哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

实验名称	利用 Wireshark 进行协议分析				
姓名	张智雄		院系	计算学部	
班级	2103601		学号	2021112845	
任课教师	刘亚维		指导教师	刘亚维	
实验地点	格物 207		实验时间	2023.11.4	
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)				
教师评语					

哈尔滨工业大学计算学部
FACULTY OF COMPUTING, HIT

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：

必做内容：

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**

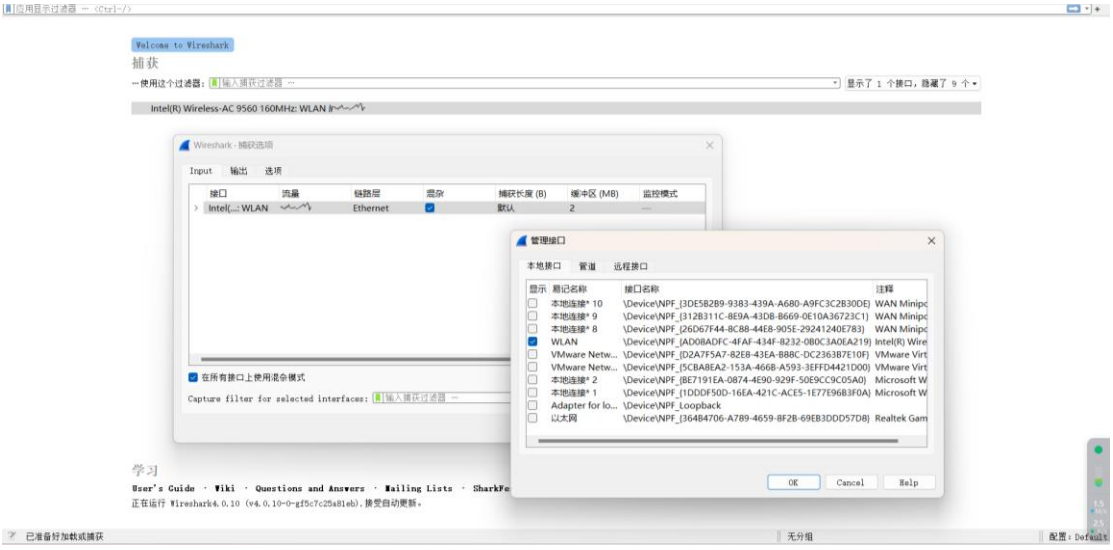
选做内容：

1. 利用 Wireshark 分析 **DNS 协议**
2. 利用 Wireshark 分析 **UDP 协议**
3. 利用 Wireshark 分析 **ARP 协议**

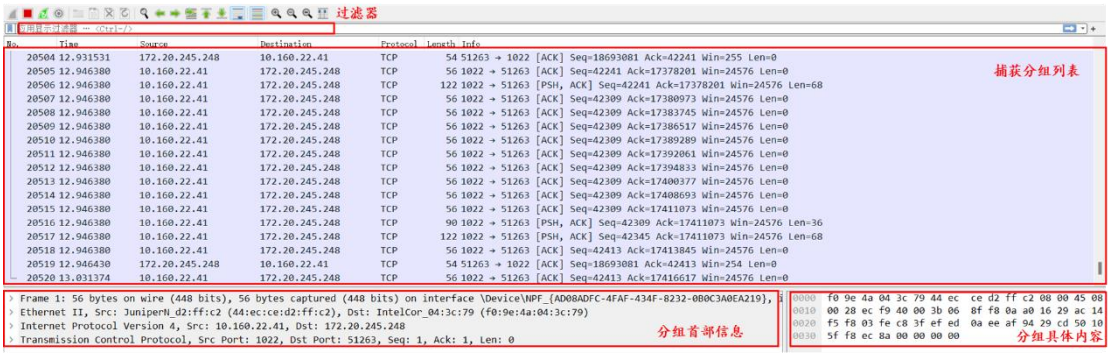
实验过程及结果：

一、Wireshark的使用

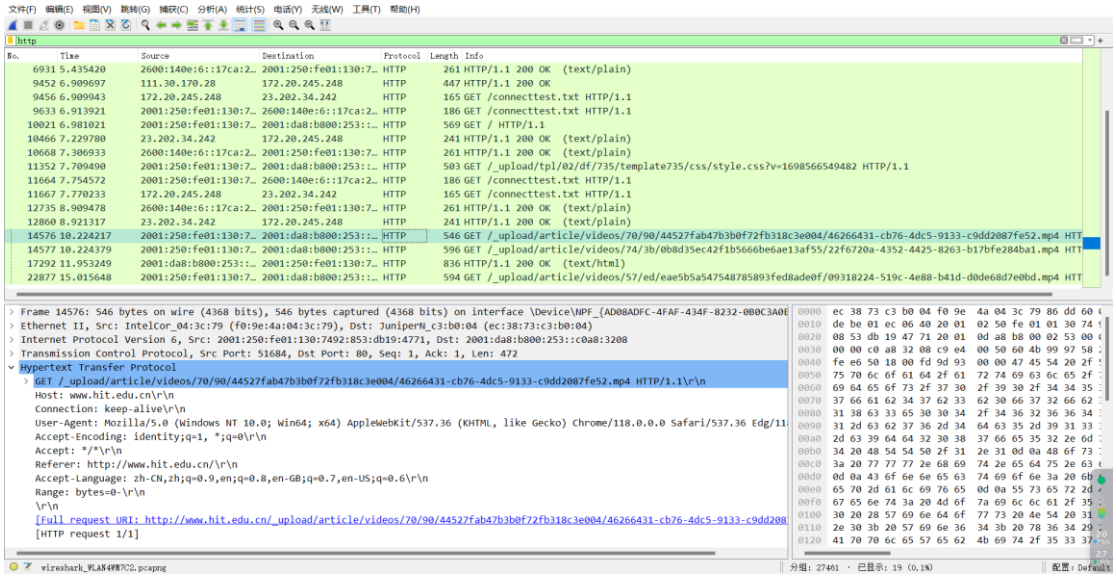
启动 Web 浏览器和 Wireshark，选择“capture”下拉菜单中的“Capture Options”命令，设置分组俘获的选项后，开始分组捕获。



开始分组捕获，出现分组捕获窗口：



浏览器打开http://www.hit.edu.cn，捕获并提取显示HTTP报文如下：



二、利用 Wireshark 分析 HTTP 协议

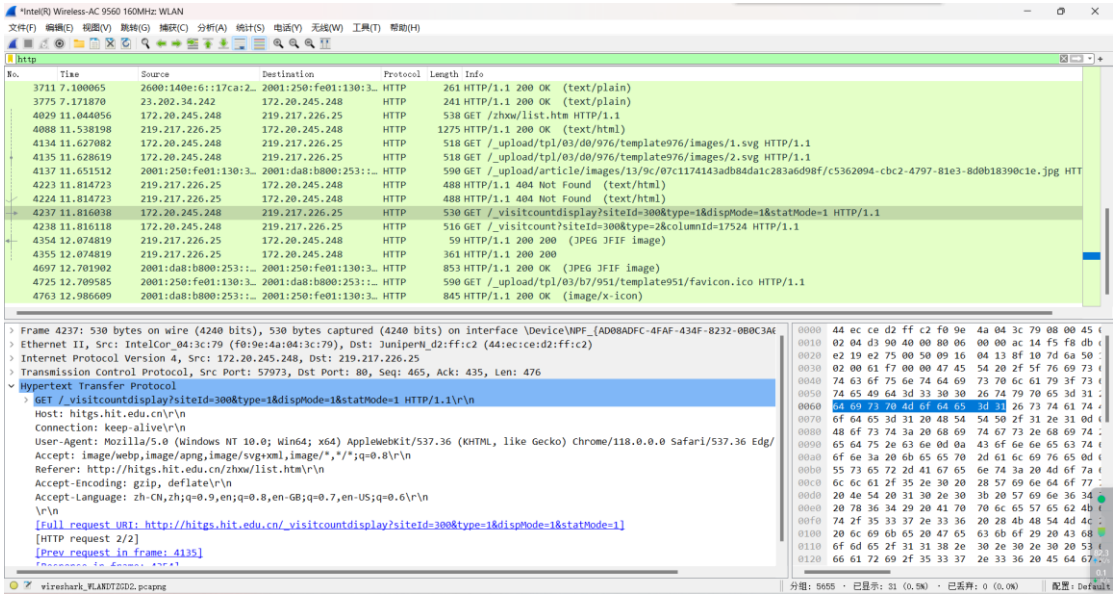
1. HTTP GET/response 交互

1) 启动 Web browser，然后启动 Wireshark 分组嗅探器；

2) 在窗口的显示过滤说明处输入“http”，开始 Wireshark 分组俘获；

3) Web browser 窗口中输入地址 http://hitgs.hit.edu.cn/zhxw/list.htm 并跳转；

4) 停止分组俘获，得到捕获结果如下：



问题思考：

a) 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？

```
4237 11.816038    172.20.245.248    219.217.226.25    HTTP    530 GET /_visitcountdisplay?siteId=300&type=1&dispMode=1&statMode=1 HTTP/1.1
4238 11.816118    172.20.245.248    219.217.226.25    HTTP    516 GET /_visitcount?siteId=300&type=2&columnId=17524 HTTP/1.1
4354 12.074819    219.217.226.25    172.20.245.248    HTTP     59 HTTP/1.1 200 200  (JPEG JFIF image)
4355 12.074819    219.217.226.25    172.20.245.248    HTTP    361 HTTP/1.1 200 200
```

答：由GET请求报文知，浏览器运行HTTP1.1；而由返回报文知，服务器同样运行HTTP1.1

b) 你的浏览器向服务器指出它能接收何种语言版本的对象？

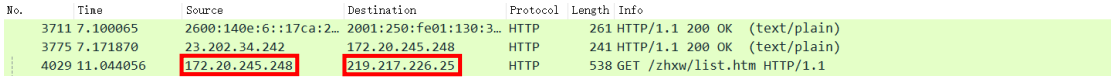
```
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
```

答：由‘Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6’可知，服务器支持 zh-CN (中文简体，中国)、zh (中文)、en (英文)、en-GB (英文，英国)、en-US (英文，美国)（按照优先级从高到低）

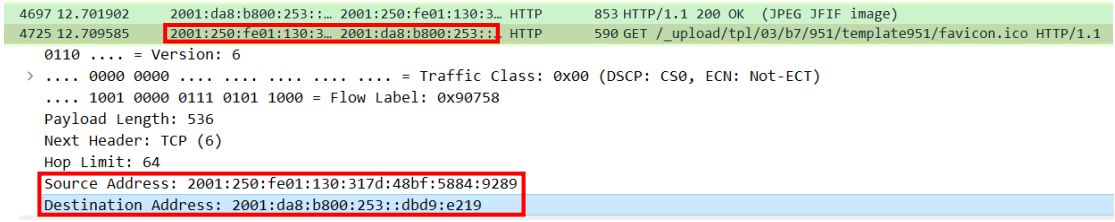
c) 你的计算机的 IP 地址是多少？服务器 http://hitgs.hit.edu.cn/zhxw/list.htm 的 IP 地址是多少？

答：根据 GET 命令的 Source 和 Destination 字段可知，

- IPv4 下，本机 IP 地址为 172.20.245.248；服务器地址为 219.217.226.25；

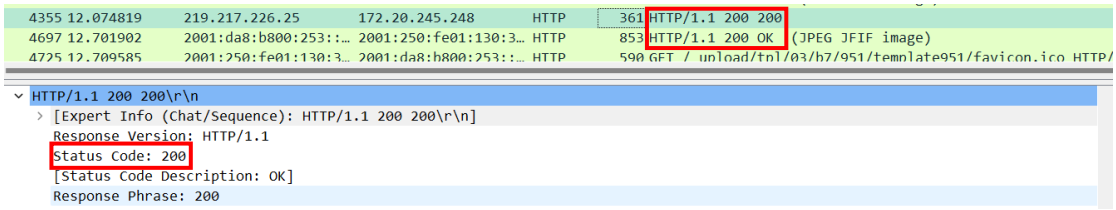


- IPv6 下，本机 IP 地址为 2001:250:fe01:130:317d:48bf:5884:9289；服务器地址为 2001:da8:b800:253::dbd9:e219

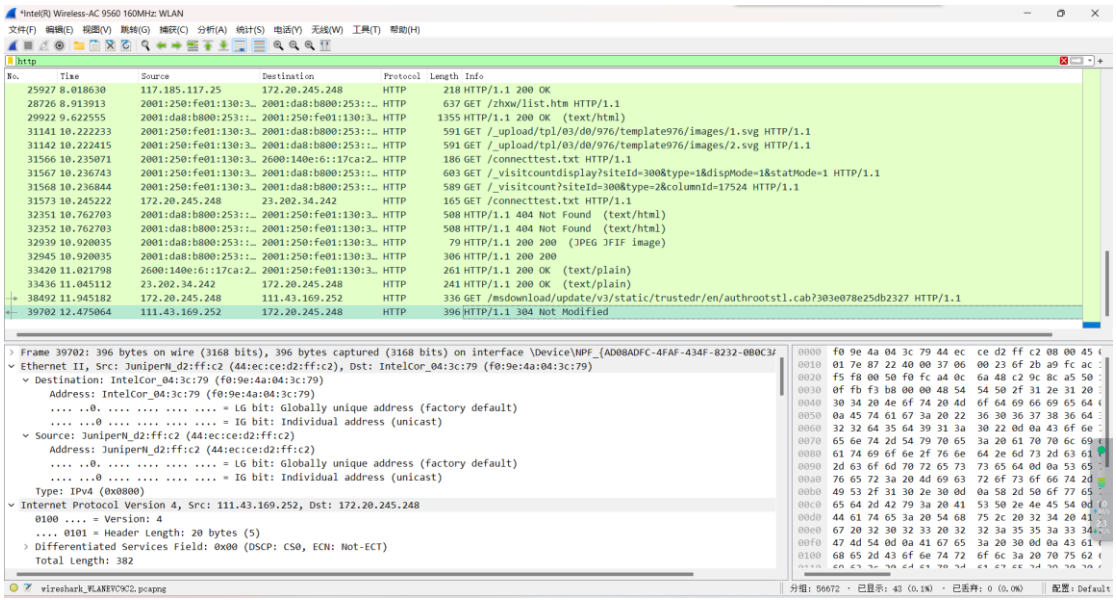


d) 从服务器向你的浏览器返回的状态代码是多少？

答：一般情况下均为 200，存在 404 等情况。



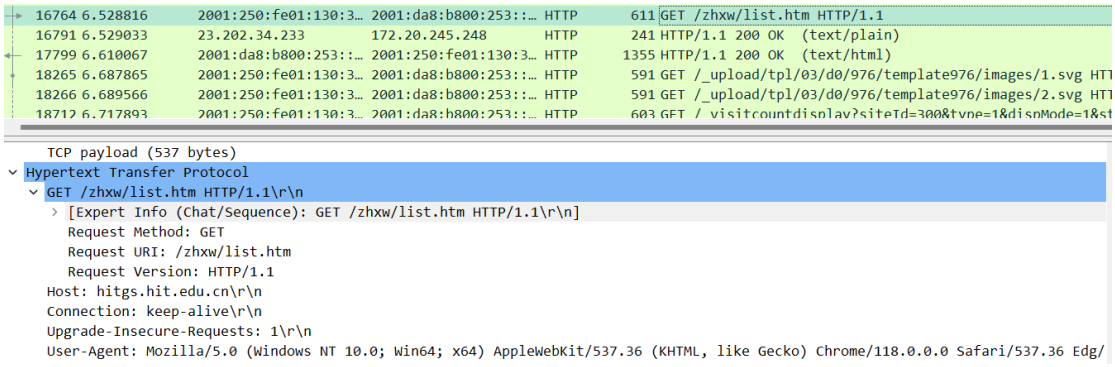
2. HTTP 条件 GET/response 交互
   1) 启动浏览器，清空浏览器的缓存。
   2) 启动 Wireshark 分组俘获器，开始 Wireshark 分组俘获。
   3) 在浏览器的地址栏中输入 URL：http://hitgs.hit.edu.cn/zhxw/list.htm 跳转并刷新。
   4) 停止 Wireshark 分组俘获，结果如下。



问题思考：

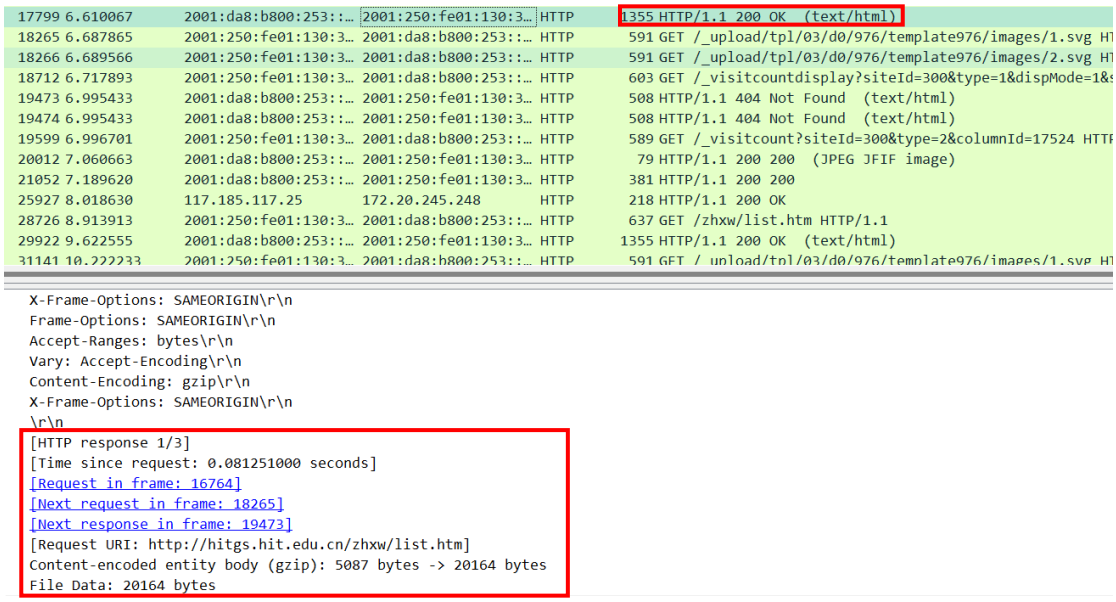
a) 分析你的浏览器向服务器发出的第一个 HTTP GET 请求的内容，在该请求报文中，是否有一行是：IF-MODIFIED-SINCE？

答：观察得，在首次向服务器发送 GET 请求的内容中，没有 IF-MODIFIED-SINCE 行。



b) 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？

答：服务器明确返回了文件内容，可以从返回报文中直接看出，也可从服务器返回的状态码 200 获知。

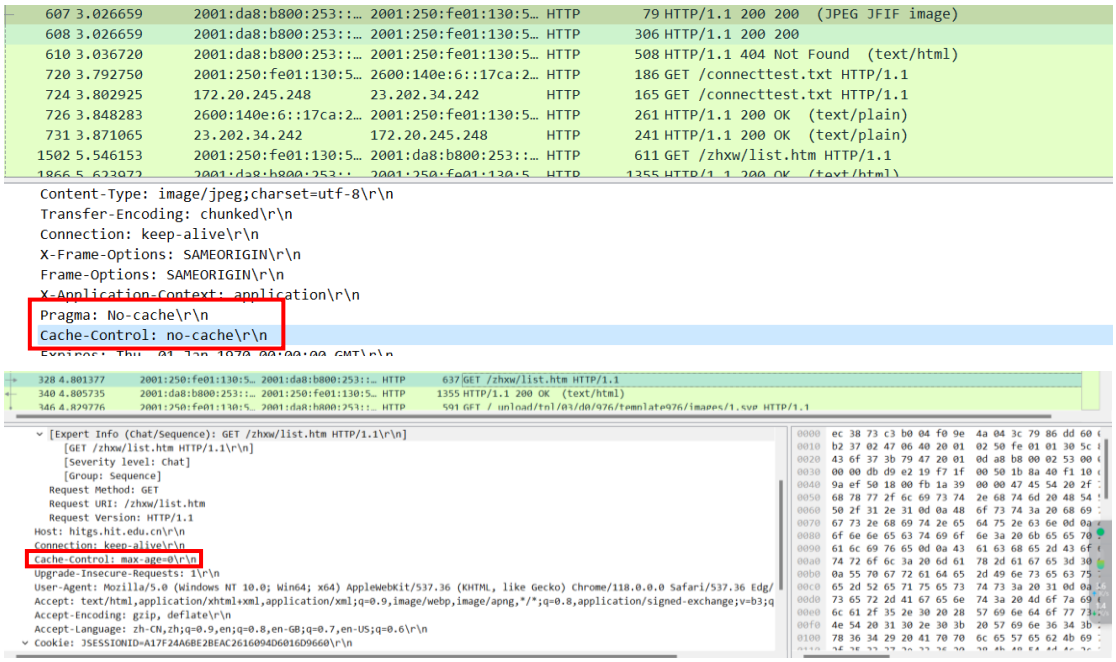


c) 分析你的浏览器向服务器发出的较晚的“HTTP GET”请求，在该请求报文中是否有一行是：IF-MODIFIED-SINCE？如果有，在该首部行后面跟着的信息是什么？

答：第二次发送的 GET 请求并不包含 IF-MODIFIED-SINCE 首部，但相较于第一次报文增加了“Cache-Control: max-age=0”字段（这通常是强制刷新导致的），指定了缓存的最大寿命（时间）。在这里，"max-age" 设置为 0，表示该响应的内容应立即过期，不应该被缓存。表示客户端不希望使用缓存数据，而是要求服务器始终提供最新的内容。

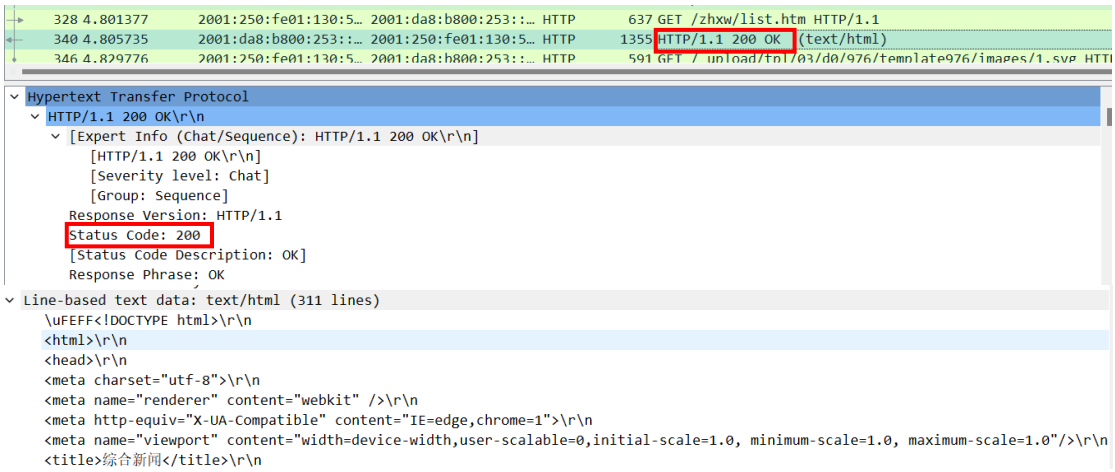
根本原因是，服务器在返回报文时返回了“Pragma: No-cache”和“Cache-Control: no-cache”字段，表示不应使用缓存来处理请求的响应，即禁用缓存。

但正常的流程是再次访问时加入 IF-MODIFIED-SINCE 首部，表示浏览器内容最后更新的时间（eg. Thu, 25 Nov 2022 09:48:50 GMT），从而向服务器发送条件 GET，询问是否可以直接使用本地缓存。



d) 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。

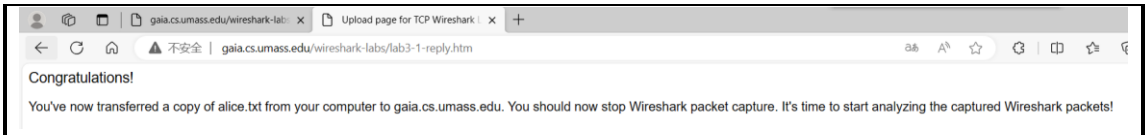
答：由于上述浏览器在第二次发送的报文中包含“Cache-Control: max-age=0”字段，因而返回的 HTTP 状态代码为 200，并返回了文件的内容。



事实上，对于条件 GET，如果服务器返回的状态码为 200，则表明内容需要更新，并明确返回了文件的内容；如果服务器返回的状态码为 304，则表示网页的内容没有更新，可以直接使用本地缓存的内容。

三、利用 Wireshark 分析 TCP 协议

1. 俘获大量的由本地主机到远程服务器的 TCP 分组
   1) 启动浏览器，打开 http://gaia.cs.umass.edu/wireshark-labs/alice.txt 网页，保存 ALICE'S ADVENTURES IN WONDERLAND 文本到本地。
   2) 打开 https://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html 网页，输入保存的本地文件。
   3) 启动 Wireshark，开始分组俘获。
   4) 单击“Upload alice.txt file”按钮，将文件上传到 gaia.cs.umass.edu 服务器。
   5) 停止俘获。



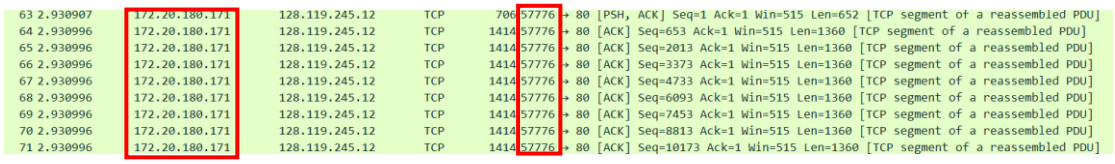
2. 浏览追踪信息

在显示筛选规则中输入“tcp”,可以看到在本地主机和服务器之间传输的一系列 tcp 和 http 报文，你应该能看到包含 SYN 报文的三次握手。也可以看到有主机向服务器发送的一个 HTTP POST 报文和一系列的“http continuation”报文。

问题思考：

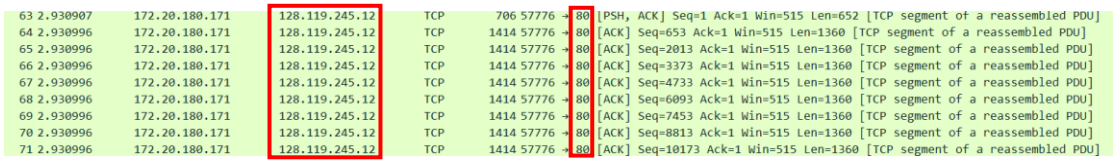
a) 向 gaia.cs.umass.edu 服务器传送文件的客户端主机的 IP 地址和 TCP 端口号是多少？

答：客户端 IP 地址为 172.20.180.171（电脑重启了 IP 地址有变化），TCP 端口号为 57776。



b) 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？

答：服务器的 IP 地址为 128.119.245.12，TCP 端口号为 80。

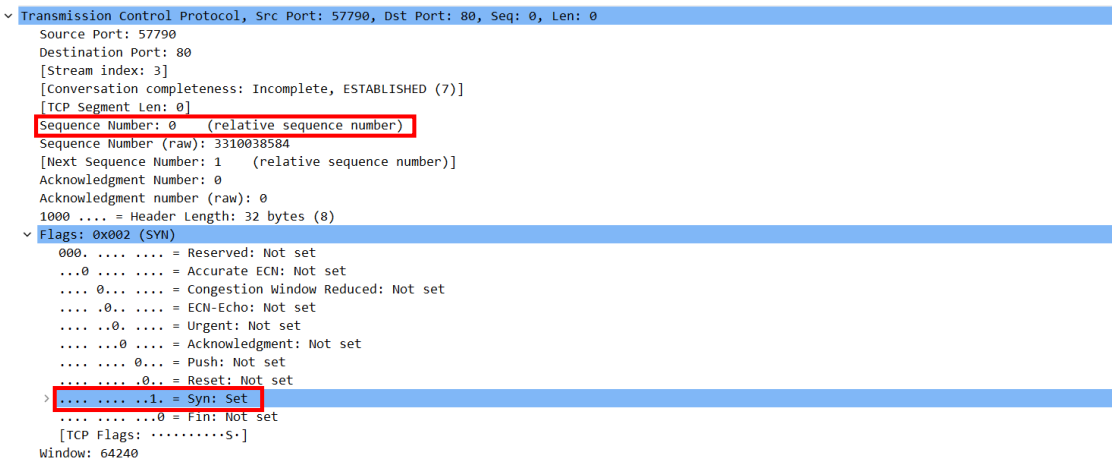


3. TCP 基础

问题思考：

a) 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？

- 客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段序号是 0；
- 报文段中将 Flags 中的 SYN 标志位置 1 以标识该报文段为 SYN 报文段。



b) 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？

- 服务器向客户端发送的 SYN ACK 报文段序号为 0；
- 报文段中，Acknowledgement 字段的值为 1；
- Gaia.cs.umass.edu 服务器通过将客户端发送过来的报文段seq + 1决定此 ACK 的值；
- 通过将 Flags 中 ACK 和 SYN 标志位同时置 1 来标示报文段是 SYN ACK 报文段。

```
Transmission Control Protocol, Src Port: 80, Dst Port: 57790, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 57790
    [Stream index: 3]
    [Conversation completeness: Incomplete, ESTABLISHED (7)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3861593938
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3310038585
    1000 .... = Header Length: 32 bytes (8)
  Flags: 0x012 (SYN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    > .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····A··S·]
    Window: 29200
```

c) 你能从捕获的数据包中分析出 TCP 三次握手过程吗？

```
36 2.165978    172.20.180.171    128.119.245.12    TCP    66 57790 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
37 2.482559    128.119.245.12    172.20.180.171    TCP    66 80 → 57790 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1360 SACK_PERM WS=128
38 2.482618    172.20.180.171    128.119.245.12    TCP    54 57790 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
```

- **第一次握手：**客户端会向服务器发送一个 SYN 报文，初始序列号 Seq 为 0，不携带其他任何数据，进入 SYN_SEND 状态，等待服务器确认连接；
- **第二次握手：**服务器收到 SYN 报文段，确认客户的 SYN（Ack = seq + 1），同时发送 SYN 报文，即 SYN ACK 报文，服务器进入 SYN_RECV 状态
- **第三次握手：**双方建立起连接，客户端确认收到服务器的 SYN ACK 报文，回复 ACK 报文段（SYN 置 0），同时可以发送数据。

d) 包含 HTTP POST 命令的 TCP 报文段的序号是多少？

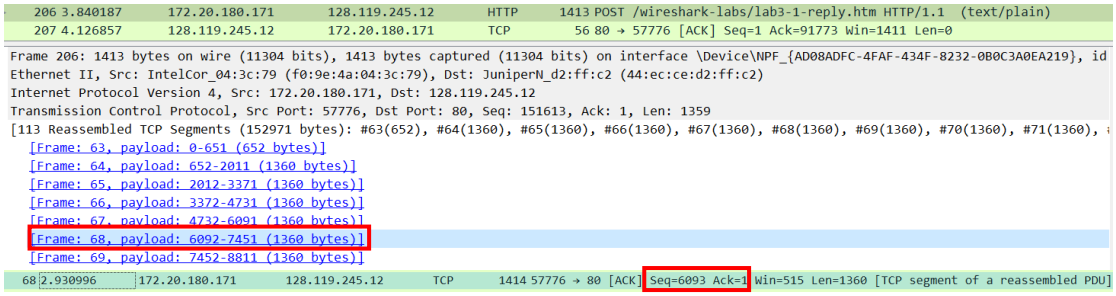
答：筛选 http 报文，发现包含 HTTP POST 命令的 TCP 报文段序号为 151613。

```
206 3.840187    172.20.180.171    128.119.245.12    HTTP    1413 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
223 4.158952    128.119.245.12    172.20.180.171    HTTP     831 HTTP/1.1 200 OK  (text/html)
```

```
> Frame 206: 1413 bytes on wire (11304 bits), 1413 bytes captured (11304 bits) on interface \Device\NPF_{AD08ADFC-4FAF-434F-8232-0B0C3A0EA219},
> Ethernet II, Src: IntelCor_04:3c:79 (f0:9e:4a:04:3c:79), Dst: JuniperN_d2:ff:c2 (44:ec:ce:d2:ff:c2)
> Internet Protocol Version 4, Src: 172.20.180.171, Dst: 128.119.245.12
v Transmission Control Protocol, Src Port: 57776, Dst Port: 80, Seq: 151613, Ack: 1, Len: 1359
    Source Port: 57776
    Destination Port: 80
    [Stream index: 7]
    [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 1359]
    Sequence Number: 151613    (relative sequence number)
    Sequence Number (raw): 506912697
    [Next Sequence Number: 152972    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 69223681
```

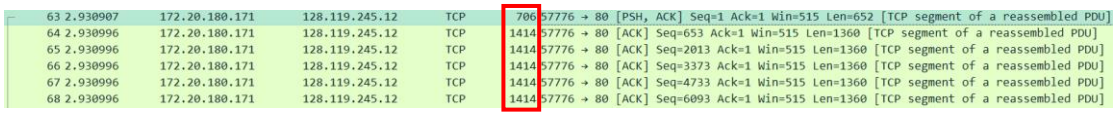
e) 如果将包含 HTTP POST 命令的 TCP 报文段看作是 TCP 连接上的第一个报文段，那么该 TCP 连接上的第六个报文段的序号是多少？是何时发送的？该报文段所对应的 ACK 是何时接收的？

- 第六个报文段的序号是 6093；
- 在 HTTP POST 命令之前，TCP 连接建立之后发送的；
- 该报文段对应的 ACK 是在该报文段发送之后，HTTP POST 命令之后接收的

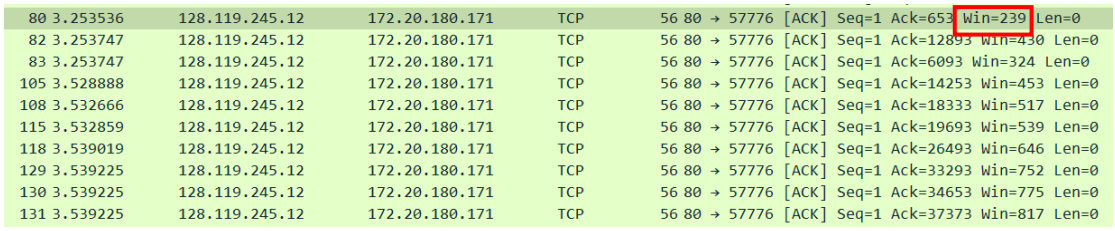


f) 前六个 TCP 报文段的长度各是多少？

答：如上图，除第一个 TCP 报文段为 706 Bytes，其余都为 1414 Bytes。实际包含数据内容为 652 Bytes 和 1360 Bytes（会有 54 Bytes 的部等信息）。



g) 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？



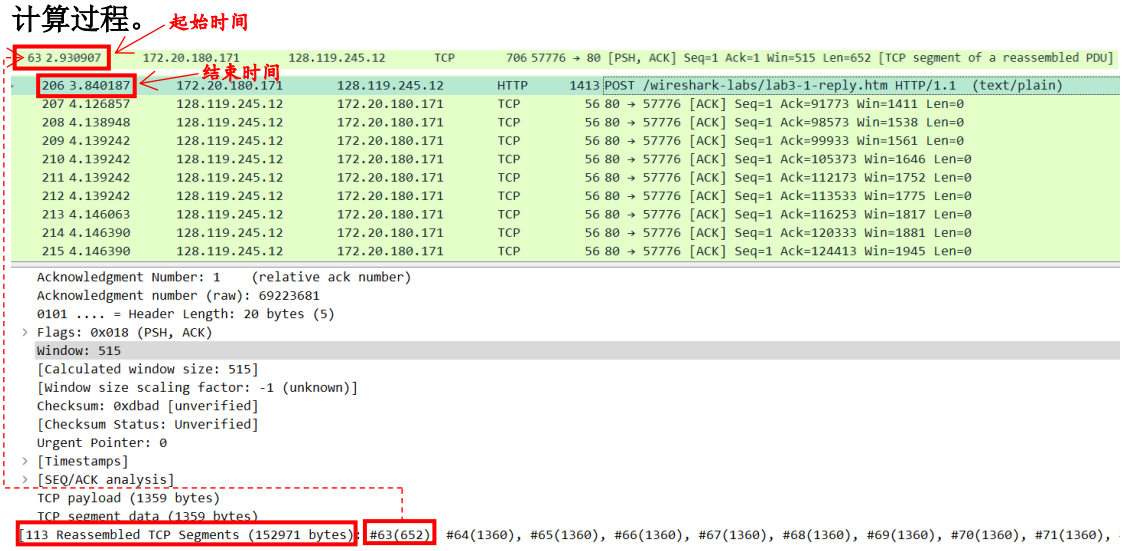
答：最小的可用缓存空间是 239 字节。本次实验数据量较小，并未出现接收端缓存不够用的情况，过程中缓存空间还呈现逐渐增大的趋势。

但是限制发送端传输后，接收端的缓存仍然可能出现不够用的现象，原因可能是数据处理速度较慢、接收端缓冲区过小等因素。

h) 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？

答：没有，因为没有出现重复的序列号。

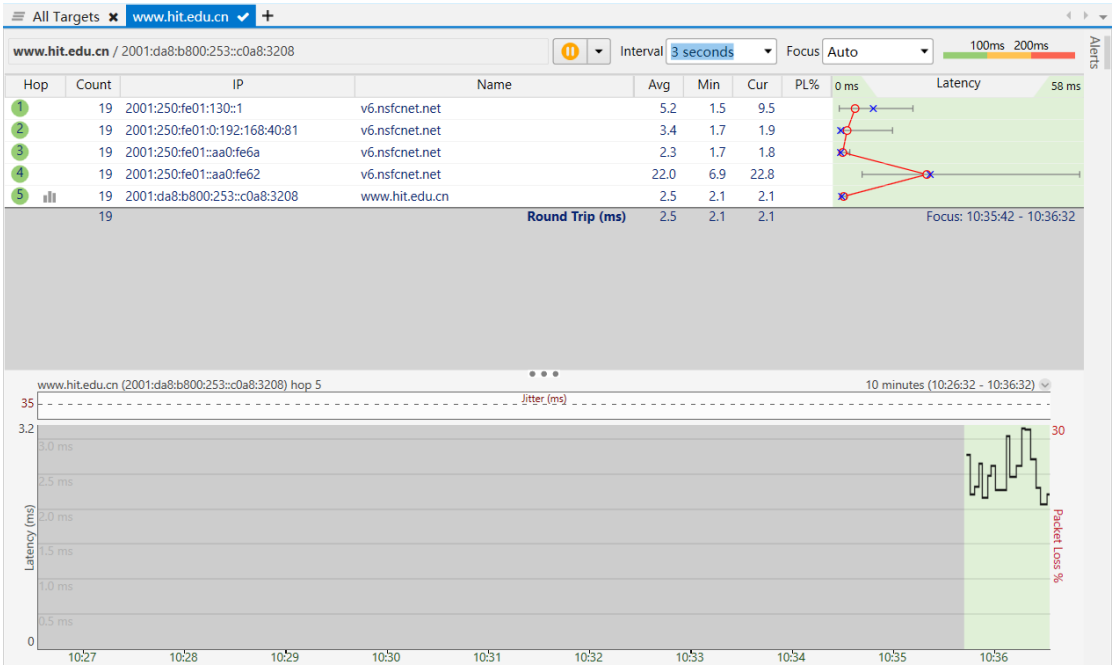
i) TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。



答：数据总长度为$152971 + 113 \times 54 = 159073$字节（算上首部行等字段），发送第一个段的时间节点为2.930907，发送最后一个段的时间节点为3.840187，发送时间间隔为$3.840187 - 2.930907 = 0.90928s$，因此吞吐率为$159073/0.90928s \approx 1.75 \times 10^6 Bps$。

四、利用 Wireshark 分析 IP 协议

1. 通过执行 traceroute 执行捕获数据包
   1) 启动 Wireshark 并开始数据包捕获；
   2) 启动 pingplotter 并“Address to Trace Window”域中输入目的地址。



2. 对捕获的数据包进行分析
   1) 第一个主机发出的 ICMP Echo Request 消息，在 packet details 窗口展开数据包的 Internet Protocol 部分。

问题思考：

a) 你主机的 IP 地址是什么？

答：172.20.184.210

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

答：在 IP 数据包头中，上层协议字段的值是“Protocol”字段，它的值是 1。这表示上层协议是 ICMP（Internet Control Message Protocol）。

c) 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？

答：IP 头部大小为 20 字节（由 IP 数据包的 Header Length 字段）。该 IP 数据包的净载大小为 36 字节，通过 Total Length 字段的值为 56 字节，减去首部字段 20 字节确定。

d) 该 IP 数据包分片了吗？解释你是如何确定该 IP 数据包是否进行了分片

答：没有分片。原因是 Flags 和 Fragment Offset 字段均为 0，表明该 IP 数据包不存在偏移，未经过分片。

2) 单击 Source 按钮对捕获的数据包按源 IP 地址排序。选择第一个主机发出的 ICMP Echo Request 消息， 在 packet details 窗口展开数据包的 Internet Protocol 部分。

问题思考：

a) 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？

- TTL 字段，指示数据包可以经过的最大跳数；
- Seq 字段，标识和追踪不同 ICMP 消息之间的关系，通常以顺序号的方式递增。
- Checksum 字段，由数据内容决定，因此会发送变化。

b) 哪些字段必须保持常量？哪些字段必须改变？为什么？

答：必须保持常量：

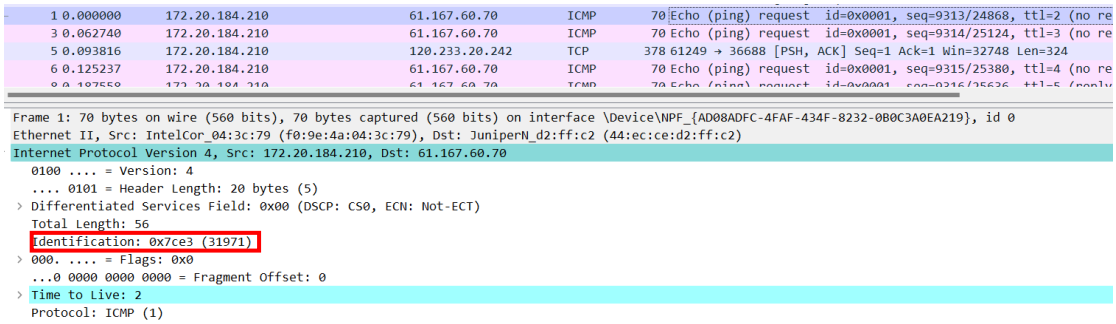
- 源端口和目标端口，这些字段在整个 TCP 连接的生命周期内保持不变。它们用于标识通信的源和目标端点；
- ICMP 消息类型和代码（ICMP Type 和 Code）；
- Identification（标识）字段，用于标识和关联 Ping 请求和 Ping 响应，在 Ping 响应中保持不变，以确保正确地关联响应与请求。

必须改变：

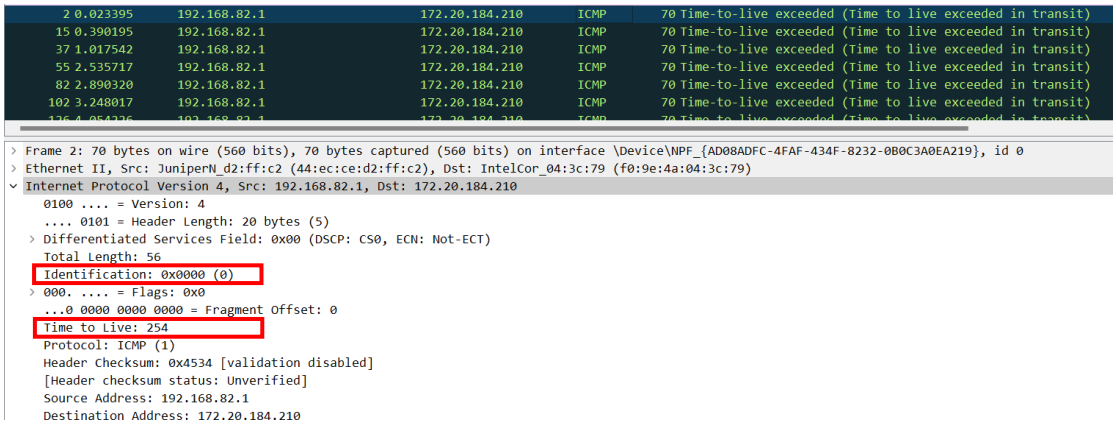
- 序列号字段 Seq，在不同的 Ping 消息中通常会递增，以标识不同的 Ping 请求；
- 校验和 Checksum 字段，由数据内容决定，因此会发生变化。

c) 描述你看到的 IP 数据包 Identification 字段值的形式。

答：16 位二进制数，用十六进制表示



3) 找到由最近的路由器（第一跳）返回主机的 ICMP Time-to-live exceeded 消息。



问题思考：

a) Identification 字段和 TTL 字段的值是什么？

答：Identifier 字段值为 0x0000；TTL 字段值为$255-1=254$

b) 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？

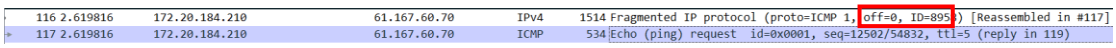
答： Identifier 字段和 TTL 字段均保持不变。相同的 Identifier 标识是为了分段后组装为一段数据，并不代表信号。每经过一个路由器（一跳），TTL 都会减小 1，因此在初始 TTL 相同的情况下，都会返回相同的TTL = 254。

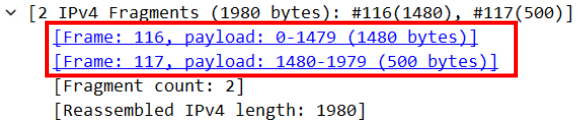
4) 单击 Time 列按钮对捕获的数据包按时间排序。找到在将包大小改为 2000 字节后主机发送的第一个 ICMP Echo Request 消息。

问题思考：

a) 该消息是否被分解成不止一个 IP 数据报？

答： 是的，该消息被分解成两片。





b) 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？

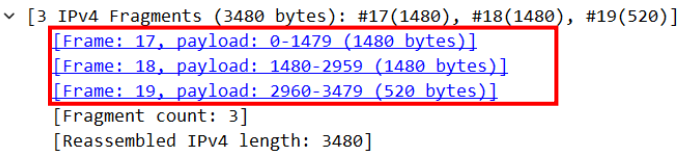
- ✧ Flags 字段的值为 0x1，其中最后一位为 1，表示"More fragments"标志被设置，表明这是一个分片。
- ✧ Fragment Offset 字段的值为 0，表示这是分片的第一个部分，因为第一个分片的偏移量通常为 0。
- ✧ Total Length 字段的值为 1500 字节，表明该分片的总长度为 1500 字节，包含 1480 字节的数据和 20 字节的首部。

3. 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。

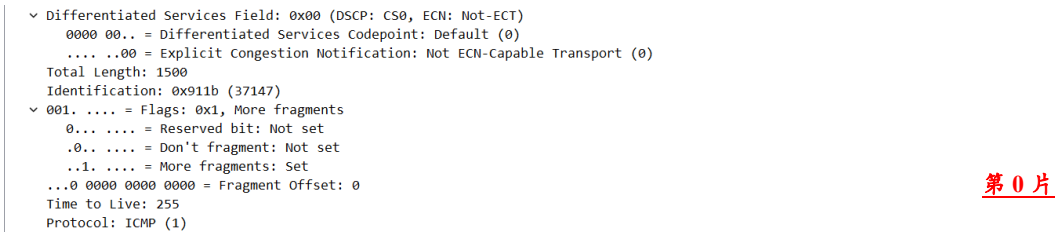
问题思考：

a) 原始数据包被分成了多少片？

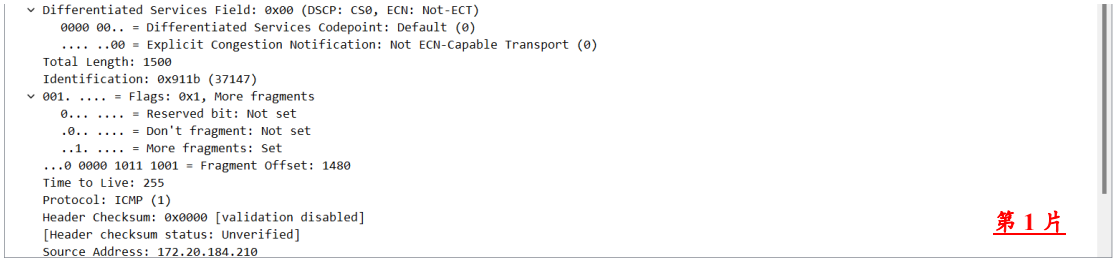
答： 原始数据包被分成了 3 片



b) 这些分片中 IP 数据报头部哪些字段发生了变化？



第 0 片



第 1 片

```
▾ 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: 540
  Identification: 0x911b (37147)
▾ 000. .... = Flags: 0x0
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  ...0 0001 0111 0010 = Fragment Offset: 2960
  Time to Live: 255
  Protocol: ICMP (1)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 172.20.184.210
```

第 2 片

✧ More Fragment 段的值：前两片为 1，最后一片 0；

✧ Fragment Offset 字段的值不同：分别为 0，1480，2960；

✧ Total Length 字段大小不同：前两片为 1500，最后一片为 540 字节；

五、利用 Wireshark 分析 ARP 协议

1) 利用 MS-DOS 命令：arp 或 c:\windows\system32\arp 查看主机上 ARP 缓存的内容

```
C:\Users\mr>arp -a

接口: 192.168.138.1 --- 0xd
  Internet 地址         物理地址              类型
  192.168.138.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     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
  239.255.255.250       01-00-5e-7f-ff-fa     静态

接口: 172.20.247.120 --- 0x10
  Internet 地址         物理地址              类型
  172.20.0.1            44-ec-ce-d2-ff-c2     动态
  172.20.223.9          44-ec-ce-d2-ff-c2     动态
  172.20.247.167        44-ec-ce-d2-ff-c2     动态
  172.20.255.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     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
  239.255.255.250       01-00-5e-7f-ff-fa     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态

接口: 192.168.234.1 --- 0x12
  Internet 地址         物理地址              类型
  192.168.234.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
```

问题思考：

a) 说明 ARP 缓存中每一列的含义是什么？

答：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) 在命令行模式下输入：ping 192.168.1.82（或其他 IP 地址）

```
C:\Users\mr>ping 172.20.56.36

正在 Ping 172.20.56.36 具有 32 字节的数据:
来自 172.20.56.36 的回复: 字节=32 时间=49ms TTL=63
来自 172.20.56.36 的回复: 字节=32 时间=73ms TTL=63
来自 172.20.56.36 的回复: 字节=32 时间=95ms TTL=63
来自 172.20.56.36 的回复: 字节=32 时间=103ms TTL=63

172.20.56.36 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 49ms, 最长 = 103ms, 平均 = 80ms
```

3) 启动 Wireshark，开始分组俘获。

No.	Time	Source	Destination	Protocol	Length	Info
38	1.514942	IntelCor_04:3c:79	Broadcast	ARP	42	Who has 172.20.56.36? Tell 172.20.247.120
39	1.516536	JuniperN_d2:ff:c2	IntelCor_04:3c:79	ARP	60	172.20.56.36 is at 44:ec:ce:d2:ff:c2

问题思考：

a) 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？

答：ARP 数据包格式如图所示，共 28 字节，具体由 9 部分构成：

部分名称	字节数	部分名称	字节数
硬件类型	2	发送端 MAC 地址	6
协议类型	2	发送端 IP 地址	4
硬件地址长度	1	目的 MAC 地址	6
协议地址长度	1	目的 IP 地址	4
OP	2		

b) 如何判断一个 ARP 数据是请求包还是应答包？

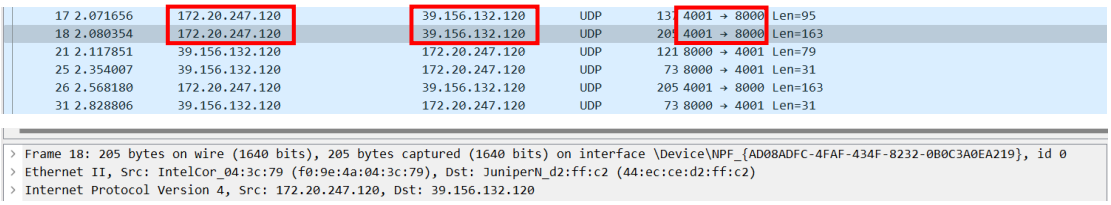
答：判断一个 ARP 分组是 ARP 请求还是应答的字段是“OP”，当其值为 0×0001 时是请求包，为 0×0002 时是应答包。

c) 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？

答：ARP 查询需要广播，因为它需要询问整个局域网中的设备来找到目标设备的 MAC 地址，而 ARP 响应是为了回应特定的 ARP 查询，局域网中的其他主机不需要此次查询的结果，所以它需要在一个有着明确目的局域网地址的帧中传送。

六、利用 Wireshark 分析 UDP 协议

1) 启动 Wireshark，开始分组捕获；
2) 发送 QQ 消息给你的好友；
3) 停止 Wireshark 组捕获；
4) 在显示筛选规则中输入“udp”并展开数据包的细节。



问题思考：

a) 消息是基于 UDP 的还是 TCP 的？ **答：**UDP。

b) 你的主机 ip 地址是什么？目的主机 ip 地址是什么？

答：主机 IP 地址为 172.20.247.120，目的主机 IP 地址为 39.156.132.120。

c) 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：主机发送端口为 4001，QQ 服务器接收端口为 8000。

d) 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？

源端口号	目标端口号	长度	校验和

答：UDP（用户数据报协议）数据报的格式如图所示，共 8 字节，具体由 4 部分构成：

部分名称	字节数
源端口号（Source Port）	2
目标端口号（Destination Port）	2
长度（Length）	2
校验和（Checksum）	2

e) 为什么你发送一个 ICQ 数据包后，服务器又回给你的主机一个 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？

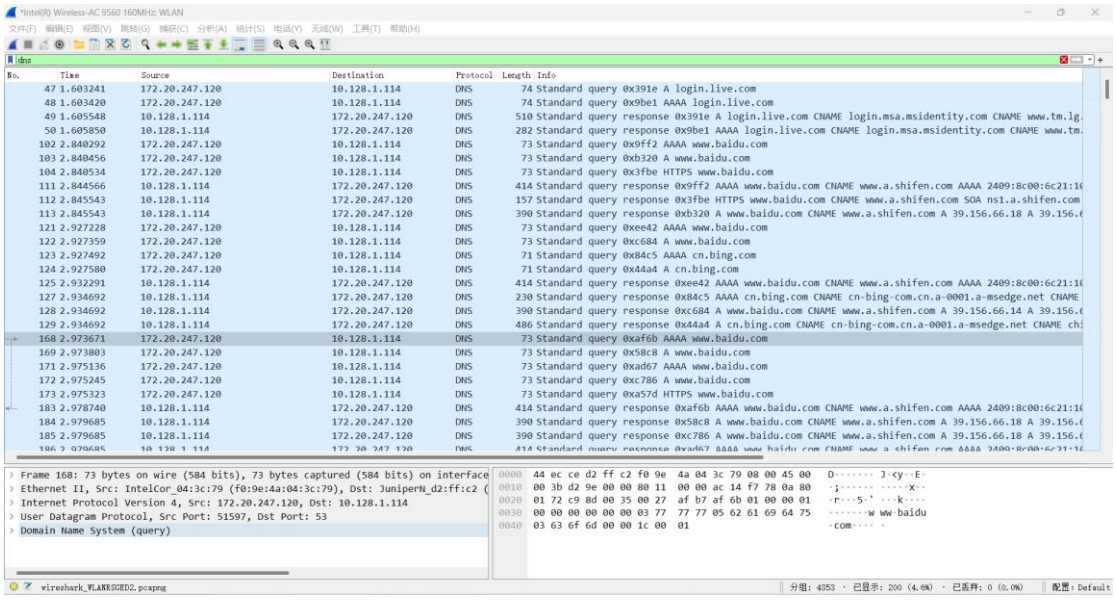
答：发送一个 ICQ 数据包后，服务器回复一个数据包，这种回应通常用于确认数据包的接收、处理请求或者传递状态信息。

服务器回复一个数据包仅确认数据包的接收，并不维护连接状态或数据包的顺序，数据有可能乱序到达或者丢失，为不可靠数据传输。

TCP 协议是一种面向连接的协议，它在通信的两端维护一个连接状态，确保数据包的可靠传递和顺序传输。TCP 使用序列号、确认号和连接建立过程来实现可靠性和连接状态维护，而 UDP 则不具备这些特性，因此是无连接的。

七、利用 Wireshark 分析 DNS 协议

1) 打开浏览器键入：www.baidu.com；
2) 打开 Wireshark，启动抓包；
3) 在控制台回车执行完毕后停止抓包，查看 wireshark 捕获的 DNS 报文如下。



分析得知，查询的目的地址均为相同的10.128.1.114，经查询是内网IP。

八、利用 Wireshark 分析 Ethernet 数据帧

以太网数据帧（Ethernet data frame）是在计算机网络中用于在物理层和数据链路层之间传输数据的基本单位，提供了底层的数据传输机制，而 TCP 和 HTTP 等协议则建立在它们之上，用于管理连接、数据分段、数据可靠性和应用层通信。

前同步码	目的地址	源地址	类型	数据	CRC

以太网数据帧结构如图所示，主要包含了物理地址（MAC 地址）和一些控制信息等六个字段，以确保数据帧的传输和完整性，在上述分析过程中已有提及。

```
Ethernet II, Src: IntelCor_04:3c:79 (f0:9e:4a:04:3c:79), Dst: JuniperN_d2:ff:c2 (44:ec:ce:d2:ff:c2)
  Destination: JuniperN_d2:ff:c2 (44:ec:ce:d2:ff:c2)
      Address: JuniperN_d2:ff:c2 (44:ec:ce:d2:ff:c2)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: IntelCor_04:3c:79 (f0:9e:4a:04:3c:79)
      Address: IntelCor_04:3c:79 (f0:9e:4a:04:3c:79)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 172.20.180.171, Dst: 128.119.245.12
```

例如上述数据帧报文，它包含了以下信息：

- Src：源 MAC 地址，指示了数据帧的发送者 f0:9e:4a:04:3c:79

- Dst：目标 MAC 地址，指示了数据帧的接收者 44:ec:ce:d2:ff:c2
- Type：这是数据帧的类型字段，指示数据帧中的数据是 IPv4（0x0800），表示数据帧中包含了 IPv4 协议的数据。

关于 MAC 地址的解释，主要用于局域网内的通信：

- LG bit（Locally/Group bit）：这位用于指示 MAC 地址是否是本地地址）还是组地址。0 表示是全局唯一地址，通常由硬件制造商分配的；
- IG bit（Individual/Group bit）：这位用于指示 MAC 地址是单播（数据帧只传递给一个设备）还是组播（数据帧被传递给一组设备）。

发送适配器在一个以太网帧中封装了一个 IP 数据报，并把该帧传递到物理层。接受适配器从物理层收到这个帧，提取出 IP 数据报，并把该 IP 数据报传递给网络层。

所有的以太网技术都需网络提供不可靠的无连接服务。

问题讨论：

1. 为什么访问网页时既有 IPv4 又有 IPv6？

在发送网页请求时，根据不同的网络配置和服务器支持，可能会出现两种 IP 版本（IPv4 和 IPv6）交替使用的情况，这被称为"双栈"（Dual-Stack）支持。

当客户端（通常是浏览器）与服务器通信时，它可以通过 DNS 解析域名来获取服务器的 IP 地址。如果域名的 DNS 记录同时包含 IPv4 和 IPv6 地址，那么双栈的客户端会尝试使用 IPv6 连接。如果 IPv6 连接失败或服务器不支持 IPv6，客户端将回退到 IPv4 连接。这种方式允许在 IPv4 和 IPv6 之间进行无缝的过渡，并确保尽可能多的设备可以访问网络资源。

2. 为什么有的网站再次访问没有 IF-MODIFIED-SINCE 字段？

网站和服务器可能使用缓存控制头（如"Cache-Control"和"Expires"）来指导浏览器是否应该缓存资源，以及在何时重新验证资源。如果服务器配置了缓存策略，浏览器可能会根据策略来处理资源的访问，而不需要 "If-Modified-Since" 字段。

心得体会：

1. 对HTTP、TCP、UDP、IP、DNS等协议的报文格式有了更深入的了解，理解了每个标志位的含义和作用，深入研究了这些协议之间的互动过程和工作原理。
2. 更加了解了网络协议实体之间的交互以及报文传递的一般流程和情景。
3. 了解学习了如何使用Wireshark工具来捕获数据包，以及如何使用PingPlotter工具来进行网络性能分析。