# 实验1.2

秦华谦21312683

1. **实验目的**
2. 熟练使用 Wireshark 软件，观察 IP数据报的基本结构，分析数据报的分片
3. 掌握基于 ICMP 协议的 ping 和 traceroute 命令及其工作原理
4. **实验内容**
5. **执行 ping 命令，观察 IP 数据报和 ICMP 询问报文的结构：**

我首先开启个人热点，接入个人电脑和iPad：

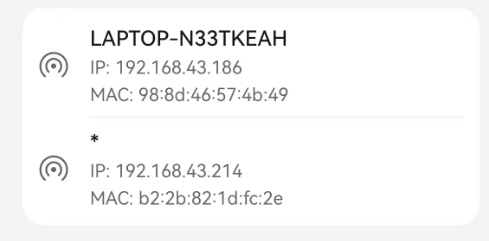


图2.1.1 电脑及iPad的IP地址

本次实验，我选择通过电脑（ip：192.168.43.186）ping iPad（ip：192：168：43：214），设置过滤条件为icmp，则可显示如下所捕获的ICMP数据包：

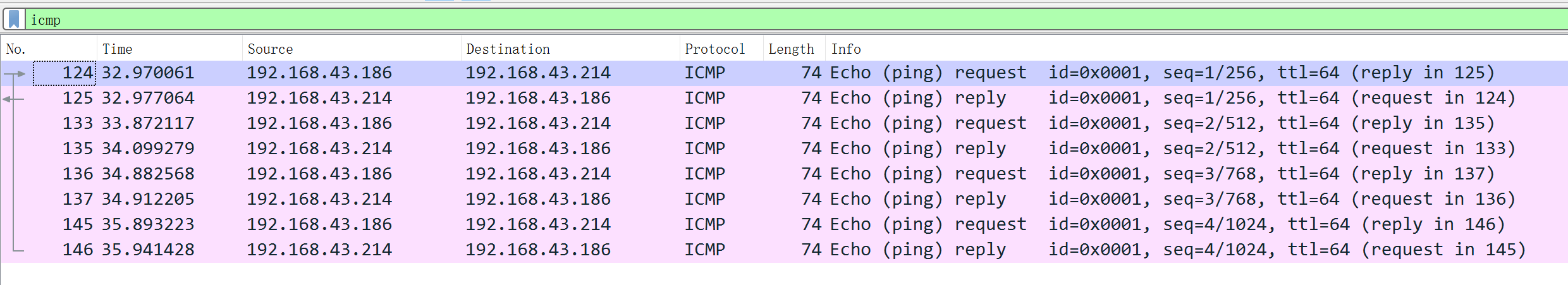


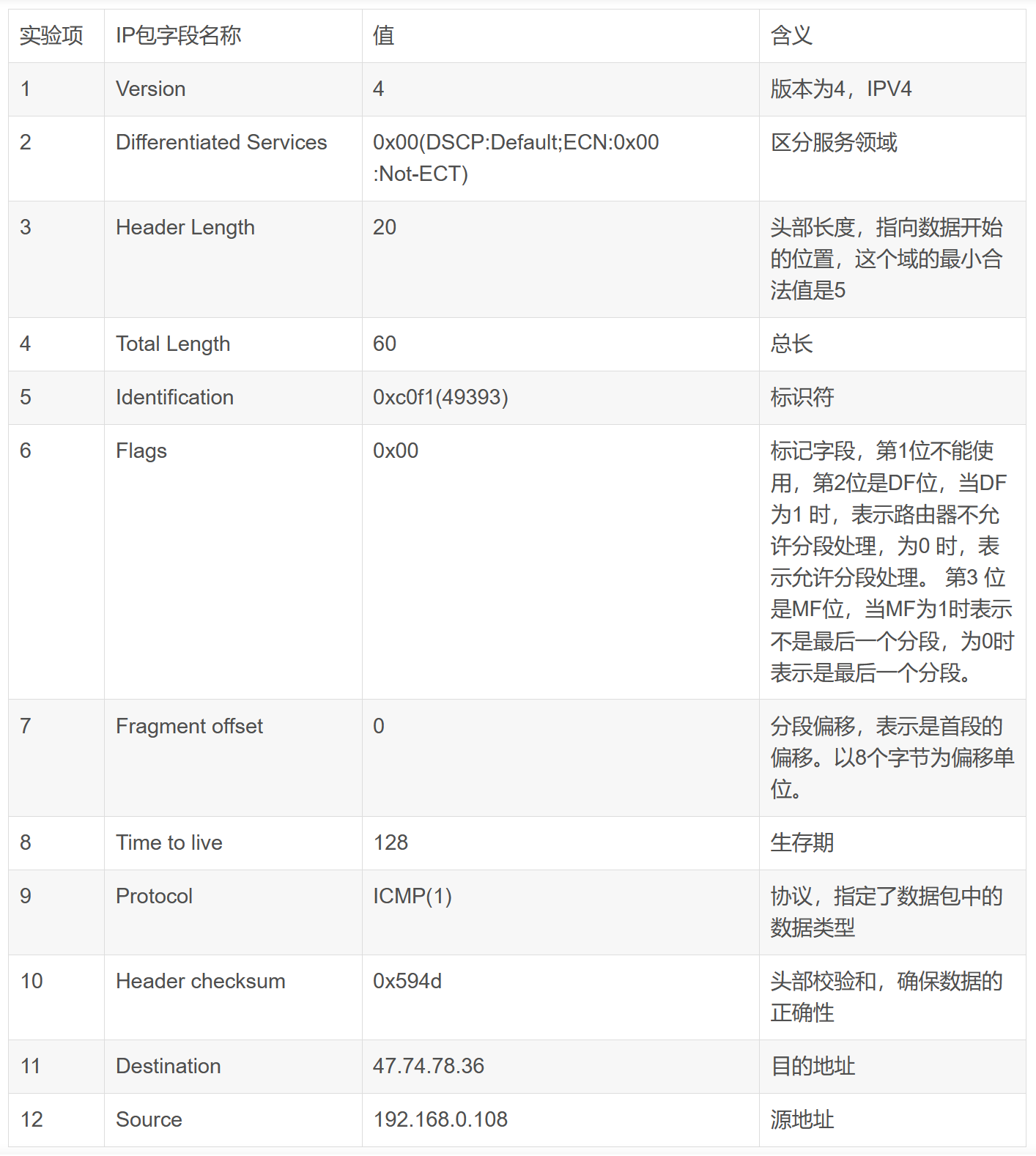
图2.1.2 Wireshark 监视器界面

选择打开第一个数据包进行分析：

* 1. **IP 数据报分析**



图2.1.3 IP 数据报结构示意图



·

表2.1.4 IP 数据报结构解释表

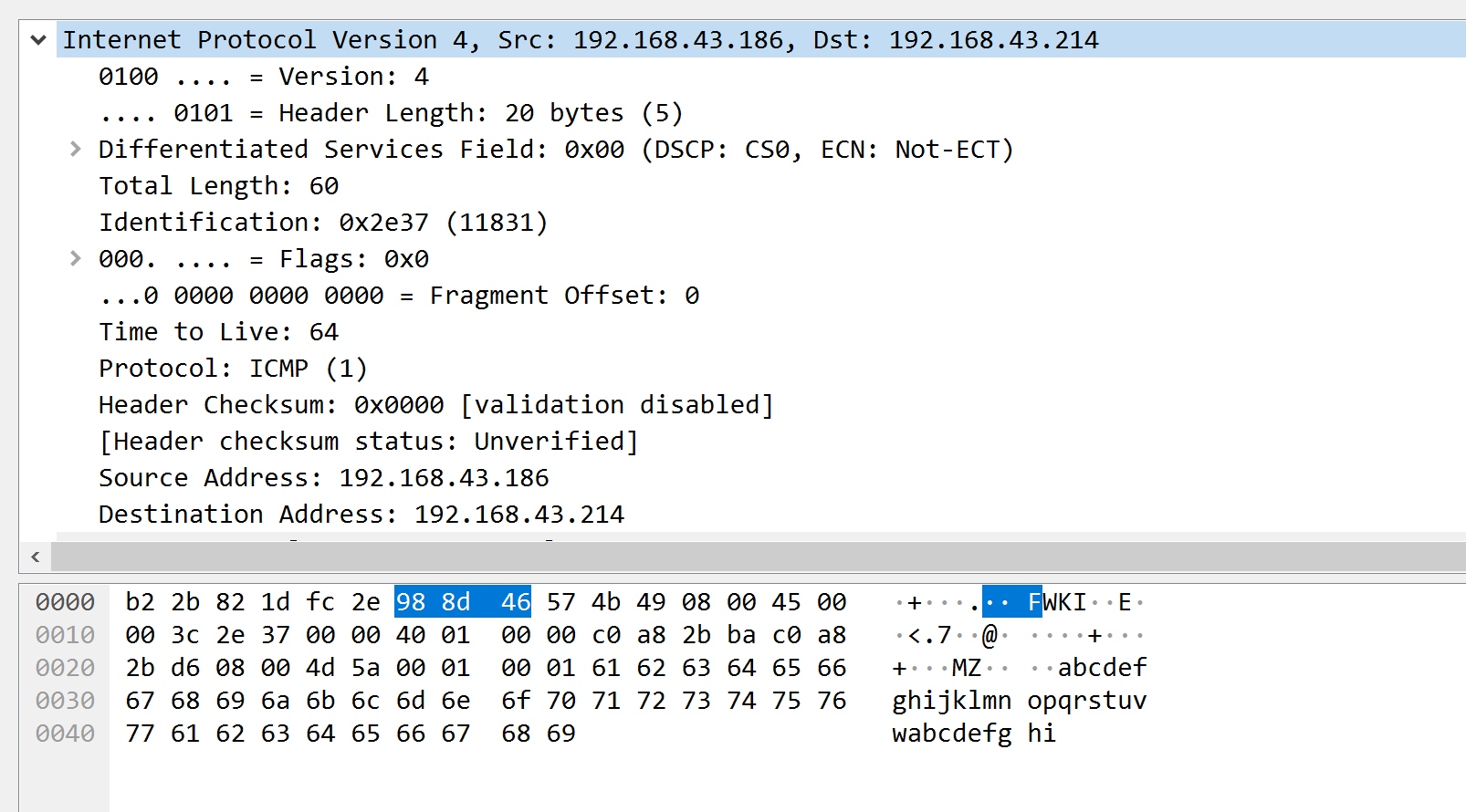


图2.1.5 查看 IP 数据报

基于此，我们便可以对本IP数据包进行分析：

<Ⅰ> 版本（version）的序号为：08:00，由文字部分也可得：版本为4，即IPv4；

<Ⅱ> 首部长度（header length）为20字节，即5个32字节；

<Ⅲ>服务类型（differentiated services）为：0x00，全为0，即没有特殊要求的一般服务；

<Ⅳ>总长度（total length）为60个字节；

<Ⅴ>标识（identification）为0x2e37（11831），即为标识符；

<Ⅵ>标记（flags）为0x00，第1位不使用，第2位DF位为0表示允许分段处理，第3位MF位为0表示这是最后一个分段；

<Ⅶ>片偏移（fragment offset）为0，表示首段偏移为0；

<Ⅸ>生存时间（time to live）为64，表示IP数据包还能生存多久，根据操作系统不同而变化；

<Ⅹ>协议（protocol）为ICMP；

<Ⅺ>头部校验和（header checksum）为0x0000，确保数据的准确性；

<Ⅻ>源地址（source）为：192.168.43.186，为我的电脑的IP地址；

<ⅩⅢ>目的地址（destination）为：192.168.43.214，为我的iPad的IP地址。

* 1. **ICMP 报文分析**

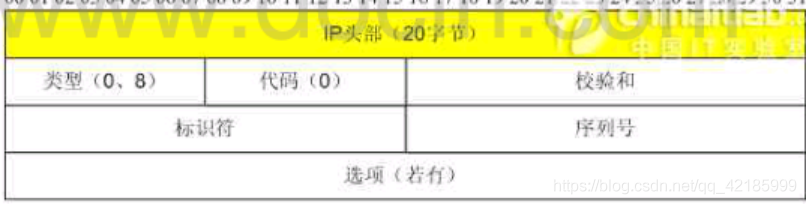


图2.1.6 ICMP 数据报结构示意图

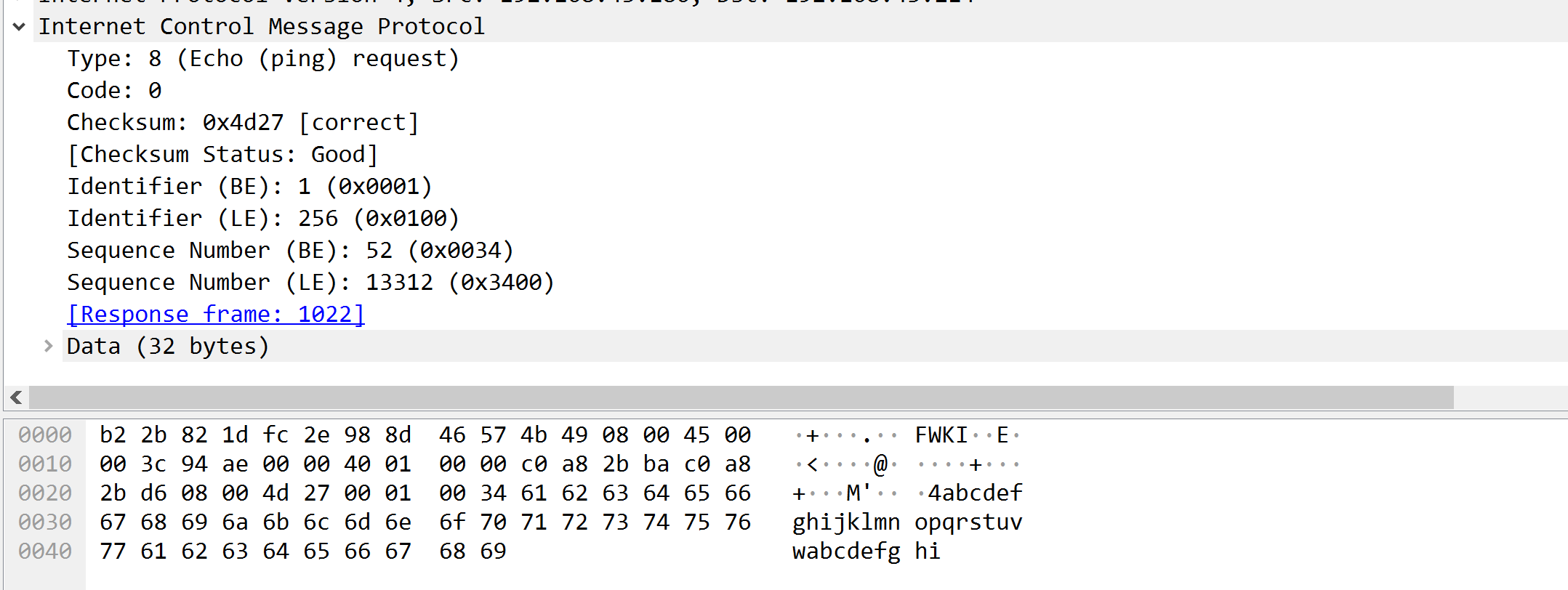


图2.1.7 查看 ICMP 报文（request）

基于此，我们便可以对本ICMP数据包（request）进行分析：

<Ⅰ> 类型（type）为：8（回显请求）；

<Ⅱ> 代码（code）为0；

<Ⅲ> 校验和（checksum）为0x4d27，确保数据的准确性，由显示可得本次正确；

<Ⅳ> 标识符，大端顺序（identifier（BE））为：1（0x0001）；

<Ⅴ> 标识符，小端顺序（identifier（LE））为：256（0x0100）；

<Ⅵ> 序列号，大端顺序（sequence number （BE））为：52（0x0034）；

<Ⅶ> 序列号，小端顺序（sequence number （LE））为：13312（0x3400）。

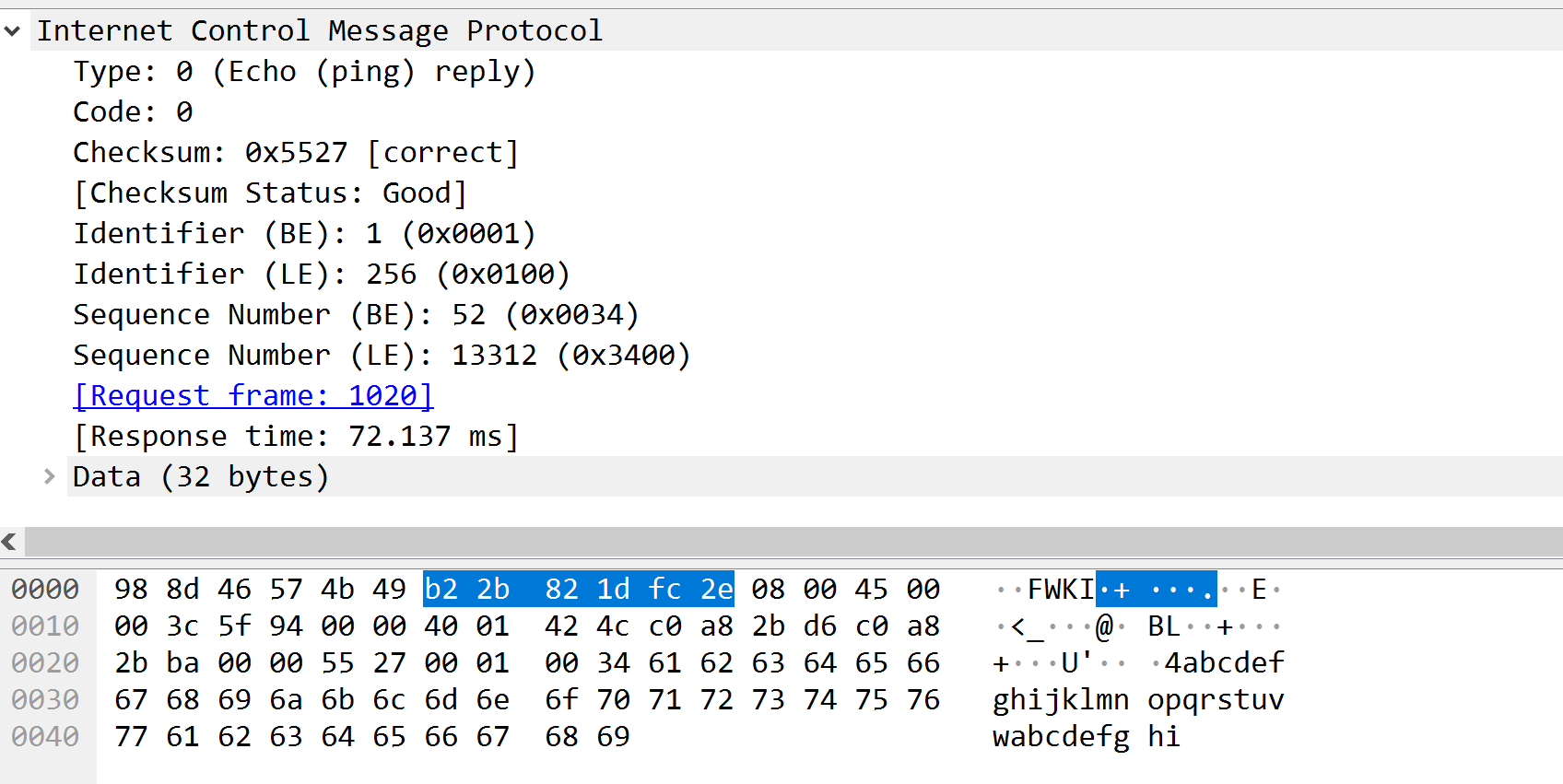


图2.1.8 查看 ICMP 报文（reply）

基于此，我们便可以对本ICMP数据包（reply）进行分析：

<Ⅰ> 类型（type）为：0（回显应答）

<Ⅱ> 代码（code）为0；

<Ⅲ> 校验和（checksum）为0x5527，确保数据的准确性，由显示可得本次正确；

<Ⅳ> 标识符，大端顺序（identifier（BE））为：1（0x0001）；

<Ⅴ> 标识符，小端顺序（identifier（LE））为：256（0x0100）；

<Ⅵ> 序列号，大端顺序（sequence number （BE））为：52（0x0034）；

<Ⅶ> 序列号，小端顺序（sequence number （LE））为：13312（0x3400）；

<Ⅸ> 回应时间（response time）为：72.127ms。

对比可得，ICMP Echo Request 和 Echo Reply的主要区别在于类型（type）和校验和（checksum）。类型不同的原因在于两个报文一个为回显请求，一个为回显应答；校验和不同的原因在于两个报文的内容不同，则校验和不同。同时，Echo Reply还多了一项回应时间（response time），单位为毫秒。

1. **改变 ping 命令的参数，观察 IP 数据报分片：**

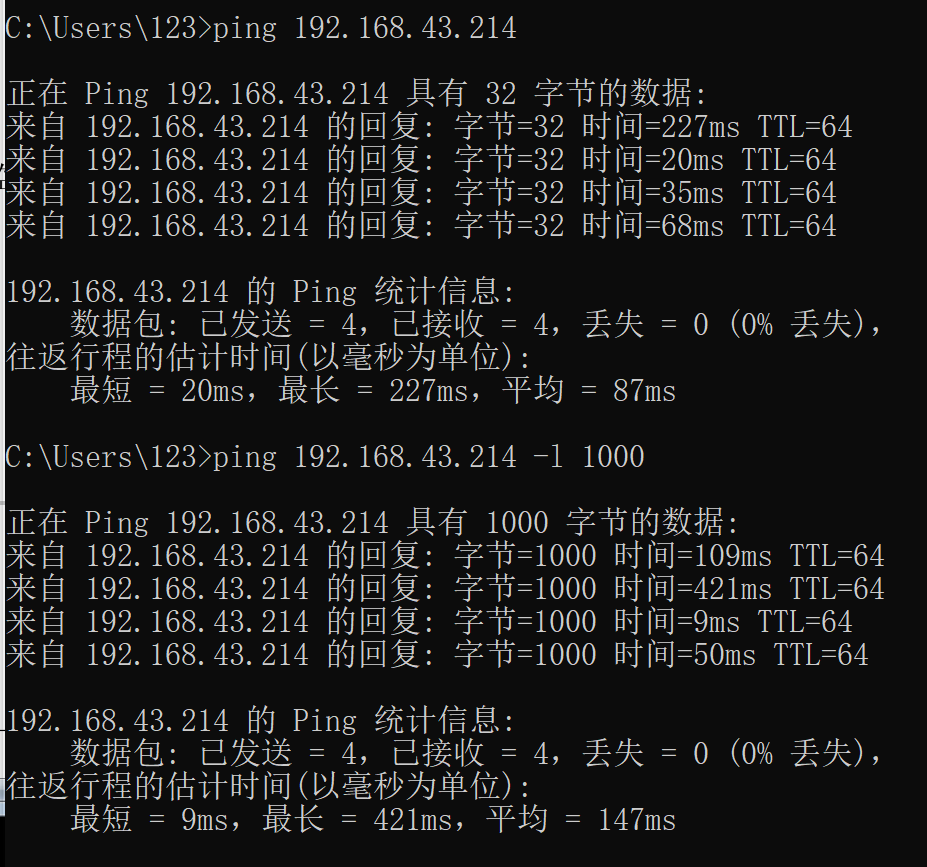
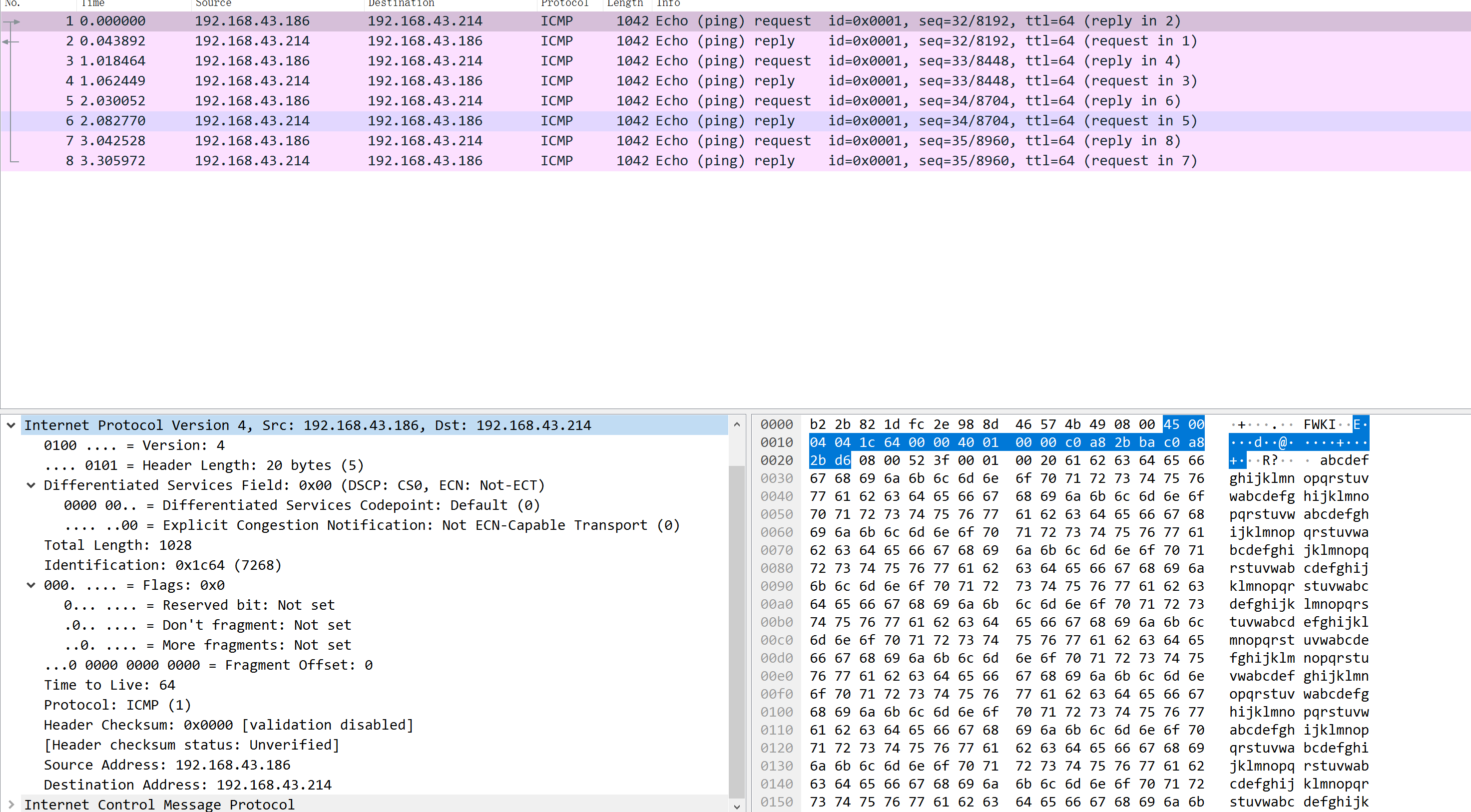


图2.2.1 ping 命令执行

如图可得，ping命令默认为32字节的数据，由于数据较少，不会出现分片情况。

* 1. **改变length长度为1000**





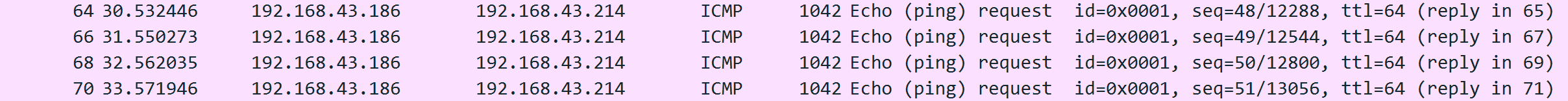


图2.2.2 改变length大小为1000结果

由图可见，当字节长度为1000时，没有分片。

* 1. **改变length长度为2000**



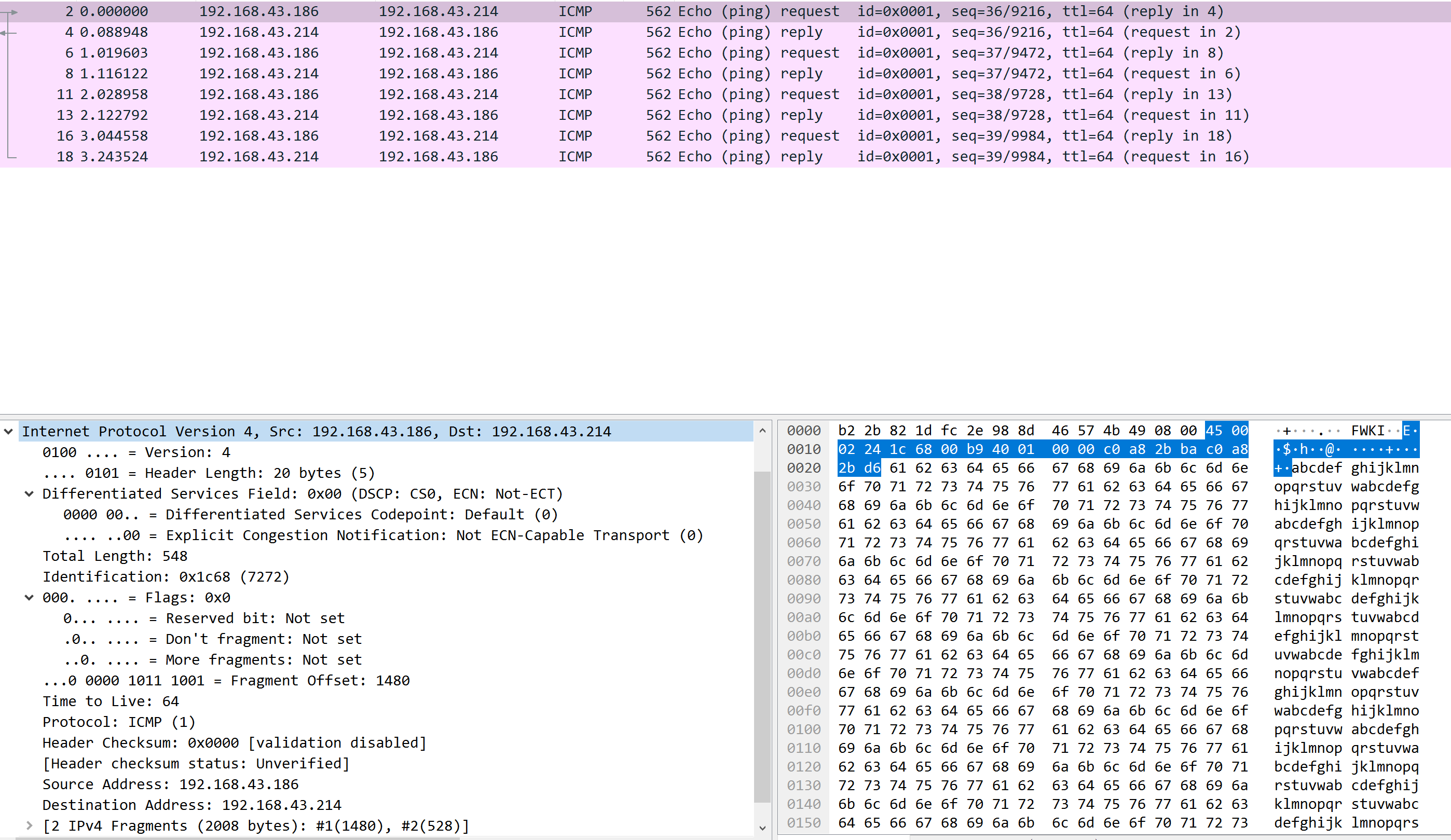




图2.2.3 改变length大小为2000结果

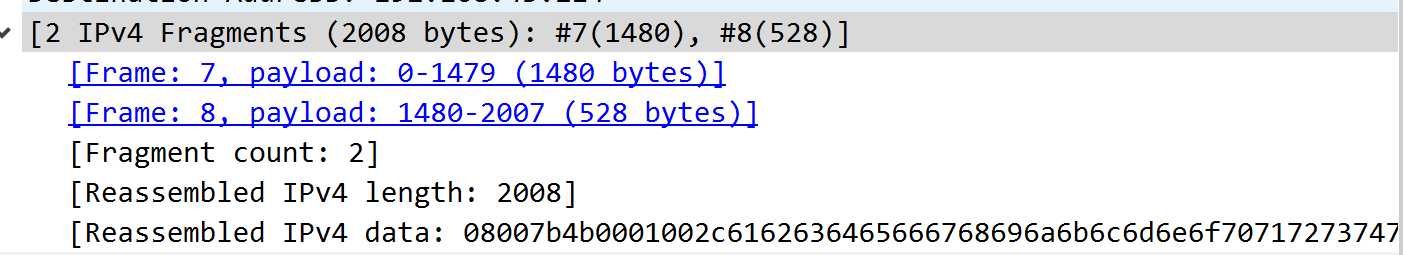


图2.2.4 改变length大小为2000结果



图2.2.5 改变length大小为2000分片情况

由图可见，当字节长度为2000时，有分片，多了一个数据包。先捕捉到的IPv4 数据包长度为1514，而接下来的ICMP包长度为 562。

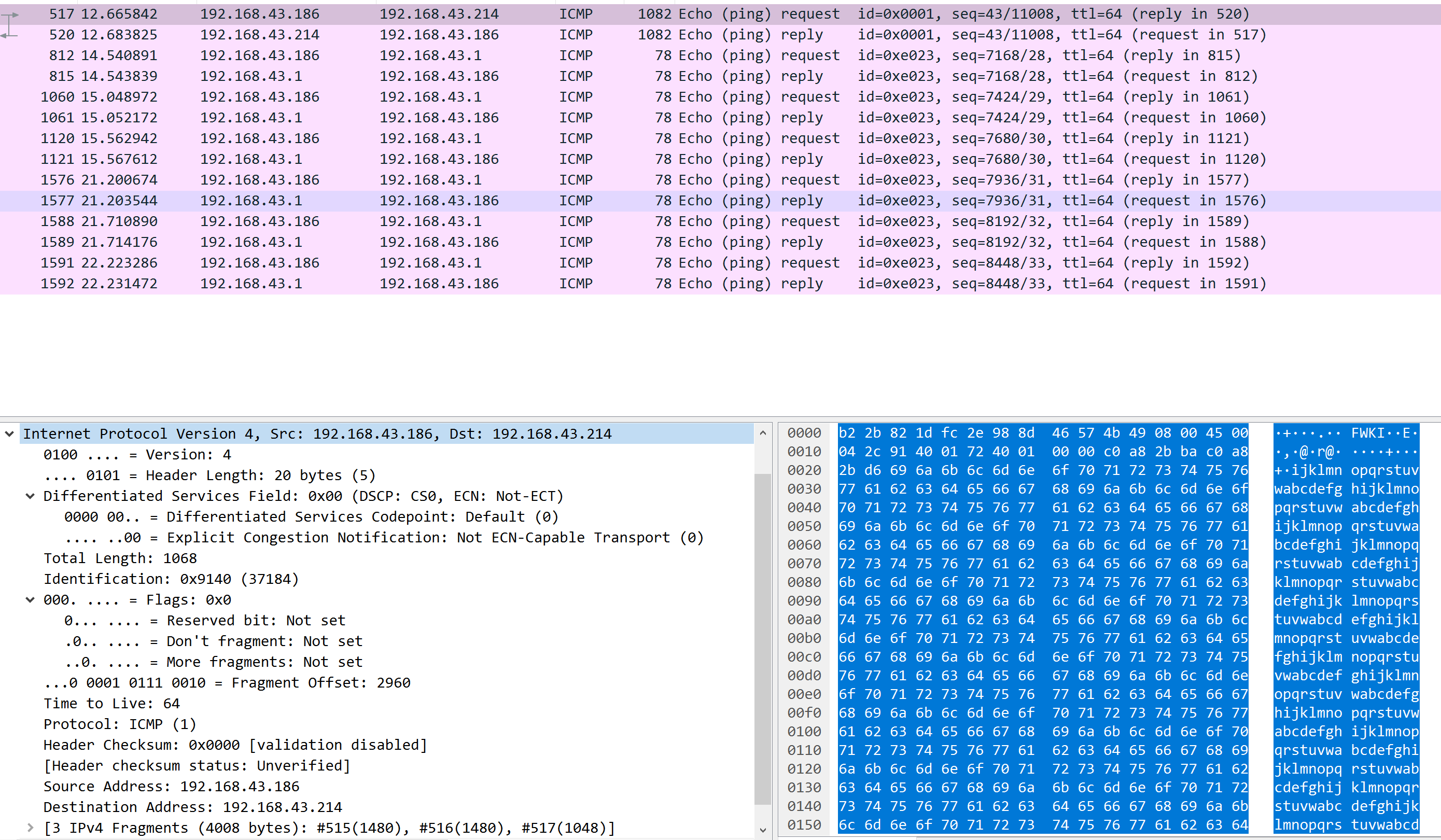




图2.2.6 改变length大小为4000结果

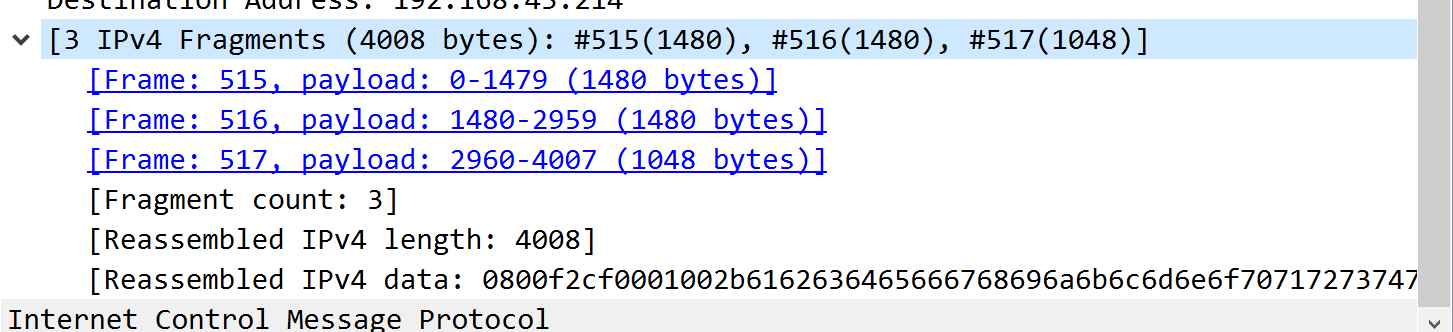


图2.2.7 改变length大小为4000结果

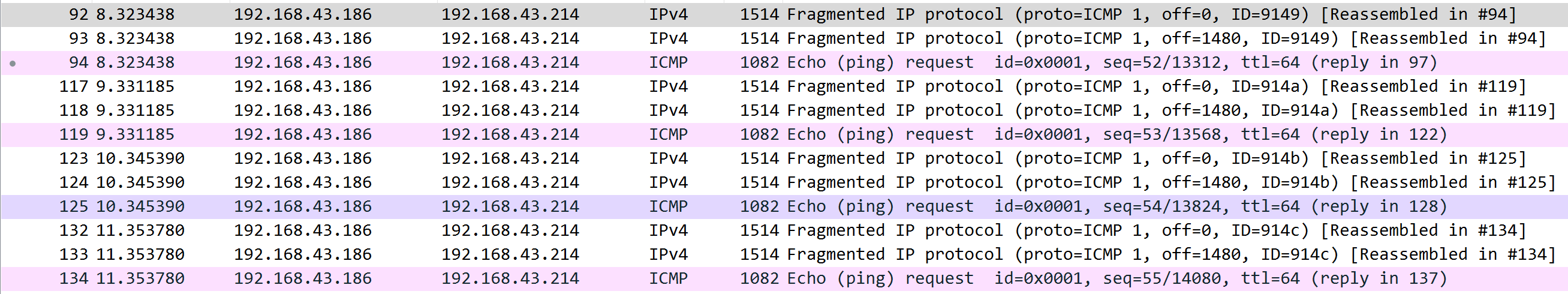


图2.2.8 改变length大小为4000分片情况

由图可见，当字节长度为4000时，有分片，多了两个数据包。先捕捉到的两个IPv4 数据包长度为1514，而接下来的ICMP包长度为 1082。

首先确认局域网下的MTU为1500字节（其限制了数据帧的最大长度，传输的IP报文的大小超过最大传输单位MTU时就会产生IP分片情况）。

经过如上数据包分析，可初步得出结果：length参数大于1480会出现分片的情况。以length参数为2000的结果为例，其首先捕捉到的IPv4数据包长度为1514字节（14字节以太网帧头+20字节IP包头+1480字节数据），fragment offset为0，代表手段偏移为0；而ICMP数据包长度为528字节（14字节以太网帧头+20字节IP包头+8字节ICMP包头+剩余520字节数据），fragment offset为1480，代表前一段已包含1480字节的数据，与前一个IPv4数据包数据长度吻合，猜想成立。

综上，本局域网网关的最大传送单元MTU是1500字节，超过则自动分片，减去IP首部20字节，所以IP数据报能发1480字节。对于一个IPv4数据包，其还需包括20字节的IP首部和14字节的以太网帧头，所以其最大大小应为1514.

同时我在实验中也发现了一个特殊的情况，就是在只发送一个数据包的情况下，想要保证不出现“fragmented IP protocol”，最大的允许数据字节长度应该为1472字节而非1480字节：

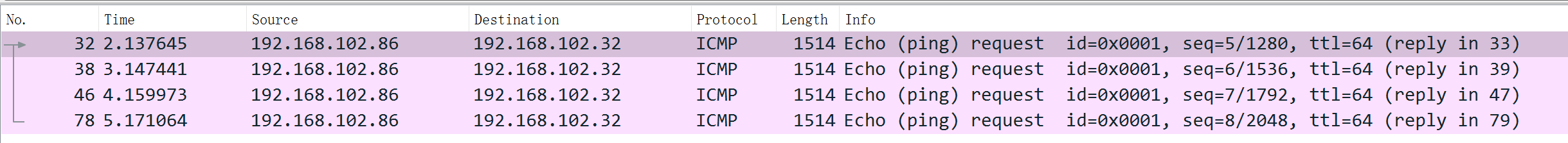


图2.2.9 改变length大小为1472分片情况

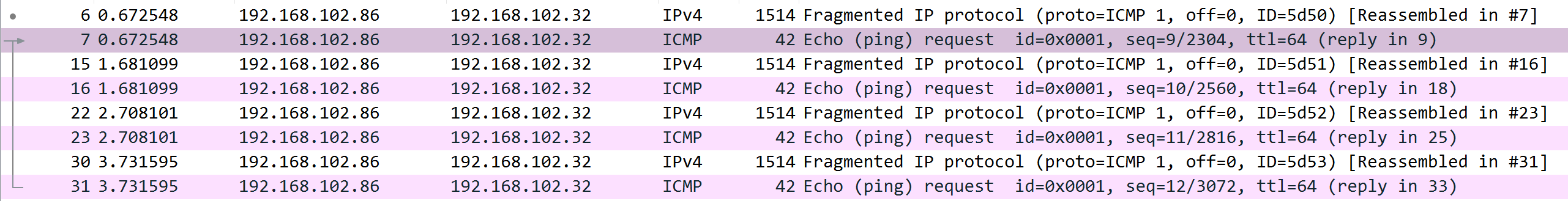


图2.2.10 改变length大小为1480分片情况

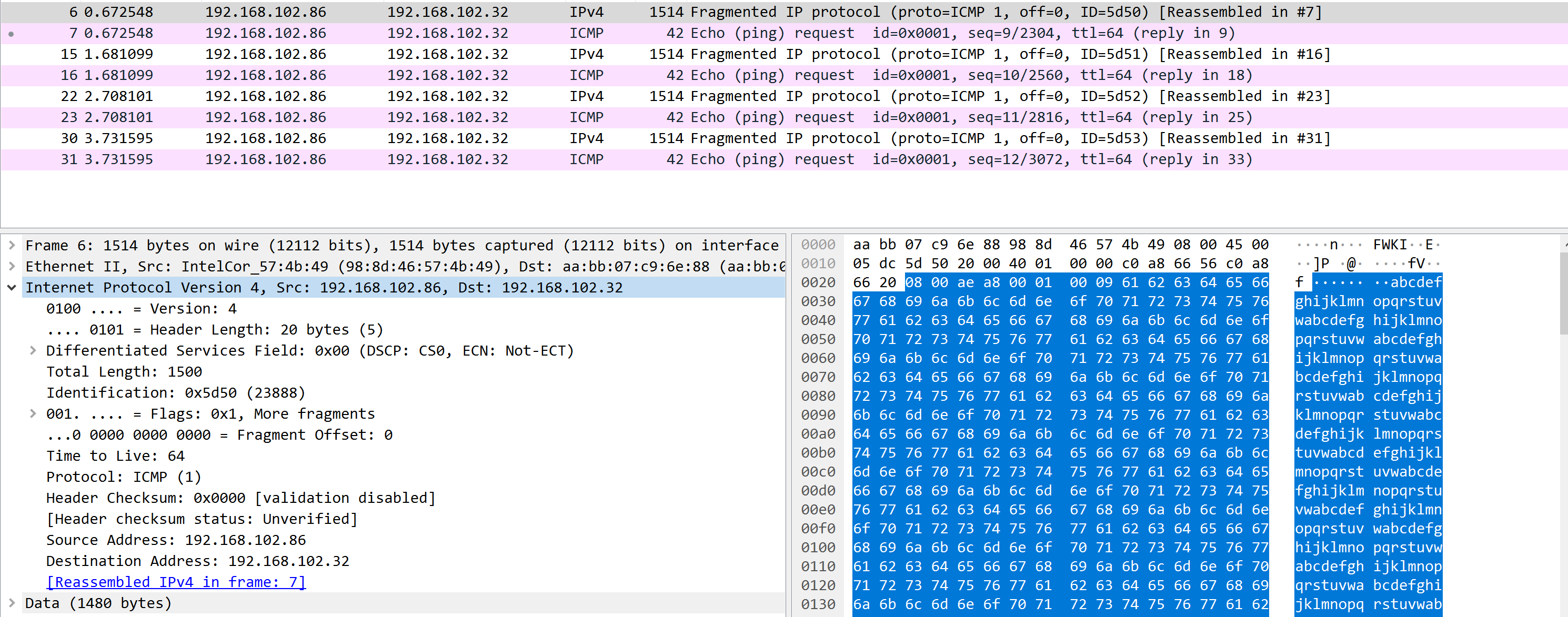




图2.2.11 改变length大小为1480的IPv4包

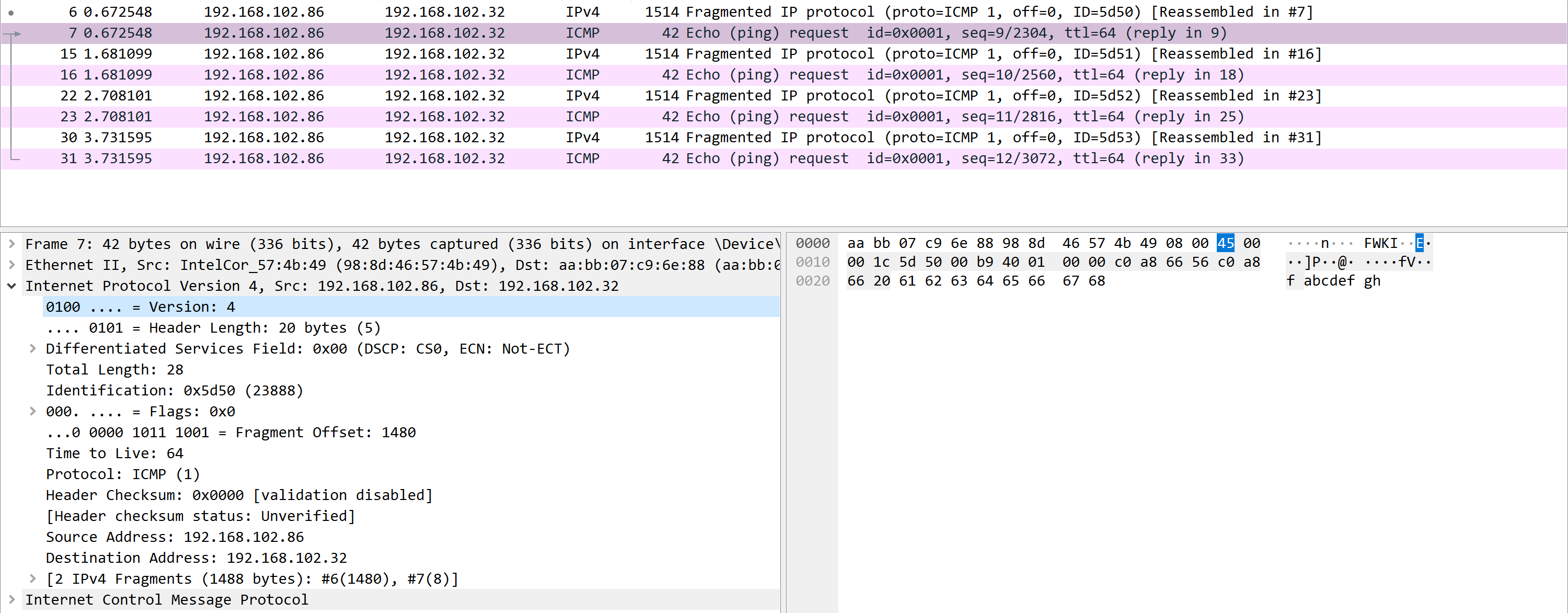




图2.2.12 改变length大小为1480的ICMP包

由图2.2.11，图2.2.12可见，当字节数为1472时不会发生分片，而字节数为1480时会发生分片。两个IP数据包数据的字节数分别为1480字节和8字节。但根据已学知识和length大小为1480的IPv4包中所显示的data为1480字节可猜想得：之所以在只发送一个数据包时，length大于1472就会分片，是因为其只会发送一个ICMP报文，而ICMP报文比起IPv4报文会增加一个8字节ICMP包头，使得data的字节长度从1480字节被压缩至1472字节。

1. **执行 Traceroute 命令，观察 ICMP 差错报文的结构，并分析其工作原理**：

ping 工具只能测试目的设备的连通性，但是看不到数据包的传输路径。所以在网络不通的情况下，无法知道网络问题发生在哪个位置。tracert工具可以查看数据包的整条传输路径，包括途中经过的中间设备。

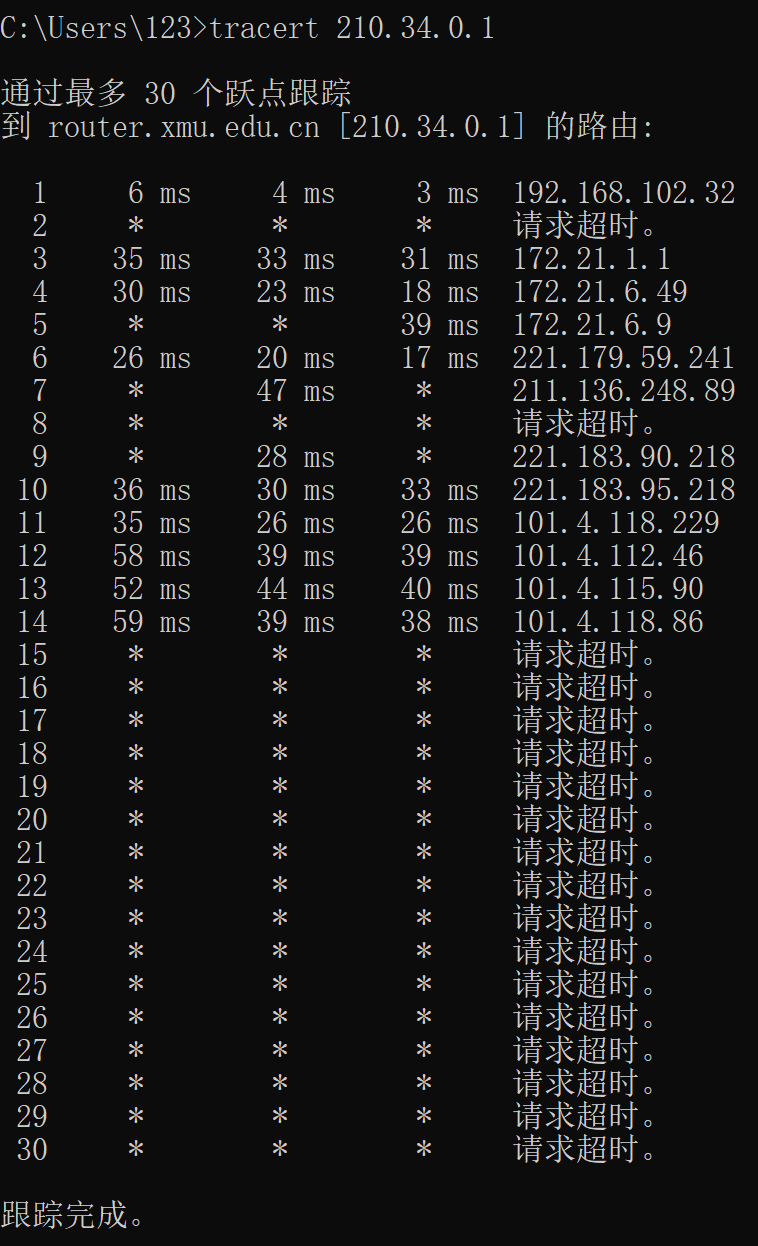


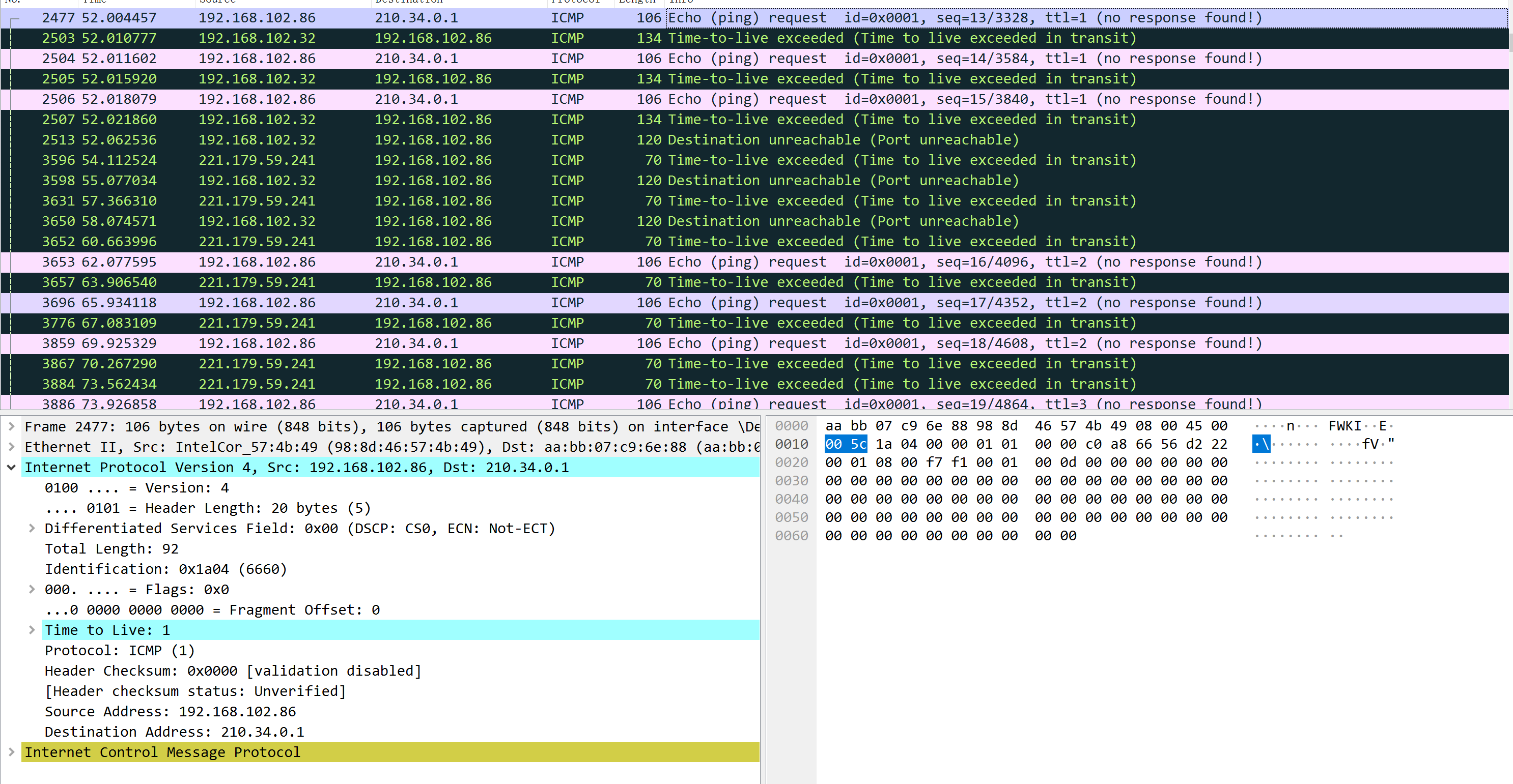
图2.3.1 tracert 结果

图2.3.2 差错报文

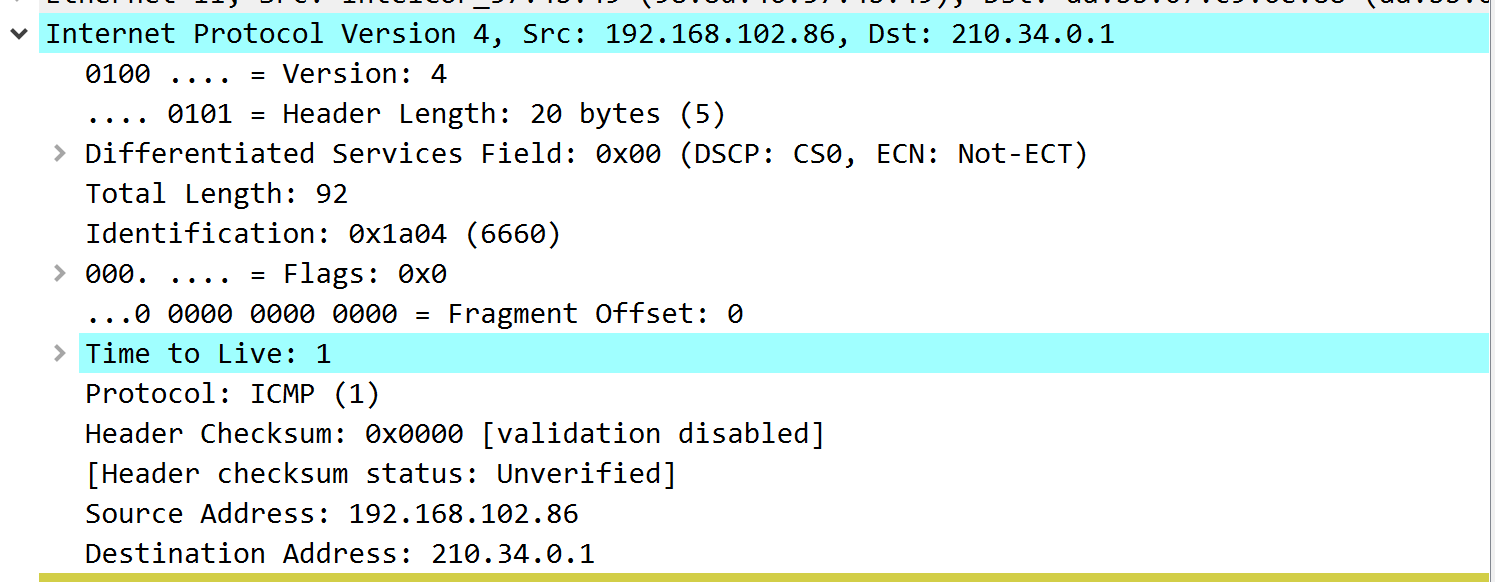




图2.3.3 发送的第一个ICMP报文

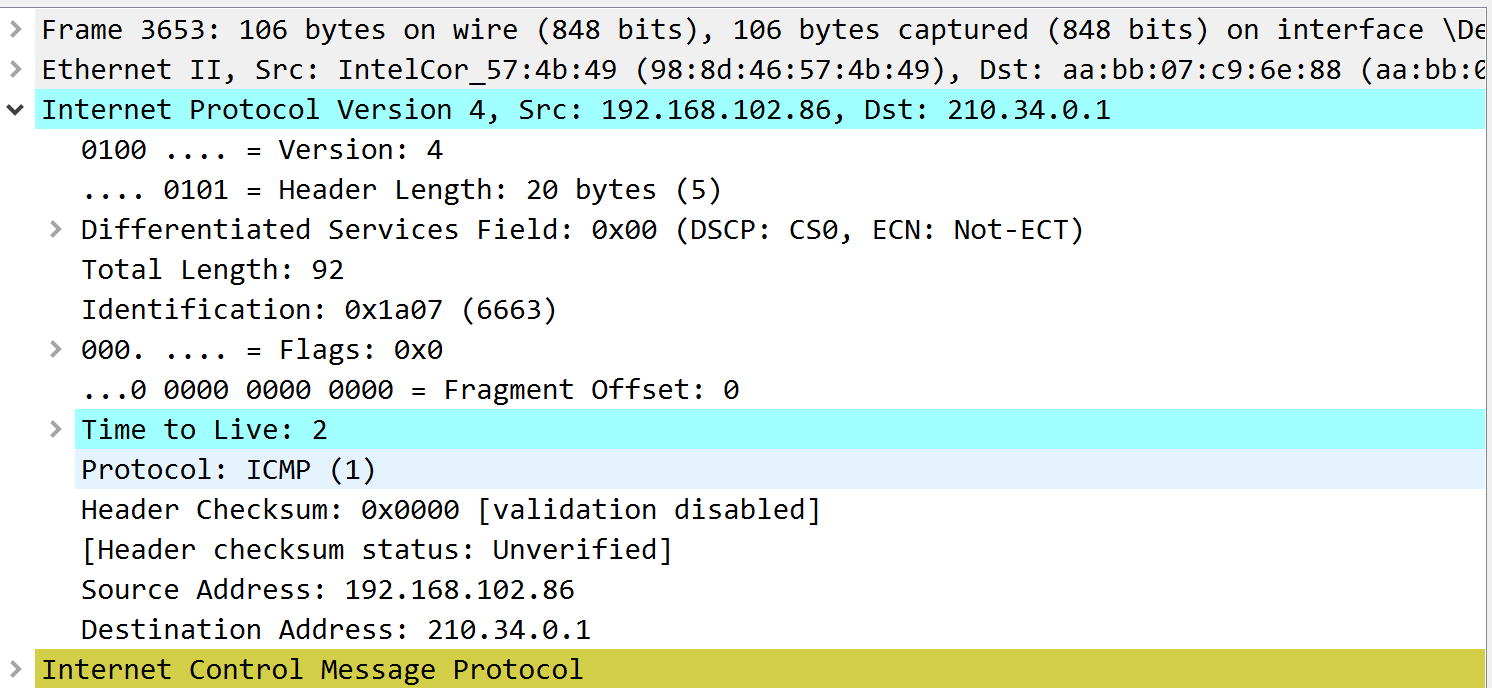




图2.3.4 发送的第四个ICMP报文

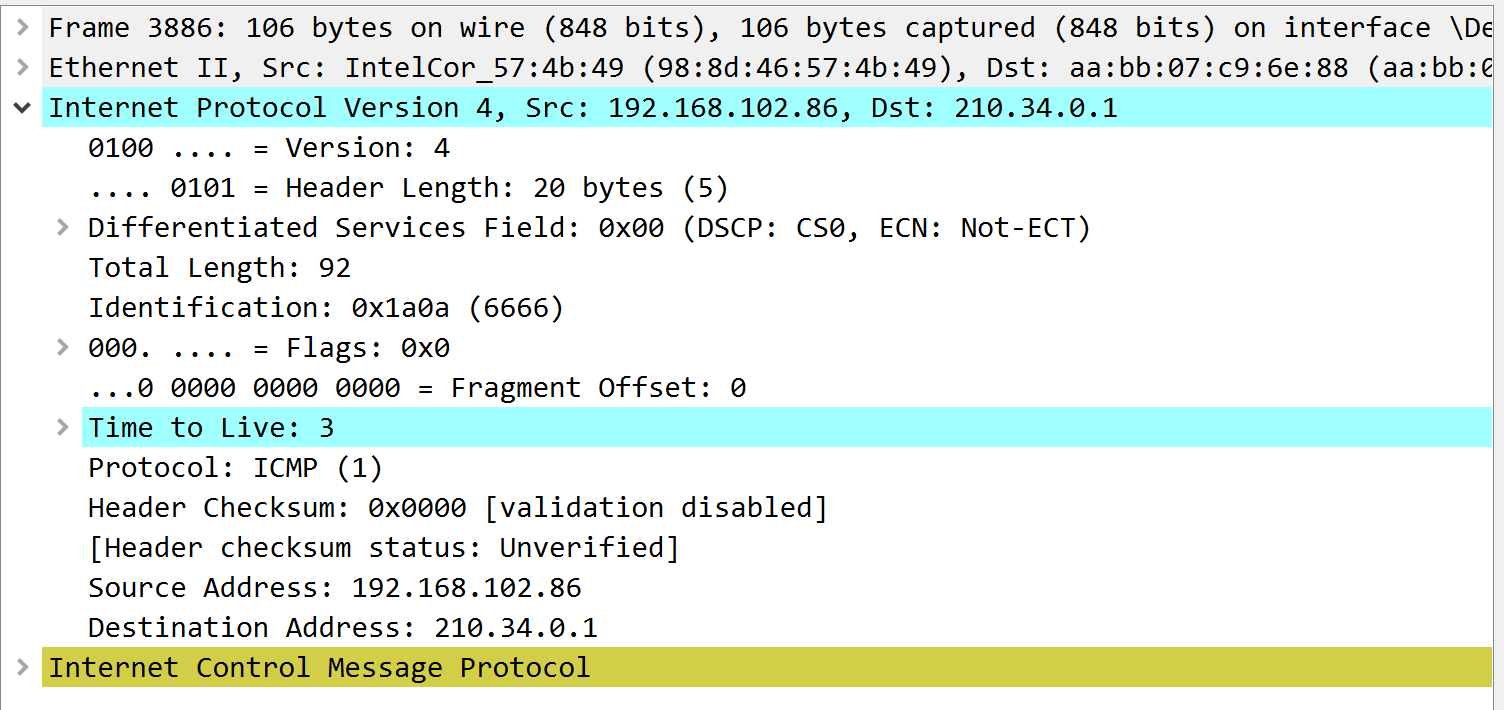




图2.3.5 发送的第七个ICMP报文



图2.3.6 Time-to-live exceed 差错报文

由以上分析可得，在使用tracert时，原设备会逐跳发送数据包，并等待响应报文。由命令界面可得，原设备会向每一个路由器发送3个数据包。给第一个路由器发送数据包时，TTL值设为1，第二个设为2，以此类推。且此数据报文是经过每一个路由器逐级送出的，每一个路由器收到数据包后会将TTL减1，随即将其发向下一个路由器；若TTL已减少至0，路由器则会将数据包丢弃，并返回一个time exceeded差错报文，原设备的tracert收到响应报文后，取出源IP地址。

由此可以去表述一个较为完整的tracert过程：

①第一个路由器收到数据包后TTL值减1，随即丢弃数据包，并返回一个time exceeded消息。

②源设备的tracert收到响应报文后，取出源IP地址，即路径上的第一个路由器地址。

③然后tracert发送一个 TTL 值为2的数据包。第一个路由器将 TTL 值减1，并转发数据包。

④第二个路由器再将TTL值减 1，丢弃数据包并返回一个time exceeded消息。

⑤tracert收到响应报文后，取出源 IP 地址，即路径上的第二个路由器地址。⑥类似步骤，tracert逐跳获得每一个路由器的地址，并探测到目的设备的可达性。

⑦当到达目的地时，目标主机返回一个 [ICMP port unreachable]的消息，使发起者确认IP数据报已经正常到达。

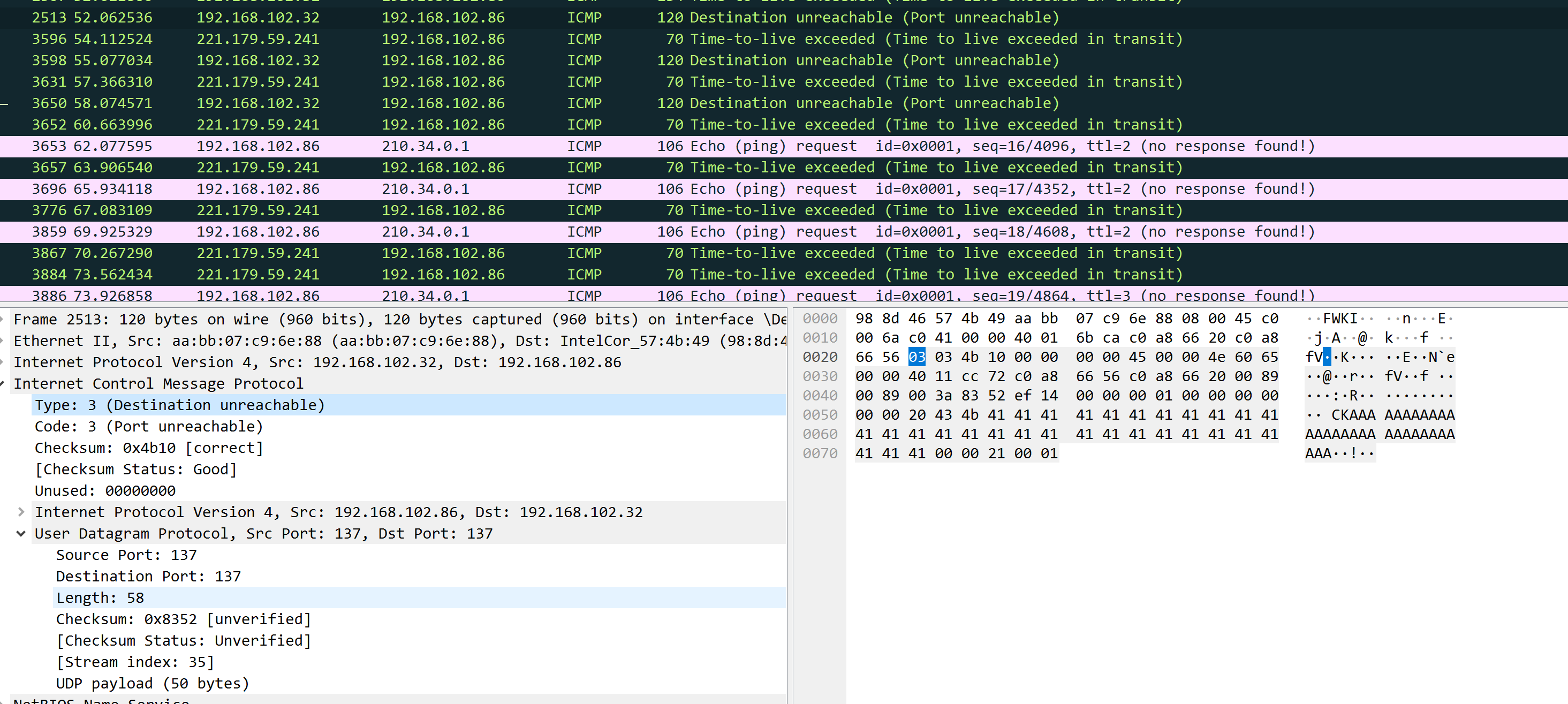


图2.3.7 destination unreachable 差错报文



图2.3.8 ICMP五种差错报文

由图2.3.8可知，差错报文主要包括重点不可达、源点抑制、时间超过、参数问题和改变路由报文等等。

由图2.3.6和图2.3.7可知，tracert的差错报文主要包括时间超过报文（time exceeded）和终点不可达报文（destination unreachable）两种：time exceeded报文主要通过设置TTL实现，以用于确认每一阶段所经过的路由器的IP地址和传输时间；destination unreachable报文主要通过填入一个不可能的端口号值作为目标端口号实现。当目的主机，收到数据包后，会返回 ICMP 差错报文消息，且这个差错报文消息的类型是“destination unreachable”，以用于确认发送方发出的数据包到达了目的主机。

由图2.3.6和图2.3.7进行ICMP差错报文分析：

1. Type和code部分不同：time exceeded报文的type为11，code为0，代表其问传输期间生存时间为0的差错；destination unreachable报文的type为3，code为3，代表其为端口不可达的差错；
2. tracert的data不像ping命令中的随机字符，内容全为0，由于其是错误分组，因此有一个unused字段，其值为0；
3. IPv4中所包含的信息与ping报文中无过大差别；
4. 路由跟踪的ICMP响应数据包（非time exceeded）的ICMP的type为0，代表ICMP响应，每次的序列号都不同。

经过如上分析，我所绘制出的tracert交互过程如图2.3.9：

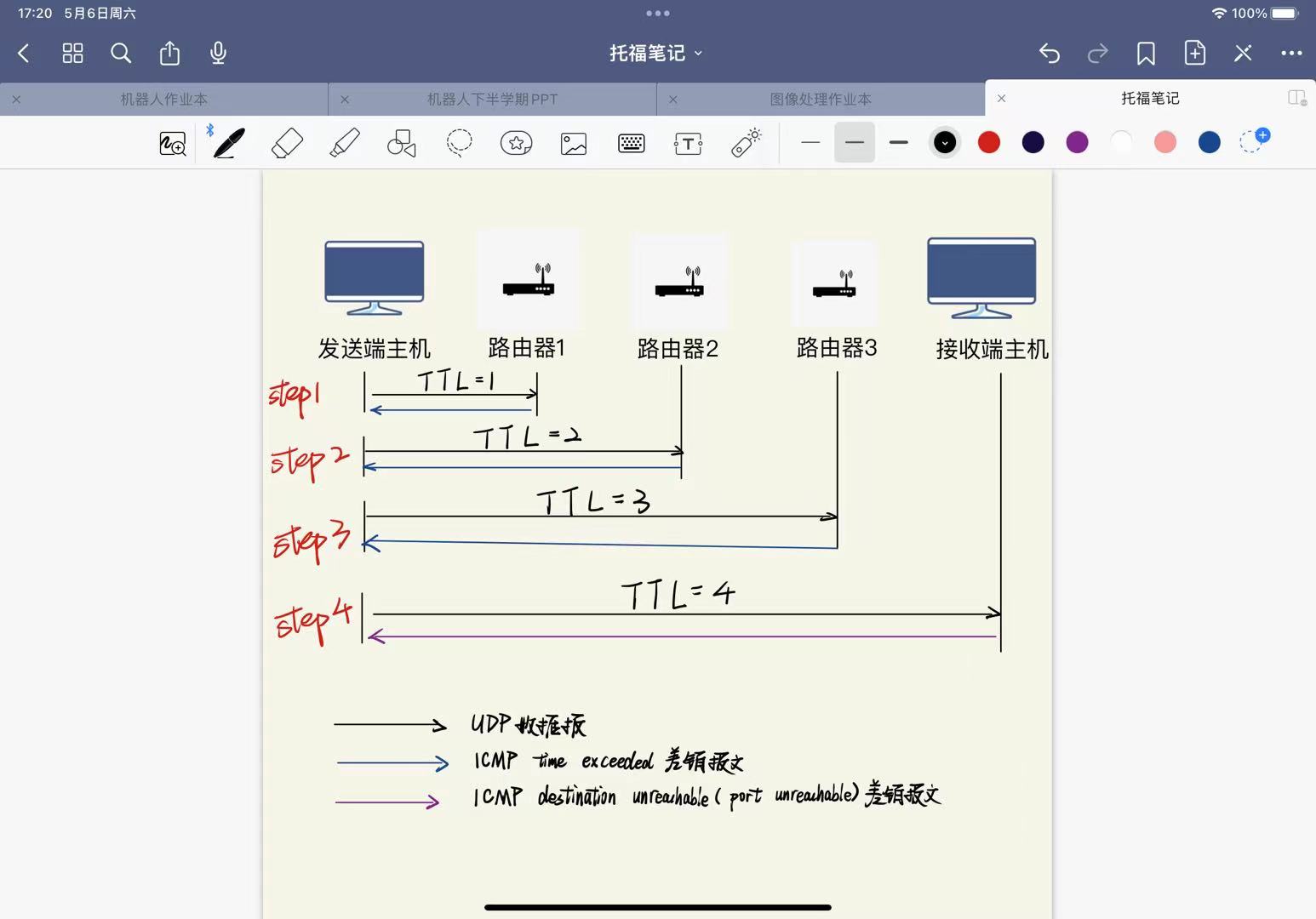


图2.3.9 tracert数据交互过程

1. **实验所遇问题**

问题：在分析切片问题时，我发现我无法通过在wireshark中查找“IPv4 fragments”来确定MTU

解决：通过在网上查找到的方式解决：

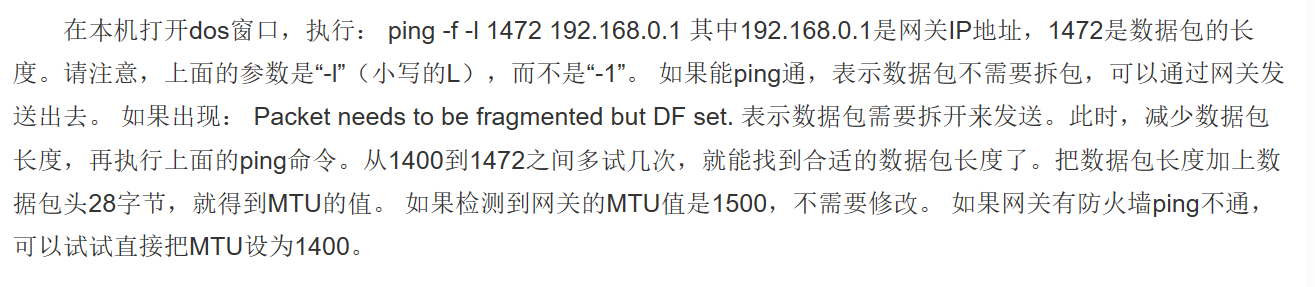




图3.1 查询MTU方法

1. **思考题**
2. 问：在实际操作中，Traceroute 命令返回的某些条目以“\*”号表示。请思考有哪些原因可能导致这样的情况。

答：

1. 可能是防火墙封掉了ICMP的返回信息，使我们得不到什么相关的数据包返回数据。
2. 可能使中间任何一个router上封了ICMP Echo Request，traceroute就不能工作；中间的router看不到，但能看到packet 到达了最后的destination；如果封了ICMP Echo Reply，中间的全能看到，最后的destination看不到。
3. 问：发送方要怎样决定 IP 数据报分组大小，才能避免因为不同网络 MTU 不一致而引起分片呢？

答：

可以探测路径最小MTU：发送探测IP包，在包头中设置不可分片，然后通过是否受到ICMP消息即可确定路径MTU。

具体操作：首先在发送端主机发送IP数据报时将IP包首部的分片禁止标志位设置为1。根据这个标志位，途中的路由器不会对大数据包进行分片，而是将包丢弃。 随后，通过一个ICMP的不可达消息将数据链路上MTU的值一起给发送主机，不可达消息的类型为“需要进行分片但设置了不分片位”。发送主机端每次收到ICMP 差错报文时就减少包的大小，以此来定位一个合适的分组大小和路径MTU值，就避免因为不同网络 MTU 不一致而引起分片。

1. **实验感悟**

本次实验，我借助wireshark工具，深入了解了IP协议，ICMP协议，分片操作和tracert操作流程。

通过本次实验，也使得抽象的理论知识形象化，使我更加可视化地了解了IP报文和ICMP报文，让我对本身较为枯燥的理论知识印象更为深刻，同时也了解了课内所不了解的分片和tracert知识，让我对本门课程产生了更为深刻的理解。