哈尔滨工业大学

<<计算机网络>>

实验报告

(2018 年度春季学期**)**

姓名:	
学号:	
学院:	计算机科学与技术学院
教师:	

实验五 利用 Wireshark 进行协议分析

一、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。

二、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

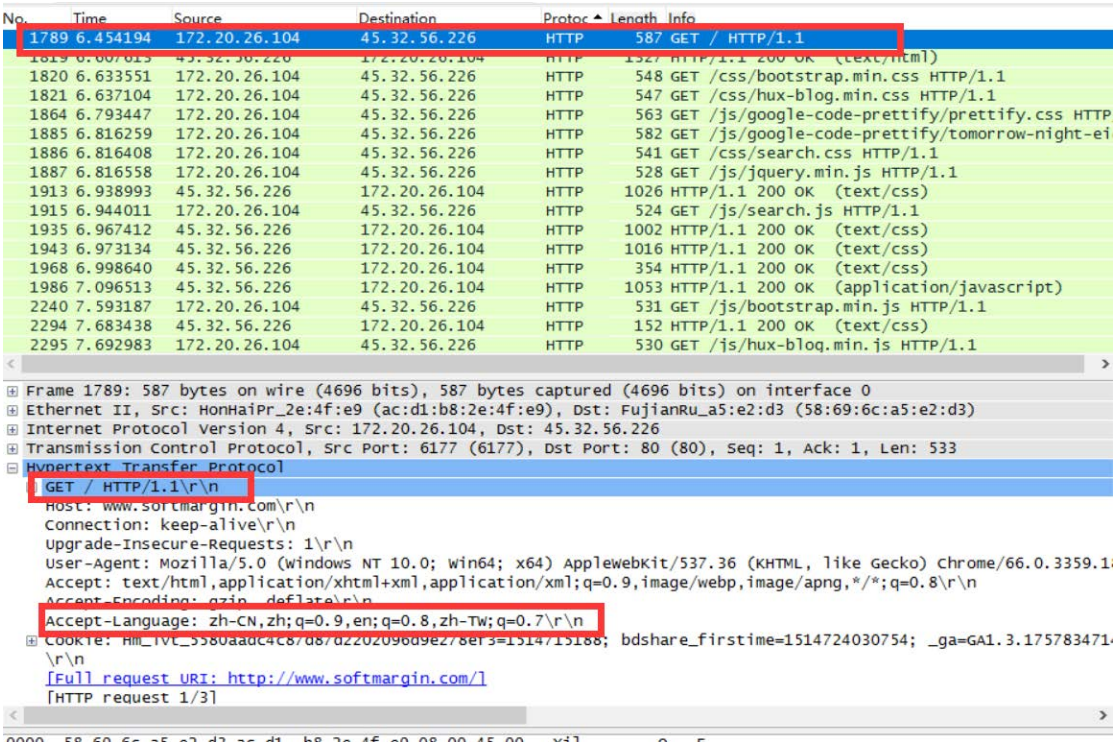
选做内容：

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. HTTP 分析

1.1 HTTP GET/response 交互



问题：

1. **你的浏览器运行的是 HTTP1.0，还是 HTTP1.1？你所访问的服务器所运行 HTTP 协议的版本号是多少？**

   都为 HTTP 1.1

2. **你的浏览器向服务器指出它能接收何种语言版本的对象？**

   zh-CN,zh;,en;,zh-TW;

3. **你的计算机的 IP 地址是什么？服务器 http://www.hit.edu.cn 的 IP 地址是多少？**

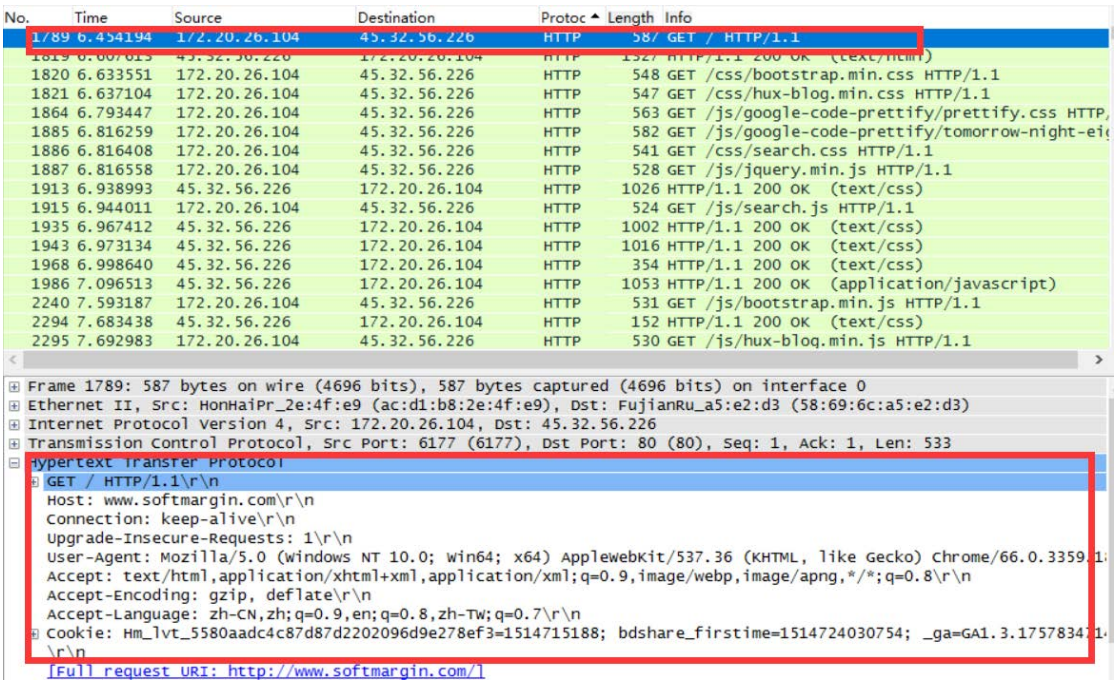
   本地计算机 IP：172.20.26.104

   服务器 IP：45.32.56.226

4. **从服务器向你的浏览器返回的状态代码是多少？**

   200

1.2 HTTP 条件 GET/response 交互



1. **分析你的浏览器向服务器发出的第一个 HTTP GET 请求的内容， 在该请求报文中，是否有一行是：IF-MODIFIED-SINCE？**

   并没有

2. **分析服务器响应报文的内容，服务器是否明确返回了文件的内容？如何获知？**

   确实返回了文件内容

   根据服务器返回报文的状态码可判断是否返回了文件内容，也可直接查看返回报文是否含文件内容

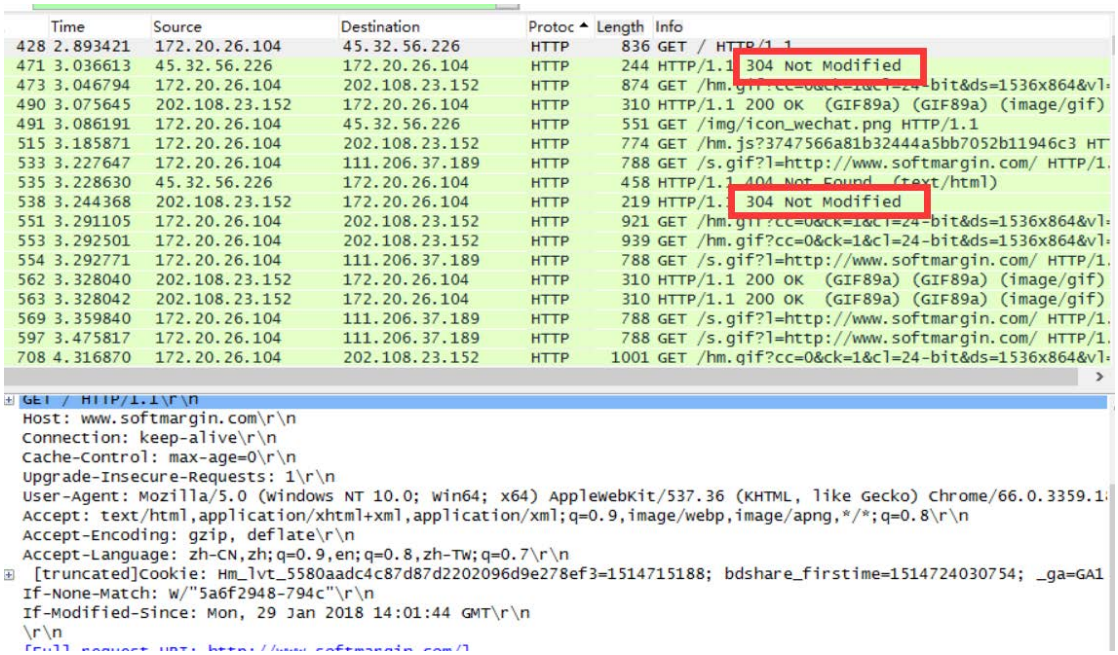
   HTTP Status Code 为 304 时不返回文件

   HTTP Status Code 为 200 时返回文件

3. **分析你的浏览器向服务器发出的较晚的“HTTP GET”请求，在该请 求报文中是否有一行是：IF-MODIFIED-SINCE？如果有，在该首 部行后面跟着的信息是什么？**

   存在 IF-Modifide-Since,后面跟着请求文件在本地缓存中的更改时间。

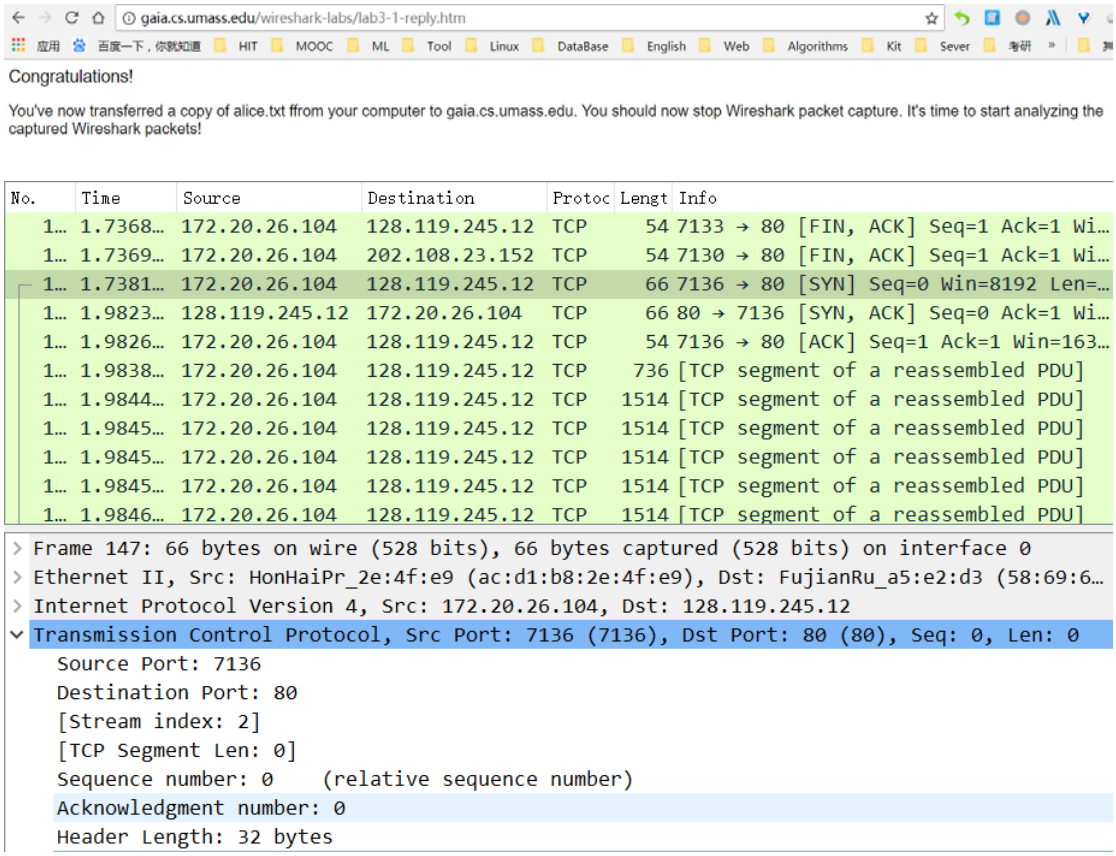
4. **服务器对较晚的 HTTP GET 请求的响应中的 HTTP 状态代码是多 少？服务器是否明确返回了文件的内容？请解释。**



304，没有返回明确文件，由于本地缓存存有该文件，并且该文件与服务器的一致，服务器此文件并没由更新。

2 TCP 分析

俘获大量的由本地主机到远程服务器的 TCP 分组：



1. 向 gaia.cs.umass.edu 服务器传送文件的客户端主机的 IP 地址和 TCP 端口号是多

少？

IP：172.20.26.104

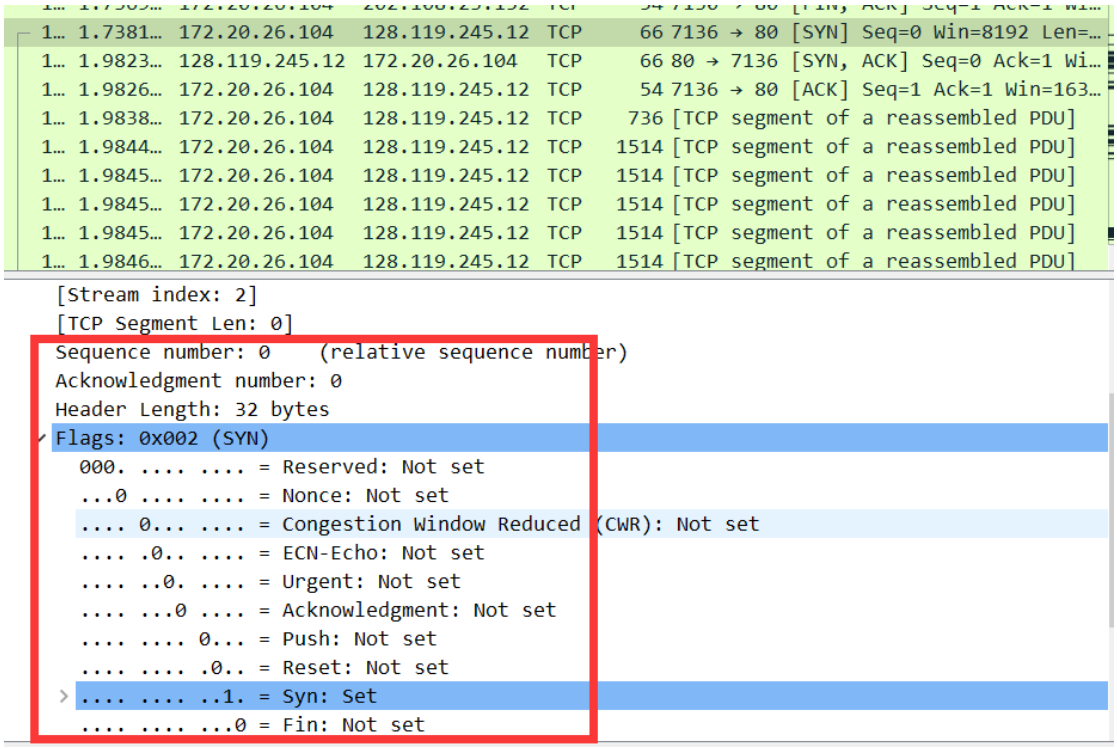
port：7136

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

IP：128.119.245.12

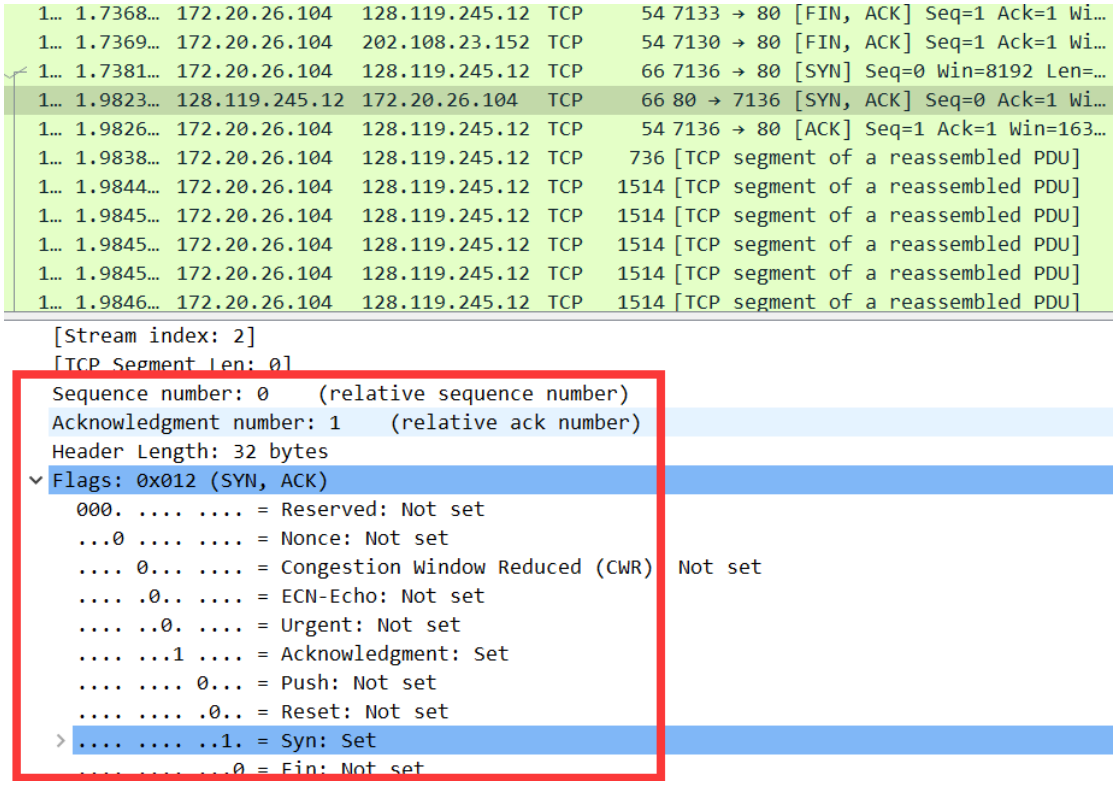
port：80

3. 客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号（sequence number）是多少？在该报文段中，是用什么来标示该报文段是 SYN 报文段的？



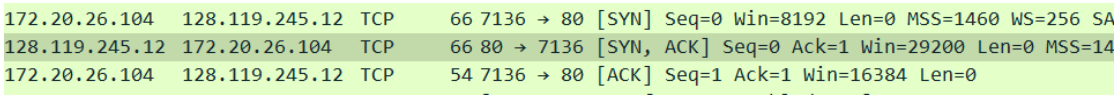
序号为 0，Flags 字段第 11 位表示 SYN 段

4. 服务器向客户端发送的 SYN ACK 报文段序号是多少？该报文段 中，Acknowledgement 字段的值是多少？Gaia.cs.umass.edu 服务器 是如何决定此值的？在该报文段中，是用什么来标示该报文段是 SYNACK 报文段的？



SYN ACK 报文段序号为 0，Acknowledgement 字段值为 1，ack 是根据上一次客户端发给服务器的 seq+1 得到的，Flags 字段第 8 位为 1 表示 ACK ， 第 11 位为 1 表示 SYNSYN

5. 能从捕获的数据包中分析出 tcp 三次握手过程吗？

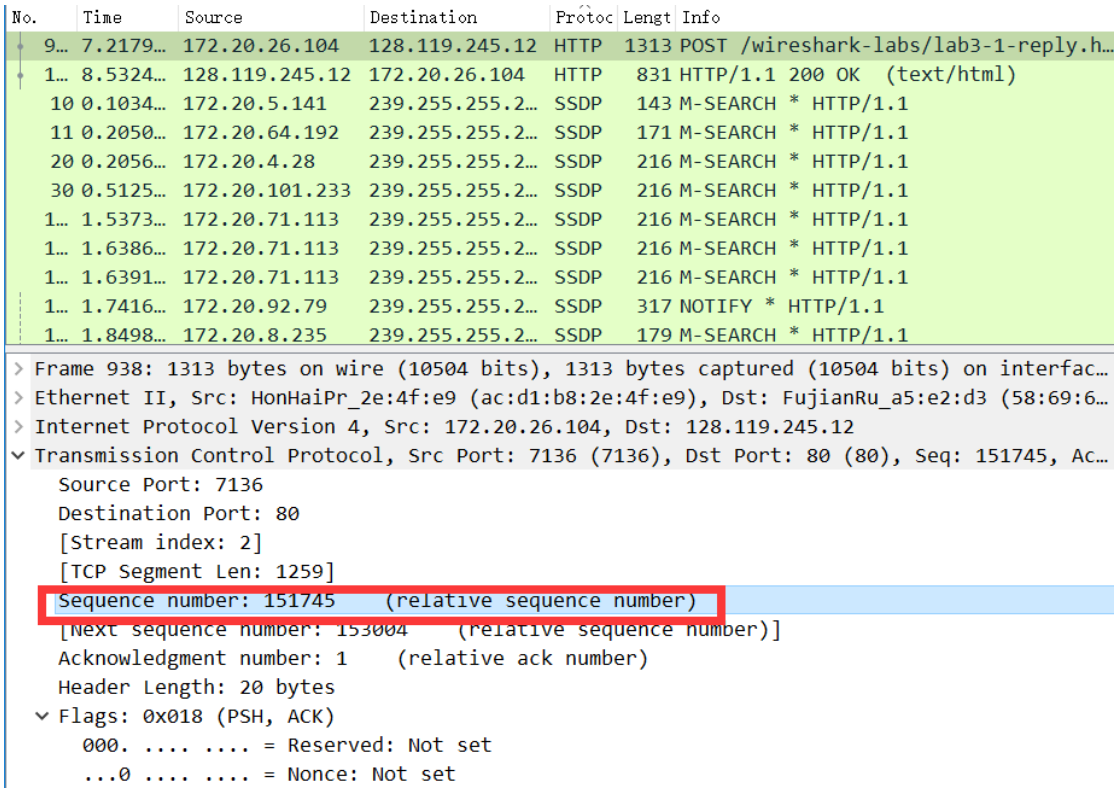


首先客户端向服务器发送 seq=0 的建立连接的请求

然后服务器向客户端返回 seq=0,ack=0+1=1 的响应

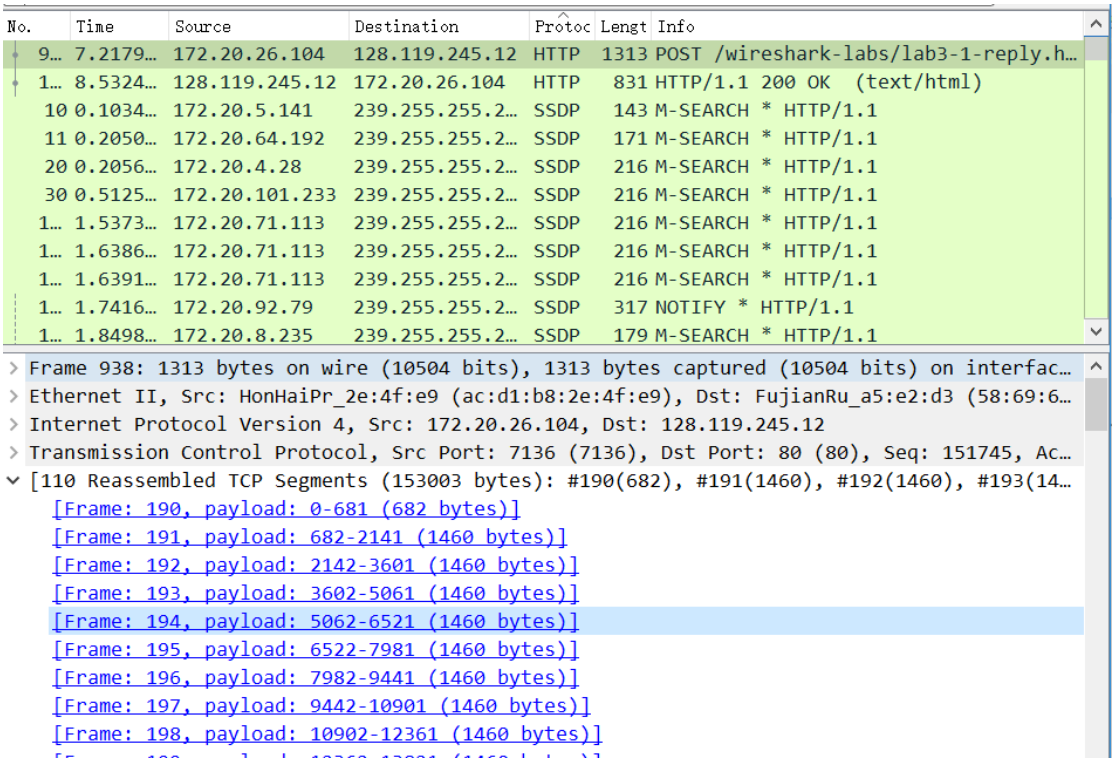
客户端收到响应，返回 seq=1,ack=0+1=1 的确认报文，连接建立

6. 包含 HTTP POST 命令的 TCP 报文段的序号是多少？



Seq=151745

7. 如果将包含 HTTP POST 命令的 TCP 报文段看作是 TCP 连接上的 第一个报文段，那么该 TCP 连接上的第六个报文段的序号是多 少？是何时发送的？该报文段所对应的 ACK 是何时接收的？



第六个报文段 Seq=6523，在 http post 发送之前，tcp 连接建立之后发送。具体发送时间

```
Arrival Time: May 29, 2018 20:52:06.210985000 ﷺ
```

No.	Time	Source	Destination	Protoc	Lengt	Info
2…	2.227446	128.119.245.12	172.20.26.104	TCP	60	80 → 7136 [ACK] Seq=1 Ack=683 Win=30592…
2…	2.227513	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
2…	2.228669	128.119.245.12	172.20.26.104	TCP	60	80 → 7136 [ACK] Seq=1 Ack=2143 Win=3353…
2…	2.228670	128.119.245.12	172.20.26.104	TCP	60	80 → 7136 [ACK] Seq=1 Ack=3603 Win=3648…
2…	2.228670	128.119.245.12	172.20.26.104	TCP	60	80 → 7136 [ACK] Seq=1 Ack=6523 Win=4236…
2…	2.228723	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
2…	2.228739	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
2…	2.228755	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
2…	2.228768	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
2…	2.228781	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]
2…	2.228794	172.20.26.104	128.119.245.12	TCP	1514	[TCP segment of a reassembled PDU]

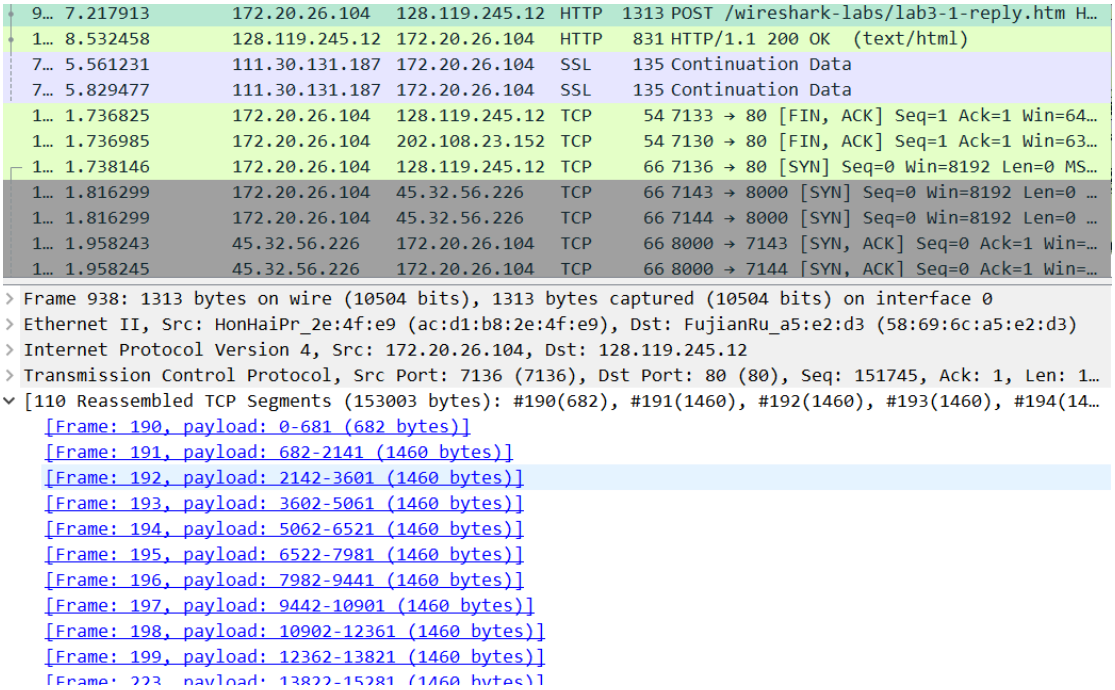
```
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.20.26.104
v Transmission Control Protocol, Src Port: 80 (80), Dst Port: 7136 (7136), Seq: 1, Ack: 6523, Len: 0
    Source Port: 80
    Destination Port: 7136
    [Stream index: 2]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 6523    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x010 (ACK)
    Window size value: 331
    [Calculated window size: 42368]
    [Window size scaling factor: 128]
  > Checksum: 0xf96b [validation disabled]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
```

对应的 ack 即为服务器返回的第六个 ack，返回时间为：

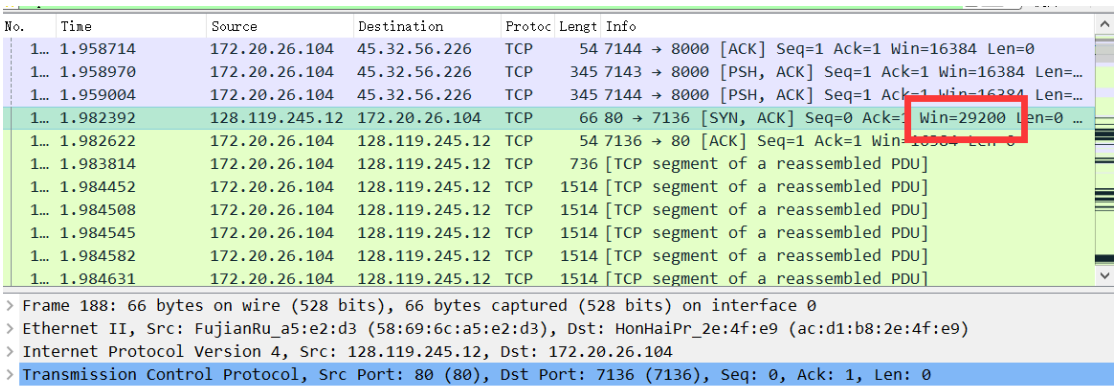
```
Arrival Time: May 29, 2018 20:52:06.455073000 ﷺ
```

8. 前六个 TCP 报文段的长度各是多少？

如图：



9. 在整个跟踪过程中，接收端公示的最小的可用缓存空间是多少？ 限制发送端的传输以后，接收端的缓存是否仍然不够用？

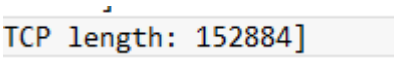


如图，接收端公示的最小的可用缓存空间是 29200，该窗口大小会一直增加，不会出现接收端的缓存不够用的情况。

10 在跟踪文件中是否有重传的报文段？进行判断的依据是什么？TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。

没有出现重传，没有重复的序号报文。

总报文大小：



开始时间：

```
Arrival Time: May 29, 2018 20:52:06.210985000 ﷺ
```

结束时间：

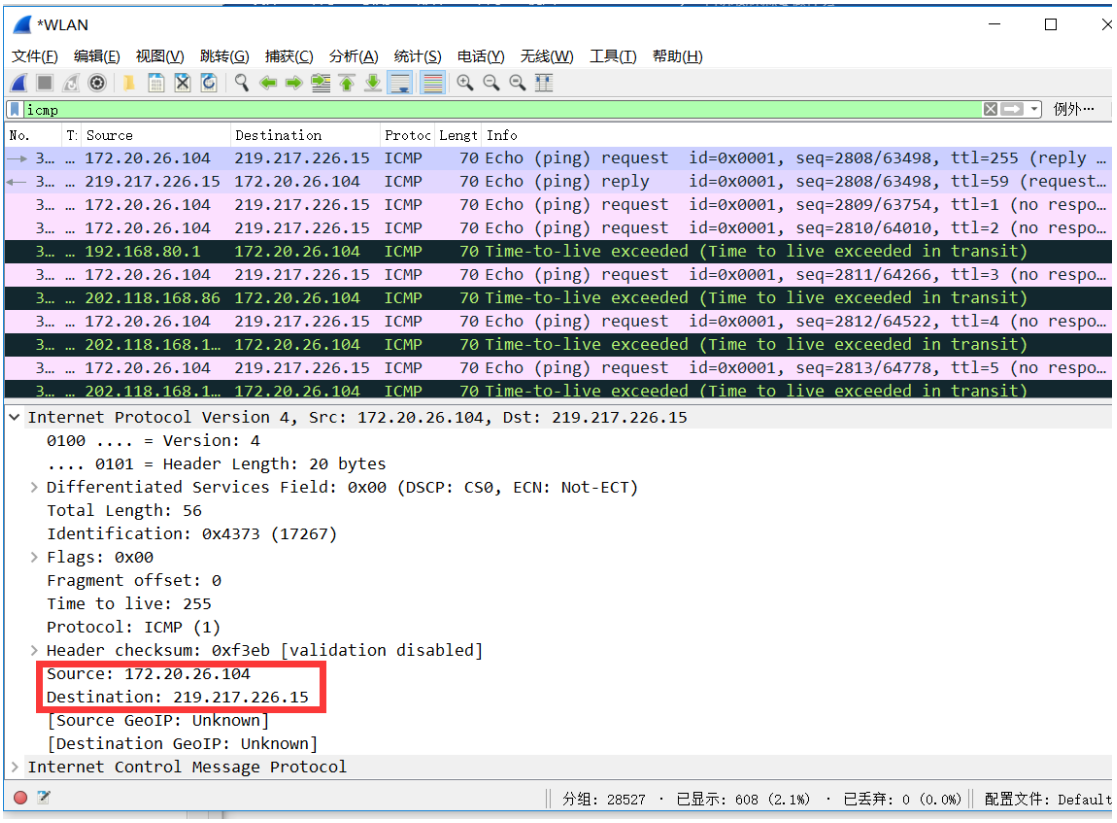
```
Arrival Time: May 29, 2018 20:52:06.455073000 ﷺ
```

throughput = 152884byte / 0.24s = 637016.66KB/S

3. IP 分析

A. 对捕获的数据包进行分析

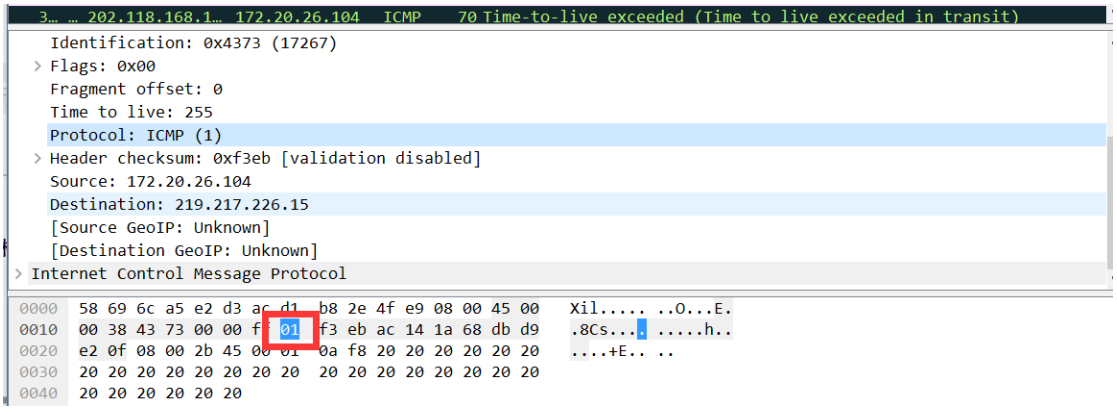
1. 在你的捕获窗口中，应该能看到由你的主机发出的一系列 ICMP Echo Request 包和中间路由器返回的一系列 ICMP TTL-exceeded 消息。选 择第一个你的主机发出的 ICMP Echo Request 消息，在 packet details 窗口 展开数据包的 Internet Protocol 部分



1. 你主机的 IP 地址是什么？

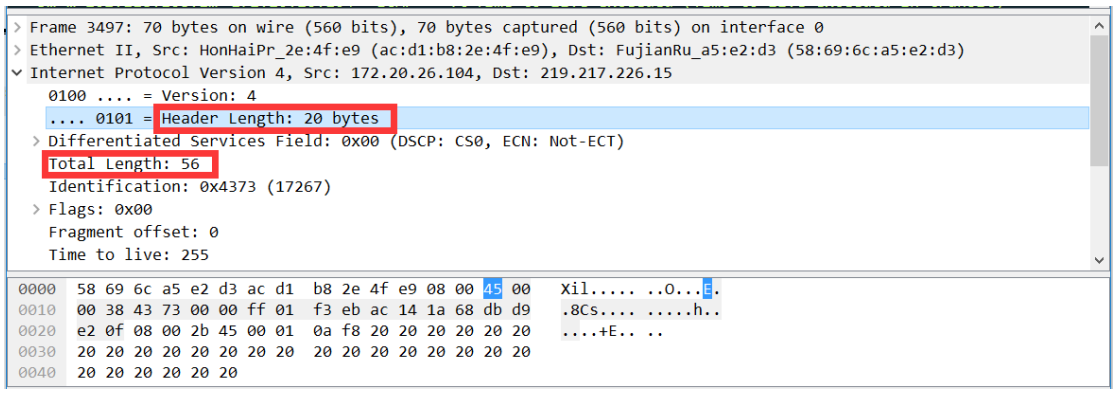
主机 IP 地址:172.20.26.104

2. 在 IP 数据包头中，上层协议（upper layer）字段的值是什么？



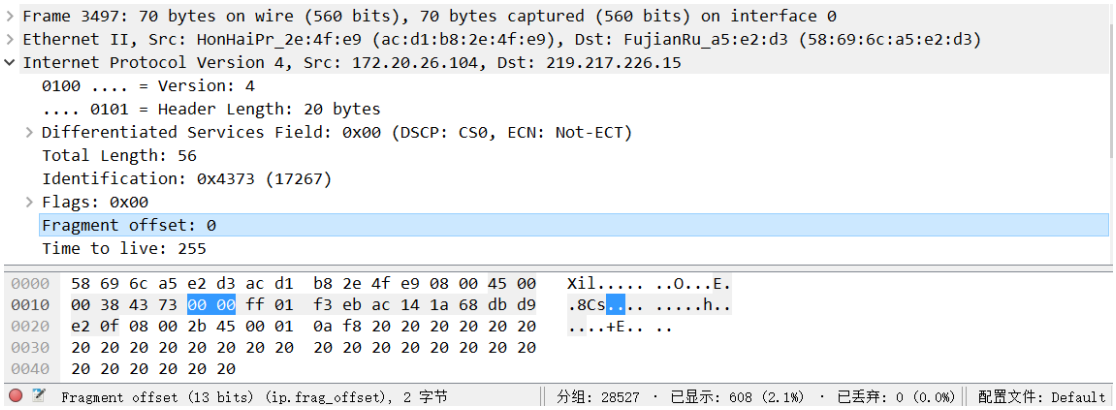
上层协议字段的值是 01。

3. **IP 头有多少字节？该 IP 数据包的净载为多少字节？并解释你是怎样确定该 IP 数据包的净载大小的？**



20 字节。净载 36 字节。可查看数据报长度字段，为 56，减去首部字段即为净载。

4. **该 IP 数据包分片了吗？解释你是如何确定该 P 数据包是否进行了分片**



没有。查看标志位。 Fragment offset: 0

B. **单击 Source 列按钮，这样将对捕获的数据包按源 IP 地址排序。选择第一个你的主机发出的 ICMP Echo Request 消息，在 packet details 窗口展开数据包的 Internet Protocol 部分。在“listing of captured packets”窗口，你会看到许多后续的 ICMP 消息（或许还有你主机上运行的其他协议的数据包）**

1. 你主机发出的一系列 ICMP 消息中 IP 数据报中哪些字段总是发生改变？

答：identification,ttl,header checksum,

2. 哪些字段必须保持常量？哪些字段必须改变？为什么？

答：保持常量：Version,header length，upper layer protocol 源 IP、目的 IP 等

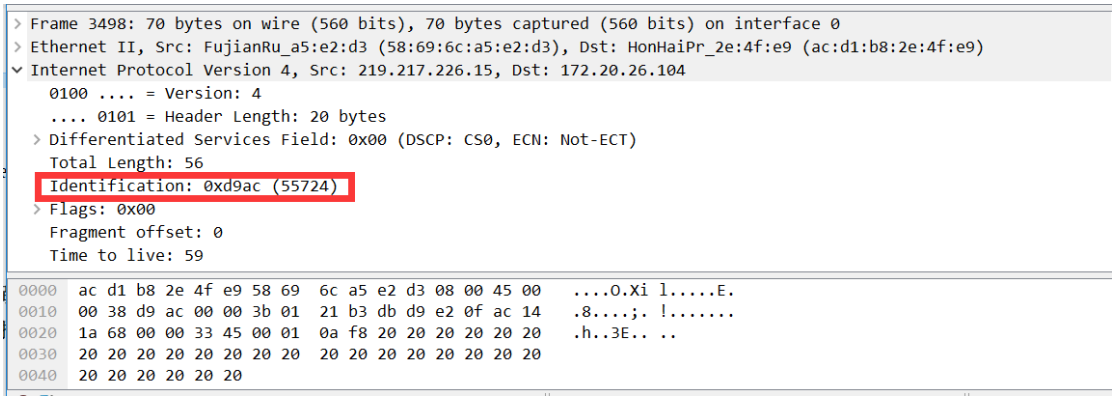
必须改变：identification,ttl,header checksum

ID 必须改变：鉴别码，用于区分不同的数据包；

TTL 必须改变：来自于 traceroute 的要求，用来测试路径上的路由信息；

Header Checksum 必须改变：首部校验和，前面的字段改变，该值也跟着改变；

3. 描述你看到的 IP 数据包 Identification 字段值的形式



答：16 位。在某范围内递增。

4. 找到由最近的路由器（第一跳）返回给你主机的 ICMPTime-to-live exceeded 消息。

思考下列问题：

Identification 字段和 TTL 字段的值是什么？ 最近的路由器（第一跳）返回给你主机的 ICMP Time-to-live exceeded 消息中这些值是否保持不变？为什么？

Identification：0x7769，TTL：254，

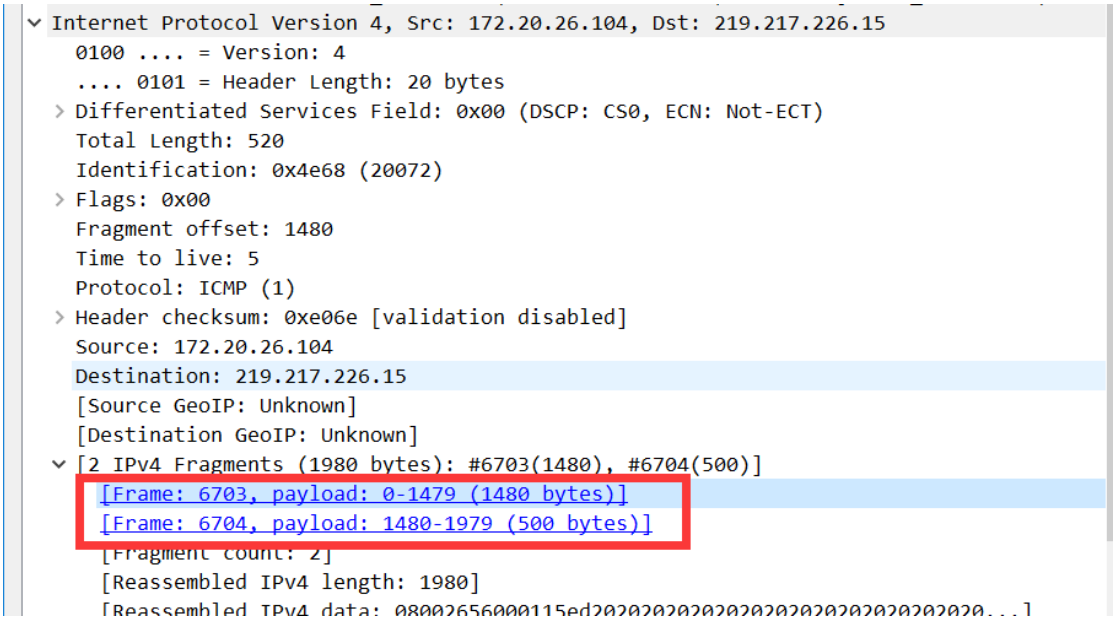
Identification 会变，TTL 保持不变。因为 Identification 标识一个报文，不同报文 Identification 不同。因为是第一跳路由器发回的数据报，故 TTL 是最大值减 1，总是等于 254。

5. 单击 Time 列按钮，这样将对捕获的数据包按时间排序。找到在将包大小改为 2000 字节后你的主机发送的第一个 ICMP Echo Request 消息。

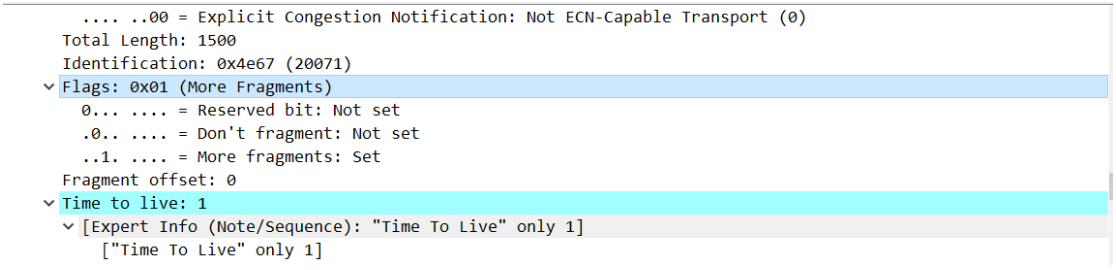
思考下列问题：

该消息是否被分解成不止一个 IP 数据报？

观察第一个 IP 分片， IP 头部的哪些信息表明数据包被进行了分片？ IP 头部的哪些信息表明数据包是第一个而不是最后一个分片？该分片的长度是多少



分成了两个。



More fragments=1 表示分片了且不是最后一片，该分片的长度是 1500B

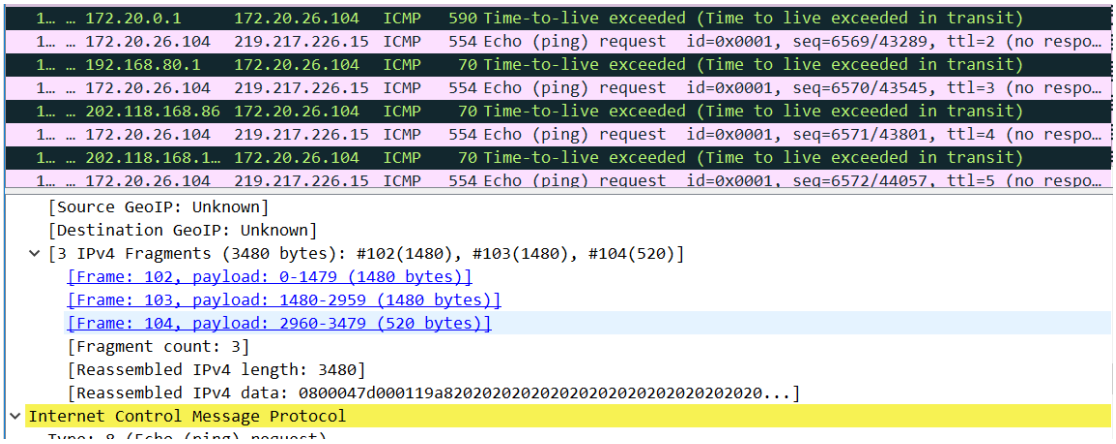
C. 找到在将包大小改为 3500 字节后你的主机发送的第一个 ICMP

Echo Request 消息。

思考下列问题：

原始数据包被分成了多少片？

这些分片中 IP 数据报头部哪些字段发生了变化？



分成三片

前 2 个分片More fragments=1，后两个分片offset 变为 1480 和 2960

4.抓取 ARP 数据包

1. 利用 MS-DOS 命令 arp 或 c:\windows\system32\arp 查看主机上 ARP 缓存的内容。说明 ARP 缓存中每一列的含义是什么?

输入 apr – a 查看主机上 ARP 缓存的内容，结果如下图所示（截图显示部分）：

管理员: 命令提示符

```
C:\Windows\system32>arp -a

接口: 192.168.244.1 --- 0x2
  Internet 地址         物理地址              类型
  192.168.244.254       00-50-56-ec-c8-d3     动态
  192.168.244.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.2             01-00-5e-00-00-02     静态
  224.0.0.5             01-00-5e-00-00-05     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.100           01-00-5e-00-00-64     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  224.0.0.253           01-00-5e-00-00-fd     静态
  224.67.40.111         01-00-5e-43-28-6f     静态
  224.117.200.182       01-00-5e-75-c8-b6     静态
  224.168.100.17        01-00-5e-28-64-11     静态
  224.193.235.115       01-00-5e-41-eb-73     静态
  225.4.76.124          01-00-5e-04-4c-7c     静态
  225.255.141.183       01-00-5e-7f-8d-b7     静态
  227.70.211.58         01-00-5e-46-d3-3a     静态
  229.255.255.250       01-00-5e-7f-ff-fa     静态
  230.4.109.47          01-00-5e-04-6d-2f     静态
```

ARP 缓存中的每一列分别表示 IP 地址所对应的物理地址和类型（动态配置或静态配置）

2. 清除主机上 ARP 缓存的内容，抓取 ping 命令时的数据包。分析数据包，回答下面的问题：

① ARP 数据包的格式是怎样的？由几部分构成，各个部分所占的字节数是多少？

数据包格式：

以太网的 ARP 请求和应答的分组格式，如图 6-11 所示。

图 6-11 ARP 请求和应答的分组格式

由 9 部分构成，分别是硬件类型（2 字节），协议类型（2 字节），硬件地址长度（1 字节），协议地址长度（1 字节），OP（2 字节），发送端 MAC 地址（6 字节），发送端 IP

地址（4 字节），目的 MAC 地址（6 字节），目的 IP 地址（4 字节）。

截取一个 ARP 数据包：

```
Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: 3c:b6:b7:c6:5f:f1 (3c:b6:b7:c6:5f:f1)
  Sender IP address: 172.20.0.148
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 172.20.108.38
```

② 如何判断一个 ARP 数据是请求包还是应答包？

从 op 字段判断：1 是请求，2 是应答

③为什么 ARP 查询要在广播帧中传送，而 ARP 响应要在一个有着明确目的局域网地址的帧中传送？

因为 ARP 进行查询工作时，并不知道目的 MAC 地址，所以要通过广播的形式，向局域网中所有节点进行查询；但 ARP 响应时已经知道源 MAC 地址（就是发送广播帧的节点），所以要在一个有着明确目的局域网地址的帧中传送。

5. 抓取 UDP 数据包

1. **消息是基于 UDP 的还是 TCP 的？**

   UDP。

2. **你的主机 IP 地址是什么？目的主机 IP 地址是什么？**

   答：本机 IP:172.20.141.150,目的 IP:123.151.13.167

```
Internet Protocol Version 4, Src: 172.20.141.150, Dst: 123.151.13.167
User Datagram Protocol, Src Port: 4015, Dst Port: 8000
```

3. **你的主机发送 QQ 消息的端口号和 QQ 服务器的端口号分别是多少？**

   答：4015，,8000

4. **数据报的格式是什么样的？都包含哪些字段，分别占多少字节？**

   由 5 部分构成，分别是源端口号（4 字节），目的端口号（4 字节），长度（4 字节），校验和（4 字节）和应用层数据。

   抓取的一个 UDP 数据报如下所示：

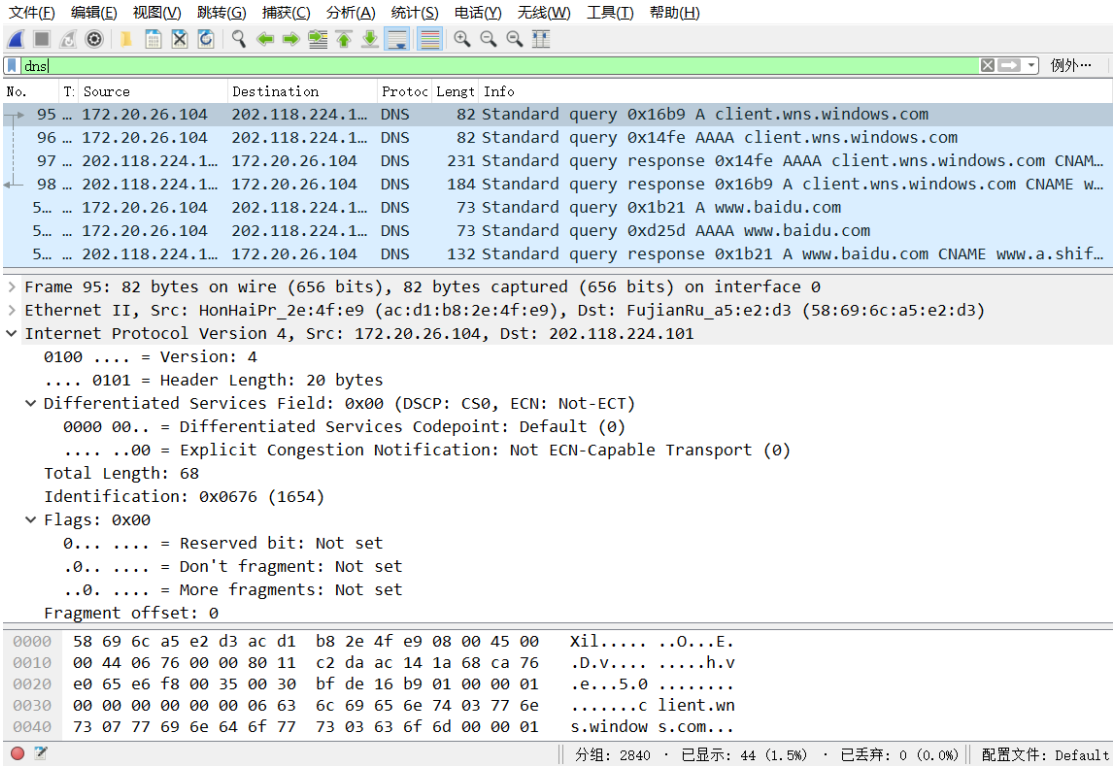
```
Source Port: 4015
Destination Port: 8000
Length: 159
Checksum: 0xa3e5 [unverified]
```

5. **为什么你发送一个 ICQ 数据包后，服务器又返回给你的主机一个 ICQ 数据包？这 UDP 的不可靠数据传输有什么联系？对比前面的 TCP 协议分析，你能看出 UDP 是无连接的吗？**

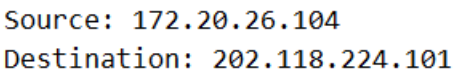
   返回的 ICQ 数据包时作确认用的，因为 UDP 是不可靠的数据传输协议，所以需要这种确认机制。能看出来，UDP 协议没有握手过程，UDP 数据包没有序列号，因此不能像 TCP 协议那样先握手再发送数据，而是直接发送数据。

6. 利用 WireShark 进行 DNS 协议分析

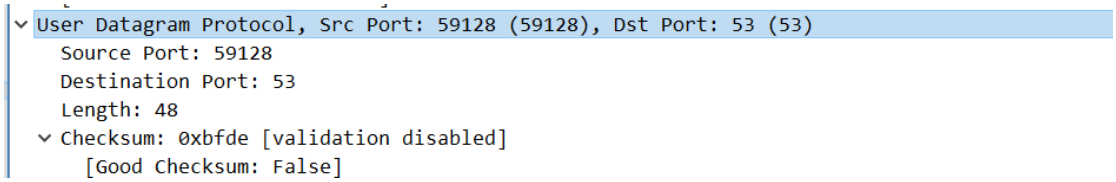
1. **打开浏览器，输入 www.baidu.com,DNS 查询消息如下图：**



2. **我的电脑 IP 地址：172.20.26.104，本地域名服务器 IP 地址：202.118.224.101.如图：**



3. **UDP 报文的源端口号 59128，目的端口号 53**



4. **DNS 报文格式**

DNS协议报文格式

会话标识（2 字节）：是 DNS 报文的 ID 标识，对于请求报文和其对应的应答报文，这个字段是相同的，通过它可以区分 DNS 应答报文是哪个请求的响应

标志（2 字节）：

QR（1bit） 查询/响应标志，0 为查询，1 为响应

opcode（4bit）0 表示标准查询，1 表示反向查询，2 表示服务器状态请求

AA（1bit） 表示授权回答

TC（1bit） 表示可截断的

RD（1bit） 表示期望递归

RA（1bit） 表示可用递归

rcode（4bit） 表示返回码，0 表示没有差错，3 表示名字差错，2 表示服务器错误（Server Failure）

数量字段（总共 8 字节）：

Questions、Answer RRs、Authority RRs、Additional RRs 各自表示后面的四个区域的数目。Questions 表示查询问题区域节的数量，Answers 表示回答区域的数量，Authoritative namesversers 表示授权区域的数量，Additional recoreds 表示附加区域的数量

```
> User Datagram Protocol, Src Port: 53 (53), Dst Port: 53912 (53912)
v Domain Name System (response)
    [Request In: 753]
    [Time: 0.002935000 seconds]
    Transaction ID: 0xadd9
  v Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v t12.baidu.com: type A, class IN
        Name: t12.baidu.com
```

5. DNS 响应报文

```
Domain Name System (response)
    [Request In: 756]
    [Time: 0.025279000 seconds]
    Transaction ID: 0x3629
  Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  Queries
    t2.baidu.com: type AAAA, class IN
        Name: t2.baidu.com
        [Name Length: 12]
        [Label Count: 3]
        Type: AAAA (IPv6 Address) (28)
        Class: IN (0x0001)
  Answers
    t2.baidu.com: type CNAME, class IN, cname simage.jomodns.com
        Name: t2.baidu.com
```

四、实验心得

经过此次实验，从应用层的 HTTP、DNS 到传输层的 TCP,UDP,在到网络层的 IP、ICMP 在到链路层的 ARP，利用 WireShark，对每一协议进行了深入的跟踪与分析，基本了解了每层的功能和不同协议的报文格式，实际的感受了每个报文设计的巧妙，每个字段的必要，以及从事计算机网络工作者所做的工作的重要性以及他们的智慧。

实验过程中由于第一次使用 Wireshark，很多功能都不太了解，抓到的数据包太多无法准确确定。除了通过包的类型的筛选以外，关闭其他的一些后台程序减少网络数据包的交互，能够更好容易的定位包。此时网络数据包相对干净，分析每个包，总能有特别发现，也是挺有趣儿的。