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	实验 3：典型协议的抓包分析				
姓名		院系			
班级		学号			
任课教师	刘亚维	指导教师	刘亚维		
实验地点	G002	实验时间	2024.04.30		
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：

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

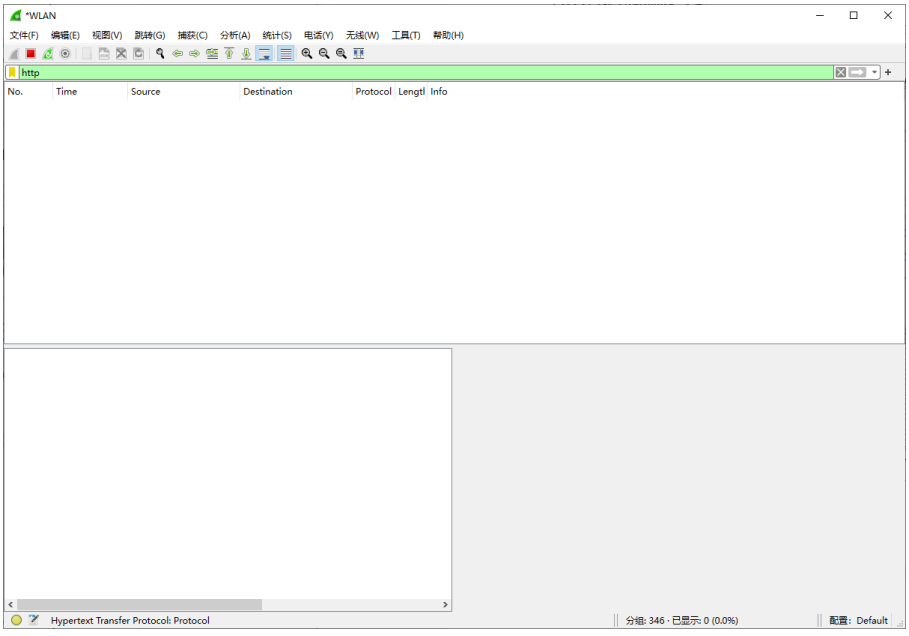
2 利用 Wireshark 分析 UDP 协议

3 利用 Wireshark 分析 ARP 协议

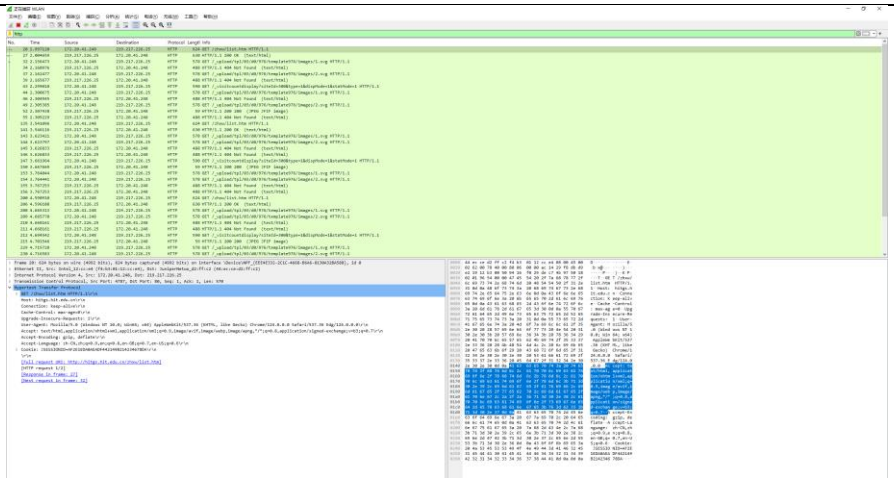
实验过程：

HTTP 分析：HTTP GET/response 交互

1. 启动Web browser，然后启动Wireshark分组嗅探器。在窗口的显示过滤说明出输入http，分组列表子窗口中将只显示所俘获到的HTTP报文。

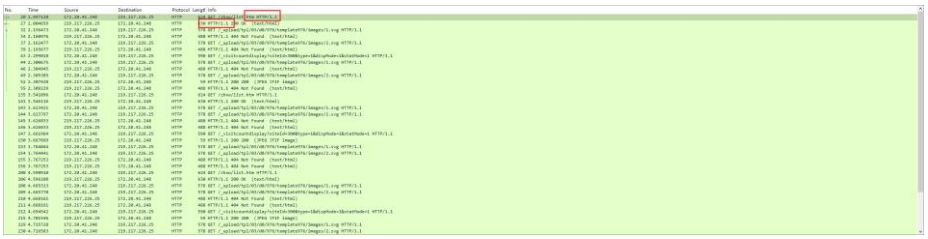


2. 开始 Wireshark 分组俘获
3. 在打开的 Web browser 窗口中输入以下地址：http://hitgs.hit.edu.cn/zhxw/list.htm
4. 停止分组俘获，结果如下所示



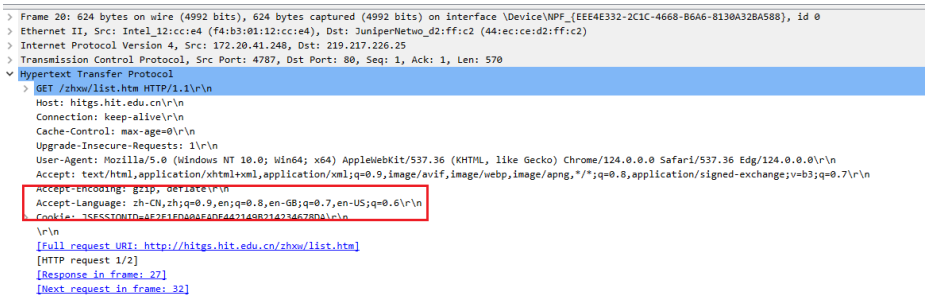
问题回答

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？



都是 HTTP1.1：第一条是 HTTP GET，浏览器运行 HTTP1.1，第二条是服务器返回，服务器也是运行 HTTP1.1

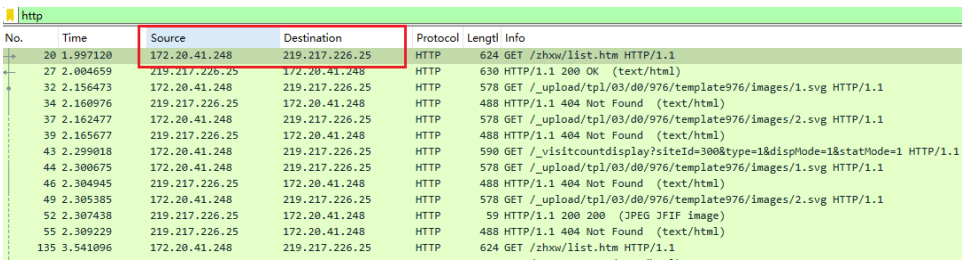
2. 你的浏览器向服务器指出它能接收何种语言版本的对象？



```
> Frame 20: 624 bytes on wire (4992 bits), 624 bytes captured (4992 bits) on interface \Device\NPF_{EEE4E332-2C1C-4668-B6A6-8130A32BA588}, id 0
> Ethernet II, Src: Intel_12:cc:e4 (f4:b3:01:12:cc:e4), Dst: JuniperNetwo_d2:ff:c2 (44:ec:ce:d2:ff:c2)
> Internet Protocol Version 4, Src: 172.20.41.248, Dst: 219.217.226.25
> Transmission Control Protocol, Src Port: 4787, Dst Port: 80, Seq: 1, Ack: 1, Len: 570
v Hypertext Transfer Protocol
  > GET /zhxw/list.htm HTTP/1.1\r\n
    Host: hitgs.hit.edu.cn\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36 Edg/124.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
    Cookie: JSESSIONID=AE2E1EDA0AEADE4421498214234678DA\r\n
    \r\n
    [Full request URI: http://hitgs.hit.edu.cn/zhxw/list.htm]
    [HTTP request 1/2]
    [Response in frame: 27]
    [Next request in frame: 32]
```

接收 zh-CN，zh 中文

3. 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？

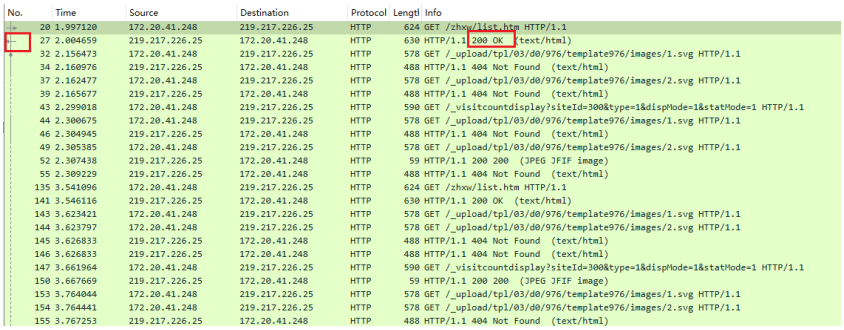


http

No.	Time	Source	Destination	Protocol	Lengtl	Info
20	1.997120	172.20.41.248	219.217.226.25	HTTP	624	GET /zhxw/list.htm HTTP/1.1
27	2.004659	219.217.226.25	172.20.41.248	HTTP	630	HTTP/1.1 200 OK (text/html)
32	2.156473	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/1.svg HTTP/1.1
34	2.160976	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
37	2.162477	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/2.svg HTTP/1.1
39	2.165677	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
43	2.299018	172.20.41.248	219.217.226.25	HTTP	590	GET /_visitcountdisplay?siteId=300&type=1&dispMode=1&statMode=1 HTTP/1.1
44	2.300675	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/1.svg HTTP/1.1
46	2.304945	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
49	2.305385	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/2.svg HTTP/1.1
52	2.307438	219.217.226.25	172.20.41.248	HTTP	59	HTTP/1.1 200 200 (JPEG JFIF image)
55	2.309229	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
135	3.541096	172.20.41.248	219.217.226.25	HTTP	624	GET /zhxw/list.htm HTTP/1.1

根据 GET 请求，我的计算机的 IP 地址是 source 即 172.20.41.248，服务器 IP 地址是 destination 即 219.217.226.25

4. 从服务器向你的浏览器返回的状态代码是多少？



No.	Time	Source	Destination	Protocol	Length	Info
20	1.997120	172.20.41.248	219.217.226.25	HTTP	624	GET /zhxw/list.htm HTTP/1.1
27	2.004659	219.217.226.25	172.20.41.248	HTTP	630	HTTP/1.1 200 OK (text/html)
32	2.156473	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/1.svg HTTP/1.1
34	2.160976	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
37	2.162477	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/2.svg HTTP/1.1
39	2.165677	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
43	2.299018	172.20.41.248	219.217.226.25	HTTP	590	GET /_visitcountdisplay?siteId=300&type=1&dispMode=1&statMode=1 HTTP/1.1
44	2.300675	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/1.svg HTTP/1.1
46	2.304945	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
49	2.305385	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/2.svg HTTP/1.1
52	2.307438	219.217.226.25	172.20.41.248	HTTP	59	HTTP/1.1 200 200 (JPEG JFIF image)
55	2.309229	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
135	3.541096	172.20.41.248	219.217.226.25	HTTP	624	GET /zhxw/list.htm HTTP/1.1
141	3.546116	219.217.226.25	172.20.41.248	HTTP	630	HTTP/1.1 200 OK (text/html)
143	3.623421	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/1.svg HTTP/1.1
144	3.623797	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/2.svg HTTP/1.1
145	3.626833	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
146	3.626833	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)
147	3.661964	172.20.41.248	219.217.226.25	HTTP	590	GET /_visitcountdisplay?siteId=300&type=1&dispMode=1&statMode=1 HTTP/1.1
150	3.667669	219.217.226.25	172.20.41.248	HTTP	59	HTTP/1.1 200 200 (JPEG JFIF image)
153	3.764044	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/1.svg HTTP/1.1
154	3.764441	172.20.41.248	219.217.226.25	HTTP	578	GET /_upload/tpl/03/d0/976/template976/images/2.svg HTTP/1.1
155	3.767253	219.217.226.25	172.20.41.248	HTTP	488	HTTP/1.1 404 Not Found (text/html)

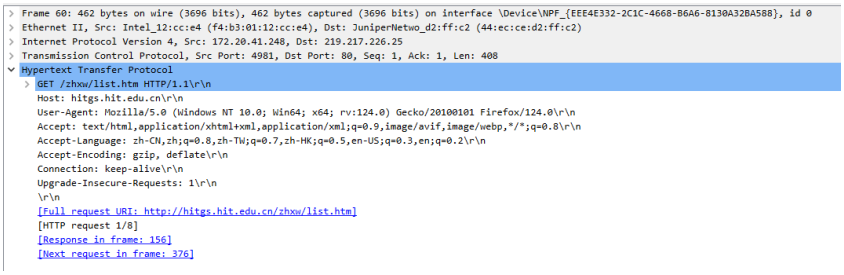
200 OK

HTTP 分析：HTTP 条件 GET/response 交互

1. 启动浏览器，清空浏览器的缓存（在浏览器中，选择“工具”菜单中的“Internet 选项”命令，在出现的对话框中，选择“删除文件”）。
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以下 URL: http://hitgs.hit.edu.cn/zhxw/list.htm,在你的浏览器中重新输入相同的 URL 或单击浏览器中的“刷新”按钮。
4. 停止 Wireshark 分组俘获，在显示过滤筛选说明处输入“http”,分组列表子窗口中将只显示所俘获到的 HTTP 报文。

回答问题

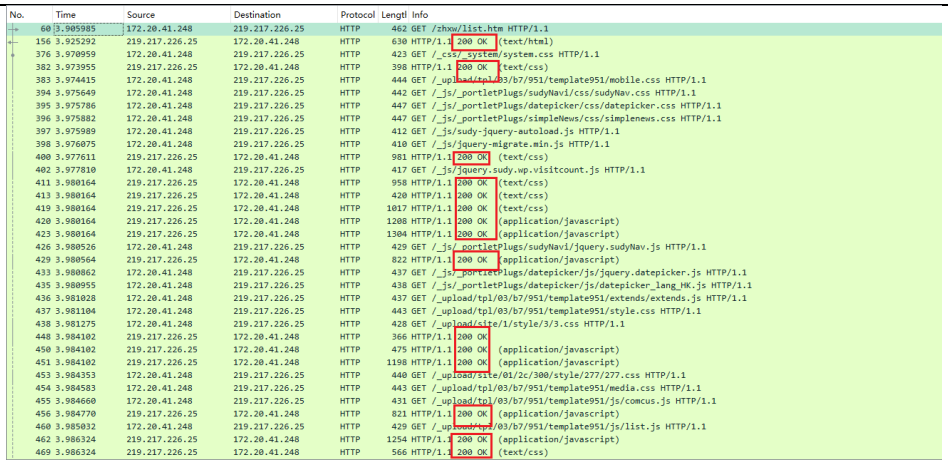
1. 分析你的浏览器向服务器发出的第一个 HTTP GET 请求的内容，在该请求报文中，是否有一行是：IF-MODIFIED-SINCE？



```
> Frame 60: 462 bytes on wire (3696 bits), 462 bytes captured (3696 bits) on interface \Device\NPF_{EEE4E332-2C1C-4668-B6A6-8130A32BA588}, id 0
> Ethernet II, Src: Intel_12:cc:e4 (f4:b3:01:12:cc:e4), Dst: JuniperNetwo_d2:ff:c2 (44:ec:ce:d2:ff:c2)
> Internet Protocol Version 4, Src: 172.20.41.248, Dst: 219.217.226.25
> Transmission Control Protocol, Src Port: 4981, Dst Port: 80, Seq: 1, Ack: 1, Len: 408
v Hypertext Transfer Protocol
  > GET /zhxw/list.htm HTTP/1.1\r\n
    Host: hitgs.hit.edu.cn\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:124.0) Gecko/20100101 Firefox/124.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://hitgs.hit.edu.cn/zhxw/list.htm]
    [HTTP request 1/8]
    [Response in frame: 156]
    [Next request in frame: 376]
```

没有 IF-MODIFIED-SINCE

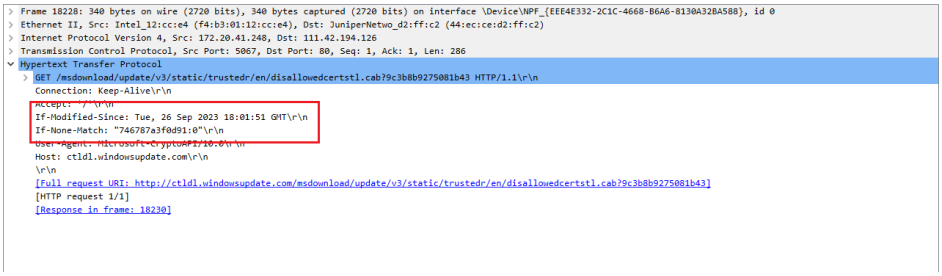
2. 分析服务器响应报文的内容，服务器是否明确返回了文件的内容？如何获知？
服务器明确返回了文件内容



如果返回状态码是 304 Not Modified 的话，说明缓存版本是最新的，消息不包含文件的内容；如果状态码是 200 OK 的话，说明缓存版本不是最新的或者缓存不命中，相应信息来自服务器端，消息中包含文件内容

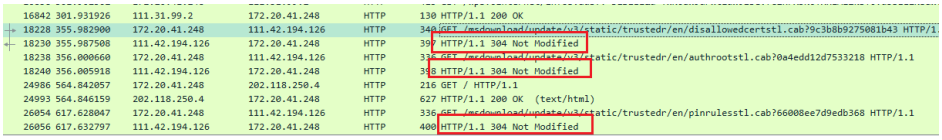
可以发现返回的所有报文状态码都是 200 OK，说明都明确返回了文件内容

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？



有，在该首部行后面跟着的信息是 TUE, 26 Sep 2023 18.01:51 GMT\r\n，代表着缓存的最后更新时间

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。



较晚的 GET 请求的 HTTP 状态码是 304 Not Modified

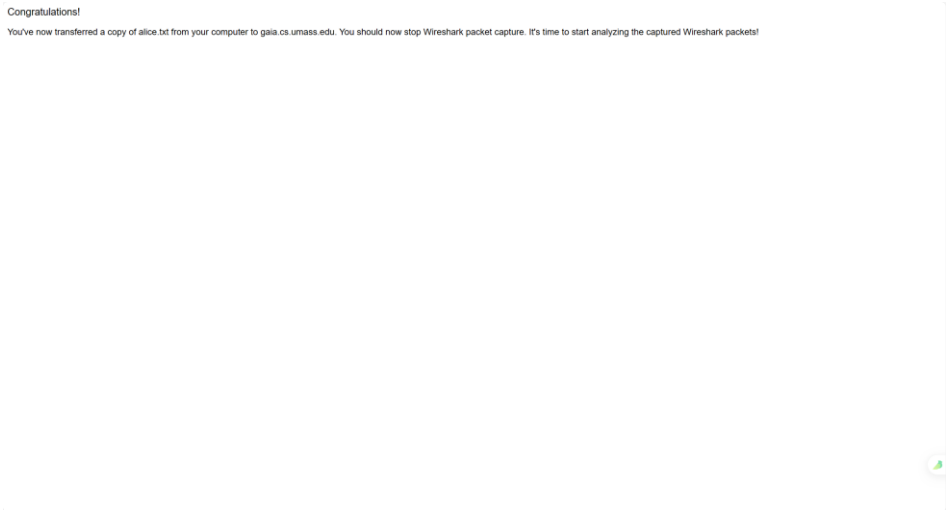
服务器不会明确返回文件，因为根据 GET 请求中 IF-MODIFIED-SINCE 字段内的时间，服务器返回 304 Not Modified，这说明客户端会使用本地没有过期的缓存文件

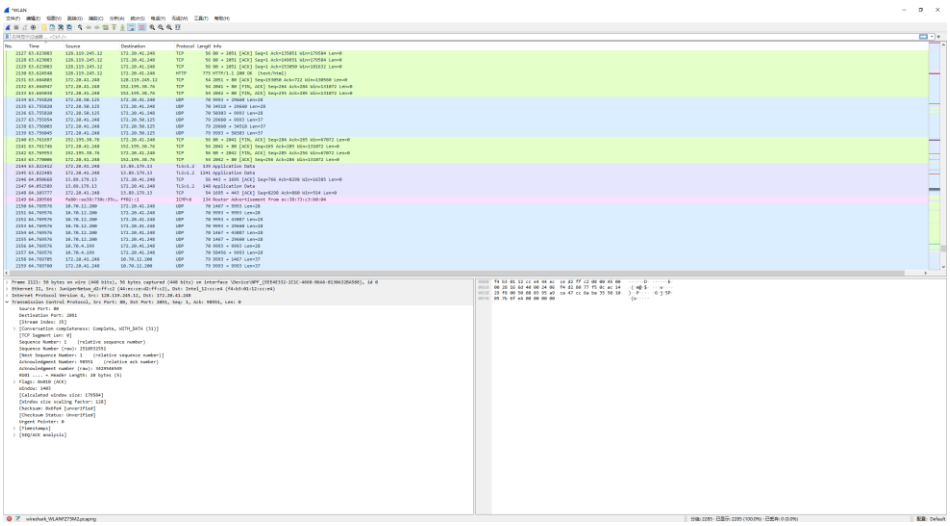
TCP 分析：俘获大量的由本地主机到远程服务器的 TCP 分组

1. 启动浏览器，打开 http://gaia.cs.umass.edu/wireshark-labs/alice.txt 网页，得到 ALICE'S ADVENTURES IN WONDERLAND 文本，将该文件保存到你的主机上。

2. 打开 http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html 网页。
3. 启动 Wireshark，开始分组俘获。
4. 在浏览器中，单击“Upload alice.txt file”按钮，将文件上传到 gaia.cs.umass.edu 服务器，一旦文件上传完毕，一个简短的贺词信息将显示在你的浏览器窗口中。
5. 停止俘获。

结果如下



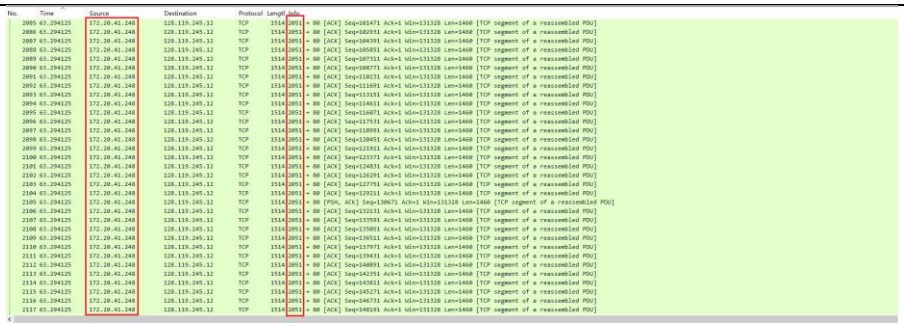


TCP 分析：浏览追踪信息

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。

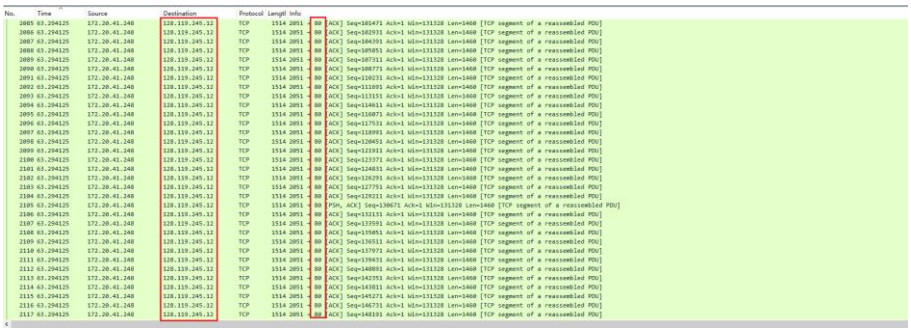
回答问题：

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？



客户端主机的 IP 地址是 172.20.41.248，TCP 端口号是 2051

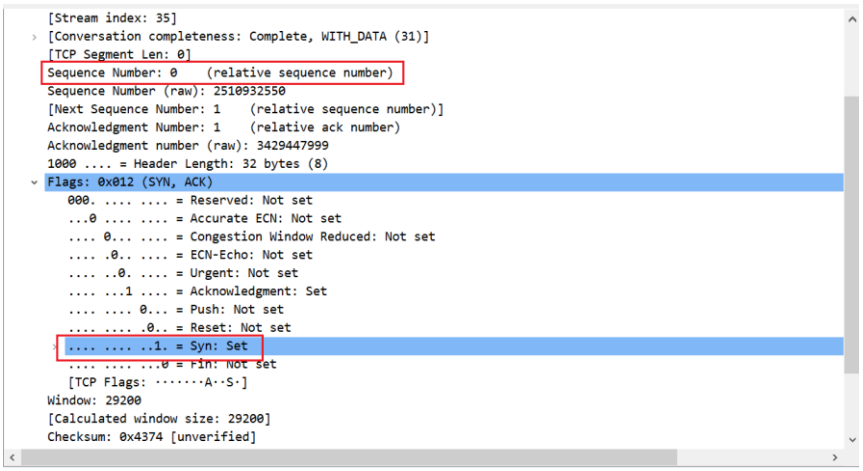
2. Gaia.cs.umass.edu 服务器的 IP 地址是多少？对这一连接，它用来发送和接收 TCP 报文的端口号是多少？



服务器的 IP 地址是 128.119.245.12，TCP 端口号是 80

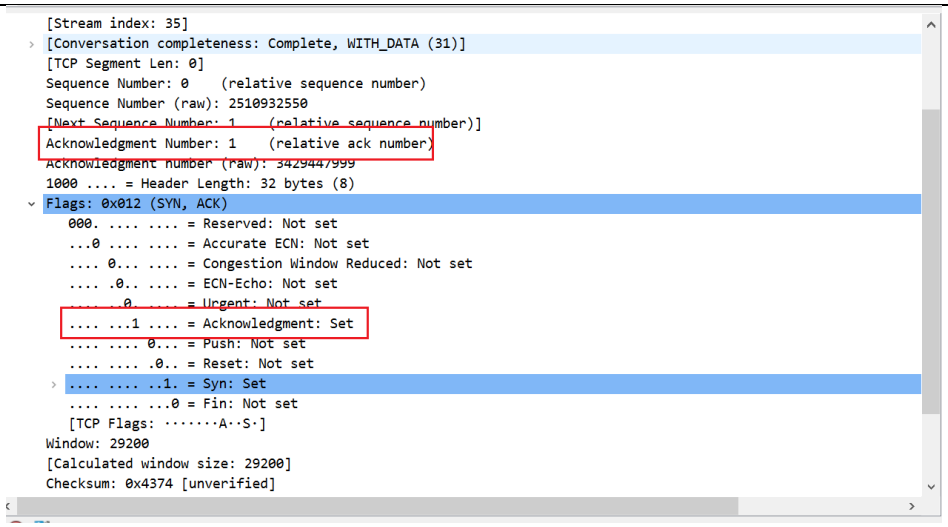
TCP 分析：TCP 基础

1. 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 报文段的序号是 0。在该报文段中通过设置 Flags 中的 SYN 标志位为 1，来标识该报文段是 SYN 报文段

2. 服务器向客户端发送的SYNACK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？



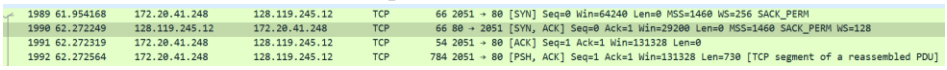
服务器发送的 SYN ACK 报文段序号为 0

报文段中，Acknowledgement 字段的值为 1

服务器通过将客户端发送过来的报文段的 seq+1 得到的 ACK 的值

通过将 Flags 位中的 ACK 标志位和 SYN 标志位同时置 1 来标识该报文段是 SYN ACK 报文段

3. 你能从捕获的数据包中分析出tcp三次握手过程吗？

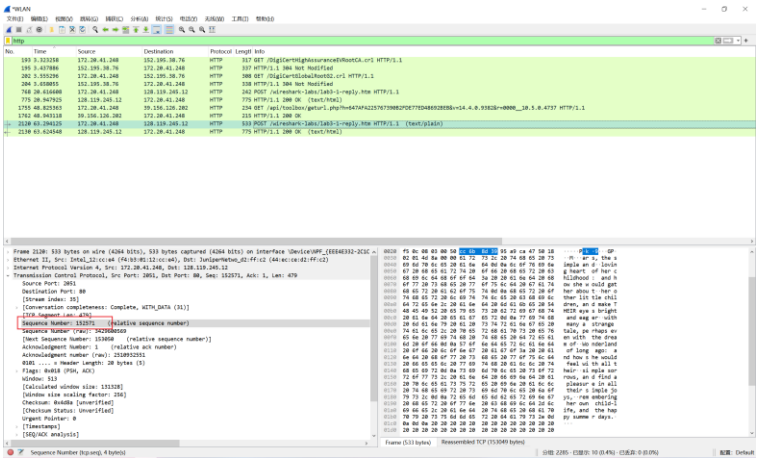


第一次：客户端向服务器发送 seq=0，SYN 标志为 1 建立请求连接

第二次：服务器收到 SYN，统一建立，回复 SYN ACK 报文段，seq 为 0，SYN 和 ACK 都为 1 以响应请求

第三次：客户端收到 SYN ACK，回复 ACK，其中 SYN 为 0，ACK 为 1，seq 为 1

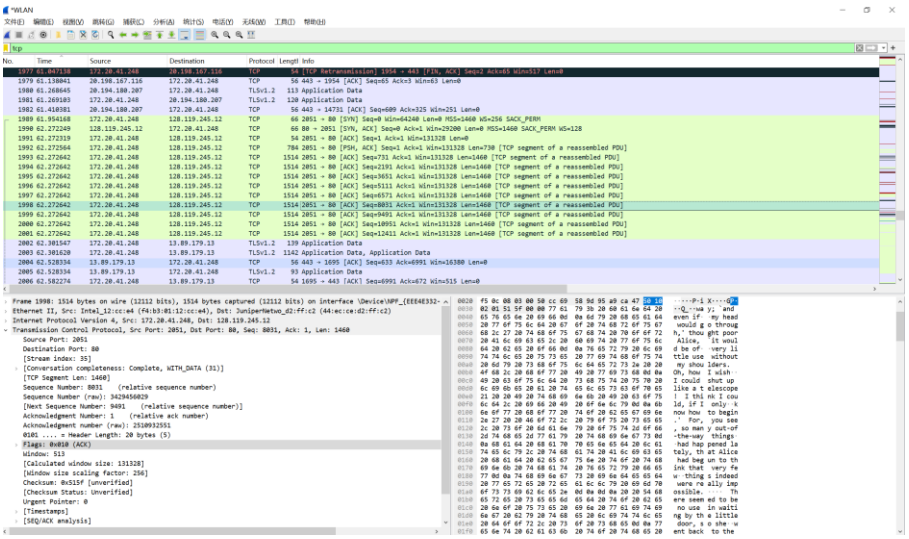
4. 包含HTTP POST命令的TCP报文段的序号是多少？



序号是 152571

5. 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？

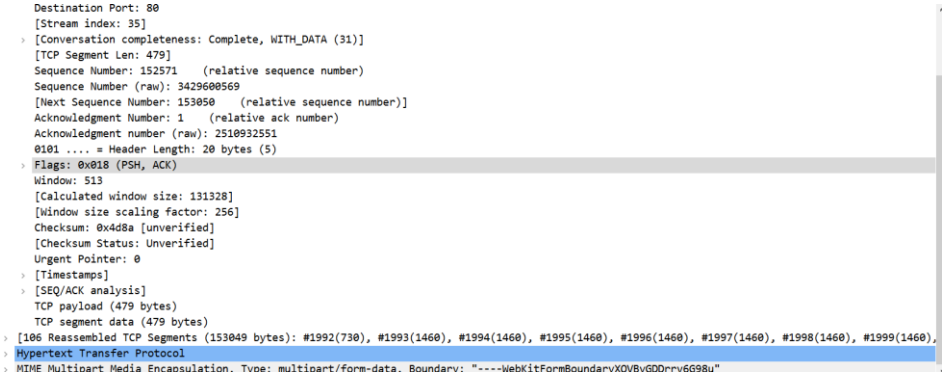


第 6 个报文段的序号是 8031,

在 HTTP POST 发送之前，TCP 连接建立之后发送的

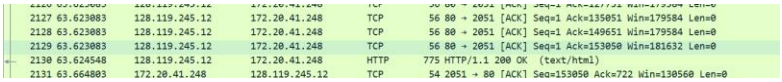
该报文段对应的 ACK 是在该报文段发送之后，HTTP POST 命令之后接收的

6. 前六个TCP报文段的长度各是多少？



长度分别为747,1460,1460,1460,1460,1460

7. 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？



接收端公示的最小可用缓存空间是 181632 字节。

不会不够用，接收端的缓存空间一直在增大

8. 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？

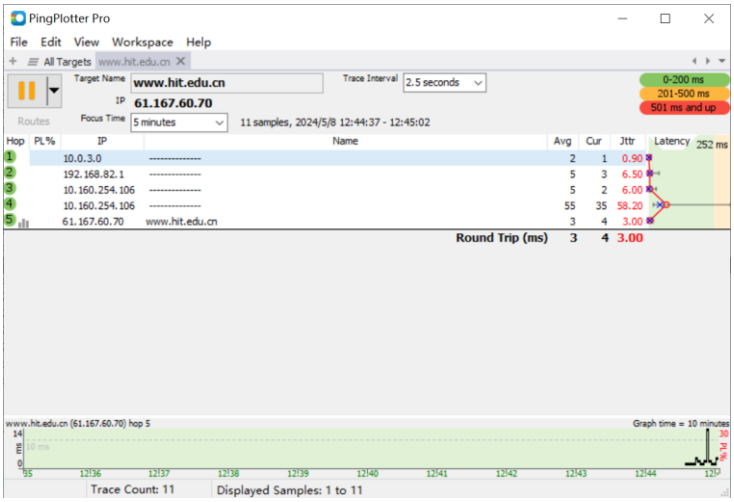
没有重传的报文段，因为发送的报文的序列号没有重复的序列号出现的情况

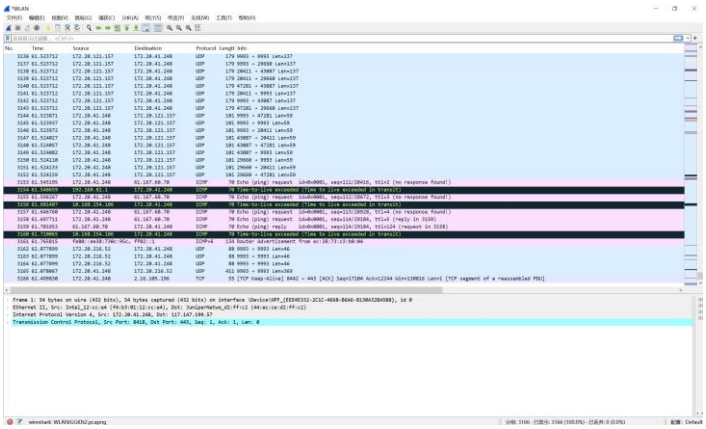
9. TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。

TCP 连接的总长度是 152940B；发送第一个段的时间是 61.954168；发送的最后一个段的时间为 62.665038；时间间隔为 62.665038-61.954168=0.71087ms；吞吐率为 152940*8/（0.71087*0.001）=1.7212Gbps

IP 分析

1. 通过执行 traceroute 执行捕获数据包





2. 对捕获的数据包进行分析

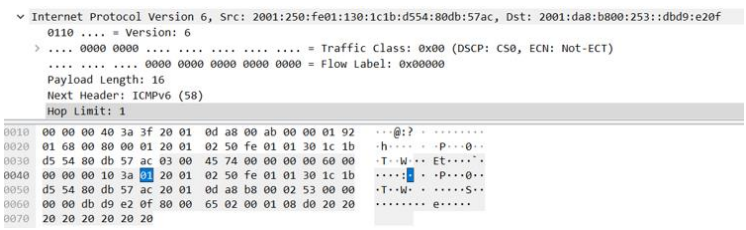
在你的捕获窗口中，应该能看到由你的主机发出的一系列 ICMPEcho Request 包和中间路由器返回的一系列 ICMP TTL-exceeded 消息。选择第一个你的主机发出的 ICMP Echo Request 消息，在 packet details 窗口展开数据包的 Internet Protocol 部分。

思考以下问题

1. 你主机的 IP 地址是什么

   172.20.41.248

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



   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？

```
> Frame 191: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NI
> Ethernet II, Src: JuniperN_d2:ff:c2 (44:ec:ce:d2:ff:c2), Dst: IntelCor_35:74:33 (bc:6e:e2:3
v Internet Protocol Version 4, Src: 10.160.254.106, Dst: 172.20.228.212
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x8418 (33816)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 252
    Protocol: ICMP (1)
    Header Checksum: 0x60b8 [validation disabled]
    [Header checksum status: Unverified]
```

IP 头有 20 字节

数据包的静载为 36 字节

4. 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？

```
v 010. .... = Flags: 0x2, Don't fragment
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 252
```

数据包分片了，有分片的偏移量

3. 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？

      Time to Live、Identification、Header Checksum 总是发生改变
   2. 哪些字段必须保持常量？哪些字段必须改变？为什么？

      Identification 用于区分不同的数据包，必须改变

      Time to Live 用于区分经过几个路由器，必须改变

      Header Checksum 由前面的部分计算而得，因此也必须改变。除此之外，其他字段保持常量
   3. 描述你看到的 IP 数据包 Identification 字段值的形式

      16 位（二字节），不断加 1 递增
   4. 找到由最近路由器（第一跳）返回给你主机的 ICMP Time-to-live exceeded 消息。

      思考下列问题

      1. Identification 字段和 TTL 字段的值是什么？

```
  Total Length: 56
  Identification: 0x45af (17839)
v 000. .... = Flags: 0x0
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 252
  Protocol: ICMP (1)
```

Identification 字段：17839（0x45af）

TTL 字段：252

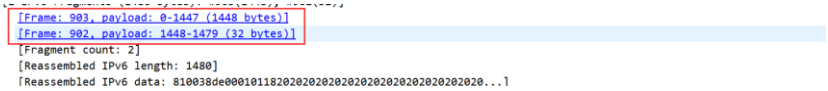
2. 最近的路由器（第一跳）返回给你主机的 ICMP Time-to-live exceeded 消息中这些值是否保持不变？为什么？

   保持不变

   对于 Identification 标识来说，相同的标识是为了分段后组装成同一段，不代表序号；因为是第一跳路由器返回的数据报，所以 TTL 也不变。

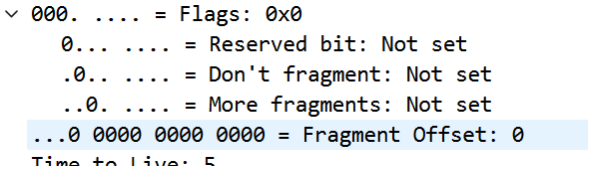
5. 单击 Time 列按钮，这样将对捕获的数据包按时间排序。找到在将包大小改为 2000 字节后你的主机发送的第一个 ICMP Echo Request 消息

   1. 该消息是否被分解成不止一个 IP 数据报？

      被分成了两个数据报

   2. 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？

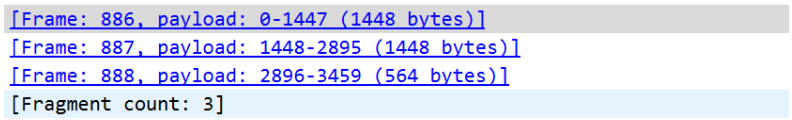
      More Fragments 位置 1，说明被分片，且不是最后一片

      Offset 为 0 说明是第一片分片

      分片的长度是 1448 字节

6. 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

   1. 原始数据包被分成了多少片？

      被分为 3 片

   2. 这些分片中 IP 数据报头部哪些字段发生了变化？

      偏移量字段、More Fragment 字段发生了变化

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
2. 在命令行模式下输入：ping 192.168.1.82（或其他 IP 地址）
3. 启动 Wireshark，开始分组俘获

回答问题

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？

```
  > arp -s 157.55.85.212   00-aa-00-62-c6-09.... 添加静态项。
  > arp -a                                .... 显示 ARP 表。

C:\Users\LEGION>arp -a

接口: 192.168.32.1 --- 0xa
  Internet 地址         物理地址              类型
  192.168.32.254        00-50-56-f4-a7-8c     动态
  192.168.32.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.2             01-00-5e-00-00-02     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: 192.168.136.1 --- 0xd
  Internet 地址         物理地址              类型
  192.168.136.254       00-50-56-ef-fd-ac     动态
  192.168.136.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.2             01-00-5e-00-00-02     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: 10.243.58.126 --- 0x13
  Internet 地址         物理地址              类型
  10.243.58.131         1a-79-9e-e6-93-2e     动态
  10.243.255.255        ff-ff-ff-ff-ff-ff     静态
  25.255.255.254        0a-00-b3-8d-ab-e2     静态
  224.0.0.2             01-00-5e-00-00-02     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: 10.241.58.126 --- 0x17
  Internet 地址         物理地址              类型
  10.241.165.164        ce-47-a9-e7-e1-6c     动态
  10.241.252.10         ce-03-d7-d1-3a-eb     动态
  10.241.255.255        ff-ff-ff-ff-ff-ff     静态
  25.255.255.254        de-00-de-ee-e9-a1     静态
  224.0.0.2             01-00-5e-00-00-02     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: 172.20.41.248 --- 0x18
  Internet 地址         物理地址              类型
  172.20.0.1            44-ec-ce-d2-ff-c2     动态
  172.20.1.2            44-ec-ce-d2-ff-c2     动态
  172.20.14.1           44-ec-ce-d2-ff-c2     动态
  172.20.17.236         f4-26-79-c3-30-84     动态
  172.20.41.52          44-ec-ce-d2-ff-c2     动态
  172.20.41.93          44-ec-ce-d2-ff-c2     动态
  172.20.41.172         44-ec-ce-d2-ff-c2     动态
  172.20.50.125         44-ec-ce-d2-ff-c2     动态
  172.20.59.197         44-ec-ce-d2-ff-c2     动态
  172.20.110.89         44-ec-ce-d2-ff-c2     动态
  172.20.115.244        44-ec-ce-d2-ff-c2     动态
  172.20.121.157        44-ec-ce-d2-ff-c2     动态
  172.20.158.94         24-41-8c-e5-cc-1b     动态
  172.20.176.66         40-f8-df-80-93-8f     动态
  172.20.181.101        44-ec-ce-d2-ff-c2     动态
  172.20.216.52         44-ec-ce-d2-ff-c2     动态
  172.20.220.49         10-63-c8-44-f0-15     动态
  172.20.227.86         44-ec-ce-d2-ff-c2     动态
  172.20.235.242        44-ec-ce-d2-ff-c2     动态
  172.20.236.183        44-ec-ce-d2-ff-c2     动态
  172.20.250.114        58-6c-25-2d-ed-58     动态
  172.20.255.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.2             01-00-5e-00-00-02     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.113           01-00-5e-00-00-71     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

C:\Users\LEGION>
```

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）。

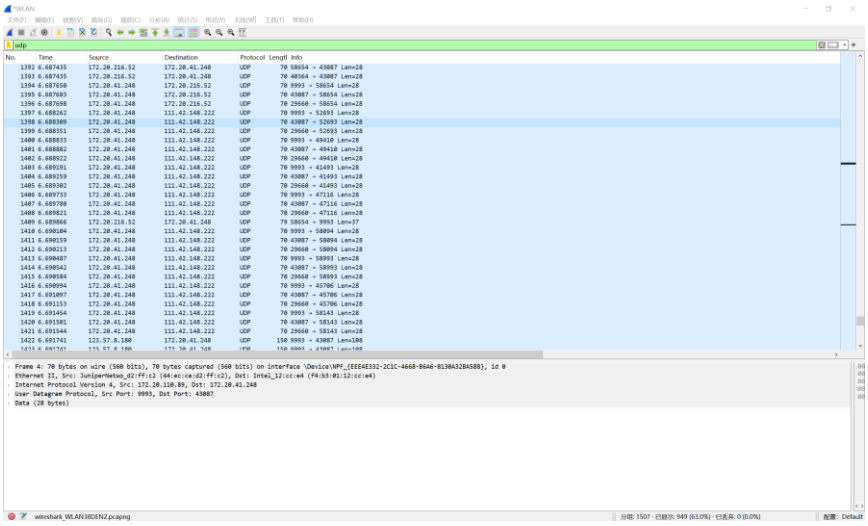
2. 清除主机上 ARP 缓存的内容,抓取 ping 命令时的数据包。分析数据包,回答下面的问题：
    1. ARP 数据包的格式是怎样的？由几部分构成，各个部分所占的字节数是多少？

ARP 数据包由 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）以及目的 IP 地址（4 字节）

2. 如何判断一个 ARP 数据是请求包还是应答包？
   可以通过 OP 字段查看。OP 为 0x0001 时表明该 ARP 数据为请求包， OP 为 0x0002 时表明该 ARP 数据是应答包。
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？
   因为在进行 ARP 查询时，发送主机并不知道目的 IP 对应的 MAC 地址，所以需要进行广播查询。而 ARP 响应报文知道查询主机的 MAC 地址，且局域网中的其他主机不需要此次查询的结果，因此 ARP 响应要在一个有着明确目的局域网地址的帧中传送。

抓取 UDP 数据包

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。



分析 QQ 通讯中捕获到的 UDP 数据包。根据操作思考以下问题：

1. 消息是基于 UDP 还是 TCP 的
   如图所示，消息是基于 UDP 的
2. 你的主机 ip 地址是什么？目的主机 ip 地址是什么？

1239 5.701991	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1240 5.702247	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1241 5.702269	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1242 5.702514	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1243 5.702538	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1244 5.702788	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1245 5.702810	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1246 5.703066	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1247 5.703090	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1248 5.703336	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1249 5.703358	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28

我的主机 IP 地址是 172.20.41.248，目的主机 IP 是 104.194.8.134

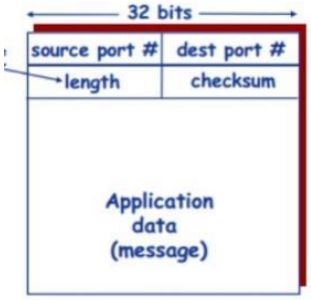
3. 你的主机发送 QQ 消息的端口号和 QQ 服务器的端口号分别是多少？

1239 5.701991	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1240 5.702247	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1241 5.702269	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1242 5.702514	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1243 5.702538	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1244 5.702788	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1245 5.702810	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1246 5.703066	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1247 5.703090	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28
1248 5.703336	172.20.41.248	104.194.8.134	UDP	46 43087 → 9993 Len=4
1249 5.703358	172.20.41.248	104.194.8.134	UDP	70 43087 → 9993 Len=28

主机端口号为 43087，QQ 服务器端口号为 9993

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？

UDP 数据报的格式如下图所示



其中 UDP 数据报由四部分组成：端口号（2 字节），目的端口号（2 字节），长度（2 字节），校验和（2 字节）

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？

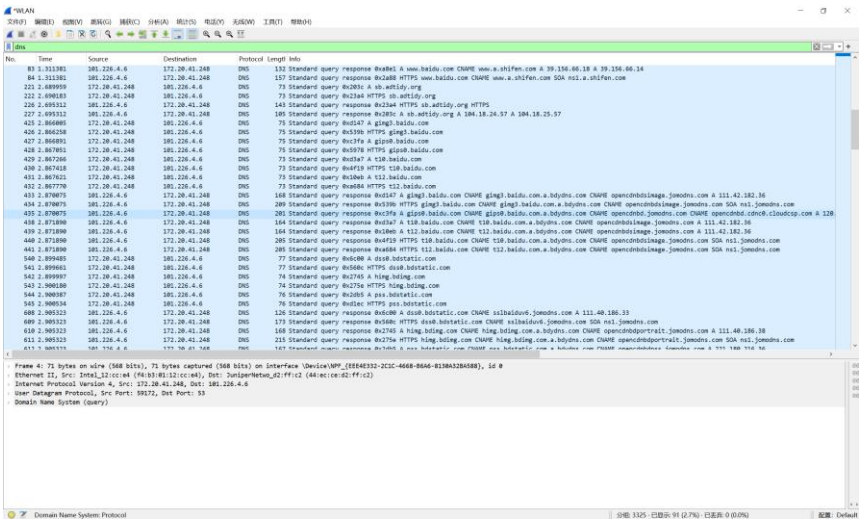
这是因为服务器需返回接收的结果给客户端。

服务器只提供了一次 ACK，所以不保证数据一定送达，即为不可靠传输；

可以看出。因为 UDP 数据包没有序列号，因此不能像 TCP 协议那样先进行握手再进行数据发送。

利用 Wireshark 进行 DNS 协议分析

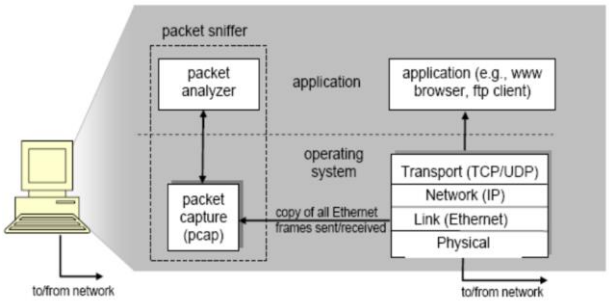
1. 打开浏览器键入：www.baidu.com；
2. 打开 Wireshark,启动抓包；
3. 在控制台回车执行完毕后停止抓包。



问题讨论：

1. 实验中的分组嗅探器的基本概念

观察在正在运行协议实体间交换报文的基本工具被称为分组嗅探器（packet sniffer）。顾名思义，一个分组嗅探器俘获（嗅探）计算机发送和接收的报文。一般情况下，分组嗅探器将存储和显示出被俘获报文的各协议头部字段的内容。下图为一个分组嗅探器的结构



如图所示，右边是计算机上正常运行的协议（在这里 Internet 协议）和应用程序（如：web 浏览器和 ftp 客户端）。分组嗅探器（虚线框中的部分）是附加计算机普通软件上的，主要有两部分组成。分组俘获库（packetcapture library）接收计算机发送和接收的每一个链路层帧的拷贝。高层协议（如：HTTP、FTP、TCP、UDP、DNS、IP 等）交换的报文都被封装在链路层帧中，并沿着物理媒体（如以太网的电缆）传输。并且图 1 假设所使用的物理媒体是以太网，上层协议的报文最终封装在以太网帧中

分组嗅探器的第二个组成部分是分析器。分析器用来显示协议报文所有字段的内容。为此，分析器必须能够理解协议所交换的所有报文的结构。例如：我们要显示图中 HTTP 协议所交换的报文的各个字段。分组分析器理解以太网帧格式，能够识别包含在帧中的 IP 数据报。分组分析器也要理解 IP 数据报的格式，并能从 IP 数据报中提取出 TCP 报文段。然后，它需要理解 TCP 报文段，并能够从中提取出 HTTP 消息。最后，它需要理解 HTTP 消息

2. Wireshark 介绍

   Wireshark 是一种可以运行在 Windows, UNIX 以及 Linux 等操作系统上的分组分析器。Wireshark（前称 Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark 使用 WinPCAP 作为接口，直接与网卡进行数据报文交换

心得体会：

1. 对计算机网络的架构有了更深层次的洞察，特别是对于应用层、传输层、网络层和数据链路层的运作机制和它们之间的相互作用有了更清晰的理解。通过捕获和分析协议数据包，我不仅对这些协议的结构和工作方式更加熟悉，还对它们在网络通信中扮演的角色有了更深入的了解。
2. 通过使用 Wireshark 工具来观察和分析网络协议的运行，我加深了对这些协议交互过程的理解，并对协议中各个字段的功能和重要性有了更深刻的认识。我现在已经能够熟练地使用 Wireshark 进行数据包的捕获和分析。我认为 Wireshark 是一个非常强大的工具，它不仅在当前的实验中非常有用，而且在未来的学习和职业生涯中也会发挥重要作用。
3. 对于计算机网络体系结构中不同协议的功能和它们在数据包传输过程中的角色有了更加明确的认识。现在我对数据包在网络中的传输流程和涉及的各种机制有了更加深入的理解。