《网络与通信》课程实验报告

实验三：数据包结构分析

姓名	孔馨怡	院系	计算机学院	学号	22122128
任课教师	何冰		指导教师	何冰	
实验地点	计算机楼 708		实验时间	2024 年 9 月 18 日	
实验课表现	出勤、表现得分(10)		实验报告得分(40)	实验总分	
	操作结果得分(50)				

实验目的：

1. 了解 Sniffer 的工作原理，掌握 Sniffer 抓包、记录和分析数据包的方法；
2. 在这个实验中，你将使用抓包软件捕获数据包，并通过数据包分析每一层协议。

实验内容：

使用抓包软件捕获数据包，并通过数据包分析每一层协议。

实验要求：（学生对预习要求的回答）（10 分）　　得分：

- 常用的抓包工具

答：

Wireshark

Wireshark 是一款功能强大的网络协议分析工具，支持实时捕获并显示网络数据包的详细内容。它支持数百种网络协议的解析，可以深入分析网络通信中的问题。Wireshark 还提供了图形化界面，便于过滤、排序和跟踪数据包流。它常用于复杂的网络故障排除、流量监控和安全分析。

Fiddler

Fiddler 是一个专门用于HTTP/HTTPS流量抓包和调试的代理工具，常用于Web开发和API测试。它能够捕获浏览器和应用程序发出的所有网络请求，并显示详细的请求和响应信息。Fiddler 还支持修改网络请求和响应，便于调试和测试不同场景下的Web应用行为。它具备强大的插件扩展功能，适合高级用户进行定制开发。

Sniffer

Sniffer 是一类用于监控、捕获和分析网络流量的工具。它可以被动地监听网络中流经的所有数据包，帮助用户分析网络性能、检测异常流量，或排查网络问题。Sniffer 能够解析不同网络协议的数据包，展示详细的包内容，比如源IP地址、目标IP地址、协议类型等信息。除了网络故障排除，它也可以用于安全领域，检测网络入侵或未授权的通信。

TShark

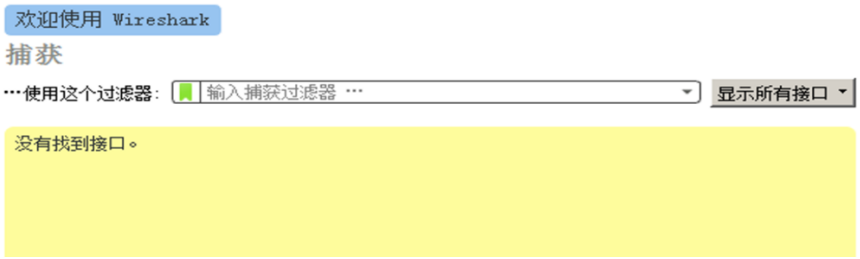
TShark 是 Wireshark 的命令行版本，适用于无图形界面的环境下进行数据包捕获和分析。它具备与Wireshark相同的强大协议解析能力，可以将数据包保存为多种格式供日后分析。TShark 还支持实时显示抓包数据，便于快速分析网络问题。它特别适合服务器、远程机器和脚本化抓包操作。

tcpdump

tcpdump 是一个命令行抓包工具，广泛用于Linux/Unix系统下的网络数据捕获。它可以抓取网络接口上流经的所有数据包，并支持将数据包保存到文件以供日后分析。tcpdump支持丰富的过滤表达式，可以选择性捕获符合条件的流量。由于其轻量级和灵活性，它被广泛应用于服务器和嵌入式设备的网络调试。

实验过程中遇到的问题如何解决的？（10 分）	得分：

问题 1：在 Win11 虚拟机中安装 wireshark 后打开找不到接口，网络配置等都没有问题。虚拟机内外的网关、ip、掩码等都没有问题。



解决一：在终端输入：net start npf

试图启动 WinPcap 驱动程序（Network Packet Filter，简称 NPF）

```
C:\Users\MAC>net start npf
服务名无效。

请键入 NET HELPMSG 2185 以获得更多的帮助。
```

失败

在终端机输入：sc start npcap

查看 Npcap 的状态。如果 Npcap 服务没有运行，尝试重新启动它

```
C:\Users\MAC>sc start npcap
[SC] StartService: OpenService 失败 5:

拒绝访问。
```

失败

此时可能就是下载的 wireshark 的版本和目前的 WinPcap 或 Npcap 不兼容等的问题。

解决二：

卸载 WinPcap/Npcap,上官网确认文档重新选择合适的版本

很遗憾，还是不行。此时我觉得问题出在我的虚拟机网络设置和匹配的 wireshark 上。

解决三：

卸载 wireshark，将虚拟机的网络设置桥接模式/共享网络等切换，尝试设置并测试。

上 wireshark 下载虚拟机适配等版本。重新安装且安装匹配 Npcap。

解决四：

设置虚拟机 ipv4/转发端口等：

问题 2：如何进行过滤？

Wireshark 数据包过滤命令使用：

一、根据 IP 地址过滤

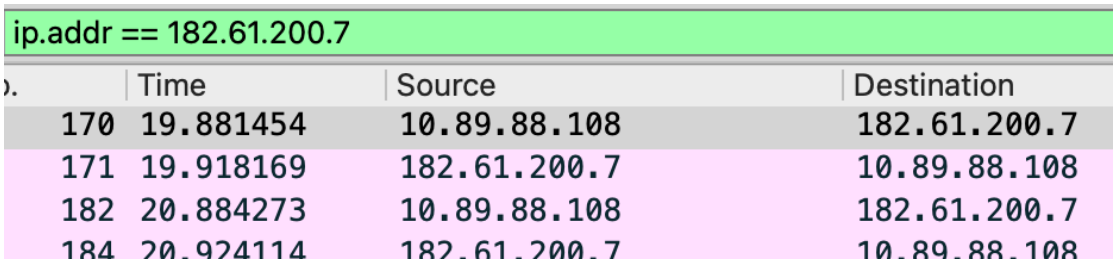
Case 1: 筛选出源或目的 IP 为 192.168.3.77 的数据包：ip.addr == 192.168.3.77

Case 2: 筛选出源 IP 为 182.254.3.77 的数据包：ip.src == 182.254.3.77

Case 3: 筛选出目的 IP 为 192.168.1.114 的数据包：ip.dst == 192.168.1.114

Eg: ping 百度 然后过滤

```
(base) kongxinyi@kongxiaoxindeMacBook-Pro ~ % ping www.baidu.com
PING www.a.shifen.com (182.61.200.7): 56 data bytes
64 bytes from 182.61.200.7: icmp_seq=0 ttl=44 time=36.908 ms
64 bytes from 182.61.200.7: icmp_seq=1 ttl=44 time=40.172 ms
64 bytes from 182.61.200.7: icmp_seq=2 ttl=44 time=40.980 ms
64 bytes from 182.61.200.7: icmp_seq=3 ttl=44 time=32.094 ms
```



二、根据端口过滤

Case 1: 筛选源或目的端口为 80 的 TCP 数据包：tcp.port == 80

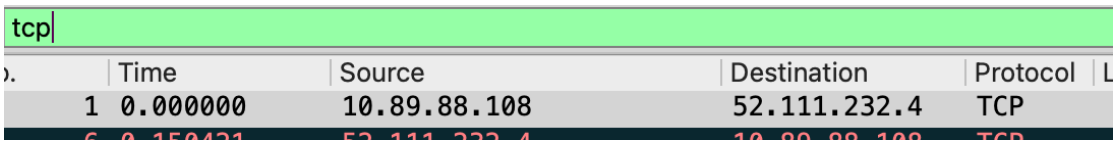
Case 2: 筛选目的端口为 80 的 TCP 数据包：tcp.dstport == 80

Case 3: 筛选源端口为 80 的 TCP 数据包：tcp.srcport == 80

Case 4: 筛选源或目的端口为 1234 的 UDP 数据包：udp.port == 1234

Case 5: 筛选源端口为 1234 的 UDP 数据包：udp.srcport == 1234

Case 6: 筛选目的端口为 1234 的 UDP 数据包：udp.dstport == 1234



三、根据协议过滤

Case 1: 筛选 HTTP GET 请求的数据包：http.request.method == GET

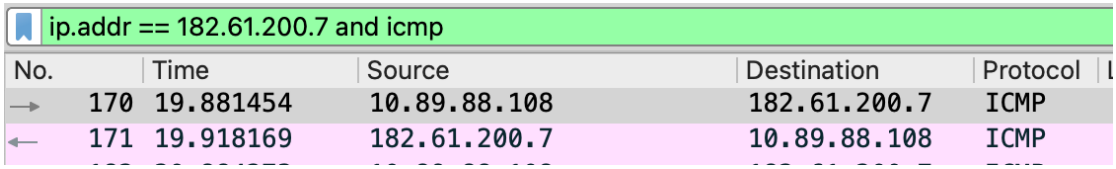
Case 2: 筛选 HTTP POST 请求的数据包：http.request.method == POST

四、根据 Payload Type 过滤

筛选 Payload Type 为 111 的数据包：rtp.p_type == 111

五、根据组合条件过滤

适当使用 && || ！ 等运算符将以上的各种过滤命令连接在一起，做到更准确的过滤。



问题 3：

无

本次实验的体会（结论）（10 分）	得分：

通过本次使用 Wireshark 进行数据包捕获和分析的实验，我对网络通信各协议层的运作机制有了更加深刻的理解。掌握了抓取不同类型数据包的操作，学会了通过精准的过滤规则筛选出特定的数据包，从而大大提升了分析效率。在逐层解析数据包时，尤其是以太网帧、IP 包以及 ICMP 协议的结构，加深了对这些协议细节的理解。整个实验过程让我更加清晰地掌握了数据从物理层到应用层的传输路径，提升了对计算机网络模型的理解，这对今后的网络学习和实践起到了重要的辅助作用。

思考题：（10 分）

思考题 1：（4 分）	得分：

写出捕获的数据包格式。

以抓包 ping www.baidu.com 的操作为例

```
> Frame 170: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Apple_86:e5:d8 (9c:3e:53:86:e5:d8), Dst: RuijieNetwor_3f:e1:1e (80:05:88:3f:e1:1e)
> Internet Protocol Version 4, Src: 10.89.88.108, Dst: 182.61.200.7
> Internet Control Message Protocol
```

Frame 170: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0

层次：物理层

内容：帧的长度和数据包捕获时的接口信息，表明该数据包在物理介质上传输时总共有 98 字节，接口 ID 为 0，接口名称为 en0（表示 Wi-Fi 网络）。

Ethernet II, Src: Apple_86:e5(9c:3e:53:86:e5), Dst: RuijieNetwor_3f:e1:1e (80:05:88:3f:e1:1e)

层次：数据链路层

内容：这是数据链路层的以太网帧信息，包括源 MAC 地址和目的 MAC 地址。这里的源 MAC 地址是 9c:3e:53:86:e5:d8，目的 MAC 地址是 80:05:88:3f:e1:1e。这一层负责在同一局域网内的节点之间传输数据帧。

Internet Protocol Version 4, Src: 10.89.88.108, Dst: 182.61.200.7

层次：网络层

内容：这是网络层的 IPv4 协议信息。源 IP 地址为 10.89.88.108，目的 IP 地址为 182.61.200.7。这一层负责在不同的网络节点间传输数据包。

Internet Control Message Protocol

层次：传输层（Transport Layer）

内容：这是传输层的信息，使用的是 ICMP 协议（Internet 控制消息协议），该协议用于诊断网络通信问题（如 Ping 请求/回应）。这一层负责在网络节点之间进行可靠的数据传输。

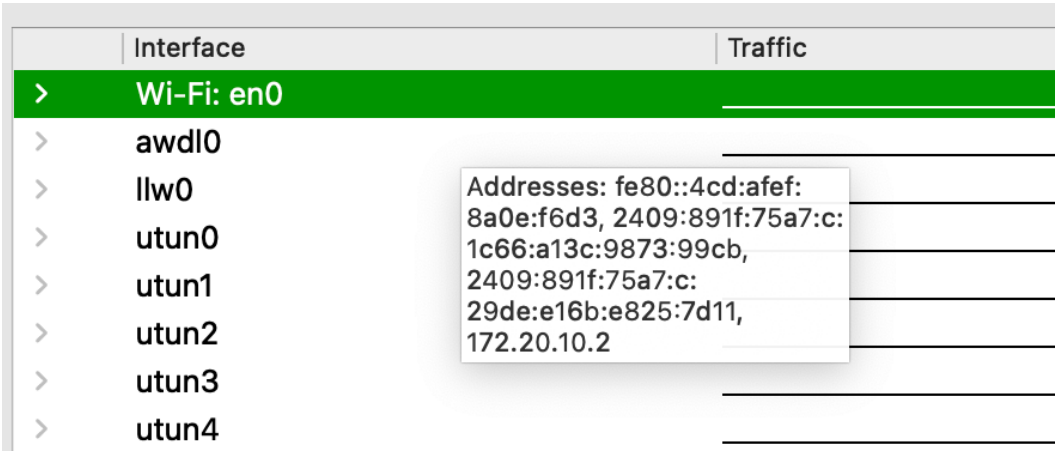
思考题2：（6分）	得分：

写出实验过程并分析实验结果。

打开Wireshark，点击菜单栏的Capture（捕获）中的Options（选项），来选择想要捕获数据包的网络接口：

这里我mac一般选的wifi：en0，如果是Win11虚拟机就是选择ip和主机一个段的以太网。

可以在选择的时候将鼠标放在上面查看ip等信息。



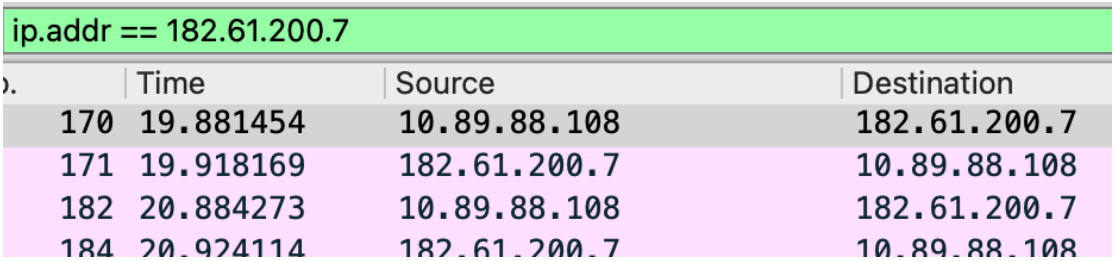
选择好之后start开始，这时候wireshark以及开始捕获数据包了。

此时开始你想要进行的操作，我这里是terminal ping了一下百度：

```
(base) kongxinyi@kongxiaoxindeMacBook-Pro ~ % ping www.baidu.com
PING www.a.shifen.com (182.61.200.7): 56 data bytes
64 bytes from 182.61.200.7: icmp_seq=0 ttl=44 time=36.908 ms
64 bytes from 182.61.200.7: icmp_seq=1 ttl=44 time=40.172 ms
64 bytes from 182.61.200.7: icmp_seq=2 ttl=44 time=40.980 ms
```

操作完成之后就可以点击菜单栏的红色方框来结束捕获的操作：

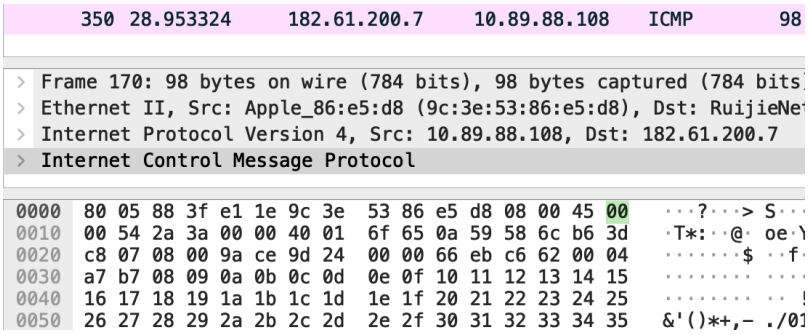
这边terminal 在 ping之后会有一个 ip，复制了以后可以作为筛选的条件：



筛选结束以后就可以看到我们的这个ping操作所发出/接收的数据包了。

此时抓包和筛选操作完成，我们可以开始分析捕获的数据包了。

选择其中一行 例如序号170的这一个数据包，下方区域会有(从上到下)数据详细区，数据字节区：



这时候我们就可以查看数据详细区的内容来分析每一层和协议等的内容了：

```
Frame 170: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
    Section number: 1
  Interface id: 0 (en0)
      Interface name: en0
      Interface description: Wi-Fi
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 19, 2024 14:36:18.305166000 CST
    UTC Arrival Time: Sep 19, 2024 06:36:18.305166000 UTC
    Epoch Arrival Time: 1726727778.305166000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.033403000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 19.881454000 seconds]
    Frame Number: 170
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
```

上图为物理层，是Wireshark中数据包的详细描述信息，包括有关数据包捕获和解码的各种数据。下面是对部分字段的详细解释：:

Frame 170: 数据包的帧编号，是Wireshark捕获的数据包的序号。

98 bytes on wire (784 bits): 数据包在网络上传输的实际字节数和比特数。

98 bytes captured (784 bits): Wireshark捕获的数据包的字节数和比特数，与传输的字节数相同，表示没有丢失数据。

Interface id: 0 (en0): 捕获数据包的网络接口名称，通常表示Wi-Fi接口。

Encapsulation type: Ethernet (1) 数据包的封装类型，这里是以太网（Ethernet），编号为 1。

Arrival Time: Sep 19, 2024 14:36:18.305166000 CST ：
数据包到达捕获设备的时间，格式为 年月日 时分秒.微秒，时区为 CST（中国标准时间）。

Frame Number: 170 数据包在捕获文件中的编号，这里是第170个数据包。

Frame Length: 98 bytes (784 bits)
数据包的总长度，包括以太网头和所有数据部分，长度为98字节（784位）。

Capture Length: 98 bytes (784 bits)
实际捕获的数据包长度，通常等于帧长度，表示没有数据丢失。

```
Ethernet II, Src: Apple_86:e5:d8 (9c:3e:53:86:e5:d8), Dst: RuijieNetwor_3f:e1:1e (80:05:88:3f:e1:1e)
  Destination: RuijieNetwor_3f:e1:1e (80:05:88:3f:e1:1e)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Apple_86:e5:d8 (9c:3e:53:86:e5:d8)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
    [Stream index: 0]
```

上图为数据传输层，主要描述的详细信息是以太网帧的基本结构：

目的MAC地址：80:05:88:3f:e1:1e

源MAC地址: 9c:3e:53:86:e5:d8

类型字段（Type）指示上层协议：IPV4

Tips：LG bit: Globally unique address (factory default): LG（Locally or Globally unique）位，0 表示这是一个全球唯一地址，通常是默认设置。

IG bit: Individual address (unicast): IG（Individual Group）位，0 表示这是一个单播地址（即特定的唯一地址，而不是广播地址）。

```
Internet Protocol Version 4, Src: 10.89.88.108, Dst: 182.61.200.7
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 84
    Identification: 0x2a3a (10810)
  000. .... = Flags: 0x0
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x6f65 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.89.88.108
    Destination Address: 182.61.200.7
    [Stream index: 9]
```

上图为网络层，这部分信息是关于IPv4数据包的详细描述。下面是对部分字段的详细分析：

Version: 4 表示IP协议的版本号。4 表示这是一个IPv4数据包。IPv6会用 6 表示。

Header Length: 20 bytes (5) ：

IP头部的长度，以字节为单位。这里是 20 bytes。头部长度字段表示IP头部的长度，单位是32位字（即4字节）。5 表示头部长度为5个32位字（5 × 4 = 20字节）。

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

用于流量优先级和拥塞通知。0x00 表示默认服务类型。

DSCP: CS0: 区分服务代码点Differentiated Services Codepoint，CS0 表示默认服务等级。

ECN: Not-ECT: 显式拥塞Explicit Congestion Notification，Not-ECT 表示没有拥塞通知能力。

Total Length: 84 ：

包含IP头和数据部分的总长度，单位为字节。这里的 84 表示整个IP数据包的长度为84字节。

Identification: 0x2a3a (10810) ：

用于唯一标识IP数据包，尤其是分片的重组。0x2a3a 是标识符的十六进制值，10810 是其十进制值。

Time to Live: 64

(TTL): 数据包的生存时间，表示数据包在网络中可以经过的最大路由器数量。每经过一个路由器，TTL 值会减 1。这里的 64 是初始TTL值。

Protocol: ICMP (1) ：

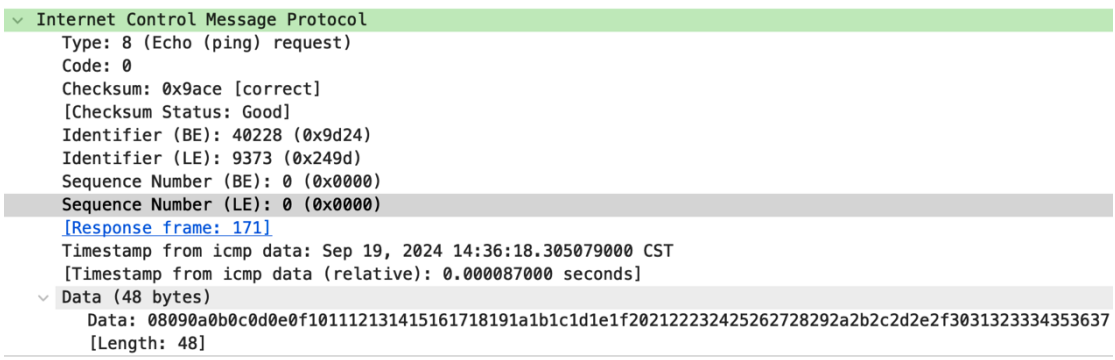
上层协议标识符。1 表示上层协议是ICMP（Internet Control Message Protocol）。

Header Checksum: 0x6f65 [validation disabled] ：

用于验证IP头部是否有错误的校验和。0x6f65 是校验和值，[validation disabled] 表示校验和验证被禁用。

Source Address: 10.89.88.108 源IP地址。

Destination Address: 182.61.200.7 目的IP地址。



上图信息描述了ICMP（Internet Control Message Protocol）数据包的内容，主要用于网络诊断和错误报告。以下是对部分字段的详细分析：

Type: 8 (Echo (ping) request) ：ICMP消息的类型。8 表示这是一个Echo请求（ping请求），用于测试网络连接或响应时间。

Code: 0 ：

ICMP消息的子类型。0 表示这是一个标准的Echo请求，没有附加的特定子代码。

Checksum: 0x9ace [correct] ：用于验证ICMP数据包在传输过程中是否出现错误的校验和。0x9ace 是校验和值，[correct] 表示校验和验证通过，数据包没有错误。

[Response frame: 171] ：这是对应的响应数据包的帧编号。171 表示这是响应数据包的帧编号，在这个案例中响应包的帧编号是171。

Timestamp from icmp data: Sep 19, 2024 14:36:18.305079000 CST ：

数据包中的时间戳，表示发送或接收数据包的精确时间。这个时间戳表示数据包发送的时间。

[Timestamp from icmp data (relative): 0.000087000 seconds] ：

Relative Timestamp: 相对于捕获开始的相对时间戳，表示数据包与前一个数据包之间的时间差。0.000087000 seconds 表示相对时间差。

Data (48 bytes) ：

ICMP消息中的数据部分。这里 48 bytes 表示数据部分的长度为48字节。

实际传输的内容（有效负载数据）：

08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c……

指导教师评语：

日期：