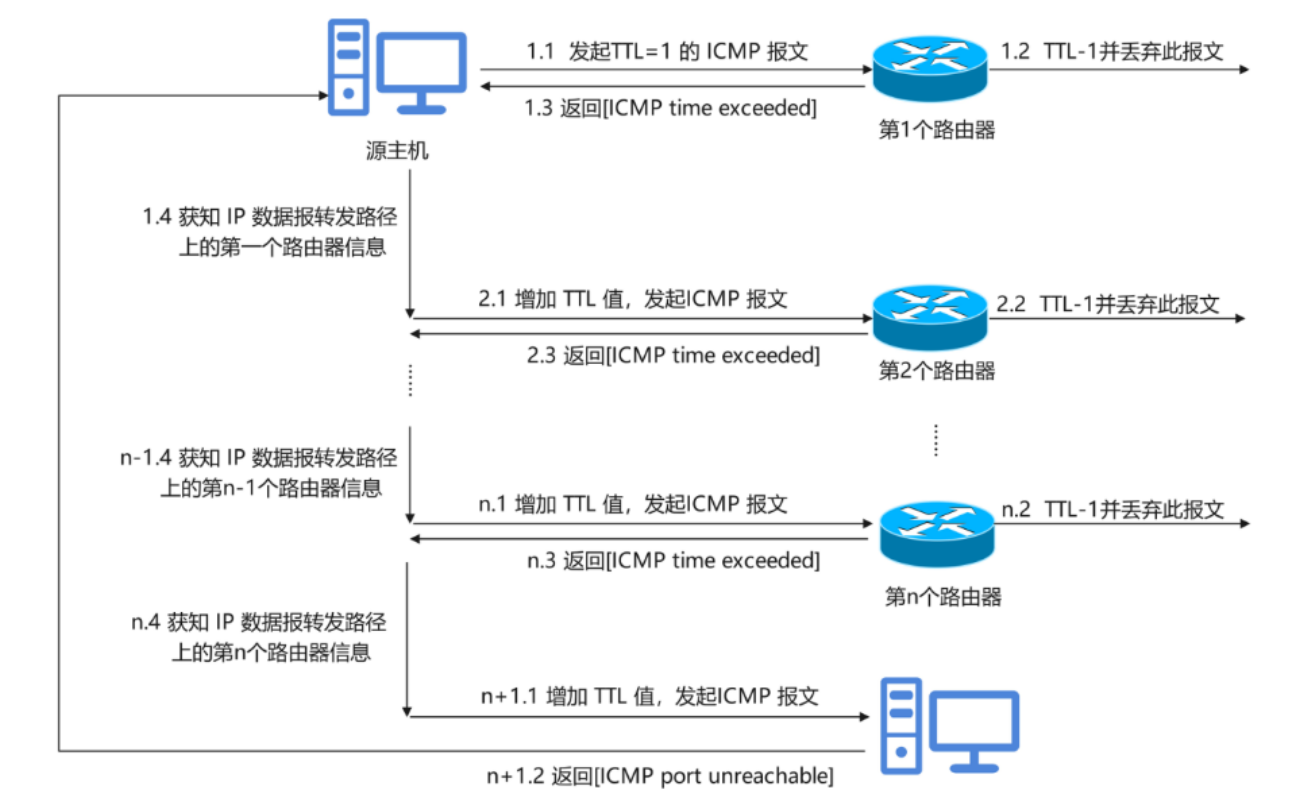


3.点击 Internet Control Message Protocol 展开，查看 ICMP 差错报文，观察并解释ICMP 报文结构和字段内容。



|  |
| --- |
| **ICMP 报文整体结构**  ICMP 报文作为 IP 数据报的数据部分进行传输，整体结构包含了多个重要字段，不同类型的 ICMP 报文会基于这个基本结构，各字段有着相应的含义来实现特定的网络控制与信息交互功能。  **各字段详细解释**   * **Type（类型）字段**：   + 在此报文中，Type 值为 8，表示这是一个 Echo (ping) request（回显请求）报文。ICMP 定义了多种不同的 Type 值来区分不同用途的报文，例如，值为 0 时对应的是 Echo Reply（回显应答）报文，用于响应回显请求；而这里的 8 明确了此报文是主动发起 ping 操作，向目标设备请求回复的请求类报文。 * **Code（代码）字段**：   + Code 值为 0，配合 Type 字段进一步细化该 ICMP 报文的具体情况。对于 Echo (ping) request 报文，当 Code 为 0 时，表示这是一个常规的、没有特殊条件或错误状态的回显请求。不同的 Type 对应的 Code 有不同含义，在回显请求这个场景下，Code = 0 就是标准的正常请求设定。 * **Checksum（校验和）字段**：   + 其值为 0xf79c ，并且标注了 [correct] 以及 [Checksum Status: Good]，说明该校验和通过验证是正确的。Checksum 字段用于对整个 ICMP 报文进行校验，发送方会计算出报文的校验和并填入该字段，接收方收到报文后同样进行校验和计算，并与该字段值对比，若一致则表明报文在传输过程中大概率没有出现错误，保证了报文的完整性和准确性。 * **Identifier (BE)（标识符 - 大端序）和 Identifier (LE)（标识符 - 小端序）字段**：   + Identifier (BE) 值为 1（0x0001 ），Identifier (LE) 值为 256（0x0100 ）。标识符用于区分不同的 ICMP 会话或者进程，在发送多个 ping 请求等情况时，可以通过这个标识符来将回应与相应的请求对应起来。大端序（BE）和小端序（LE）是计算机存储数据时字节顺序的不同表示方式，这里同时给出两种表示便于不同系统环境（有些按大端序处理数据，有些按小端序）下准确识别该标识符。 * **Sequence Number (BE)（序列号 - 大端序）和 Sequence Number (LE)（序列号 - 小端序）字段**：   + Sequence Number (BE) 值为 98（0x0062 ），Sequence Number (LE) 值为 25088（0x6200 ）。序列号主要用于对一连串的 ICMP 报文（比如多次连续的 ping 请求）进行排序，接收方可以根据序列号来判断报文是否有丢失、重复等情况，同样以大端序和小端序两种形式呈现，方便不同系统准确处理。 * **Data（数据）字段**：   + 此处显示 Data 有 64 字节，这个字段可以携带一些额外的数据信息，在 ping 操作中，通常可以用来填充一些自定义内容，比如发送一些测试用的字符等，接收方在回复时也可以把这些数据原样返回，便于测试网络传输的准确性等情况。 |

4.结合ICMP报文记录画出数据交互示意图，并描述tracert工作原理



1. 实验数据及数据的解释与对比分析：

*【考察1）是否能收集和记录有效数据。2） 是否能对数据进行解释和对比分析。】*

如上

1. 实验结论：

*【应给出有效的实验结论，并需含对“网络行为进行分析、模拟和验证，提高解决工程问题的能力和效率”的体会和评价】*

在本次关于网络层协议 IP 和 ICMP 的实验中，我收获颇丰，不仅在知识层面上对相关概念有了更深入透彻的理解，在实践技能和思维拓展方面也得到了显著的提升。

通过实际操作 Wireshark 软件进行数据包的捕获与分析，我第一次如此近距离地 “窥探” 到网络数据传输的内在奥秘。在剖析 IP 数据报时，亲眼目睹其首部中各个字段的设置与作用，如版本号明确了所采用的 IP 协议版本，首部长度界定了首部的大小范围，总长度告知整个数据报的规模，标识、标志与片偏移则巧妙地处理了数据报的分片与重组问题，生存时间确保数据报不会在网络中无休止地循环，协议字段揭示了上层所承载的协议类型，首部校验和保障数据报首部的完整性，源地址和目的地址精准定位了数据的发送端与接收端。这些原本抽象的理论知识，在真实的数据包面前变得鲜活而具体，让我深切领悟到网络层在数据传输过程中是如何精心 “包装” 与 “投递” 数据的。

对于 ICMP 报文，我从其工作原理的理论认知深入到了实际应用场景的深刻体会。ICMP 作为网络层的 “情报员”，它通过各种类型的报文来传达网络状态信息。在实验中，观察到 Echo Request 和 Echo Reply 报文在 ping 操作中的交互，清晰地理解了它们如何用于检测网络的连通性。当发送 Echo Request 后，焦急地等待 Echo Reply 的过程中，我仿佛看到了网络中的数据请求与回应在悄然进行。而当遇到网络故障时，ICMP 的差错报文，如目的不可达、超时等类型的报文，及时地反馈问题所在，让我意识到它在网络故障诊断和调试中的关键价值。这使我明白，在实际的网络运维中，ICMP 是一把不可或缺的 “利器”，能够帮助我们迅速定位和解决网络问题。

在实验过程中，我也遇到了不少挑战。例如，面对海量的数据包，如何准确地筛选出与研究目标相关的数据包成为一大难题。起初，我在设置捕获过滤器和显示过滤器时常常出现错误，导致要么捕获到过多无用数据，要么错过关键信息。但通过不断地尝试、查阅资料和请教同学，我逐渐掌握了过滤器的使用技巧，能够根据源地址、目的地址、协议类型等多种条件精准地筛选数据包。这一过程不仅提升了我的问题解决能力，也培养了我的耐心和细心。

此外，这次实验还让我对网络的复杂性和精密性有了全新的认识。网络中的每一个数据包都像是一个精心编排的 “旅行者”，它在网络层的指引下，穿越各种网络设备，跨越不同的网络拓扑结构，最终准确无误地抵达目的地。任何一个环节的失误，都可能导致数据包的 “迷失” 或传输错误。这使我深刻体会到网络工程是一个需要高度专业素养和严谨态度的领域，每一个细节都关乎整个网络的稳定与高效运行。

总的来说，本次实验是一次极具价值的学习经历。它将理论知识与实践操作紧密结合，让我在网络领域的学习道路上迈出了坚实的一大步。我相信，这次实验所积累的知识和经验，将为我今后进一步深入学习网络技术，如路由与交换技术、网络安全等打下坚实的基础，也让我对未来在网络相关领域的探索充满了期待和信心。