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	冯开来		院系	计算学部	
班级	1903602		学号	1190201215	
任课教师	李全龙		指导教师	李全龙	
实验地点	格物 207		实验时间	2021.11.24	
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					

计算机科学与技术学院 SINCE 1956...
School of Computer Science and Technology

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

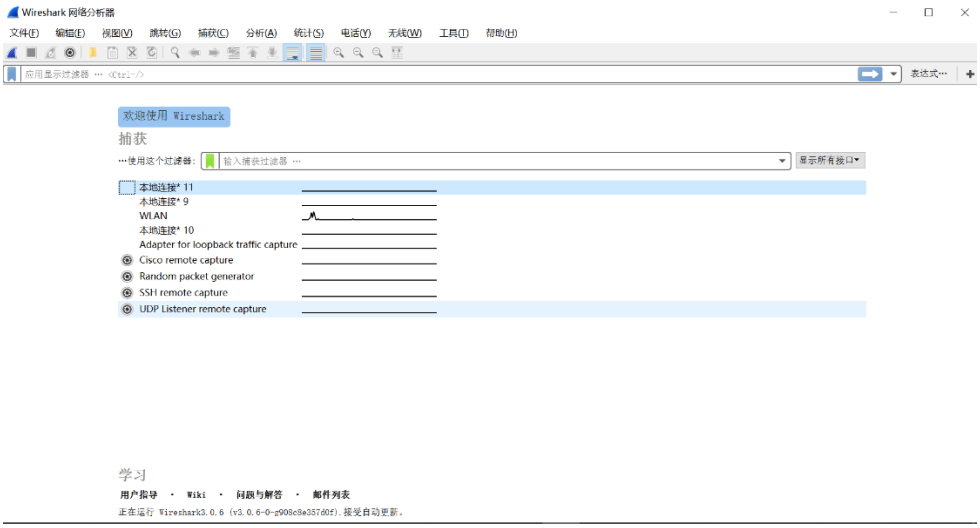
选做内容：

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

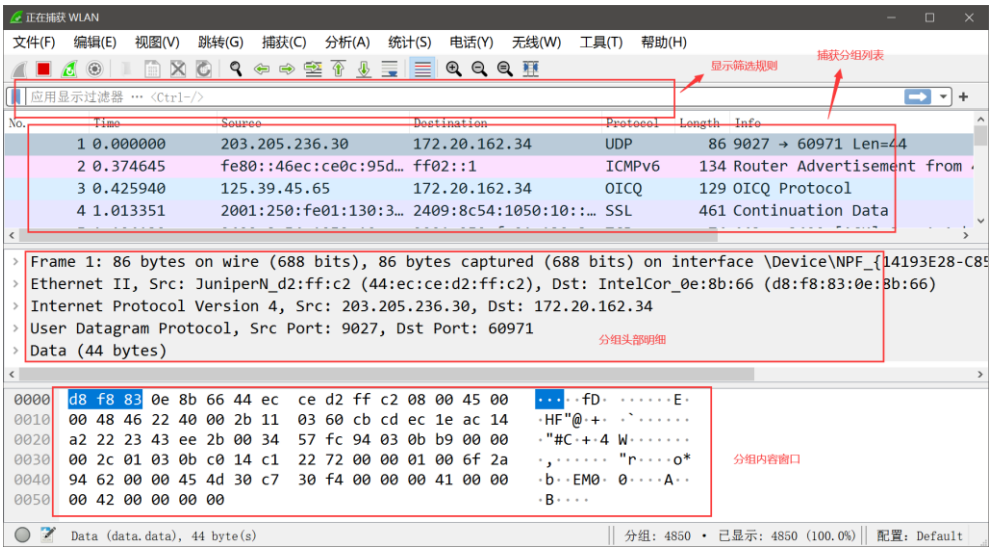
实验过程：

(一) Wireshark的使用

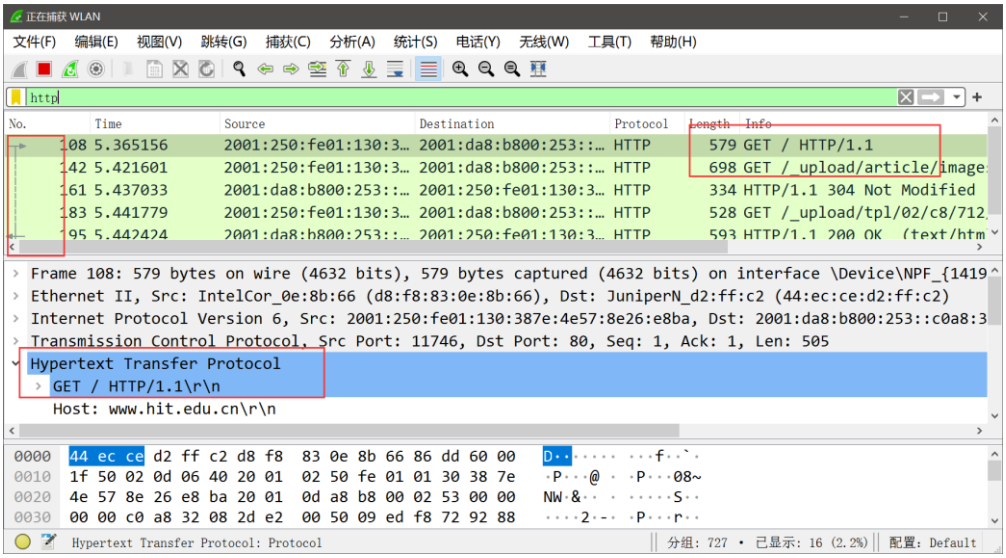
启动Web浏览器，启动Wireshark



开始分组捕获，出现分组捕获窗口：



输入URL（http://www.hit.edu.cn），捕获HTTP GET报文：



(二) HTTP分析

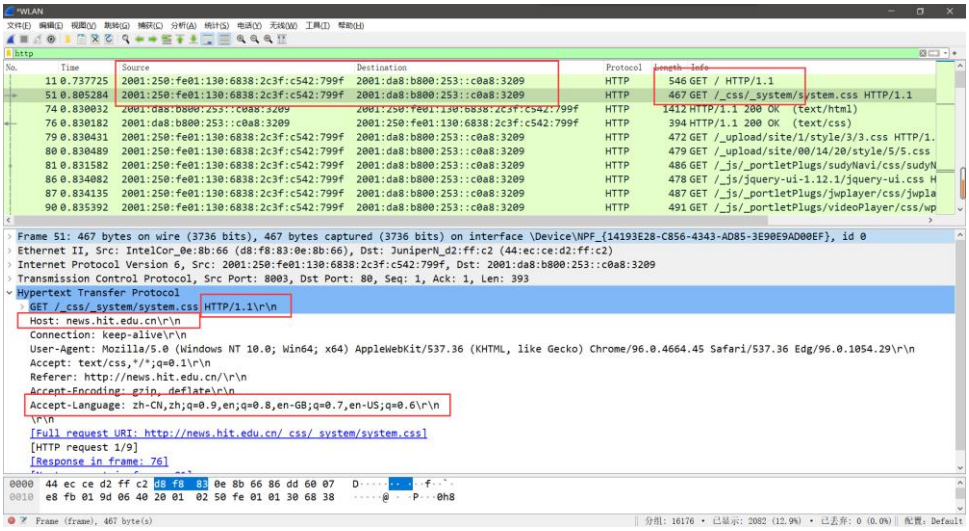
1) HTTP GET/response交互

a) 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。

b) 开始Wireshark分组俘获。

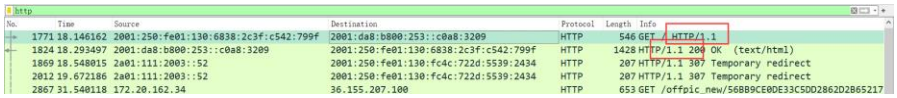
c) 在打开的Web browser窗口中输入一下地址：http://news.hit.edu.cn

d) 停止分组俘获。



根据俘获窗口内容，思考以下问题：

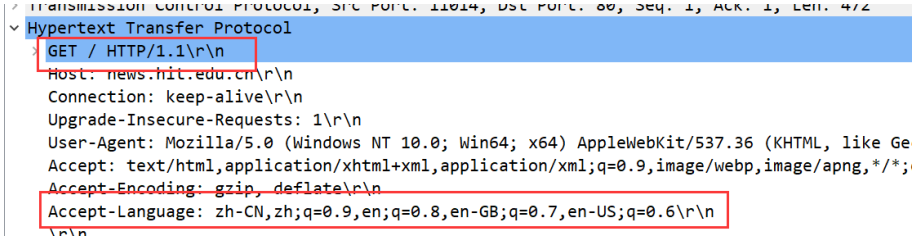
a) 你的浏览器运行的HTTP1.0还是HTTP1.1？访问的服务器HTTP协议版本号？



第一条是HTTP GET，浏览器运行HTTP1.1

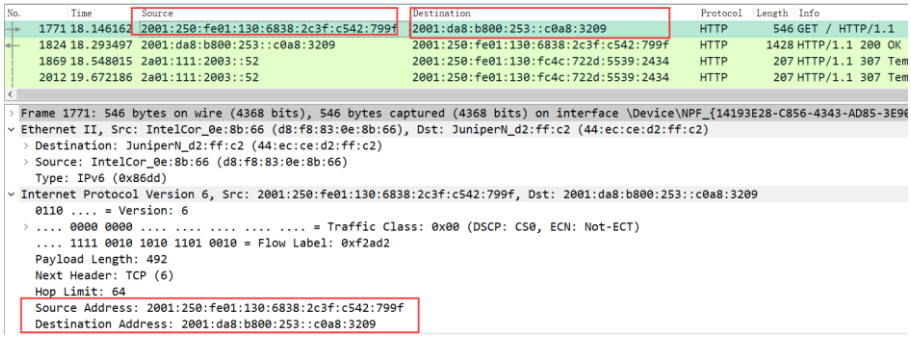
第二条是服务器端返回，服务器运行HTTP1.1

b) 你的浏览器向服务器指出接受何种语言版本



查看GET报文，Accept-Language包含简体繁体中文，英语（美）

c) 你的计算机IP地址？服务器http://news.hit.edu.cn/news的IP地址？

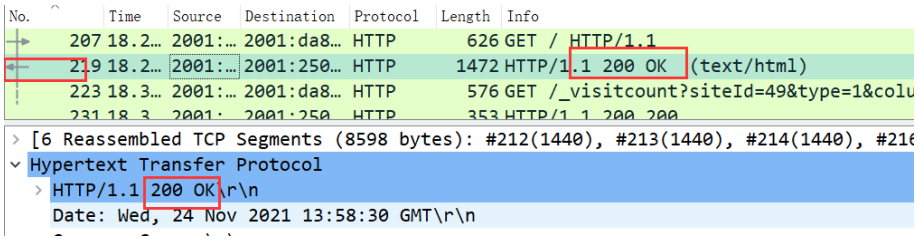


根据GET请求，我计算机IP地址即为Source，服务器为Destination

计算机IP：2001:250:fe01:130:6838:2c3f:c542:799f

服务器IP：2001:da8:b800:253::c0a8:3209

d) 服务器返回你的浏览器的状态码？



状态码为200。

2) HTTP 条件GET/response交互

e) 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”）。

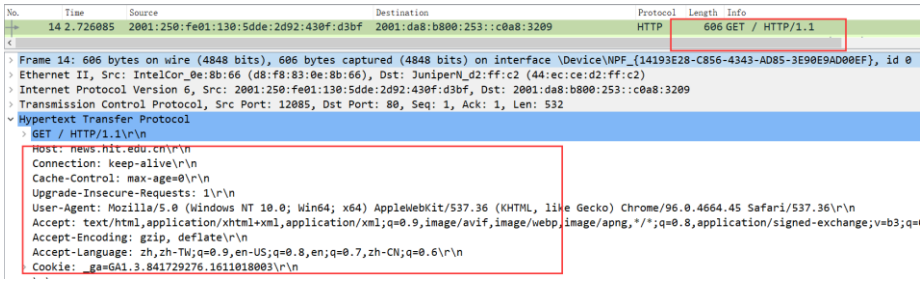
f) 启动Wireshark 分组俘获器。开始Wireshark 分组俘获。

g) 在浏览器的地址栏中输入以下URL: http://hitgs.hit.edu.cn/news,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。

h) 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。

根据俘获窗口内容，思考一下问题：

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？



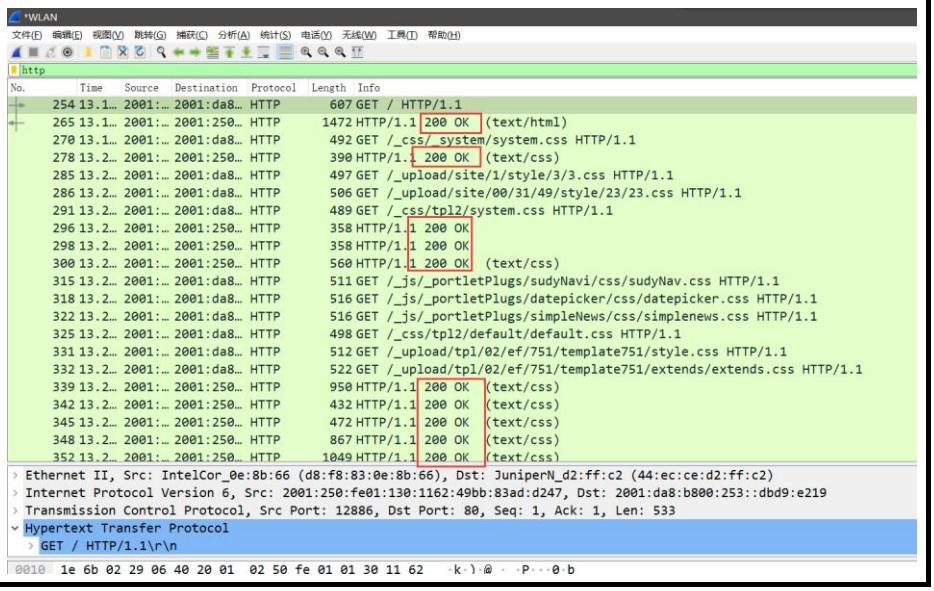
并没有IF-MODIFIED-SINCE

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？

服务器明确返回了文件的内容。

返回报文的状态码是304 Not Modified 的话，说明缓存的版本是最新的，相应的消息中不包含文件内容；

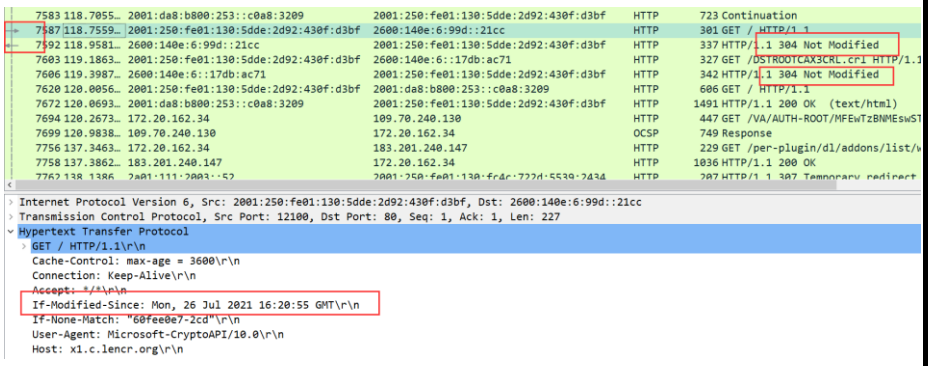
若服务器返回的报文的状态码是 200 OK 的话，说明缓存的版本不是最新，或者缓存 不命中，相应的消息来自服务器端，消息中包含文件内容。若返回的所有报文消息状态码都是 200 OK，所以说明都明确返回了文件的内容。



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？

有IF-MODIFIED-SINCE（本人刷新很多次好不容易刷新出304，太不容易）

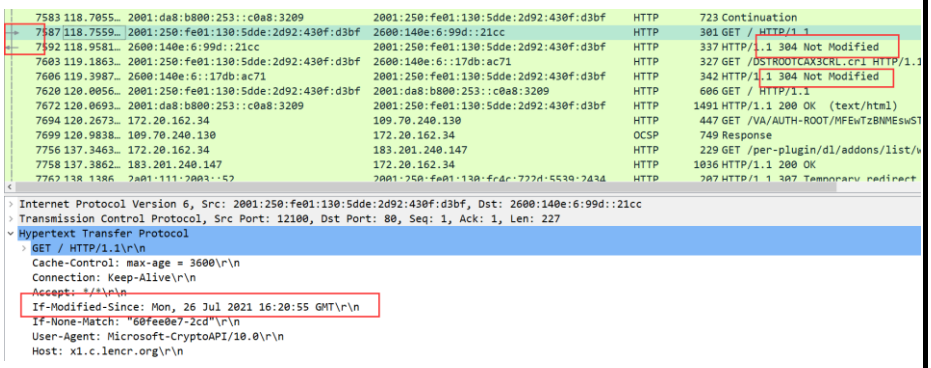
在 IF-MODIFIED-SINCE 后面跟着的是一个时间，目的是询问服务器在这个时间之后是否有修改，如果有修改，则向服务器端请求；如果没有修改则由代理从缓存中找到文件数据发送给客户端，并在返回的时候状态码改为 304 Not Modified。



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。

较晚的 GET 请求的 HTTP 状态码是 304。

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



(三) TCP分析

1) 俘获大量的由本地主机到远程服务器的TCP分组

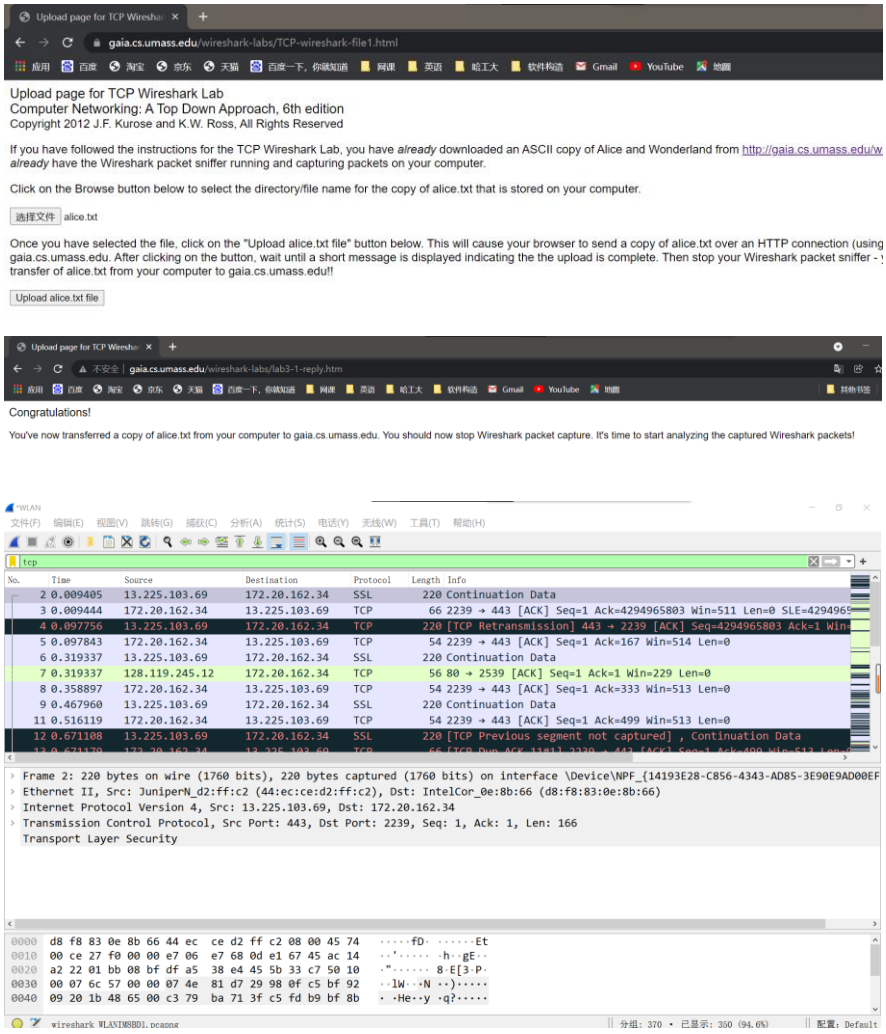
a) 启动浏览器，打开http://gaia.cs.umass.edu/wireshark-labs/alice.txt网页，得到ALICE'S ADVENTURES IN WONDERLAND文本，将该文件保存到你的主机上。

b) 打开https://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html网页。

c) 启动Wireshark，开始分组俘获。

d) 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。

e) 停止俘获。

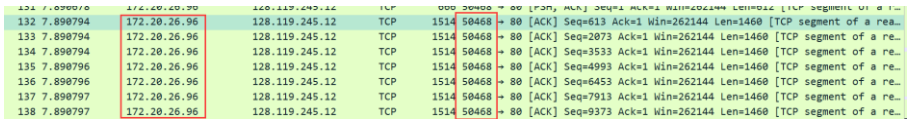


2) 浏览追踪信息

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。

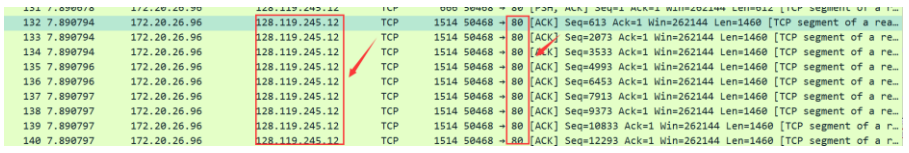
根据操作思考以下问题：

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？



客户端主机IP地址为172.20.26.96，TCP端口号为50468

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？

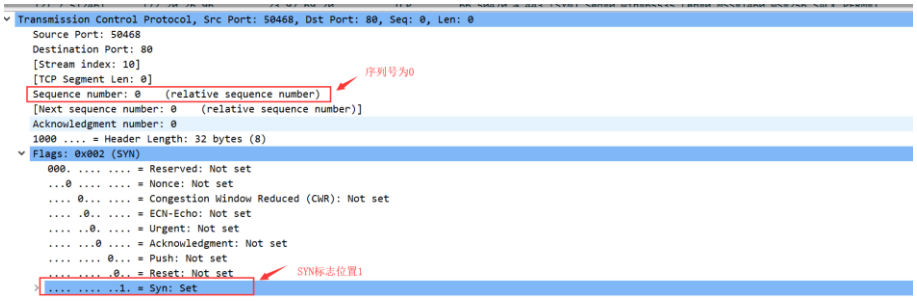


服务器的IP地址为128.119.245.12，端口号为80。

3) TCP基础

根据操作思考以下问题：

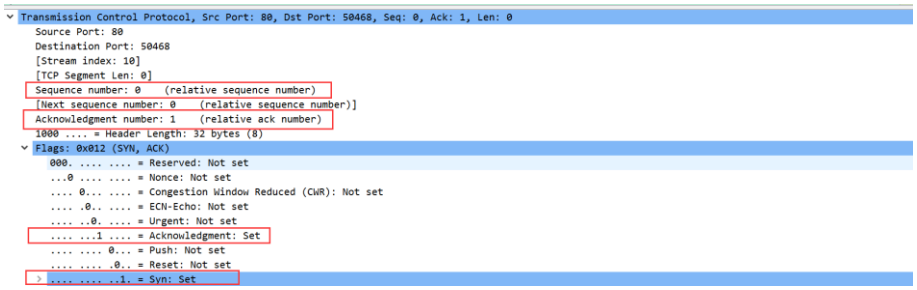
a) 客户服务器之间用于初始化TCP连接的TCP SYN报文段的序号（sequence number）是多少？在该报文段中，是用什么来标示该报文段是SYN报文段的？



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

报文段中利用一个SYN标志位，该标志位置1时，标识该报文段是SYN报文段

b) 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？



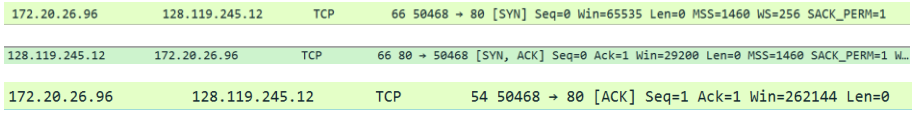
服务器发送的SYNACK报文段序号为0

报文段中，Acknowledgement字段的值为1

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 ACK 标志位和 SYN 标志位同时置 1 来标识该报文段是 SYNACK 报文段

c) 你能从捕获的数据包中分析出tcp三次握手过程吗？



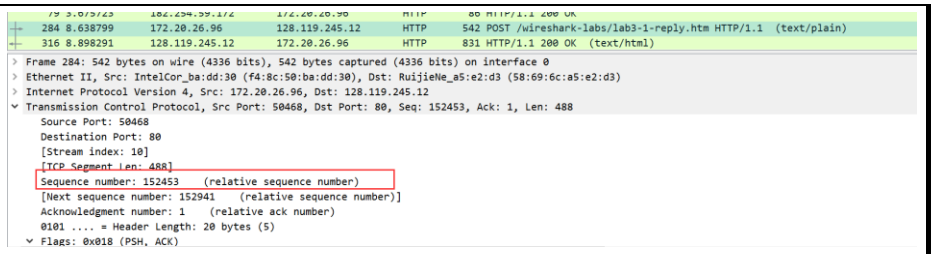
第一次握手：客户端会向服务器发送一个TCP SYN报文段，其中SYN标志位置1，初始序列号Seq为0，不携带其他任何数据，请求与服务器建立TCP连接

第二次握手：服务器收到SYN报文段，同意建立连接，则回复一个SYNACK报文段，其中SYN和ACK标志位都置1，初始序列号Seq为0，Ack为1，以此响应客户端请求

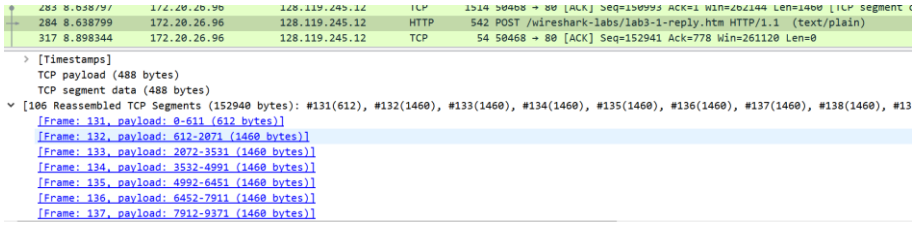
第三次握手：客户端收到服务器的SYNACK报文段，回复ACK报文段，其中SYN标志位置0，ACK标志位置1，Seq=1，Ack=1，可以携带数据

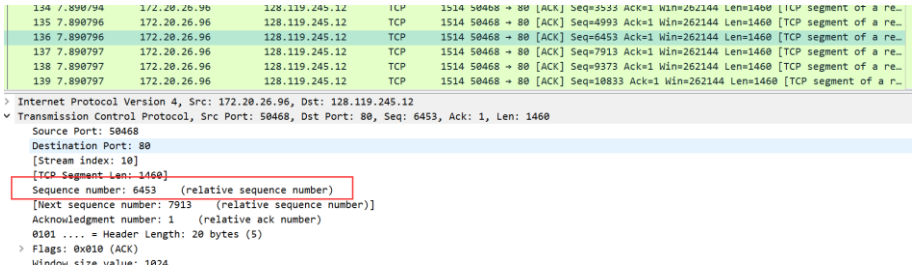
d) 包含HTTP POST命令的TCP报文段的序号是多少？

序列号为152453



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？



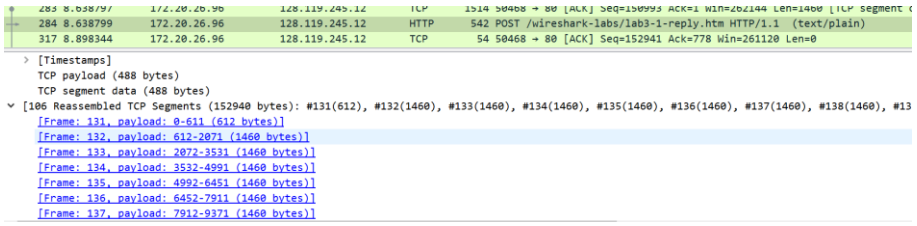


第六个报文段的序号为6453

在HTTP POST命令之前，HTTP连接建立之后发送的

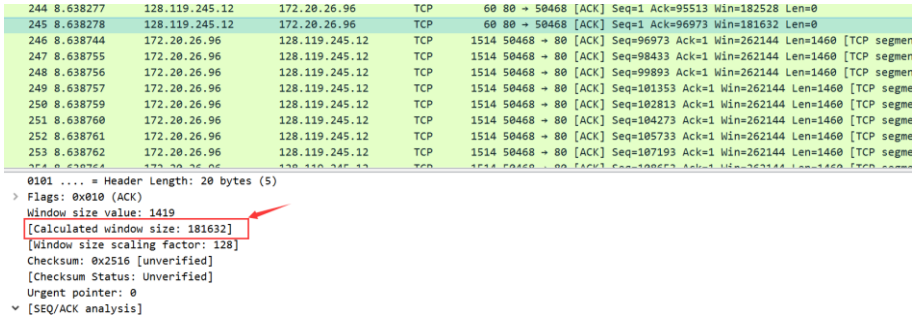
该报文段对应的ACK是在该报文段发送之后，HTTP POST命令之后接收的

f) 前六个TCP报文段的长度各是多少？



第一个TCP报文段长度为612字节，剩下的五个均为1460字节

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？



接收端公示的最小的可用缓存空间是181632字节

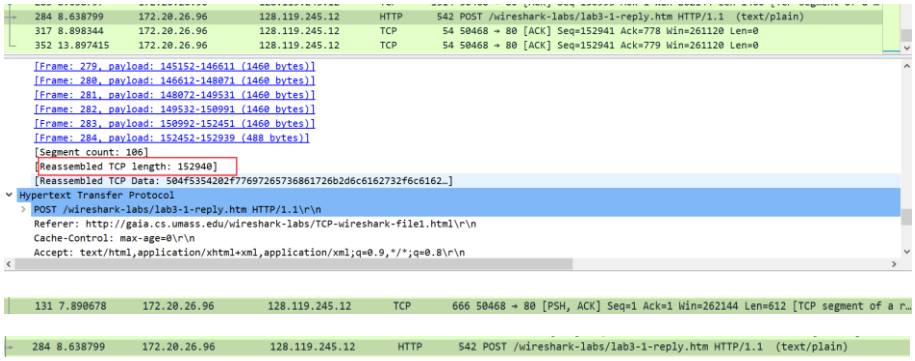
```
Window size value: 1477
[Calculated window size: 189056]

Window size value: 1523
[Calculated window size: 194944]

Window size value: 1591
[Calculated window size: 203648]

Window size value: 1865
[Calculated window size: 238720]

Window size value: 2139
[Calculated window size: 273792]
```

接收端的缓存不会不够用，如图可见，接收端的缓存空间一直在增大

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？

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；

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



TCP 段的总长度为 152940 字节

发送第一个段的时间为 7.890678

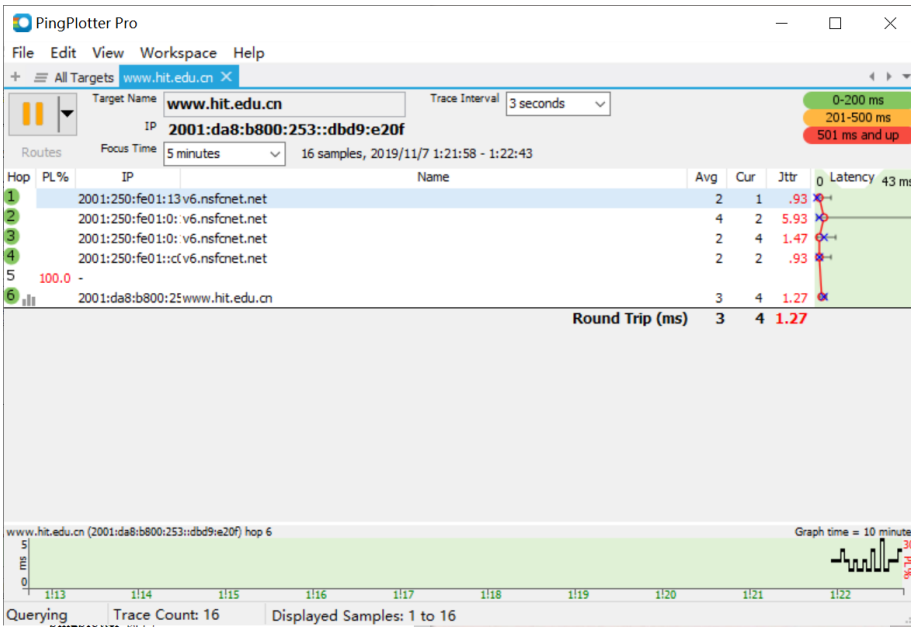
发送最后一个段的时间为 8.638799

时间间隔为 8.638799 - 7.890678 = 0.748121 ms

因此吞吐率为 152940*8/(0.748121*10^(-3))≈1.635Gbps

(四) IP分析

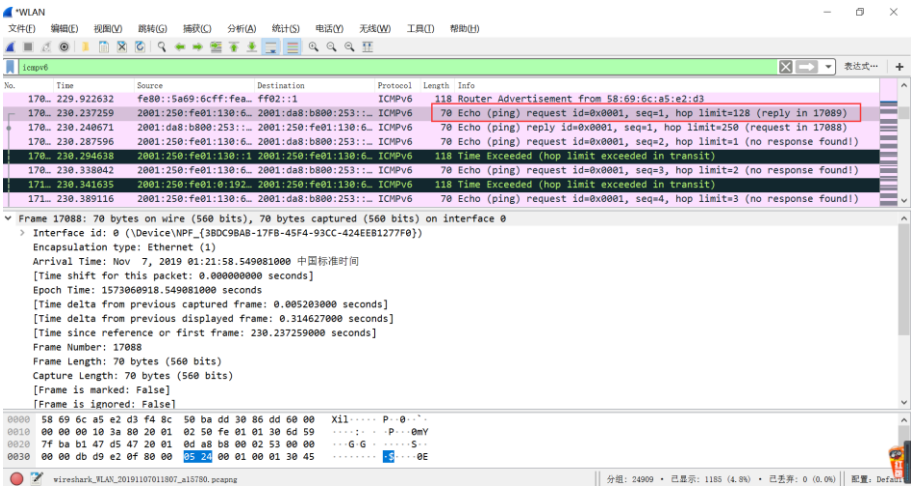
1) 通过执行traceroute 执行捕获数据包



2) 对捕获的数据包进行分析

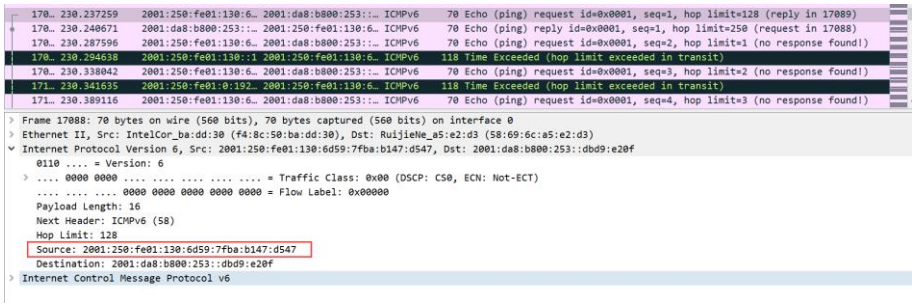
在你的捕获窗口中，应该能看到由你的主机发出的一系列ICMP Echo Request包

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。



思考下列问题：

a) 你主机的IP地址是什么？

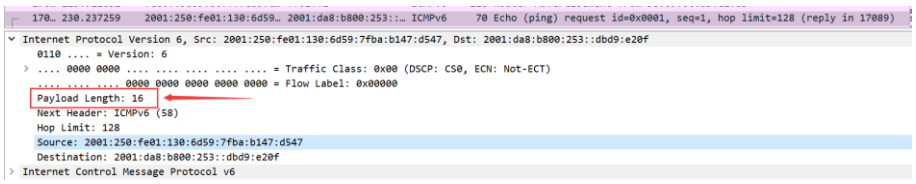


主机的IP地址为2001:250:fe01:130:6d59:7fba:b147:d547

b) 在IP数据包头中，上层协议（upper layer）字段的值是什么？

IPv6数据包没有上层协议upper layer

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？



IPv6头部有40字节；

该IP数据包净载为16字节；

如图。

d) 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

没有分片，因为IPv6数据包不允许分片，没有标识位，也没有片偏移。

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）

思考下列问题：

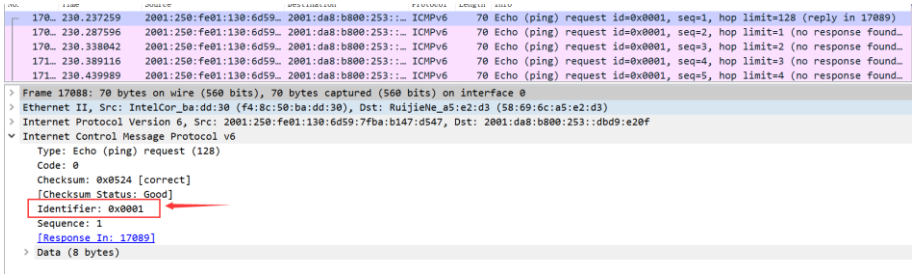
a) **你主机发出的一系列ICMP消息中IP数据报中哪些字段总是发生改变？**

Hop Limit字段

b) **哪些字段必须保持常量？哪些字段必须改变？为什么？**

源 IP 地址、版本号必须保持常量。因为ICMP消息从同一个主机端口发出下一个首部字段必须改变。因为Next Header的一致会使顺序发生冲突

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



16位(二字节)数，用十六进制表示

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

思考下列问题：

a) **Identification字段和TTL字段的值是什么？**

Identifier字段值为0x0001

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？**

保持不变

因为Identifier是从1开始记数的，而TTL是从255开始，每过一跳减1，因此定为255-1=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：

a) **该消息是否被分解成不止一个IP数据报？**

被分成两片。

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**

```
Fragment Header for IPv6
    Next header: ICMPv6 (58)
    Reserved octet: 0x00
    0000 0000 0000 0... = Offset: 0 (0 bytes)
    .... .... .... .00. = Reserved bits: 0
    .... .... .... ...1 = More Fragments: Yes
    Identification: 0x00004870
```

```
[2 IPv6 Fragments (1480 bytes): #903(1448), #902(32)]
  [Frame: 903, payload: 0-1447 (1448 bytes)]
  [Frame: 902, payload: 1448-1479 (32 bytes)]
  [Fragment count: 2]
```

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

分片的长度为 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。**

思考下列问题：

a) **原始数据包被分成了多少片？**

```
[3 IPv6 Fragments (3460 bytes): #886(1448), #887(1448), #888(564)]
  [Frame: 886, payload: 0-1447 (1448 bytes)]
  [Frame: 887, payload: 1448-2895 (1448 bytes)]
  [Frame: 888, payload: 2896-3459 (564 bytes)]
  [Fragment count: 3]
```

分成3片

b) **这些分片中IP数据报头部哪些字段发生了变化？**

```
Fragment Header for IPv6
    Next header: ICMPv6 (58)
    Reserved octet: 0x00
    0000 1011 0101 0... = Offset: 362 (2896 bytes)
    .... .... .... .00. = Reserved bits: 0
    .... .... .... ...0 = More Fragments: No
    Identification: 0x401e3640
```

偏移量字段、More Fragment字段发生了变化。

(五) 抓取ARP数据包

1) 利用MS-DOS命令查看主机APR缓存的内容

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

接口: 172.20.26.96 --- 0x8
  Internet 地址         物理地址              类型
  172.20.0.1            58-69-6c-a5-e2-d3     动态
  172.20.37.163         58-69-6c-a5-e2-d3     动态
  172.20.73.91          58-69-6c-a5-e2-d3     动态
  172.20.119.15         58-69-6c-a5-e2-d3     动态
  172.20.127.255        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
  239.11.20.1           01-00-5e-0b-14-01     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
```

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

```
C:\Users\ASUS>ping 192.34.56.23

正在 Ping 192.34.56.23 具有 32 字节的数据:
来自 192.34.56.23 的回复: 字节=32 时间=240ms TTL=46
来自 192.34.56.23 的回复: 字节=32 时间=240ms TTL=46
来自 192.34.56.23 的回复: 字节=32 时间=240ms TTL=46
来自 192.34.56.23 的回复: 字节=32 时间=239ms TTL=46

192.34.56.23 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 = 239ms, 最长 = 240ms, 平均 = 239ms
```

3) 启动Wireshark，开始分组俘获。

思考下面问题

a) 说明ARP缓存中每一列的含义是什么？

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）。

b) 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：

i. 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

由 9 部分构成：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）。

ii. 如何判断一个ARP数据是请求包还是应答包

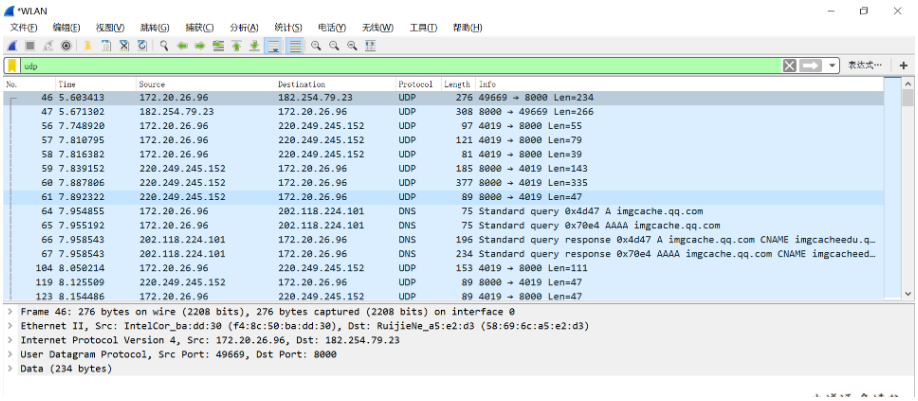
通过 OP 字段。当 OP 字段值为 0x0001 时是请求包，当 OP 字段值为 0x0002时是应答包。

iii. 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？

因为进行 ARP 查询时是不知道目的 IP 地址对应的 MAC 地址的，因此需要广播查询；而 发送ARP 响应报文时是知道查询主机的 MAC 地址（通过查询主机发出的查询报文获得），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。

(六) 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，如图所示。



分析QQ通讯中捕获到的UDP数据包。根据操作思考以下问题：

a) 消息是基于UDP的还是TCP的？

UDP

b) 你的主机ip地址是什么？目的主机ip地址是什么？

```
 56 7.748920    172.20.26.96      220.249.245.152   UDP      97 4019 → 8000 Len=55
 57 7.810795    172.20.26.96      220.249.245.152   UDP     121 4019 → 8000 Len=79
 58 7.816382    172.20.26.96      220.249.245.152   UDP      81 4019 → 8000 Len=39
```

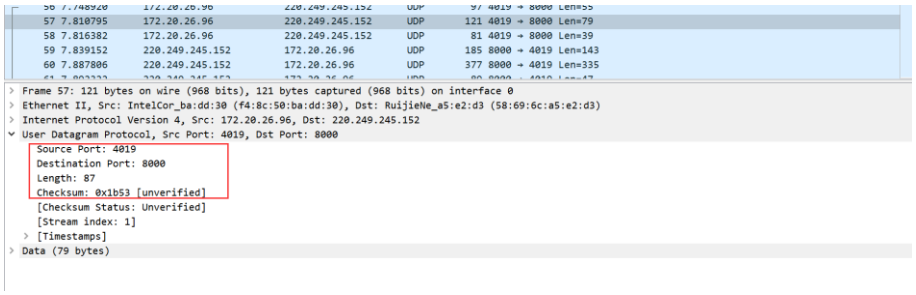
主机IP地址为172.20.26.96，目的主机IP地址为220.249.245.152

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？

```
 56 7.748920    172.20.26.96      220.249.245.152   UDP      97 4019 → 8000 Len=55
 57 7.810795    172.20.26.96      220.249.245.152   UDP     121 4019 → 8000 Len=79
 58 7.816382    172.20.26.96      220.249.245.152   UDP      81 4019 → 8000 Len=39
```

主机端口号为4019，QQ服务器端口号为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？



首部8个字节。源端口号2字节；目的端口号2字节；长度2字节；校验和2字节

e) 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？

因为服务器需返回接收的结果给客户端。

因为服务器只提供了一次返回的 ACK，所以不保证数据一定送达，即为不可靠传输；

对比前面的TCP可以看出，因为UDP 数据包没有序列号，不能像 TCP 协议那样先三次握手建立连接再发送数据，这样每次只发送一个数据报，然后等待服务器响应。

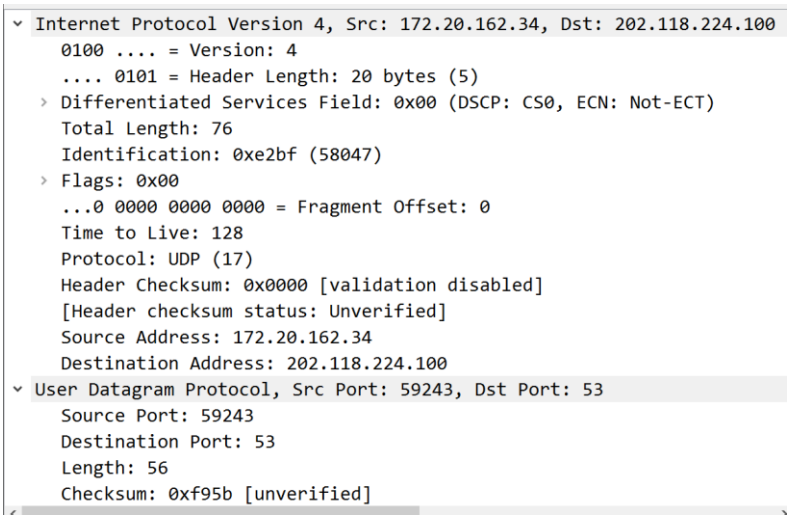
(七) 利用WireShark进行DNS协议分析

打开浏览器键入:www.baidu.com，并打开Wireshark进行抓包。

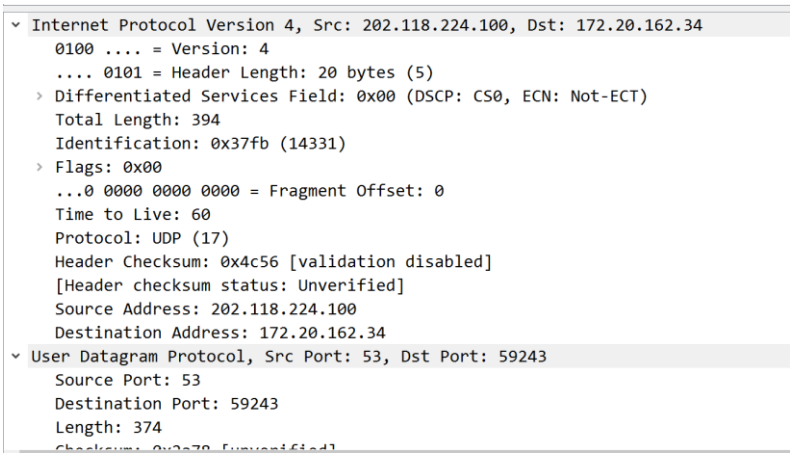
```
 14 2.363902    172.20.53.231     202.118.224.101   DNS      73 Standard query 0x5be4 A www.baidu.com
 15 2.364690    172.20.53.231     202.118.224.101   DNS      73 Standard query 0x3cdc AAAA www.baidu.com
 16 2.369292    202.118.224.101   172.20.53.231     DNS     132 Standard query response 0x5be4 A www.baidu.c
 17 2.369834    202.118.224.101   172.20.53.231     DNS     157 Standard query response 0x3cdc AAAA www.baid
115 2.549304    172.20.53.231     202.118.224.101   DNS      75 Standard query 0xced5 A b1.bdstatic.com
116 2.549304    172.20.53.231     202.118.224.101   DNS      75 Standard query 0x7f65 A s1.bdstatic.com
117 2.549447    172.20.53.231     202.118.224.101   DNS      72 Standard query 0x31e2 A t1.baidu.com
118 2.549676    172.20.53.231     202.118.224.101   DNS      72 Standard query 0x26be AAAA t1.baidu.com
119 2.549868    172.20.53.231     202.118.224.101   DNS      75 Standard query 0x03be AAAA b1.bdstatic.com
```

1. 查询的目的地址均为相同的202.118.224.101
2. 可以知道这是哈工大的dns服务器，经过ip地址查询确实这是一个来自哈尔滨市南岗区的教务网ip地址。

DNS查询报文：



DNS回复报文：



心得体会：

1. 虽然本次实验比较复杂琐碎，耗时比较长，学会了如何合理安排时间（虽然上学期计统和软构已经够恶心人了，计网显得非常和善）
2. HTTP 协议、TCP 协议、UDP 协议、IP 协议、DNS 协议等等的报文格式有了更深入的了解，明白了每一个标志位的含义与作用，深入地了解了协议交互的过程以及工作方式；
3. 另外，我更加理解了网络协议实体间进行交互及报文交换的大致流程与情形，也学习了简单的 Wireshark 工具的抓包方法和 PingPlotter 工具的使用。