哈尔滨工业大学

**<<计算机网络>>**

**实验报告**

**(2019年度春季学期)**

|  |  |
| --- | --- |
| **姓名：** | **吴昊** |
| **学号：** | **1170300527** |
| **学院：** | **计算机科学与技术学院** |
| **教师：** | **刘亚维** |

## 实验四 利用 Wireshark 进行协议分析

## 实验目的

熟悉并掌握 Wireshark 的基本操作，了解网络协议实体间进行交互以 及报文交换的情况。

## 实验内容

1) 学习 Wireshark 的使用

2) 利用 Wireshark 分析 HTTP 协议

3) 利用 Wireshark 分析 TCP 协议

4) 利用 Wireshark 分析 IP 协议

5) 利用 Wireshark 分析 Ethernet 数据帧

选做内容：

a) 利用 Wireshark 分析 DNS 协议

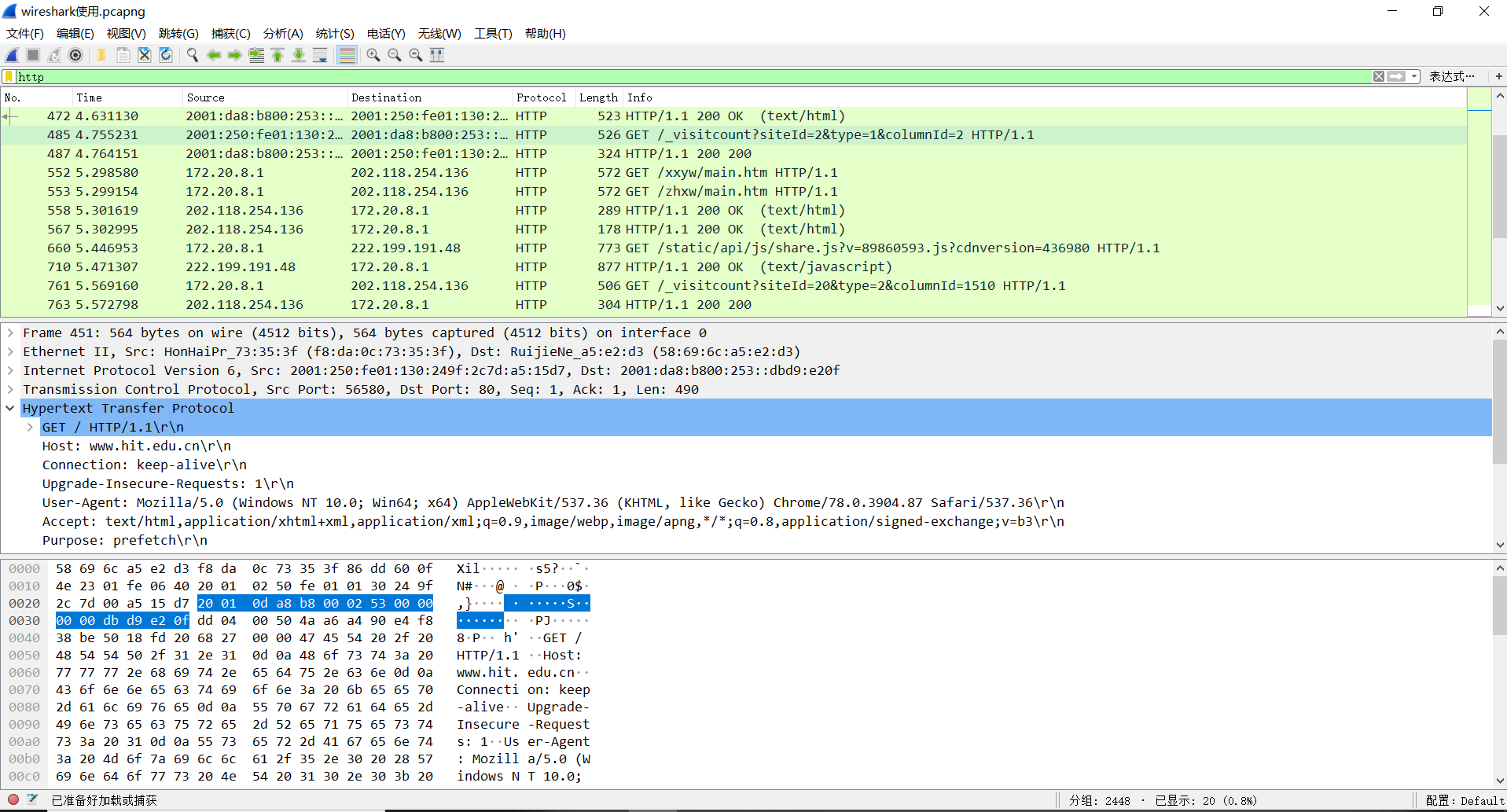
b) 利用 Wireshark 分析 UDP 协议

c) 利用 Wireshark 分析 ARP 协议

## 三、实验过程及结果

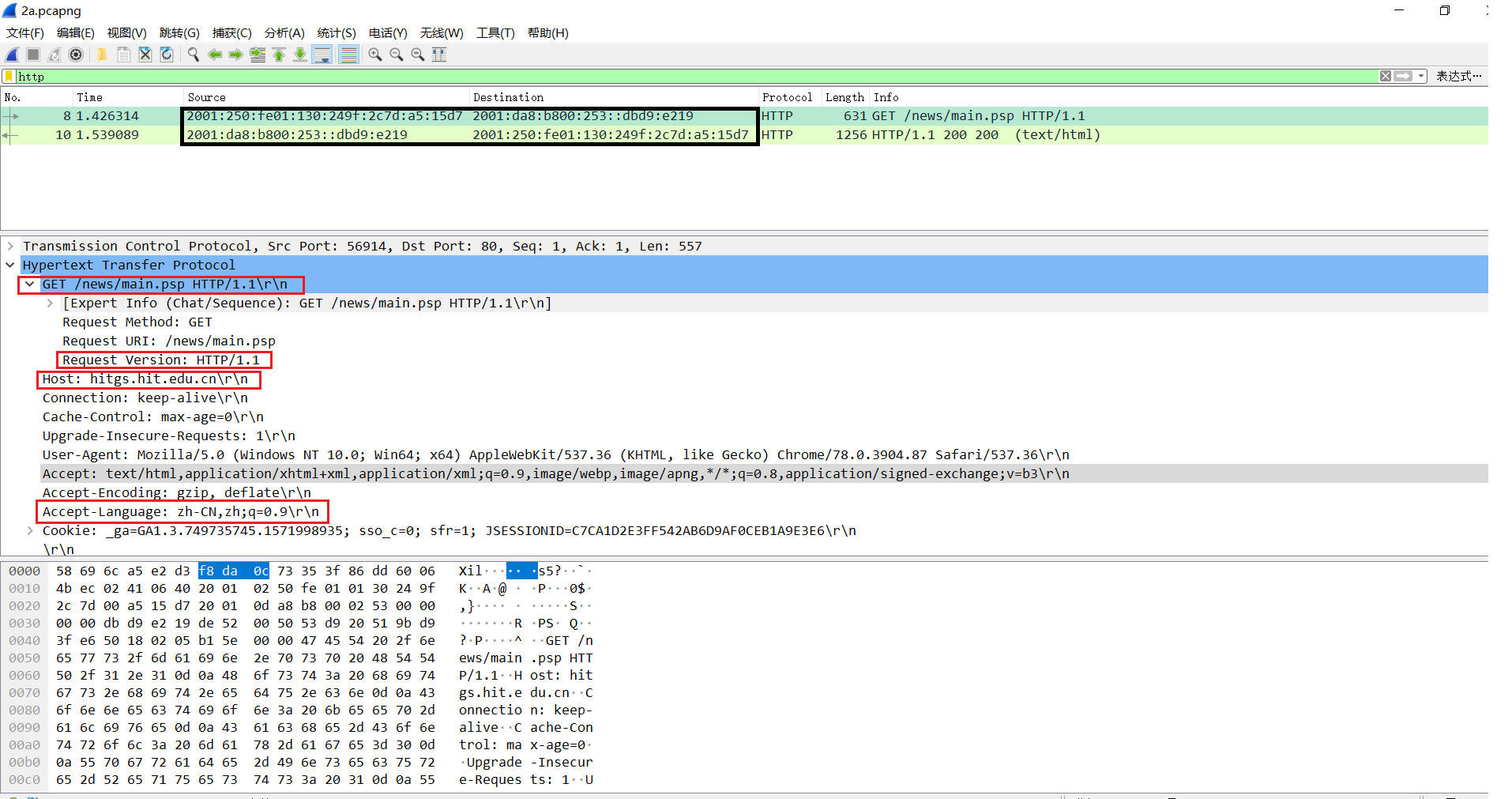
1. Wireshark 的使用

首先打开wireshark，由于使用wifi连接网络，故选用WLAN接口，输入http过滤并打开浏览器进入[www.hit.edu.cn](http://www.hit.edu.cn)图片如下



可以看出发送的是http的get请求，使用http版本是1.1，并收到200的成功请求码。并且使用的已经是IPv6地址了。

1. HTTP 分析
   1. HTTP GET/response 交互



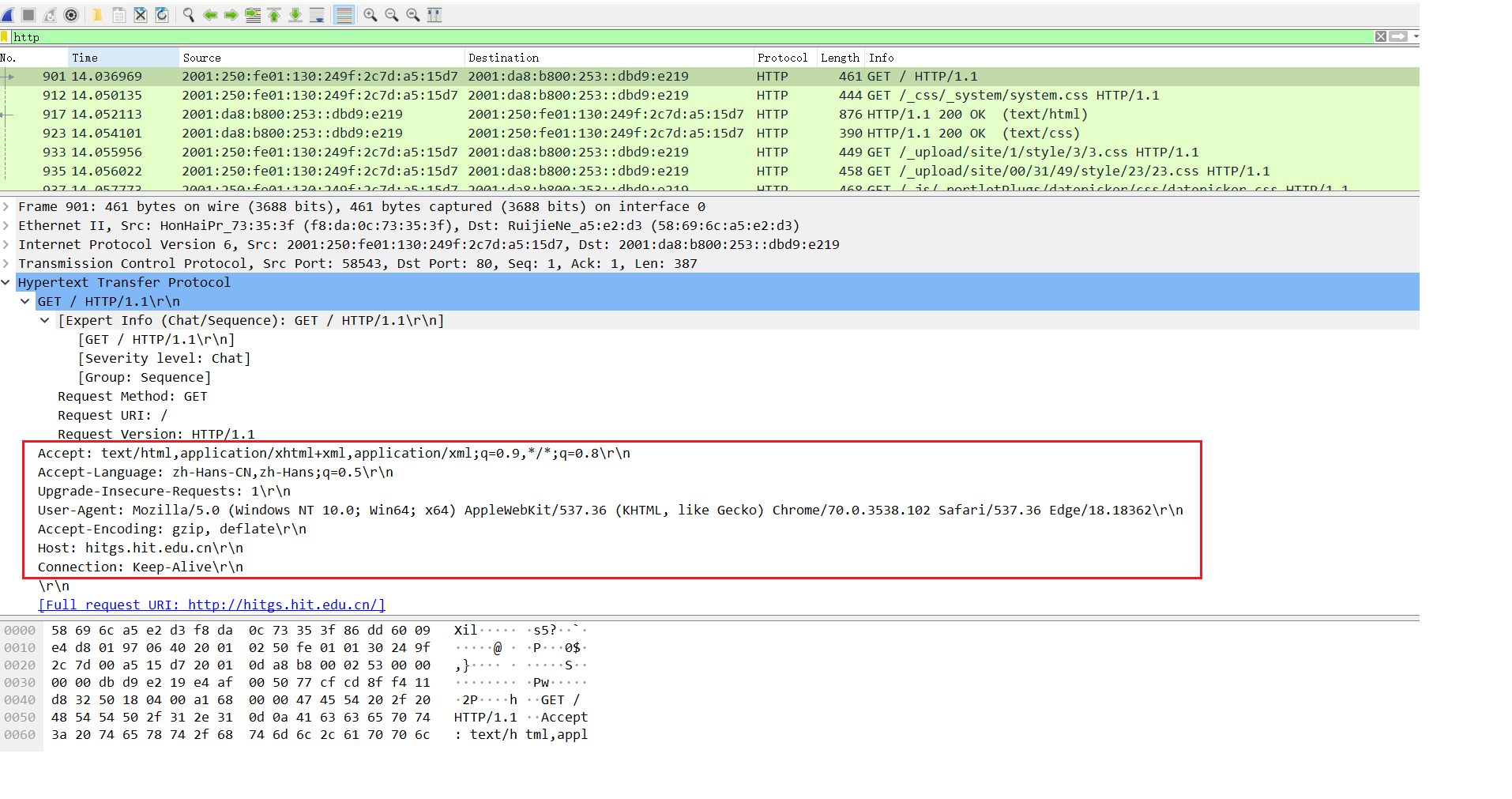
浏览器运行HTTP/1.1，访问的服务器也是HTTP/1.1

接收的语言版本是：Accept-Language：zh-CN,zh;q=0.9

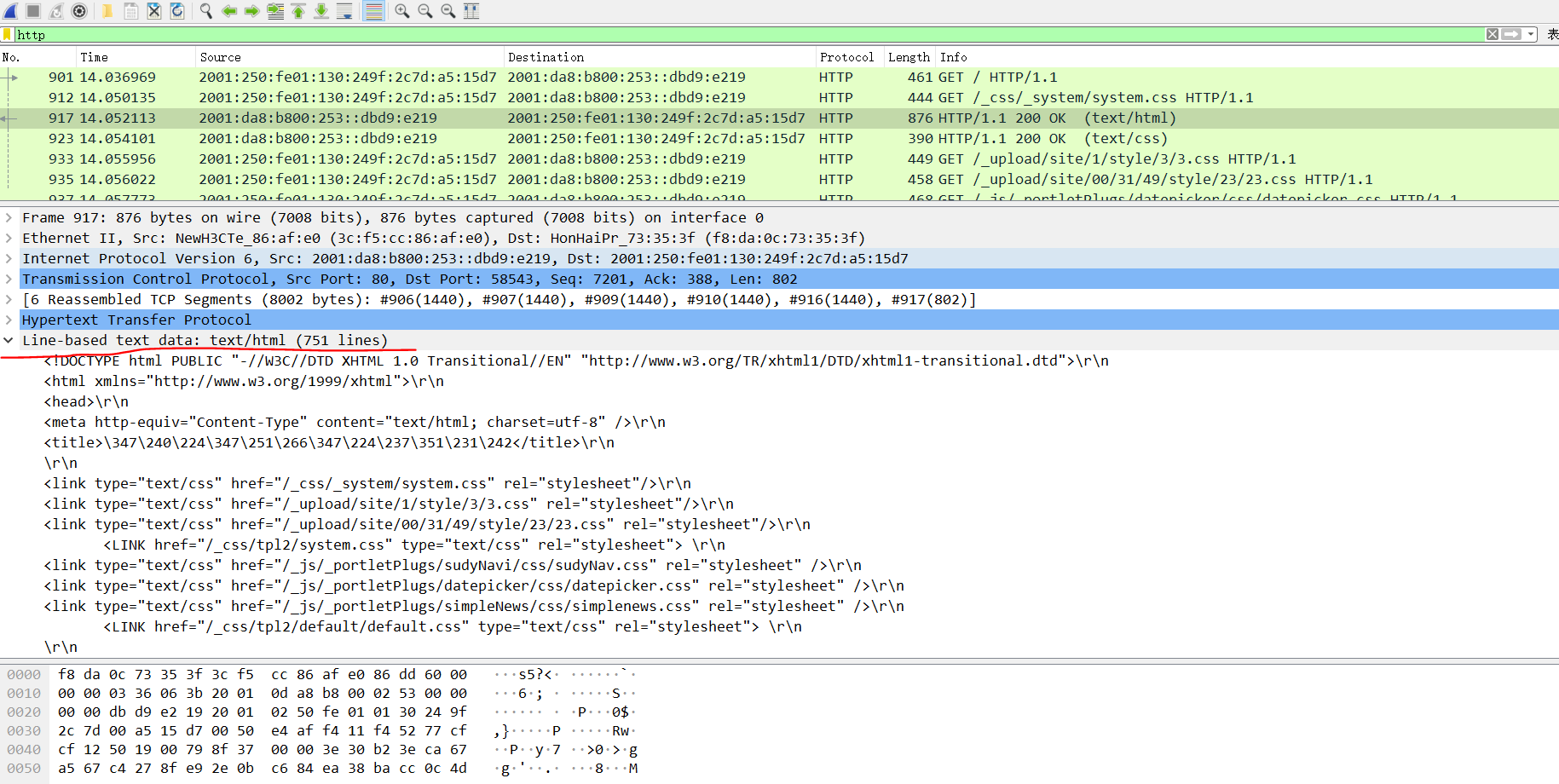
我的IP为2001:50:fe01:130:249f:2c7d:a5:15d7,服务器IP为：2001：da8:b800:253::dbd9:e219可以看出已经使用了IPv6地址

返回状态码是200

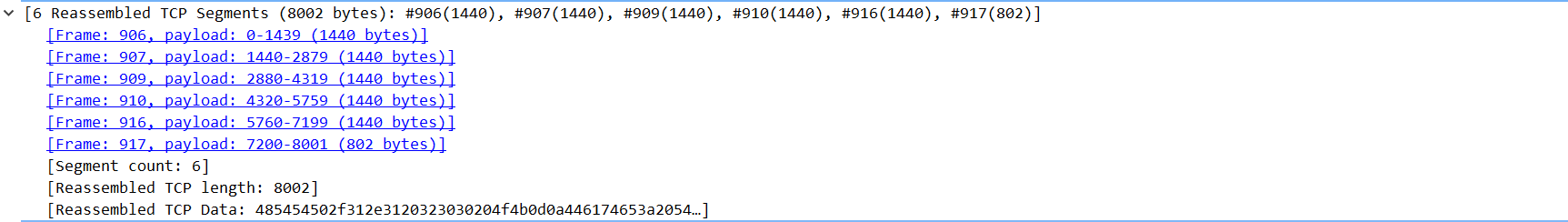
* 1. HTTP条件GET/response交互



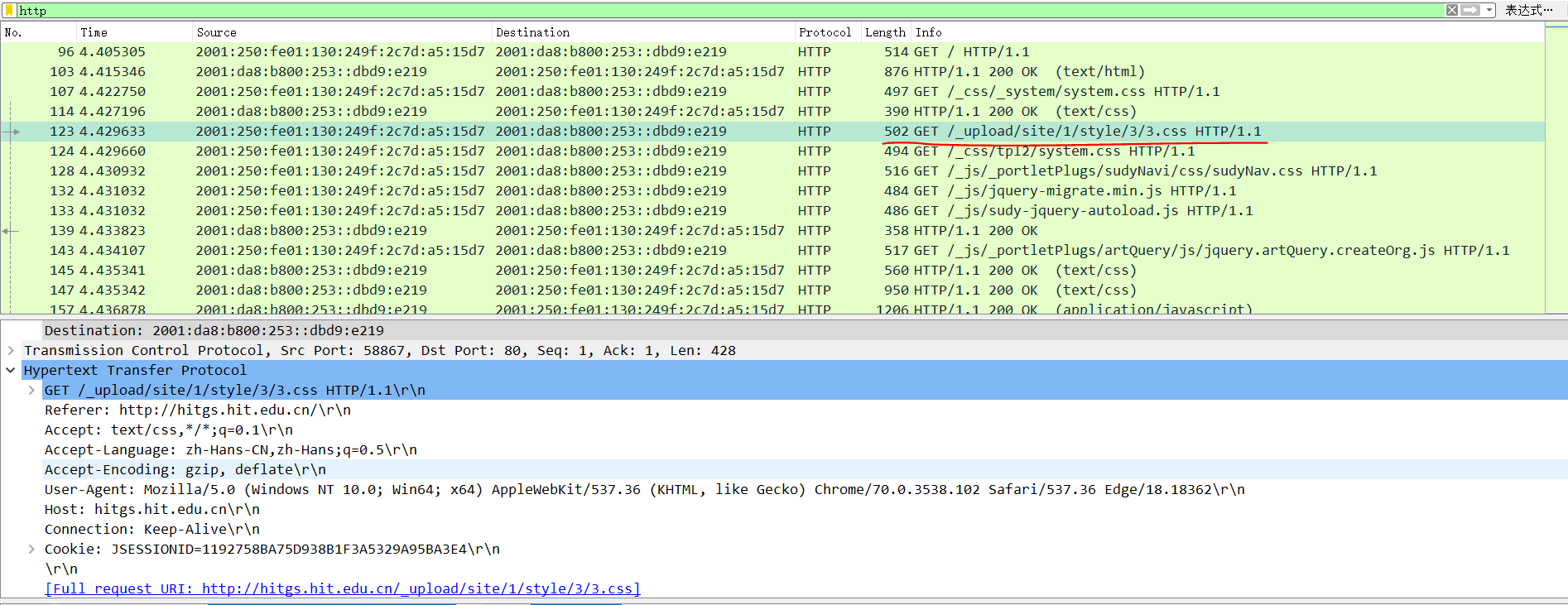
1.可以看出发送的get请求没有if-modified-since字段，所有返回的请求的状态码均为200

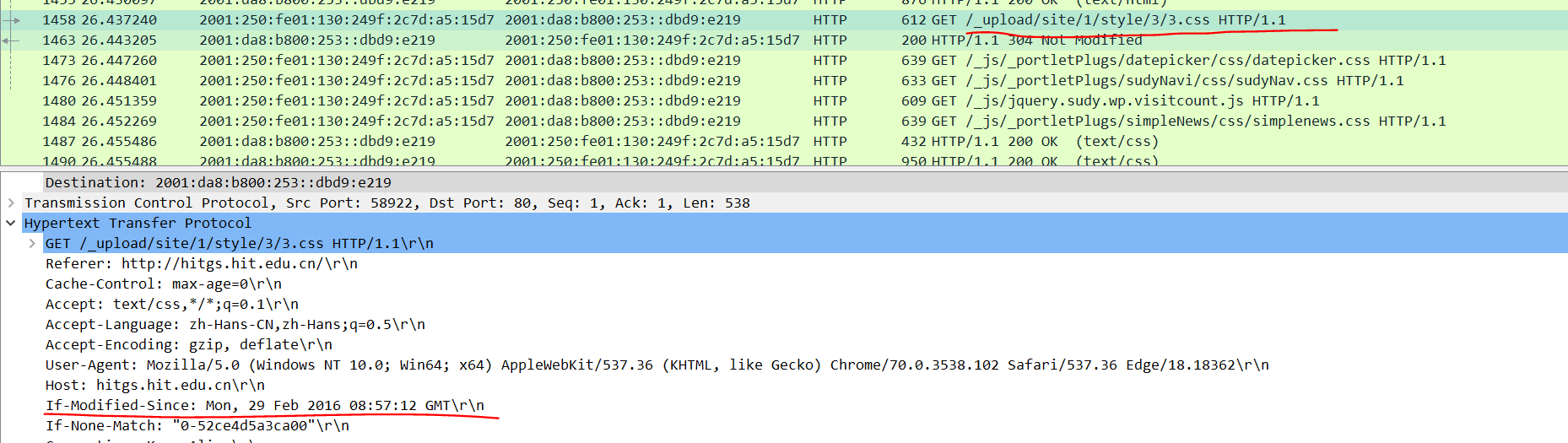


2.服务器已经明确返回了文件内容，在Line-based text data中



一共有6个TCP数据段发送



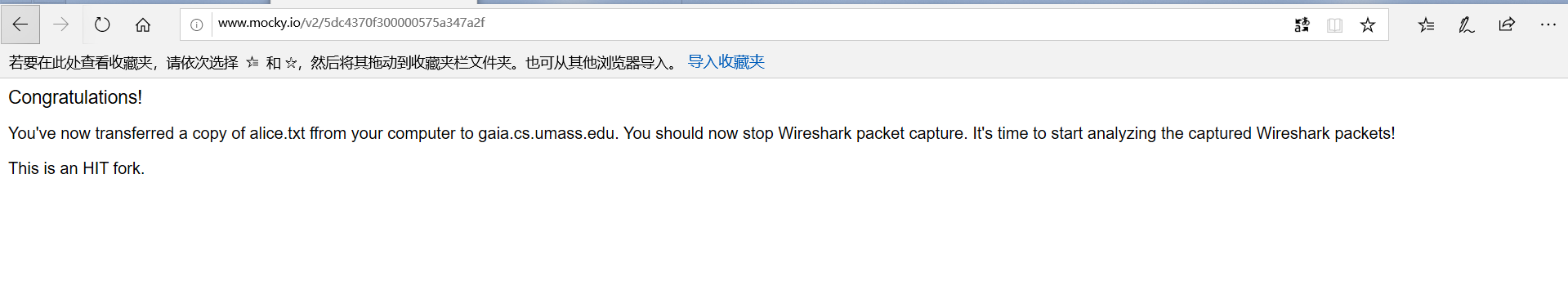


3.上面两个图片分别为第一次和第二次请求时的数据，可以看出，都请求了一个3.css文件，首次发送头没有if-modified-since而第二次请求头则存在，houm 跟着的信息为最晚版本的时间

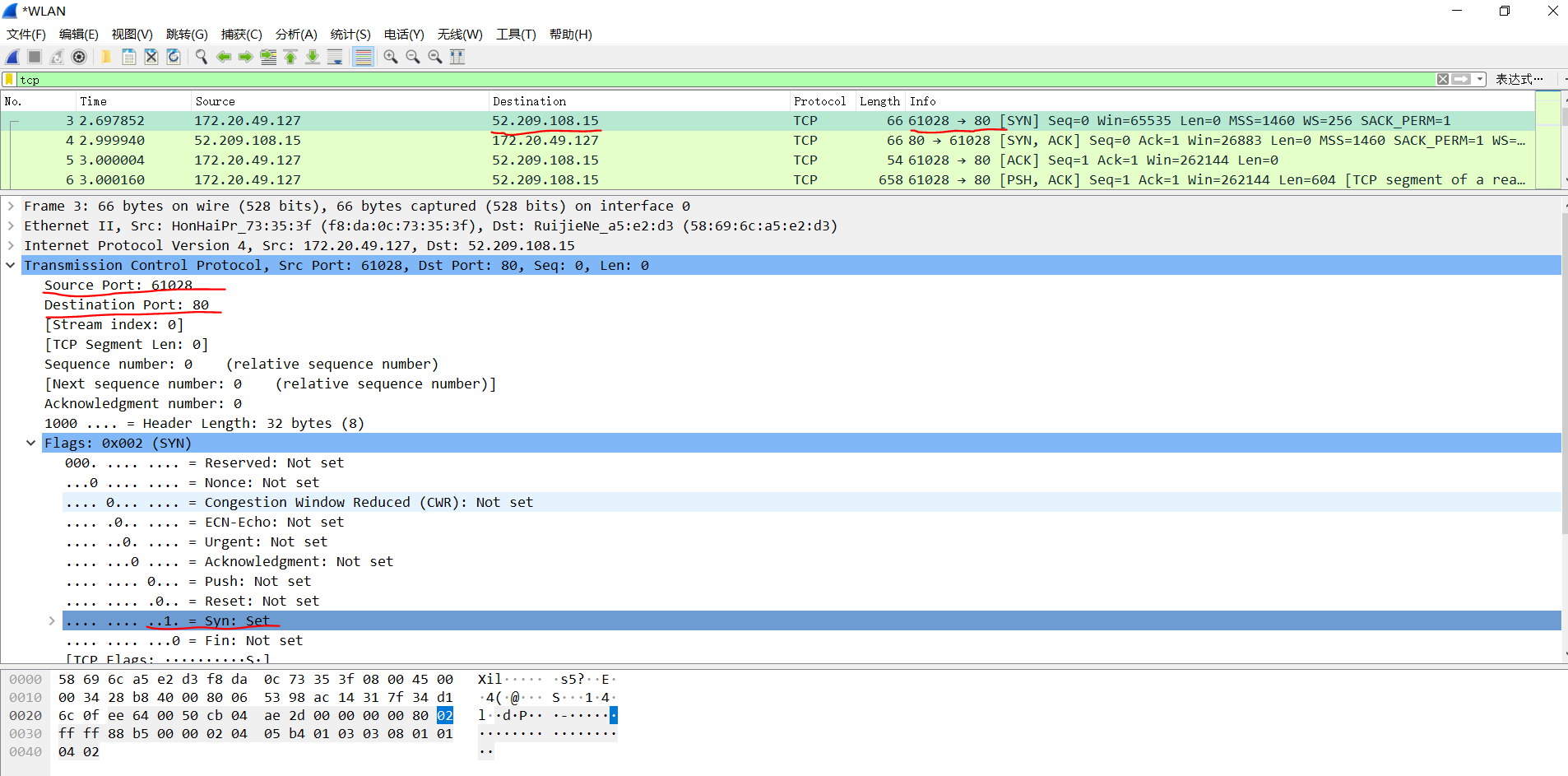
4.对较晚返回的请求码是304，因为浏览器已有缓存，并没有返回明确的内容

1. TCP 分析

A．俘获大量的由本地主机到远程服务器的 TCP 分组



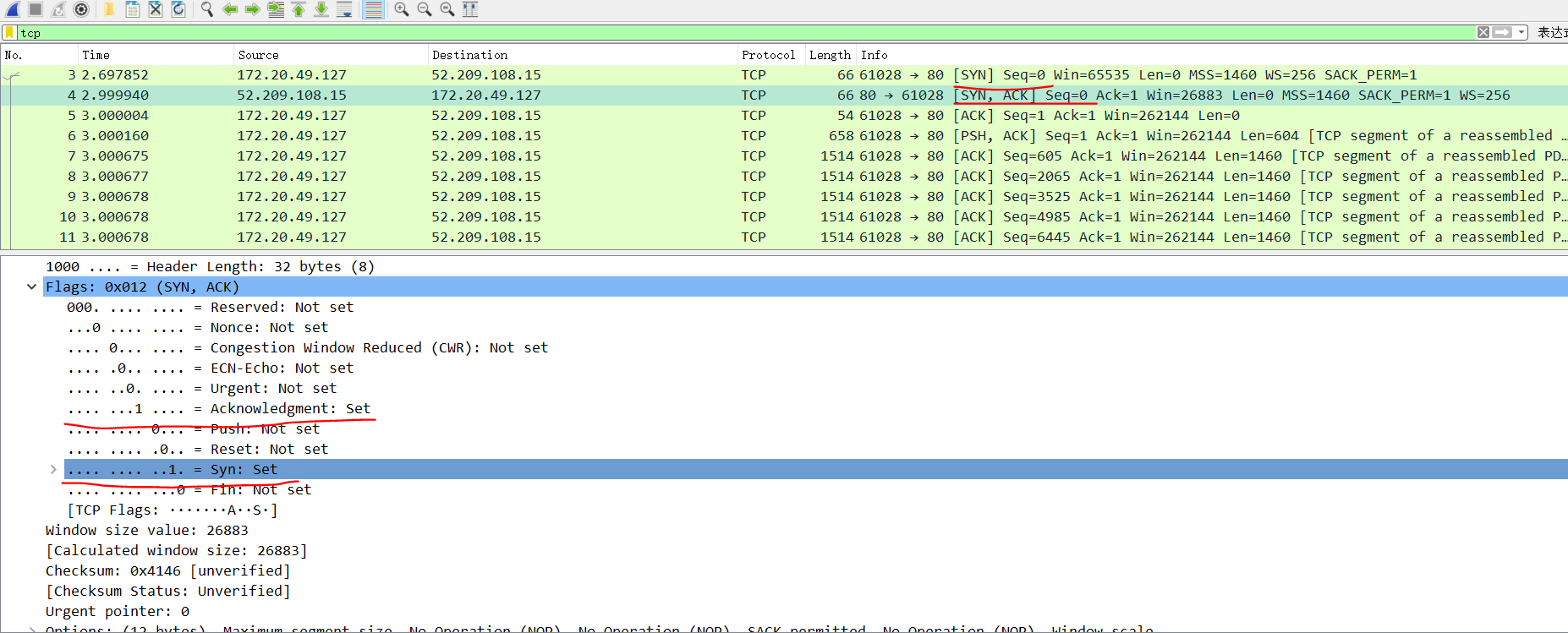
B. 浏览追踪信息



* + 1. 向 gaia.cs.umass.edu 服务器传送文件的客户端主机的IP为172.20.49.127端口为61028

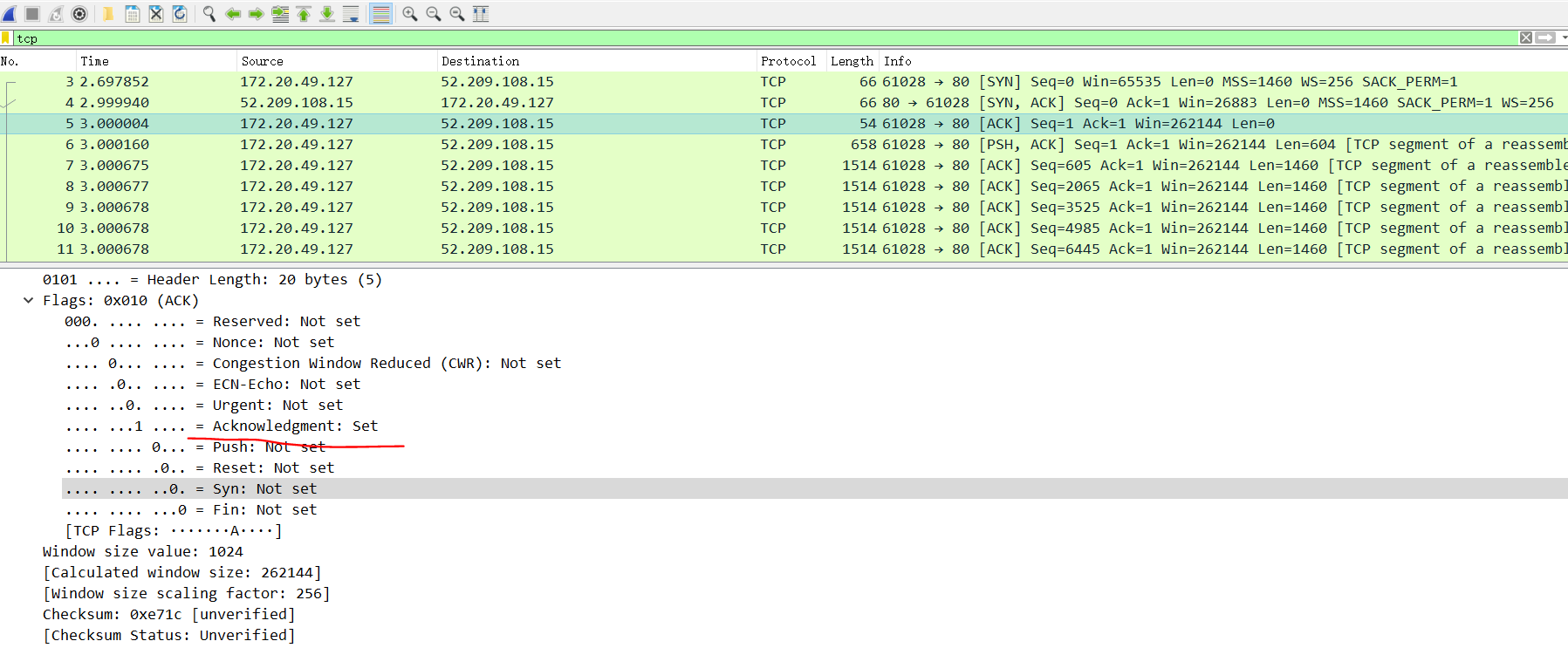
2. Gaia.cs.umass.edu 服务器的 IP 地址是52.209.108.15，对这一连接，它用来 发送和接收 TCP 报文的端口号是80

C．TCP 基础



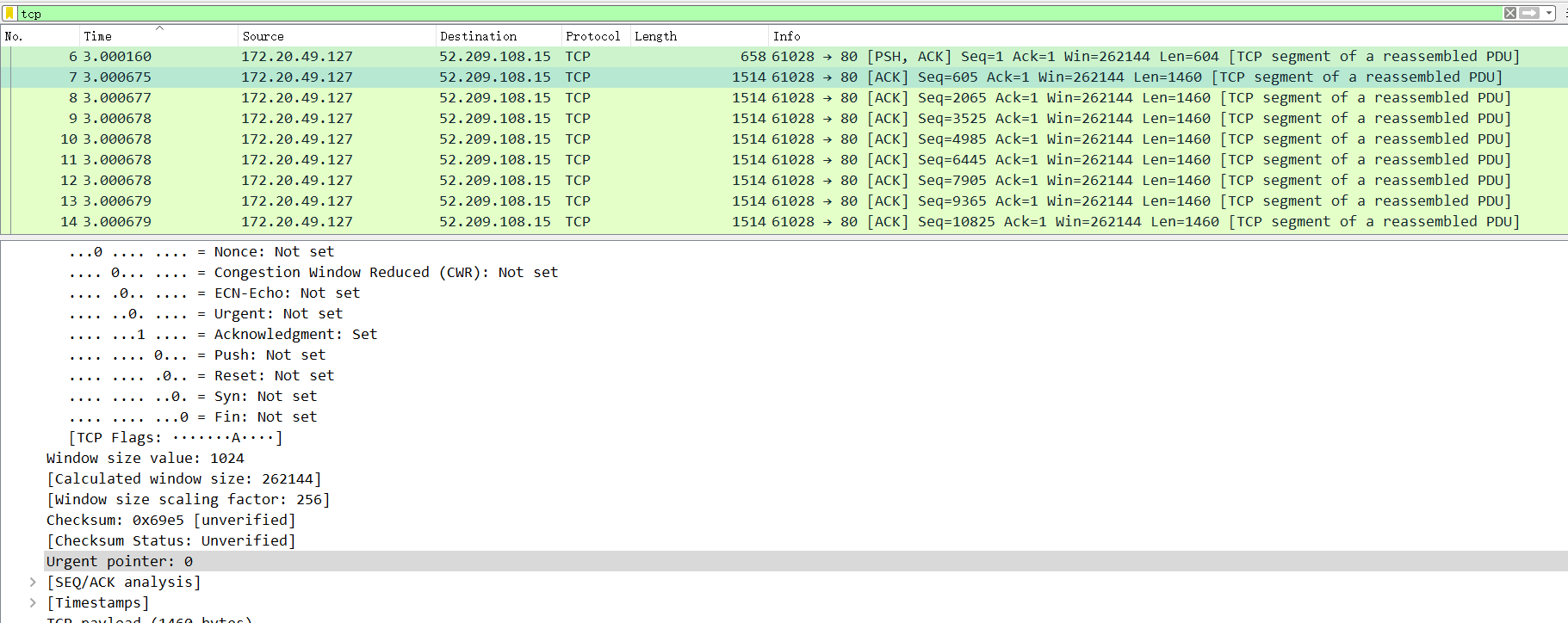
1.从上图也可以看出，初始化tcp连接时tcp的syn报文段的序号为0（随机值）；该报文段将SYN标志位置为1，表示该报文段为SYN段用于tcp建立连接

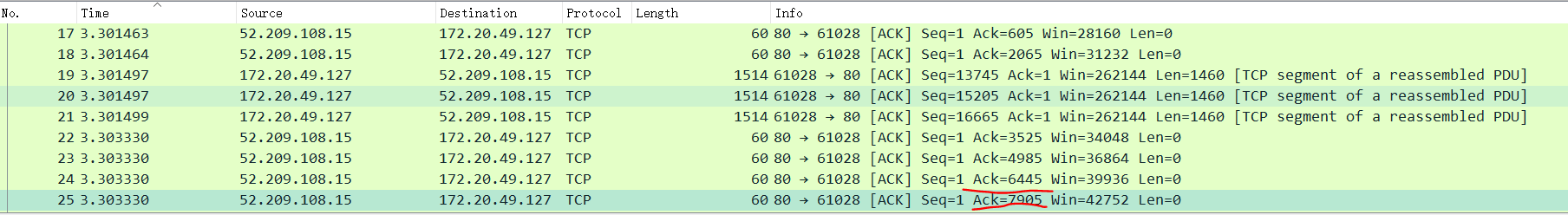
2.从上图中也可以看出SYNACK报文段的序号为0，acknowledgement字段为1，服务器端通过SYN请求报文段的seq序号加1确定acknowledgement字段；在该报文段中，使用flags部分的ack和SYN标志位置为1表示，该报文段为SYNACK报文段。

3. 

上三张图一起即为tcp三次握手的过程，前两次SYN=1第三次SYN=0，Seq为1完成握手。

4.



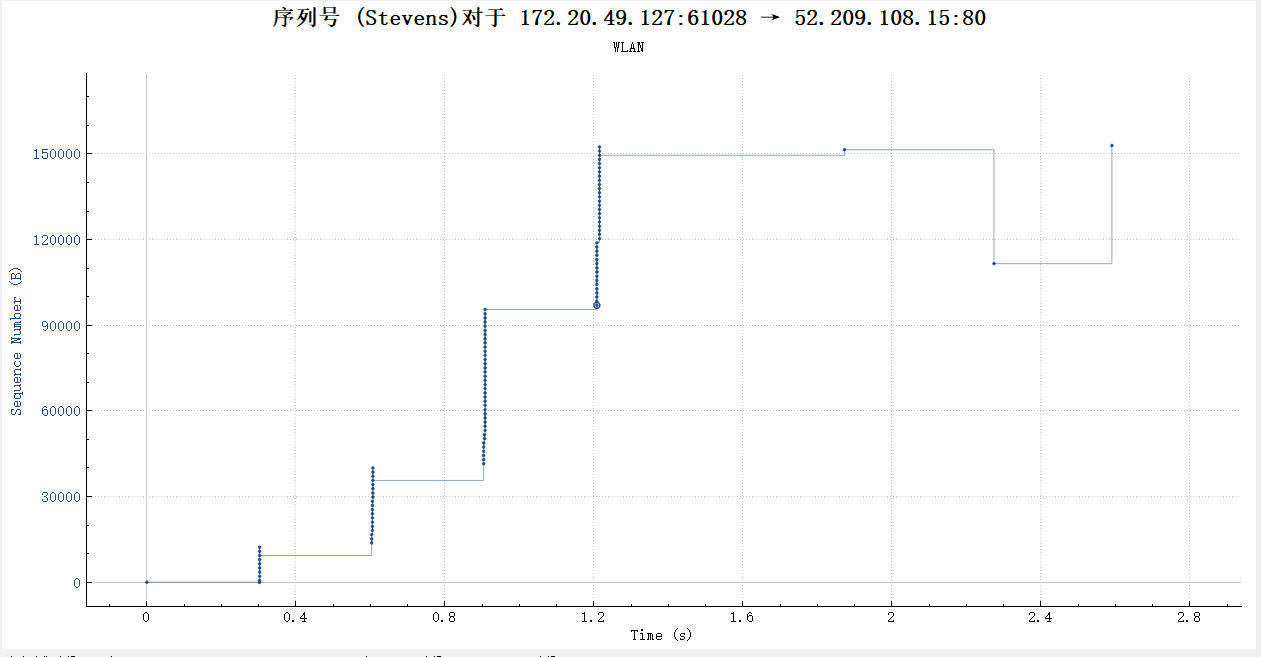


Tcp序列号是1，第六个报文段序列号是6445，在第一个报文的发送的3.000678s发送的，在3.303330s收到回复

5.从之前的图片可以看出，第一个长度为604，其余5个均为1460

6.缓存最小为28160，该窗口大小一直增大，说明限制发送端的传输以后接收端的缓存够用

7.发生了重传，看序列号的统计：如下图序列号在增加后再次变小，重传了数据



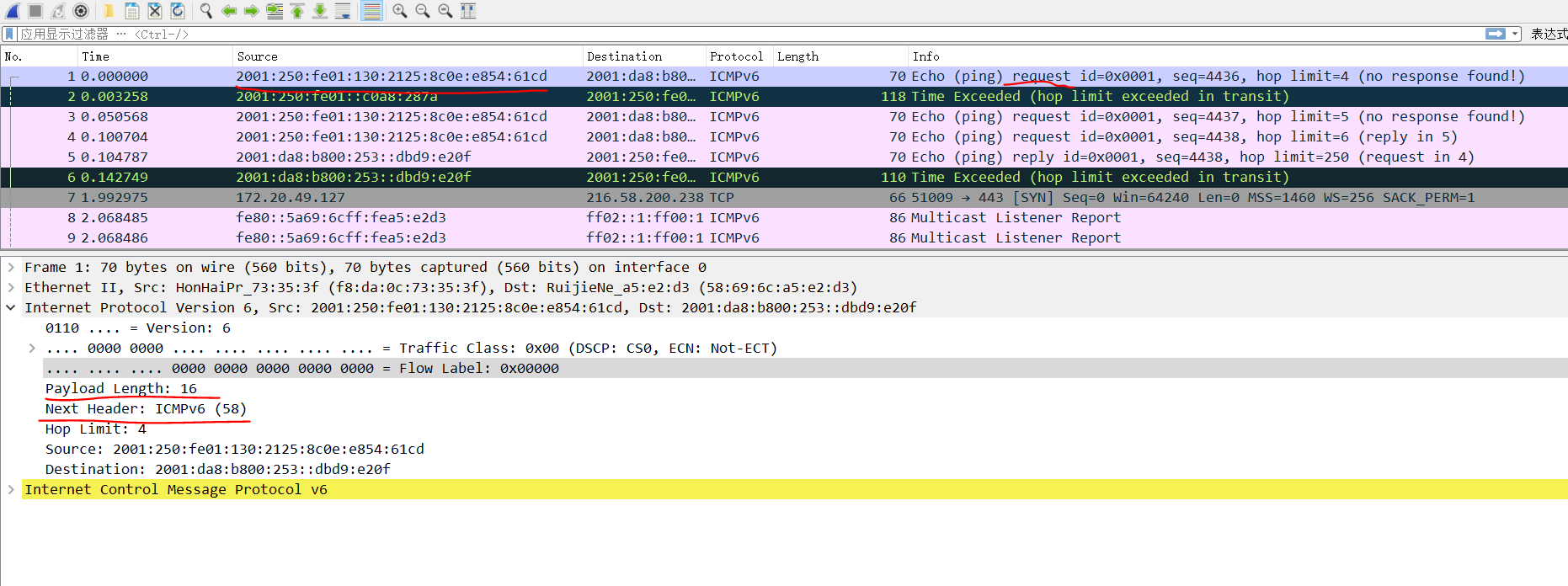
8.根据第一条和最后一条计算





吞吐率计算：

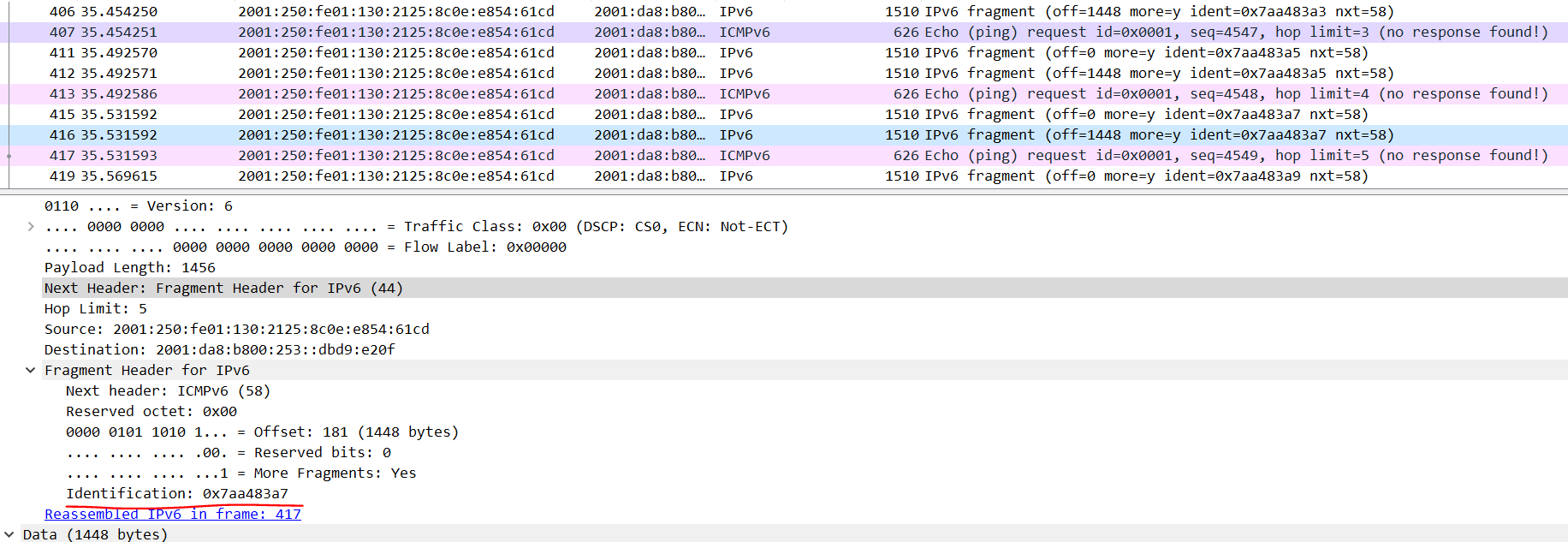
1. IP分析



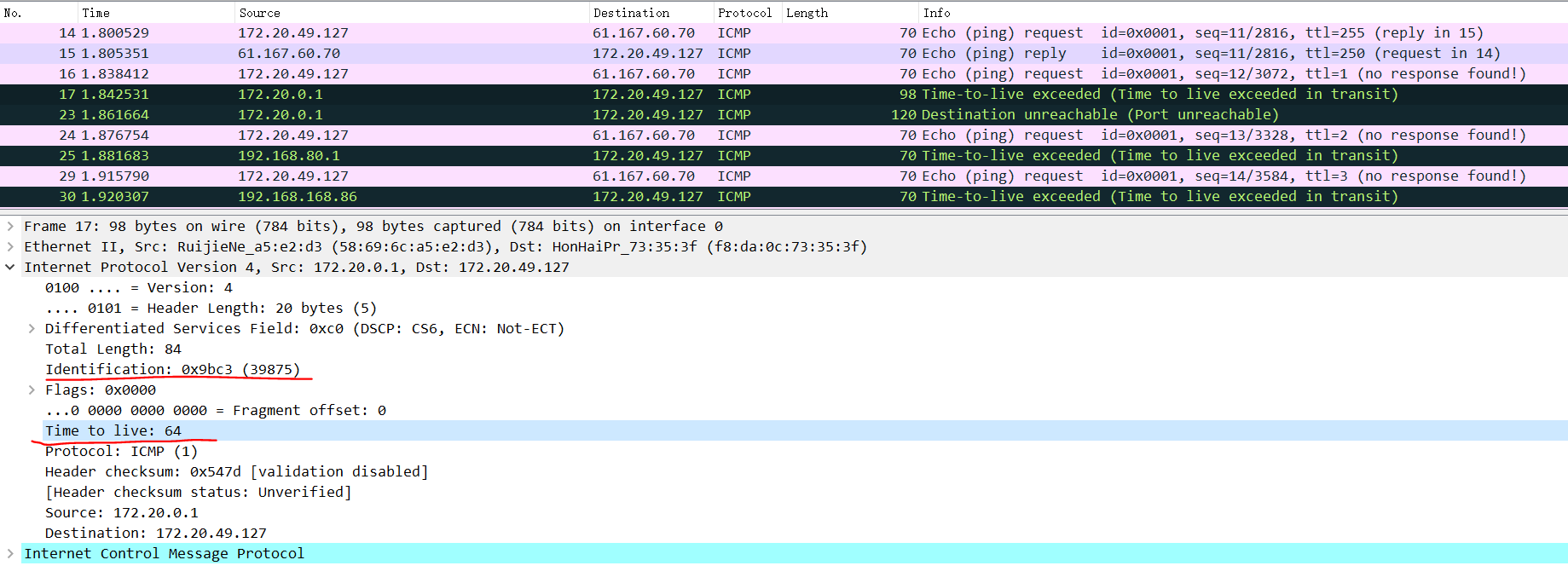
* + 1. 主机IP是2001:250:fe01:130:2125:8c0e:e854:61cd
    2. 上层协议：ICMPv6(58)
    3. IP头有56字节，由于使用的是IPv6其有固定40字节首部，Payload length为16字节，共56字节，与当初设置的56字节相同
    4. 该IP数据包净载大小为16字节，有payload length确定
    5. 并没有分片，通过Fragment Header for IPv6可以看到

思考题：

1. Hop Limit、checksum和sequence number总是发生改变。
2. 必须保持常量的是版本号、首部长度、以及协议（始终为ICMPv6）；必须改变的是Hop limit、checksum和sequence number，Hop limit为生存时间，每次转发必然改变，由于hop limit的改变，checksum也会改变，sequence number是变化的区分不同的ICMP报文。
3. Identification字段只有分片的字段才显示，且由一个数据段分片成的多个数据的值相同，并以2个字节呈递增形式。

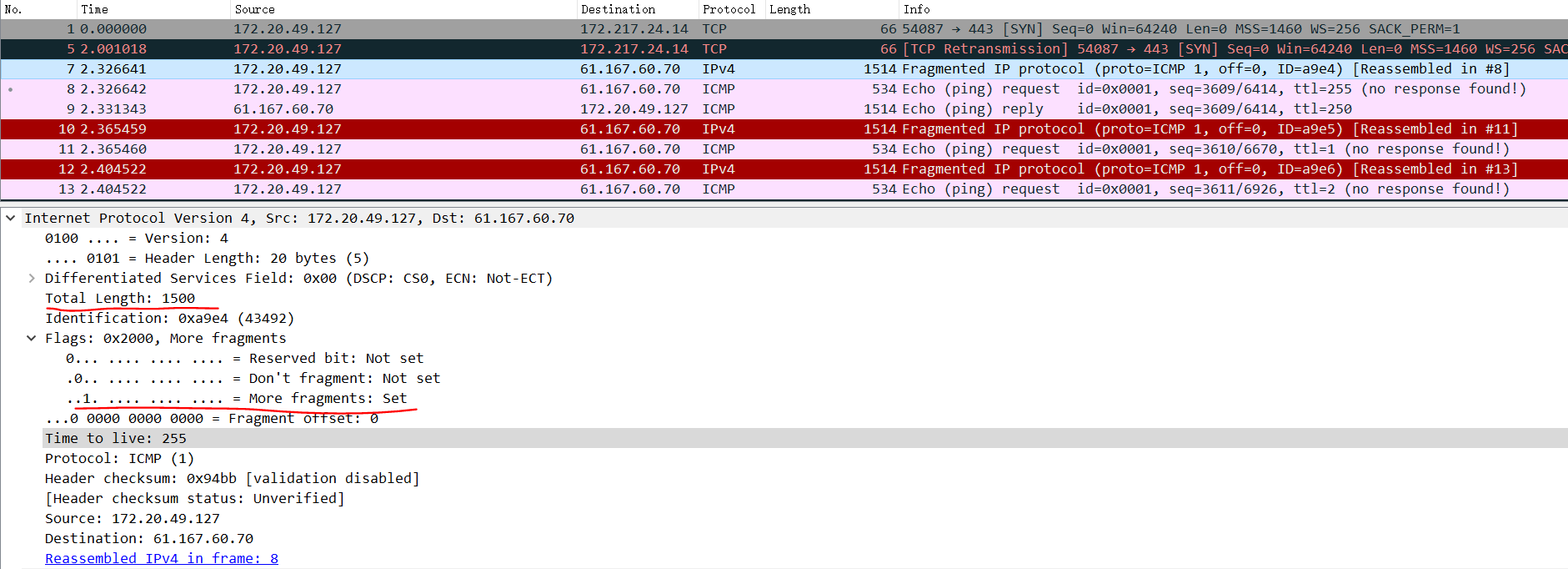


1. 由最近路由器得到的ICMP如图

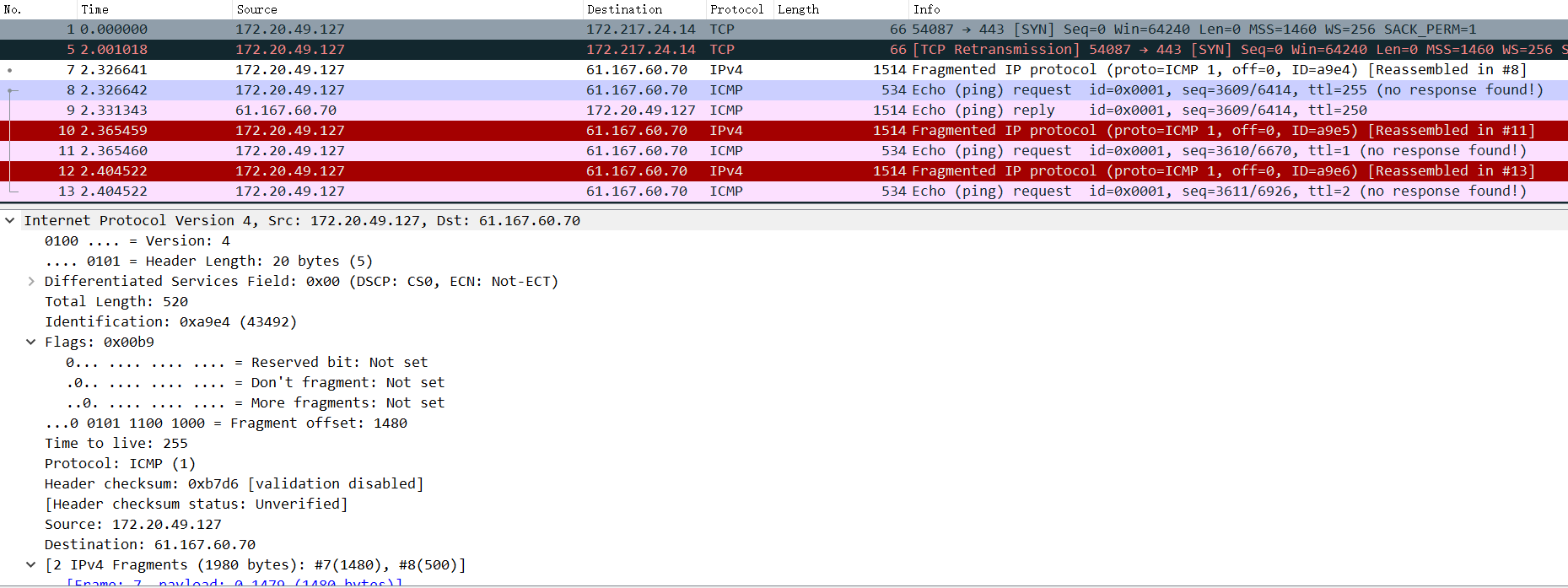


identification段为0x9bc3，TTL为64

1. identification段变化，为了区分不同的ICMP time-to-live exceeded消息；但TTL保持不变，均为一次转发
2. 当包的大小变为2000字节后发生了分片如图

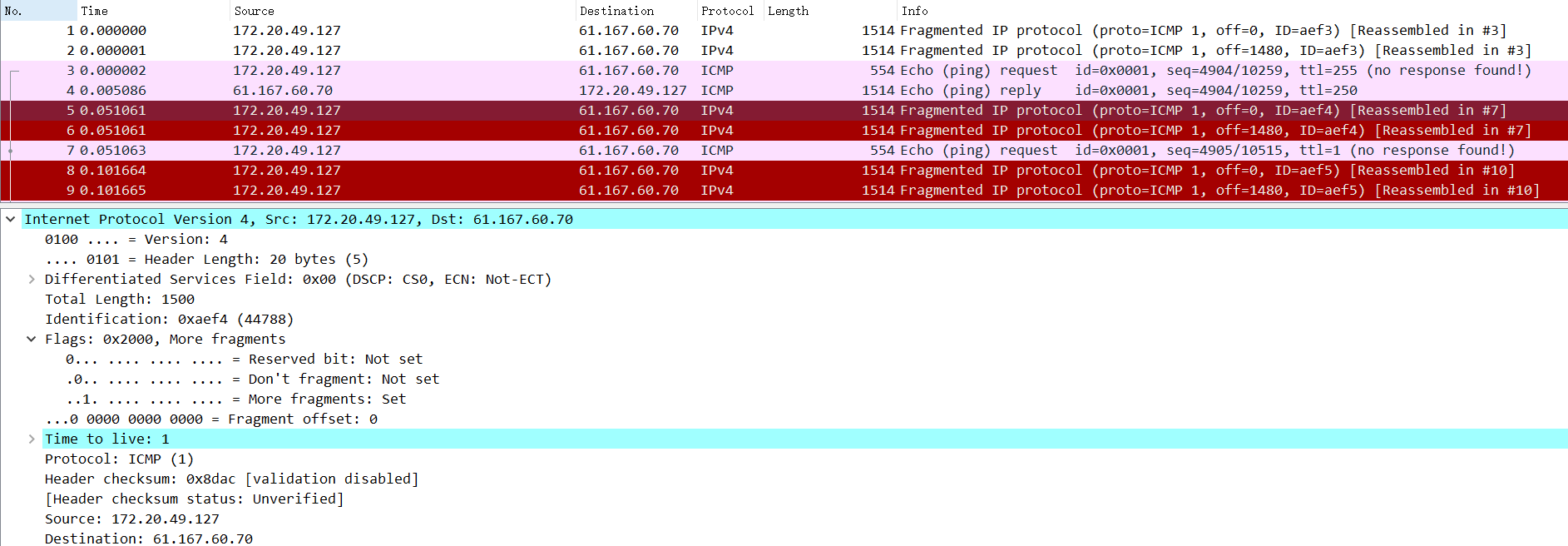


其中more fragment被置为1表示发生分片且不是最后一个总长度为1500



第二片identification段相同，并且第二片为最后一组且有偏移，more fragments为0表示最后一片

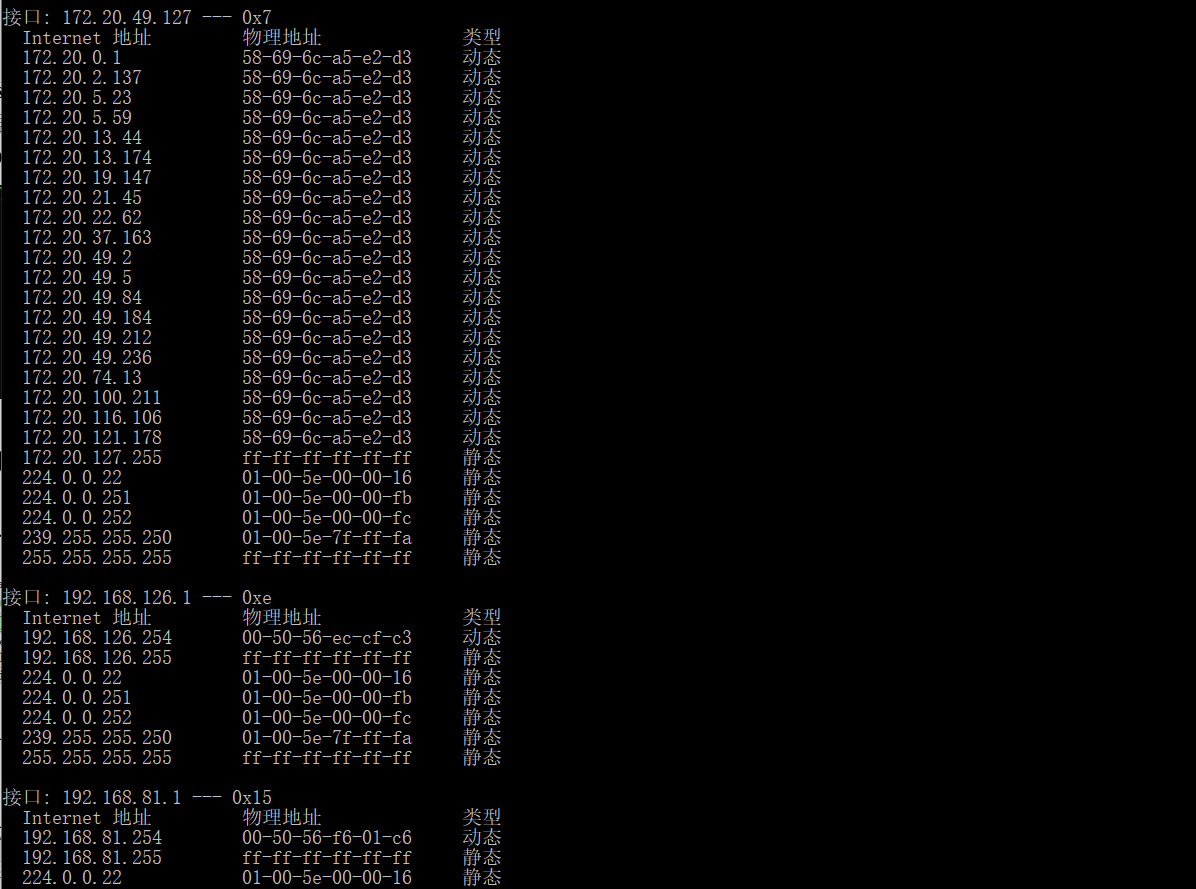
1. 当设置为3500时如图：



可以看出被分为了3片

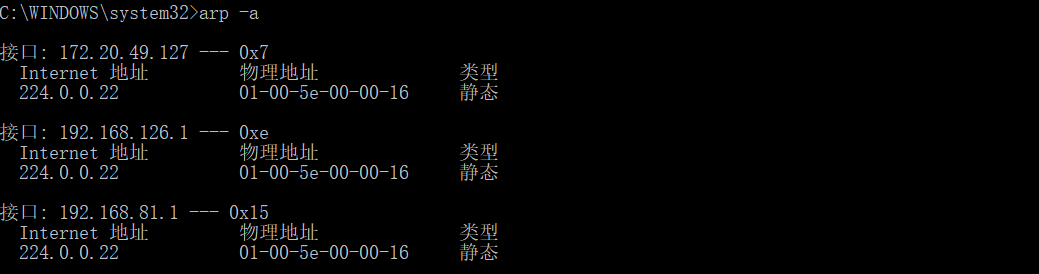
标志位部分和checksum部分发生了变化。

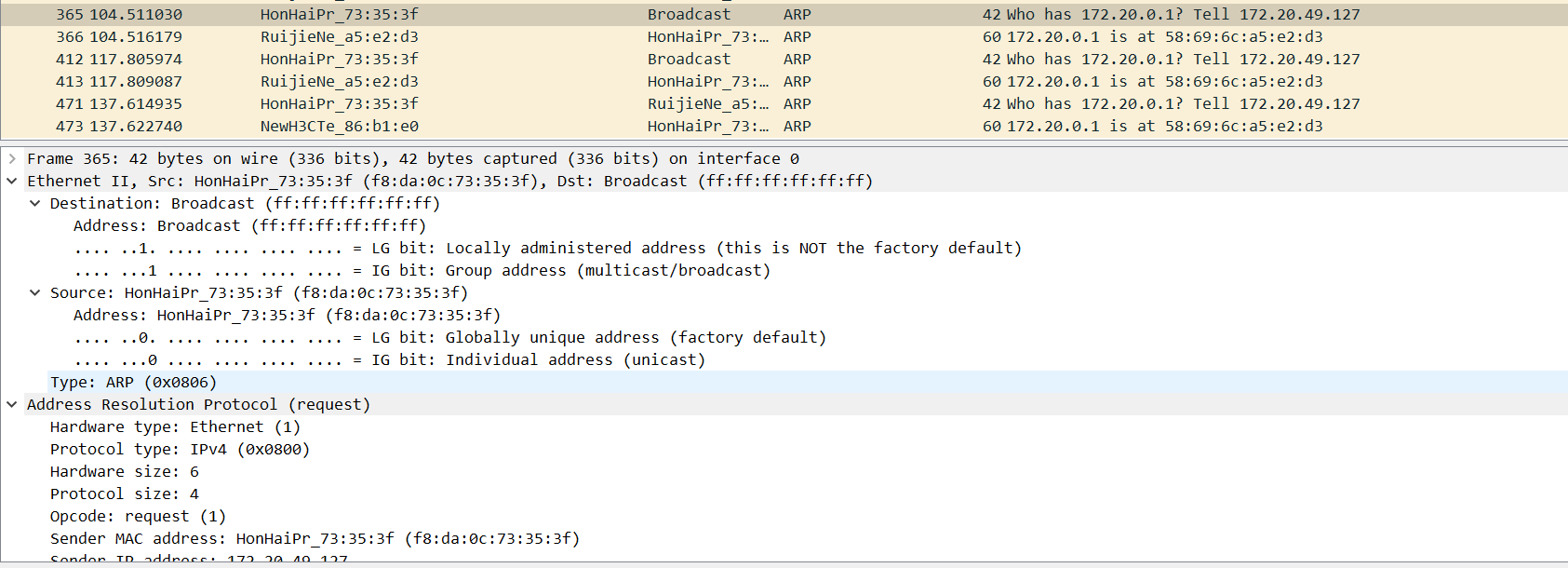
1. 抓取 ARP 数据包
   * 1. 查看arp表内容：



每一列均由其表头决定，如地址等；每一行为IP地址对应的MAC地址以及类型

* + 1. 删除arp表：

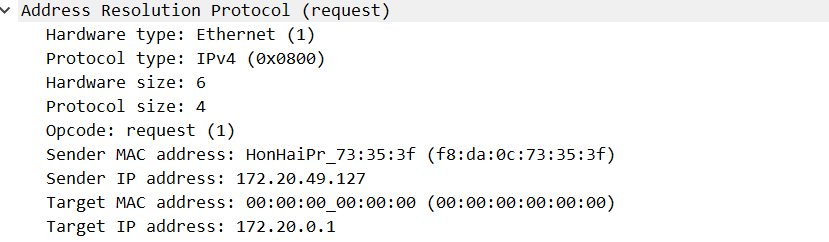




1)由上图看到第一条为ARP请求，广播“谁是172.20.0.1”告诉172.20.49.127，下一条即为回复的MAC地址，也可以看到广播的物理地址为ff:ff:ff:ff:ff:ff

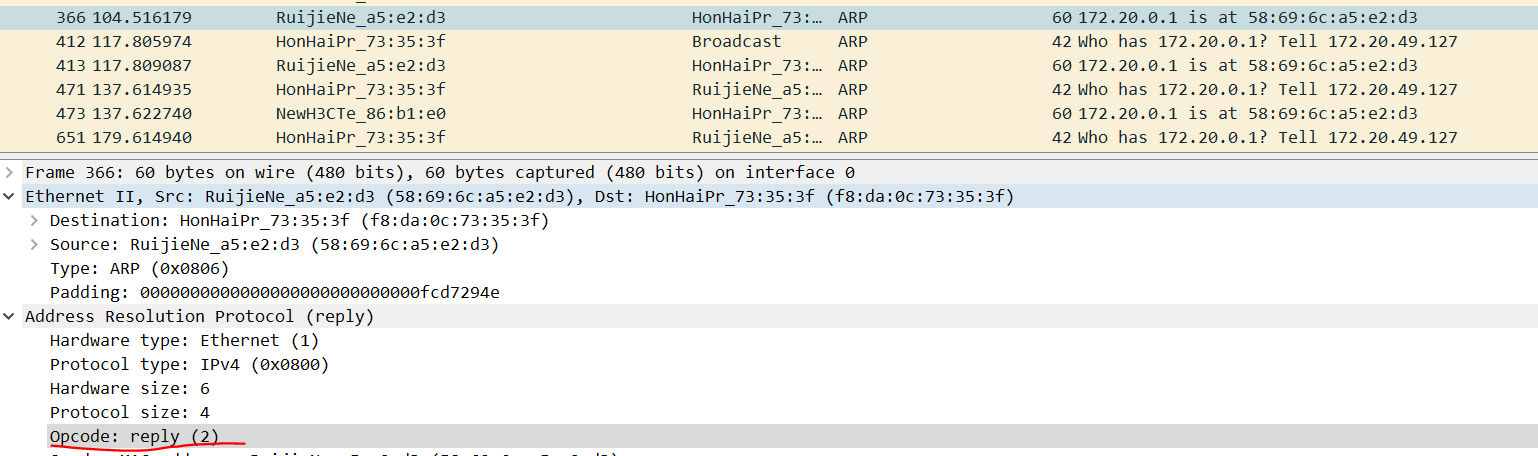
ARP格式：





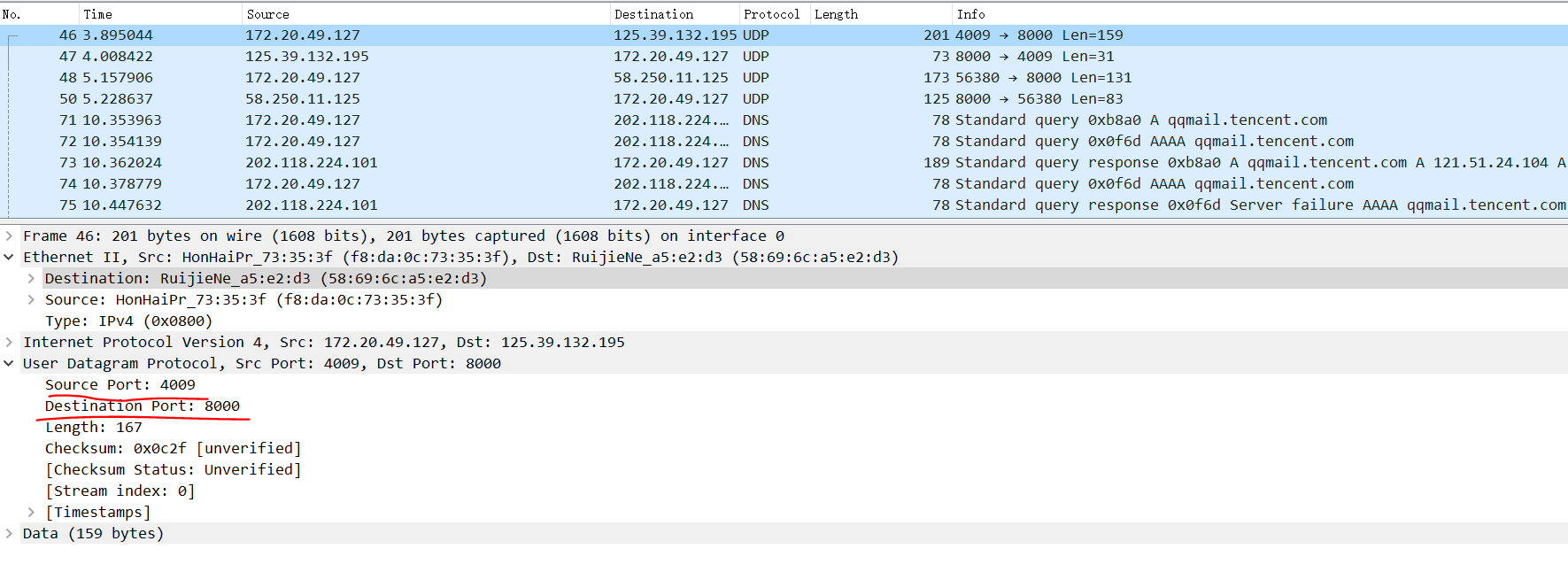
组成共有10部分，长度见上图

2)可以通过Opcode判断是请求还是应答，请求为request(1)应答为reply(2)，请求见上图，回答见下图



3)在查询时由于不知道目的IP对应的MAC地址故需要广播地址进行广播，而响应的报文可以从查询的报文中获得IP和MAC，故可以通过明确的地址发送

1. 抓取 UDP 数据包



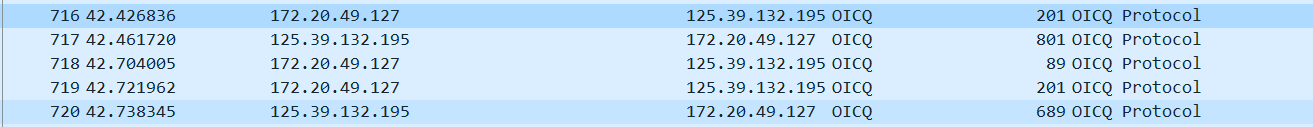
* + 1. 可以看出qq消息基于的是UDP
    2. 本机IP为172.20.49.127，目的主机IP为125.39.132.195
    3. 我的主机发送的端口号为4009，qq服务器的端口号是8000
    4. 由上图看出UDP数据包的格式，头部包含源端口号、目的端口号、长度和checksum组成。各占2个字节



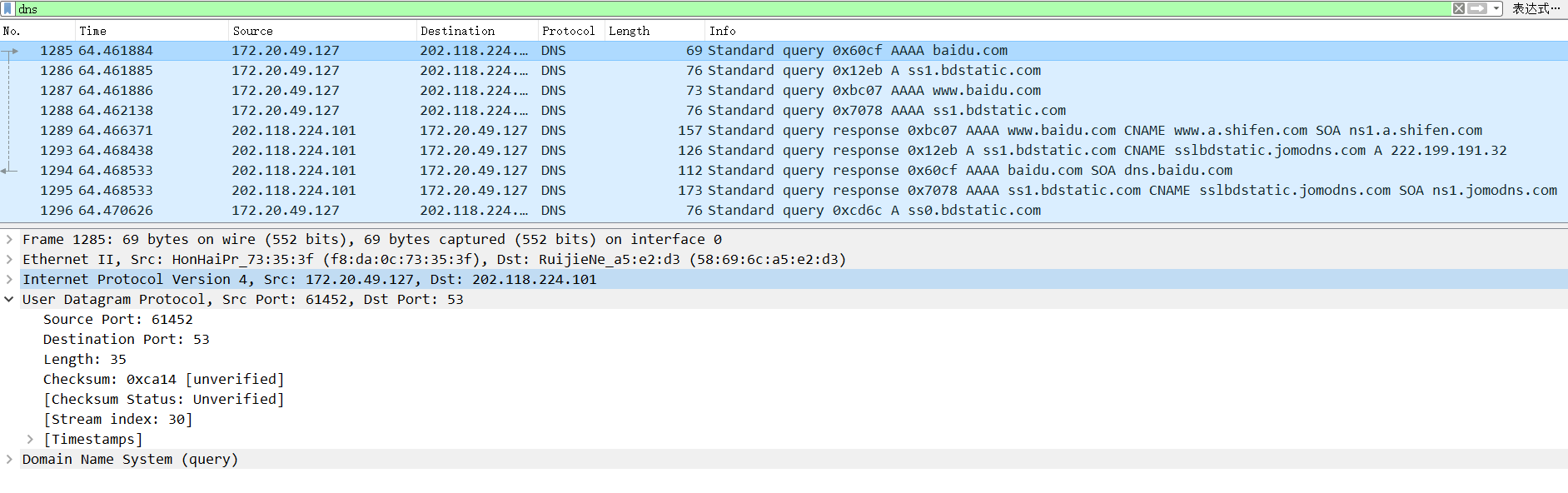
* + 1. 因为udp是不可靠数据传输，当服务器端接收到数据后要将接收结果返回给客户端。

可以看出UDP是不可靠数据传输，其只提供了一次返回的基本ACK，没有保证数据一定被送达，

可以看出，UDP数据包是没有序列号的，因此不能像TCP协议那样先进行握手再发送数据，从而发送的数据可能是乱序的。



1. 利用 WireShark 进行 DNS 协议分析



查询的地址为202.118.224.101，是哈工大的DNS服务器。



其中可以看到A，AAAA，CNAME查询等，A为查询IPv4下指向的地址，AAAA为IPv6下指向的地址，CNAME为别名，上面是4条查询，紧接着就是返回的四条查询结果。

## 四、实验心得

通过使用wireShark进行抓包，进行分析，深刻的，非常深刻的体会到了各协议交换信息的过程，原本以为只要按照操作做完就好，实则遇到很多问题，只有真正理解其原理才能看懂抓取的包中的含义。又增添了很多原本比较模糊的知识，实践非常有意义。