# 计算机网络 IP 实验报告

姓名：陈鹤影　　学号：PB21061287　　日期：2022.11.12

## 一、 实验目的：

1. 了解 ICMP 发送回显请求得到通往目的地路由的原理。
2. 了解 ICMP 分片原理。
3. 分析 ICMP 分片中不同分片 IP 头差异。
4. 掌握 pingplotter 的应用。
5. 观察 TTL 超时报文。

## 二、 实验流程及问题回答：

### Part 1：Capturing packets from an execution of traceroute

1. **实验原理：**

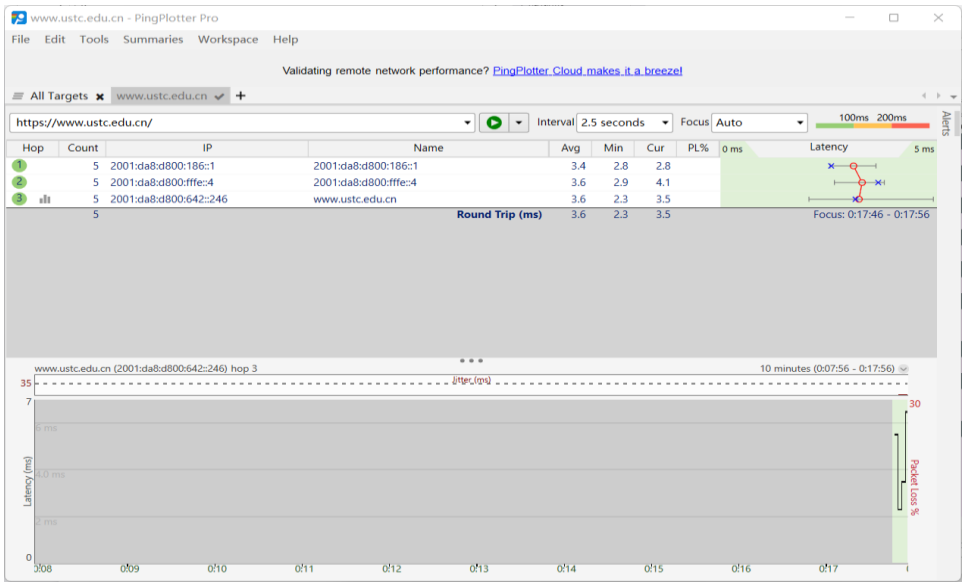
通过 traceroute 程序我们可以知道信息从本地主机到互联网另一端的主机的路径（当然每次数据包由某一相同源（source）发出，到达某一相同的目的地(destination)走的路径可能会不同，但大部分时候经过的路由是相同的）。linux 系统中一般使用 traceroute,在 MS Windows 中使用 tracert。通过向目标发送不同 IP 生存时间 (TTL) 值的“Internet 控制消息协议 (ICMP)”回应数据包，Tracert 诊断程序确定到目标所采取的路由。要求路径上的每个路由器在转发数据包之前至少将数据包上的 TTL 递减 1。数据包上的 TTL 减为 0 时，路由器应该将“ICMP 已超时”的消息发回源系统。Tracert 先发送 TTL 为 1 的回应数据包，并随后的每次发送过程将 TTL 递增 1, 直到目标响应或 TTL 达到最大值, 从而确定路由。通过检查中间路由器发回的“ICMP 已超时”的消息确定路由。某些路由器不经询问直接丢弃 TTL 过期的数据包，这在 Tracert 实用程序中看不到。

2. **实验流程：**

为了改变 ICMP 回显请求（ping）消息的长度，后续实验选用 pingplotter 而不使用 Windows 提供的 tracert。在 pingplotter 中可以修改 ping 消息的长度、追踪间隔时间和间隔数。

1）启动 Wireshark 抓包。

2）启动 pingplotter，在“Address to Trace Window.”处输入目的地址，在“# of times to Trace”处输入 3 以避免收集过多数据（新版中没有这一选项，故在实验中选择 Auto 显示所有数据）。修改 Packet Size 为 56，执行程序。可以看到如下界面（以 https://www.ustc.edu.cn/为例）：

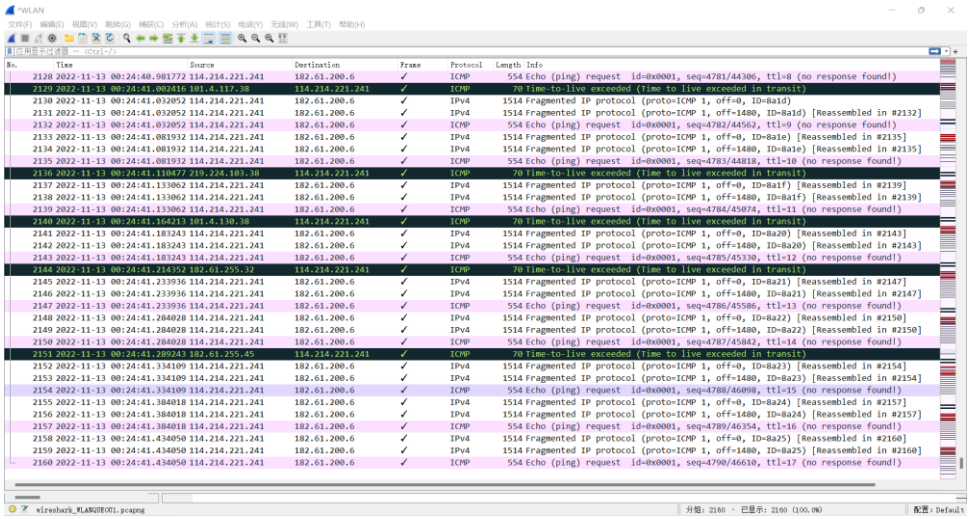



（图 1-1 pingplotter 执行界面）

3）修改 Packet Size 为 2000，重新运行 pingplotter。

4）修改 Packet Size 为 3500，重新运行 pingplotter。

5）结束 Wireshark 抓包。抓包结果如图所示：




（图 1-2 Wireshark 抓包结果）

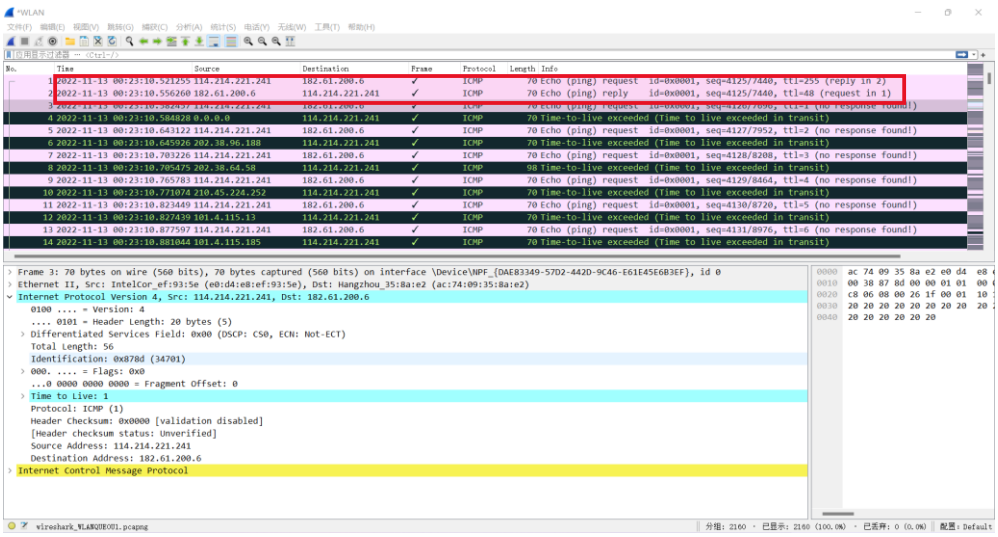
## Part 2：A look at the captured trace

1. **实验内容：**

   分析获得的 trace。

2. **实验流程及问题回答：**

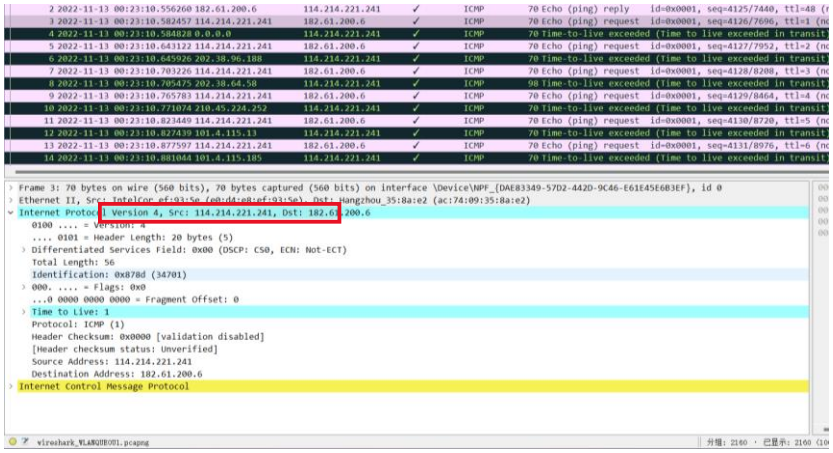
   选择计算机发送的第一条 ICMP 回显请求消息（其实在实验中 ttl=1 的 ping 请求是第二条 ICMP 回显请求），并在数据包详细信息窗口中展开数据包的 Internet 协议部分。



（图 1-3 “第一条”ICMP 回显请求消息）

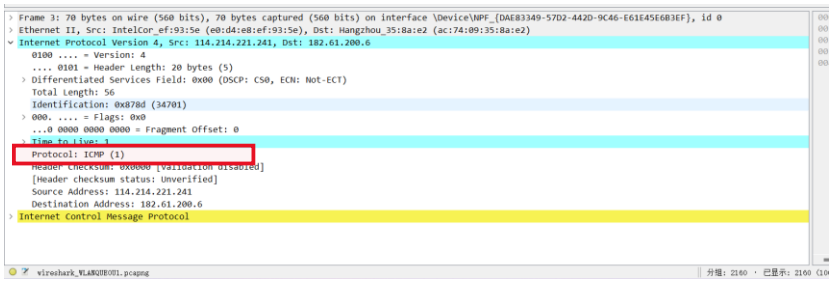
1） **What is the IP address of your computer?**

**Ans：**本机地址：114.214.221.241。



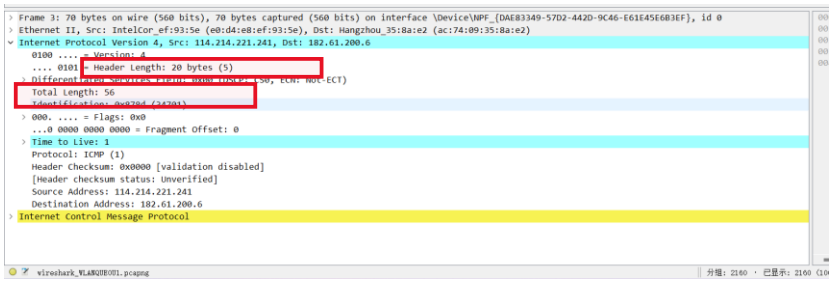
2） **Within the IP packet header, what is the value in the upper layer protocol field?**

**Ans：**1（ICMP 协议）。



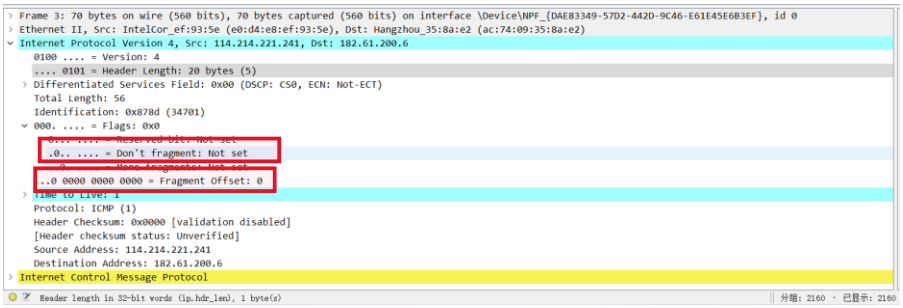
3） **How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.**

**Ans：**Hearder length：20 bytes。Payload length：36 bytes（总长度 56-IP 头长度 20=36）。

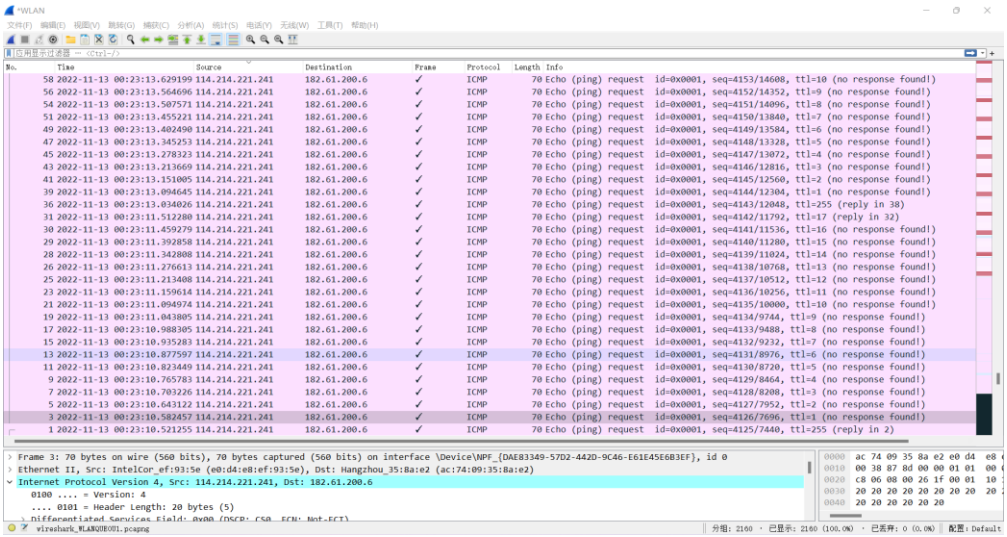



**4) Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.**

**Ans：** IP 数据报没有被分片。可以看到 MF 标志并未置 1，且偏移量为 0。



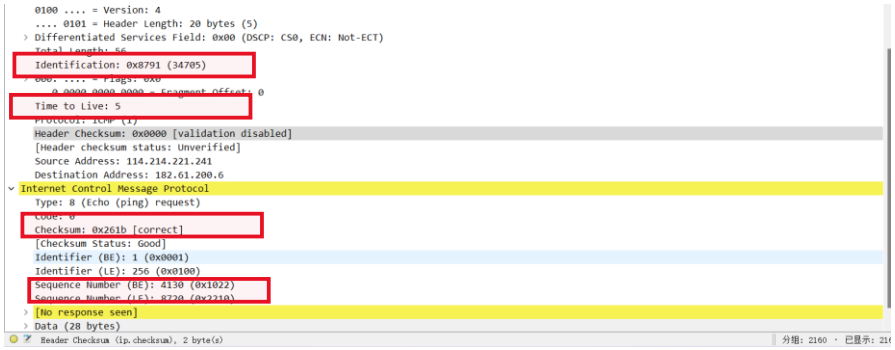

根据 IP source address 对数据包进行分类。选择计算机发送的第一条 ICMP 回显请求消息，并在“所选数据包头部详细信息”窗口中展开 Internet 协议部分。在“捕获的数据包列表”窗口中，可以看到第一个 ICMP 消息下面的所有后续 ICMP 消息（可能还夹杂着由计算机上运行的其他协议发送的数据包）。



**（图 1-4 根据 IP source 分类的显示结果）**

**5) Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?**

**Ans：** Identification、ttl、Checksum、seq。




6) **Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?**

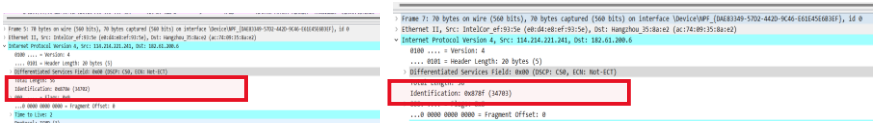
**Ans：** Stay constant：Version、Header Length、Differentiated Services Field、Protocol、address、Type、identifier。

Must Stay constant： Header Length、Protocol、address、Type。

Must change：Identification、ttl、Checksum、seq（如果分片还有 Fragment Offset）。

7) **Describe the pattern you see in the values in the Identification field of the IP datagram?**

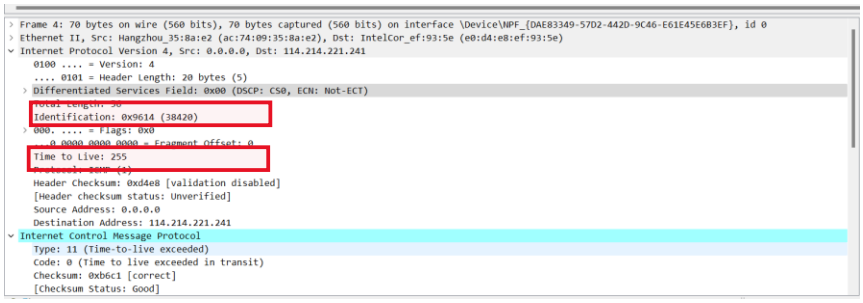
**Ans：** 每发送一个新的回显请求报文 identification 值加 1。



接下来分析最近路由发送回来的 TTL-exceeded replies 数据包：

8) **What is the value in the Identification field and the TTL field?**

**Ans：**第一个 ICMP TTL-Exceeded 消息中 IP 数据报头部的 Identification 字段是 38420，TTL 字段的值是 255。




9) **Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?**

**Ans：**Identification 的值会改变，因为每发送一个新的报文 Identification 的值会加一。观察到在同一次连续响应内 TTL 的值不会改变，但在不同次之间发生改变。经查询资料发现：ICMP 回显的报文中的 ttl 生成是根据主机上的系统和一些配置来决定的。比如：UNIX 及类 UNIX 操作系统 ICMP 回显应答的 TTL 字段值为 255，Compaq Tru64 5.0 ICMP 回显应答的 TTL 字段值为 64。

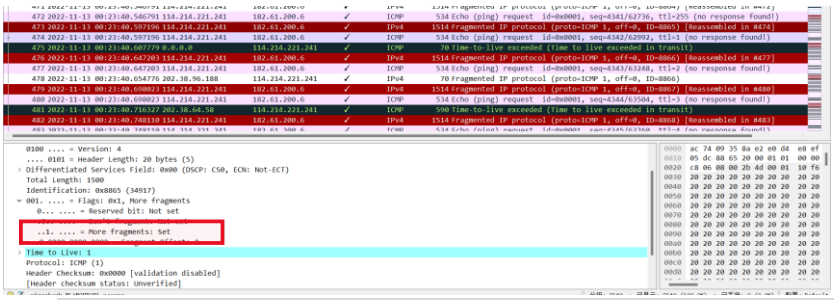
## Part 3：Fragmentation

1. **实验流程及问题回答：**

重新按照时间对数据包进行排序。

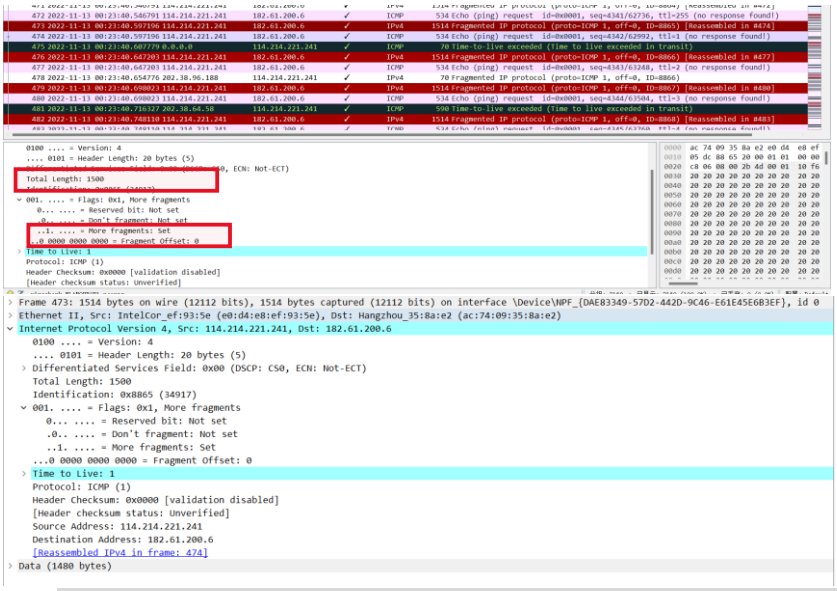
10) **Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?**

**Ans：**上述消息被分片了。



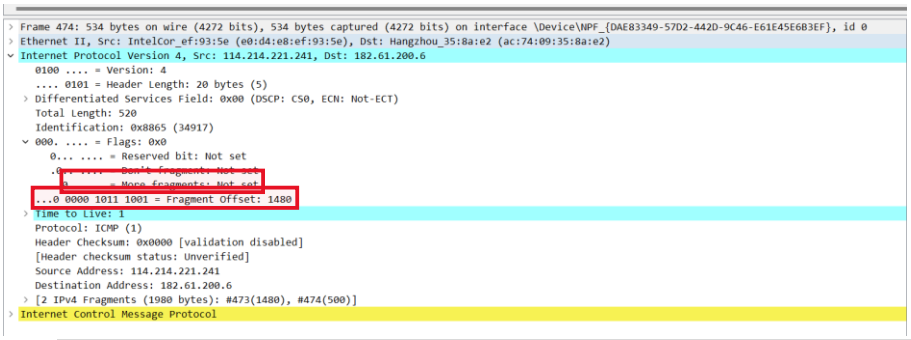
11) **Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?**

**Ans：**MF 标志置 1 但是偏移量为 0；Total length：1500 bytes。报文内容如下：



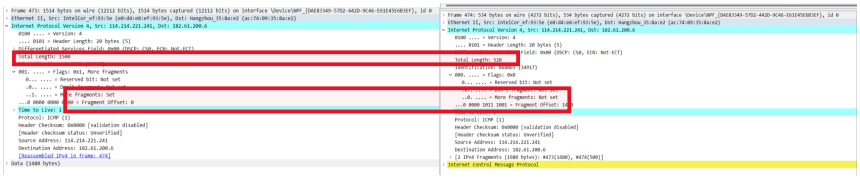
12) **Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are the more fragments? How can you tell?**

**Ans：**偏移量不为 0（Fragment Offset：1480）；之后没有更多的分片了，因为 MF 标志被置为 0。报文内容如下：




13） **What fields change in the IP header between the first and second fragment?**

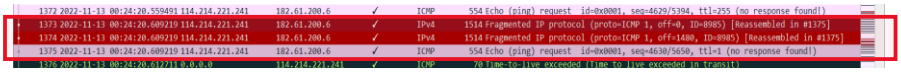
**Ans：** Total Length、MF、Fragment Offset。



下面对大小为 3500 的第一个回显请求消息进行分析。

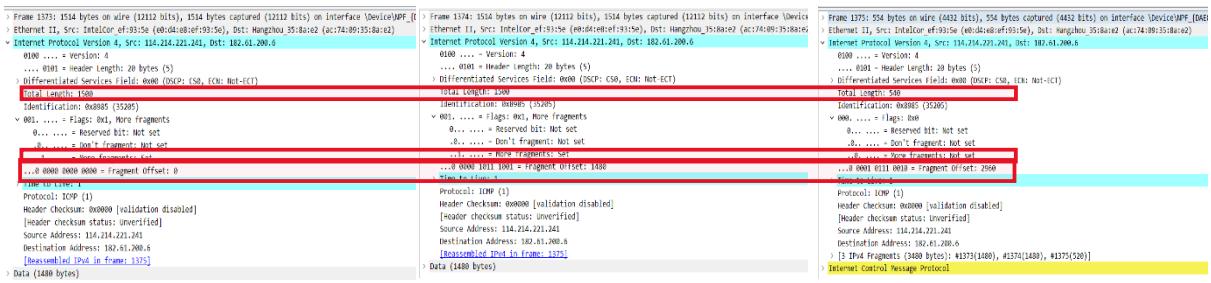
14） **How many fragments were created from the original datagram?**

**Ans：** 原先的数据报被分成了 3 个分片。




15） **What fields change in the IP header among the fragments?**

**Ans：** Total Length、MF、Fragment Offset。



# 补充内容：

## IP Protocol 字段详解（常用部分）：

| 数值 | 值描述 |
| --- | --- |
| 0 | 保留字段，用于IPv6(跳跃点到跳跃点选项) |
| 1 | Internet控制消息 (ICMP) |
| 2 | Internet组管理 (IGMP) |
| 3 | 网关到网关 (GGP) |
| 4 | 1P中的IP(封装) |
| 5 | 流 |
| 6 | 传输控制 (TCP) |
| 7 | CBT |
| 8 | 外部网关协议 (EGP) |
| 9 | 任何私有内部网关(Cisco在它的IGRP实现中使用) (IGP) |
| 10 | BBNRCC监视 |
| 11 | 网络语音协议 |

## IP 数据头结构

**版本（Version）**

4bit 的版本字段表示 IP 的版本号。如果为 0100 表示 IPv4，如果为 0110 表示 IPv6。

**首部长度（header length）**

4 比特的首部长度字段表示 IP 首部的总长度，其中包括选项字段（如果有）。该字段的表示的长度单位为 32bit（4 bytes），因此首部长度最大为 15（1111）*32bit=60bytes。在没有选项时，该字段的值为 5，表示普通的 IP 包头长度 20bytes。

**服务类型（Differentiated Service Field）**

8bit 的服务器类型（TOS）字段，其中前 3 个 bit 表示优先权（现在已经忽略该字段），随后的 4 个 bit 表示服务类型，按顺序分别表示为最小时延、最大吞吐量、最高可靠性、最小费用四种。这个 4 个 bit 中最多只能有 1 个 bit 置位，如果全是 0 则表示一般服务。最有 1 个 bit 为未用位，必须置 0。

**总长度（Total Length）**

16 个 bit 的总长度字段表示整个 IP 数据报的长度，以字节为单位。所以 IP 数据报的最大长度为 16 个 1=65535bytes。

此处的总长度字段与 MTU 是密切相关的，MTU 表示 IP 数据报的最大传输单元的大小，我们知道一般默认的 MTU 值为 1500，也就是说单个 IP 数据报的最大长度为 1500bytes。常规的无选项的 IP 包头长度为 20bytes，可以通过 MTU 计算出 DATA 数据字段的最大长度为 1480bytes。不仅如此，MTU 还与 TCP 协议报头中的 MSS 有关联，MSS 表示 TCP 数据报的最大传输长度，常规的 TCP 报头长度为 20bytes，又因为 TCP 报头是封装在 IP 包头上的，所以 MSS 一般为 1460bytes=MTU1500-IP 包头 20-TCP 报头 20。由于默认 MTU 为 1500bytes，所以网卡发送的数据帧的大小为 1518bytes，多了 18bytes 分别为：目的 MAC（6bytes）、源 MAC（6bytes）、type（2bytes）、CRC（4bytes）。

**标识（Identification）**

16bit 的标识字段唯一的标识主机发送的每一份数据报，由主机生成具有唯一性。

通常每发送一份报文该值加 1。该值在数据包分片时，会复制到每一个片中。所以在重组分片包的时候会观察该值，把该值相同的分片收集到一起重组，后面会继续讨论分片。

**标志（Flag）**

3bit 的标识字段每一位都有特定的含义，该字段主要用来分片和重组。第 1 个 bit 为保留位（Reserved Bit），一般置位 0。第 2 个 bit 为不分片位（Don't Fragment），该位在置 1 时表示不分片。第 3 个 bit 为更多片位（More Fragment），该位表示后面是否还有更多的分片，置位 1 时表示后面还有，所以除了最后一片报文，其他分片报文该位全部置 1。

**片偏移（Fragment Offset）**

13 比特的片偏移字段表示分片时，每一个分片的数据字段偏移原始数据报开始处的位置。比如原始数据报的数据字段总长为 1461bytes，使用 TCP 传输协议，那么对该数据报进行分片，第一片的 offset 字段为 0，第二片的 offset 字段就应该为 1460。就是这么理解就对了。

把一份超过 MTU 的数据报分片以后，这些分片只有在到达目的地才进行重新组装。在一组分片报文传输的过程时中间设备有可能会对该分片再次分片。如果采用 TCP 传输协议，在传输过程中丢失一个分片报文，那么整个 TCP 数据报都将被重新传送，不会只发送丢失的分片。其原因是 TCP 协议有自己的超时和重传机制。

在一组分片报文中，任何传输层的首部只会出现在第一个分片中。并且由于数据报被分片后 IP 包头的首部校验和将会重算。在分片时，除最后一片外，其他每一片的数据部分必须是 8 字节的整数倍。

**生存时间（Time To Live）**

8bit 的生成时间字段表示该 IP 数据包可以经过的路由器的最大数量。最大为 255 表示可以穿越 255 台路由，该字段采用减法的方式赋值，比如在开始时 8 个 bit 全部置位 1，没经过一台路由器该字段的值减 1。如果该字段的值减到了 0 还没有送达目的地，那么该 IP 数据包将被丢弃。最初设计这个字段是为了防止 IP 报文在网络中循环无限传输，占用带宽等问题。

**协议（Protocol）**

8bit 的协议字段表示在 IP 上层承载的是什么协议。比如：0x01 表示 ICMP 协议、0x06 表示 TCP 协议、0x11 表示 UDP 协议等。

**首部校验和（Header Checksum）**

16bit 的首部校验和字段用来使接收端检验收到的报文是否正确。该字段只对 IP 首部计算校验和不包含后面的数据字段。原因是 IP 的上层协议比如 ICMP、IGMP、TCP、UDP 协议的各自首部中均含有同时覆盖首部和数据的校验和。

计算方法：首先把首部中的该字段全部置 0，然后对首部中的每个 16bit 进行反码求和，得到的值就是该字段的值，填入后。将该数据包发给接收端后，接收端将进行相同的操作，对每个 16bit 进行反码求和（此时首部校验和字段为非 0 字段），所以计算后的值若为全 1 表示正确，否则表示收到的数据包不正确，丢弃数据包。

**选项（Option）**

一般不使用该字段，该字段的值以 32bit 为单位，不足时以 0 补充。(参考 IP 包头详解-Vinedeslly)