<<计算机网络>>

实验报告

(2017 年度春季学期)

姓名:	赵浩宁
学号:	1140310226
学院:	计算机科学与技术学院
教师:	聂兰顺

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

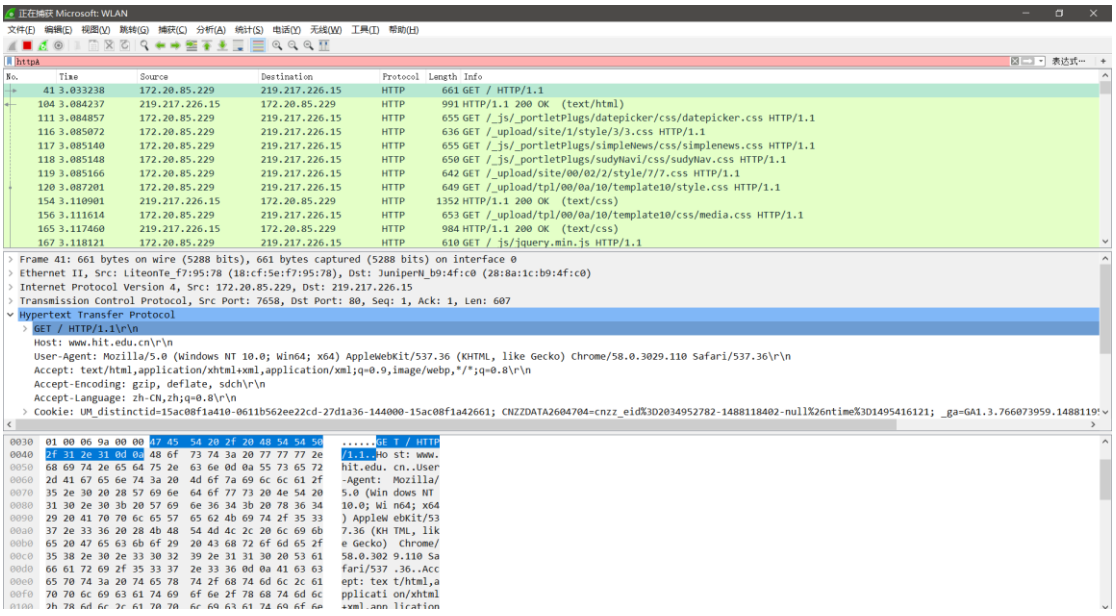
选做内容：

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

(一) HTTP 分析

1) HTTP GET/response 交互



问题：

- **"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？**

  浏览器和所访问的服务器都是 HTTP 1.1

- **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

```
Accept-Encoding: gzip, deflate, sdch\r\n
Accept-Language: zh-CN,zh;q=0.8\r\n
```

- **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？**

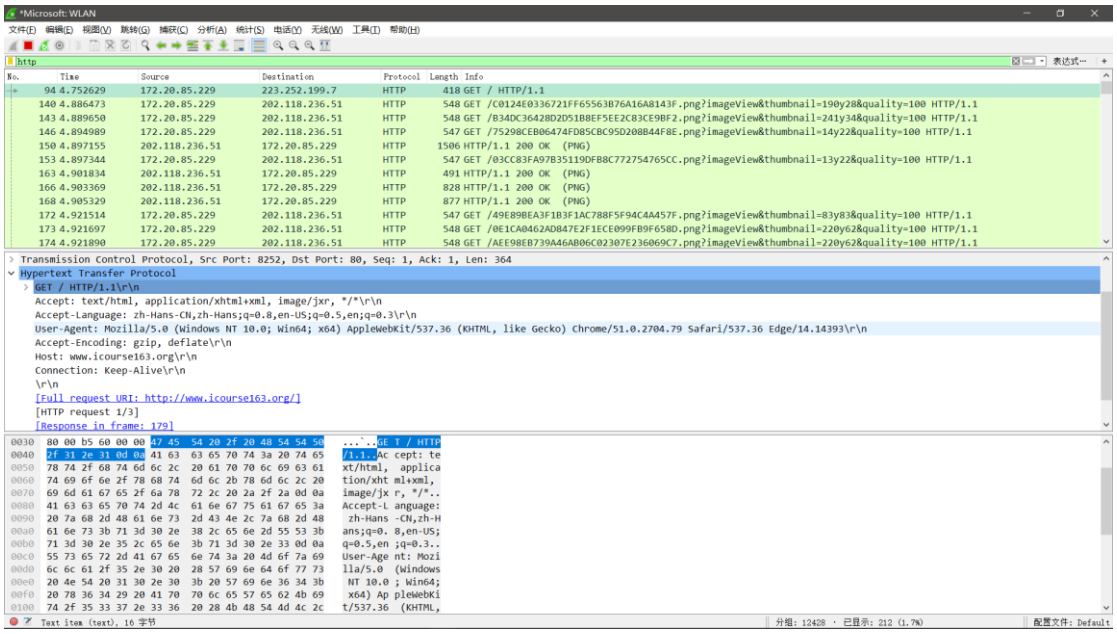
  本地计算机 IP：172.20.85.229

  服务器 IP：219.217.226.15

- **从服务器向你的浏览器返回的状态代码是多少？**

  200

2） HTTP GET/response 交互



- **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？**

  没有

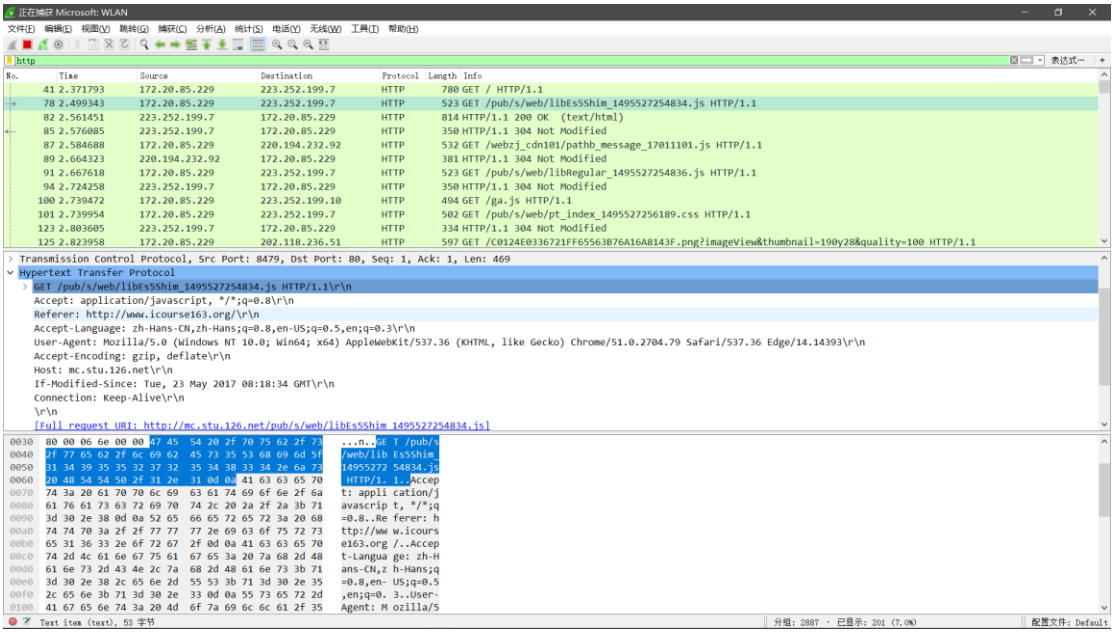
- **分析服务器响应报文的内容，服务器是否明确返回了文件的内容？如何获知？**

  服务器明确返回了内容，观察服务器返回的信息可以看到状态码

  HTTP Status Code 为 304 时不返回文件

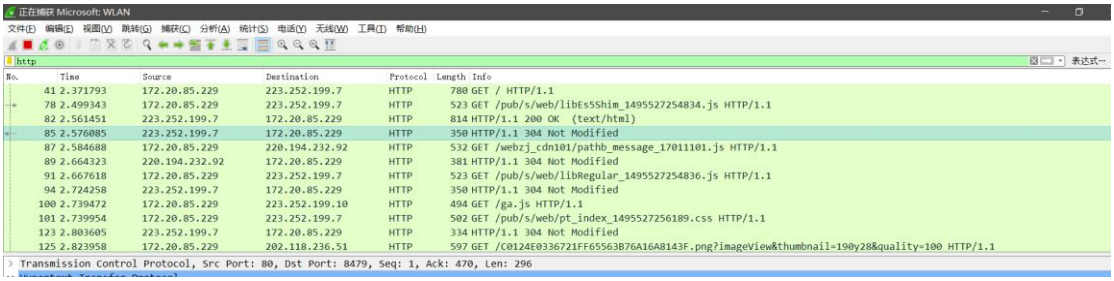
  HTTP Status Code 为 200 时返回文件

- **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，表示浏览器有缓存，后面代表时间，即服务器在这个时间点更新

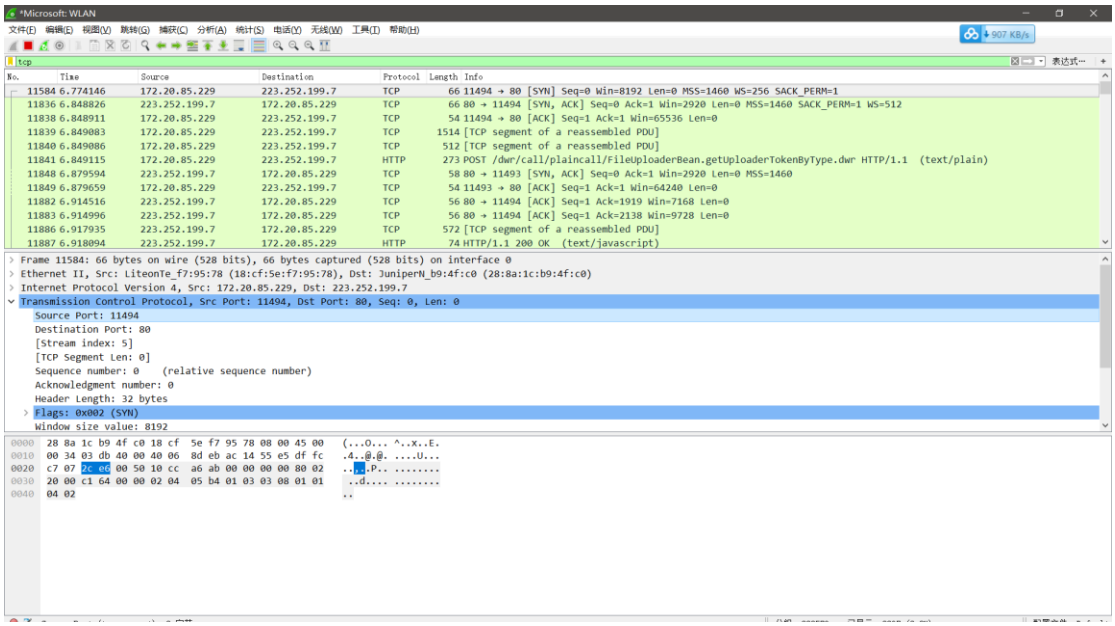
- **服务器对较晚的 HTTP GET 请求的响应中的 HTTP 状态代码是多 少？服务器是否明确返回了文件的内容？请解释。**



状态码是 304，不会返回明确文件，浏览器使用没有过期的缓存文件

(二) TCP 分析

报文捕获:



1)http://www.icourse163.org/learn/BIT-1001870001?tid=1001962001#/learn/hw?id=1002650045

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？

172.20.85.229

11494

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？

223.252.199.7

80

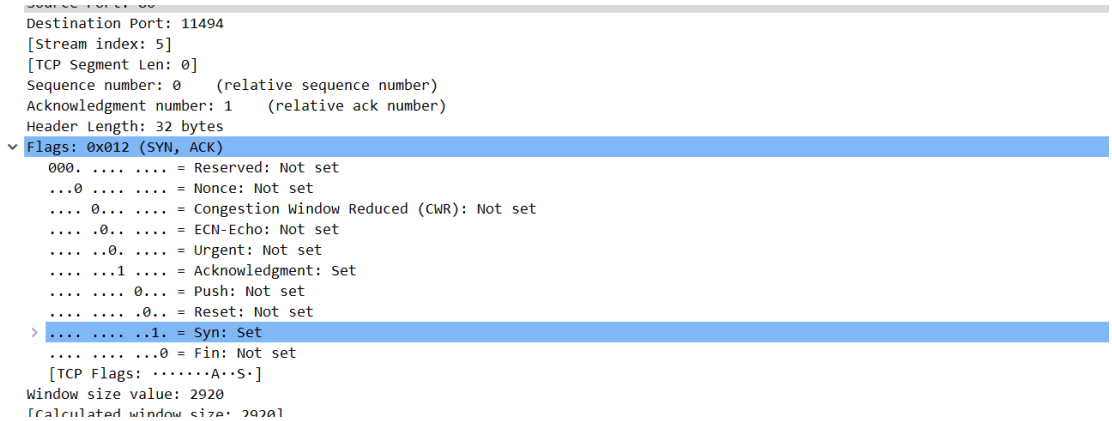
3)客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号（sequence number）是多少？在该报文段中，是用什么来标示该报文段是 SYN 报文段的？

```
Destination Port: 80
[Stream index: 5]
[TCP Segment Len: 0]
Sequence number: 0    (relative sequence number)
Acknowledgment number: 0
Header Length: 32 bytes
Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  > .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
    [TCP Flags: ··········S·]
Window size value: 8192
[Calculated window size: 8192]
Checksum: 0xc164 [unverified]
```

初始序号为 0，用 seq 来标识 SYN 段

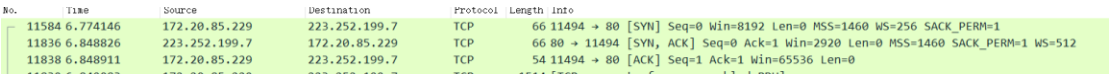
4) 服务器向客户端发送的 SYN ACK 报文段序号是多少？该报文段 中，Acknowledgement 字段的值是多少？Gaia.cs.umass.edu 服务器 是如何决定此值的？在该报文段中，是用什么来标示该报文段是

SYNACK 报文段的？



SYNACK 报文段的序号是 0，ACKnowledgement 序号是 1，Flags:0x012(SYN,ACK)来标识的

5）你能从捕获的数据包中分析出 tcp 三次握手过程吗？

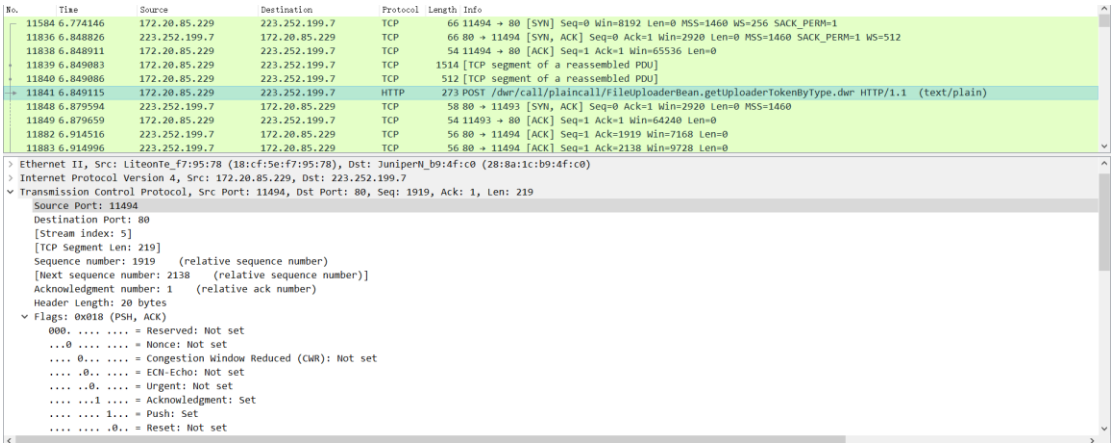


首先客户端向服务器发送 seq=0 的建立连接的请求

然后服务器向客户端返回 seq=0,ack=1 的响应，允许建立连接

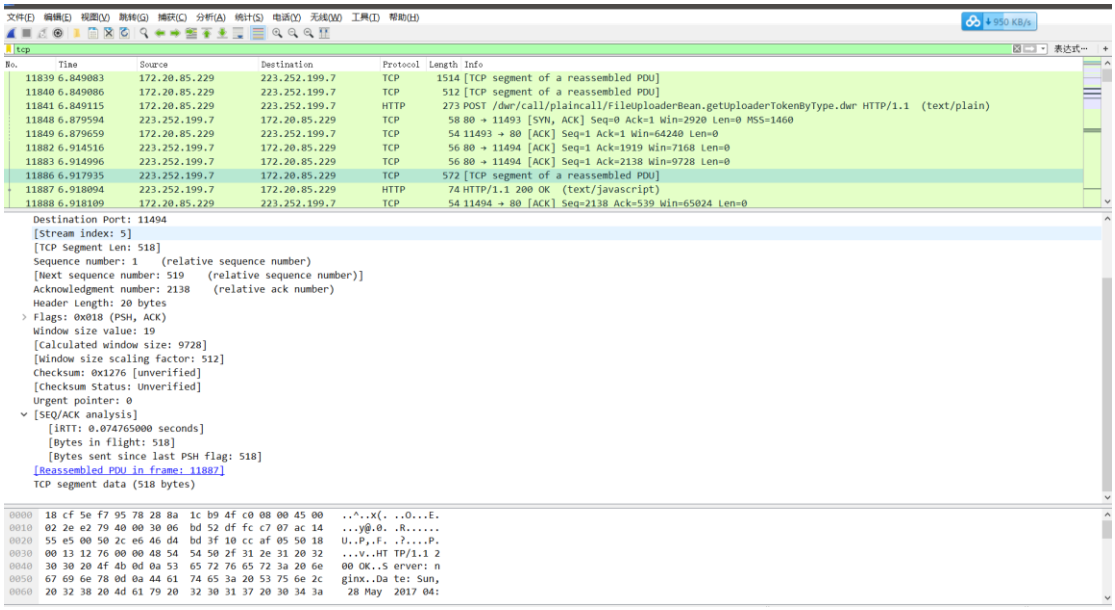
客户端收到响应，返回 seq=1,ack=1 的确认报文，连接建立

6)包含 HTTP POST 命令的 TCP 报文段的序号是多少？

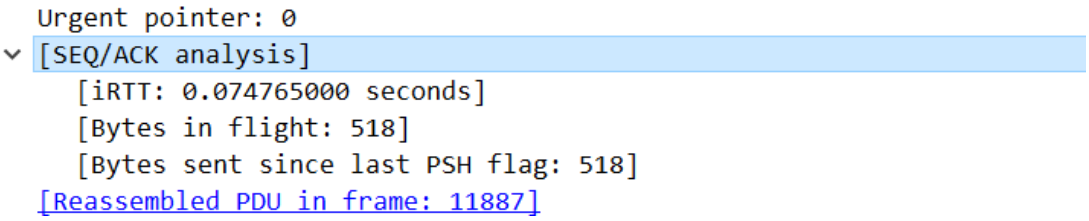


Seq=1919

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？



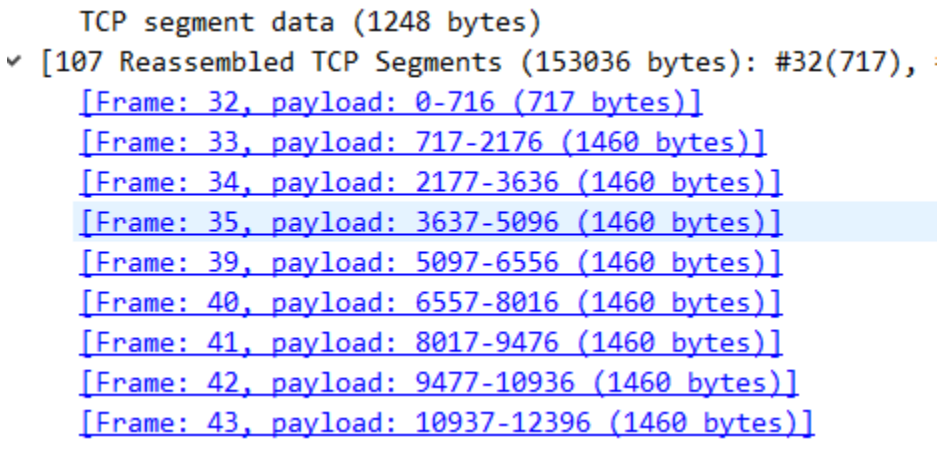
第六个报文段 Seq=1，发送时间 Arrival Time: May 28, 2017 12:58:22.013438000 中国标准时间



对应的 ack 经过 RTT 延迟后接受，时间为 May 28, 2017 12:58:22.088203 中国标准时间

8）前六个 TCP 报文段的长度各是多少？

717,1460,1460,1460,1460,1460



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？

```
    .... .... ...0 = Fin: Not set
    [TCP Flags: *******A****]
  Window size value: 513
  [Calculated window size: 131328]
  [Window size scaling factor: 256]
> Checksum: 0xde39 [validation disabled]
  Urgent pointer: 0
v [SEQ/ACK analysis]
```

如图，接收端公示的最小的可用缓存空间是 513，限制发送的传输后，接收端缓存仍然够用

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，因为客户端发送的报文序列号没有出现重复。

```
Transmission Control Protocol, Src Port: 14499 (14499), Dst Pc
[109 Reassembled TCP Segments (152871 bytes): #7(1460), #8(146
   [Frame: 7, payload: 0-1459 (1460 bytes)]
   [Frame: 8, payload: 1460-2919 (1460 bytes)]
   [Frame: 9, payload: 2920-4379 (1460 bytes)]
   [Frame: 10, payload: 4380-5839 (1460 bytes)]
   [Frame: 14, payload: 5840-7299 (1460 bytes)]
```

由图可知，发送数据总的长度为 152871B+109*54B=158757B

发送时间间隔约为 1.673847s

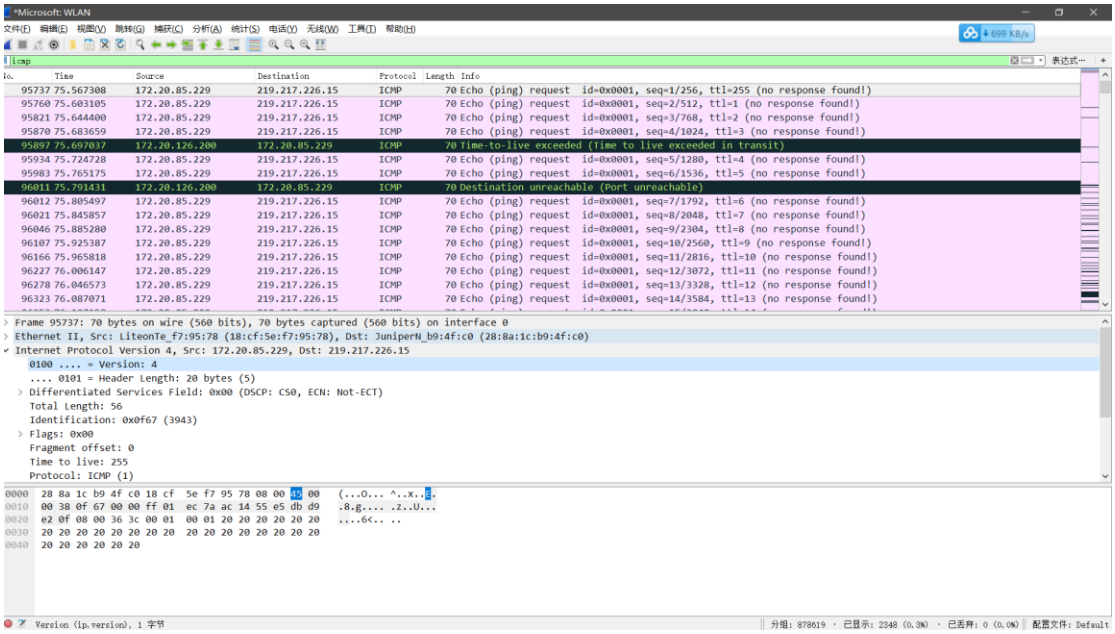
因此吞吐量为 158757B/1.673847s=94845.59bps

（三）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:172.20.85.229

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

```
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 56
  Identification: 0x0f67 (3943)
> Flags: 0x00
  Fragment offset: 0
  Time to live: 255
  Protocol: ICMP (1)
  Header checksum: 0xec7a [validation disabled]
  [Header checksum status: Unverified]
  Source: 172.20.85.229
  Destination: 219.217.226.15
  [Source GeoIP: Unknown]
```

上层协议字段的值是 01。

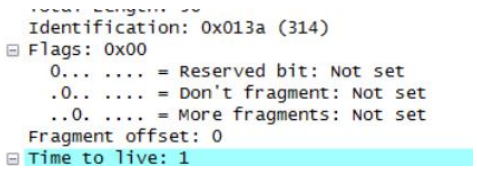
3)IP 头有多少字节？该 IP 数据包的净载为多少字节？并解释你是怎样确定该 IP 数据包的净载大小的？

```
Internet Protocol Version 4, Src: 192.168.4.26, Dst: 219.217.226.15
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes
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: 56
  Identification: 0x7a63 (31331)
> Flags: 0x00
  Fragment offset: 0
> Time to live: 1
  Protocol: ICMP (1)
```

IP 头有 20 字节。

IP 包的净载为 Total Length-Header Length=56B-20B=36B

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，分片位移为 0，More fragments 为 0 表示后面无分片。

（四）ICMP 分析

2.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）

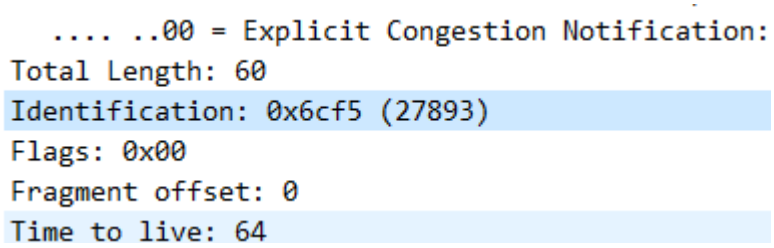
1)你主机发出的一系列 ICMP 消息中 IP 数据报中哪些字段总是发生改变？

查看多个 ICMP 消息，发现 ID、TTL、Header checksum 这三个字段总在变化。

2)哪些字段必须保持常量？哪些字段必须改变？为什么？

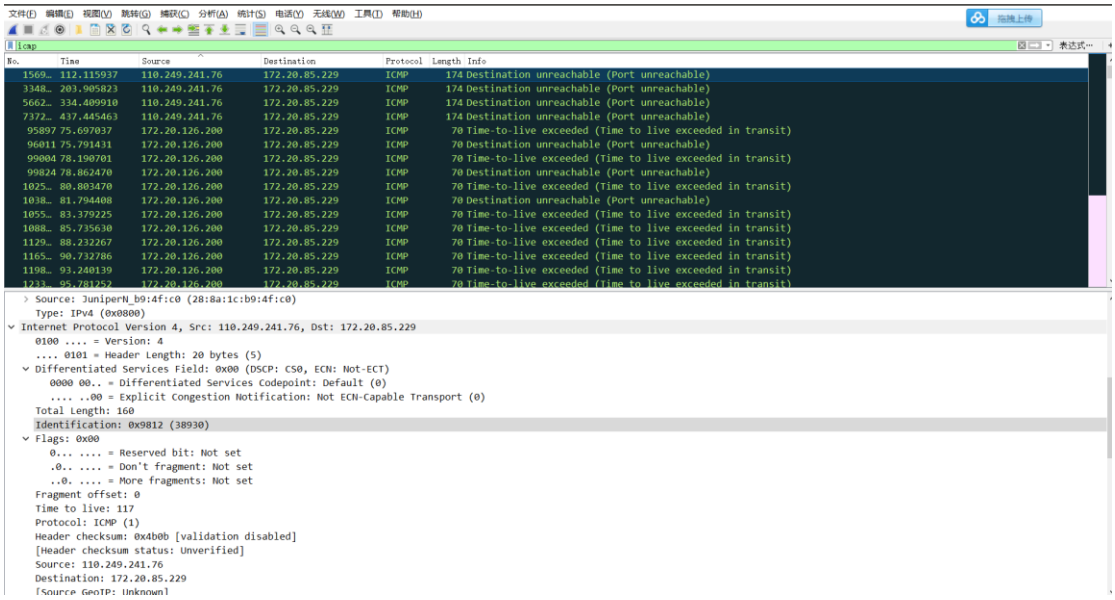
ID,TTL,头部段校验和字段会发生变化，目的 IP 地址和源 IP 地址不会发生变化。ID 数据包 ID；TTL 在经过路由器的时候需要减一，头部发生变化，因此校验和不一样。源主机和目的主机未变化，因此源 IP 目的 IP 不发生变化

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



16 位，在某一范围内是+1 递增的。

（ 3）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。



Time to live :117

Identification:0x9812

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 是标识符，TTL 是生存时间，

ICMP Time-to-live exceeded 不变， IP 是无连接服务，相同的标识是为了分段后组装成同一段，给同一个主机返回的 ICMP，

标识不代表序号，TTL 消息是相同的，因此 Identification 不变;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。

（ 4）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

```
    .... 0101 = Header Length: 20 bytes
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x7ac0 (31424)
  > Flags: 0x01 (More Fragments)
    Fragment offset: 0
    Time to live: 255
    Protocol: ICMP (1)
  > Header checksum: 0x98b4 [validation disabled]
```

```
    Protocol: ICMP (1)
  > Header checksum: 0xbbcf [validation disabled]
    Source: 192.168.4.26
    Destination: 219.217.226.15
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
  v [2 IPv4 Fragments (1980 bytes): #1021(1480), #1022(500)]
      [Frame: 1021, payload: 0-1479 (1480 bytes)]
      [Frame: 1022, payload: 1480-1979 (500 bytes)]
      [Fragment count: 2]
      [Reassembled IPv4 length: 1980]
000  44 94 fc 9c 2f e3 6c 71  d9 82 7c c3 08 00 45 00   D.../.lq ..|...E.
```

消息被分片了，分成了两片

```
Flags: 0x01 (More Fragments)
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..1. .... = More fragments: Set
```

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？

```
  [Destination GeoIP: Unknown]
  [3 IPv4 Fragments (3480 bytes): #267(1480), #268(1480), #269(520)]
    [Frame: 267, payload: 0-1479 (1480 bytes)]
    [Frame: 268, payload: 1480-2959 (1480 bytes)]
    [Frame: 269, payload: 2960-3479 (520 bytes)]
    [Fragment count: 3]
    [Reassembled IPv4 length: 3480]
```

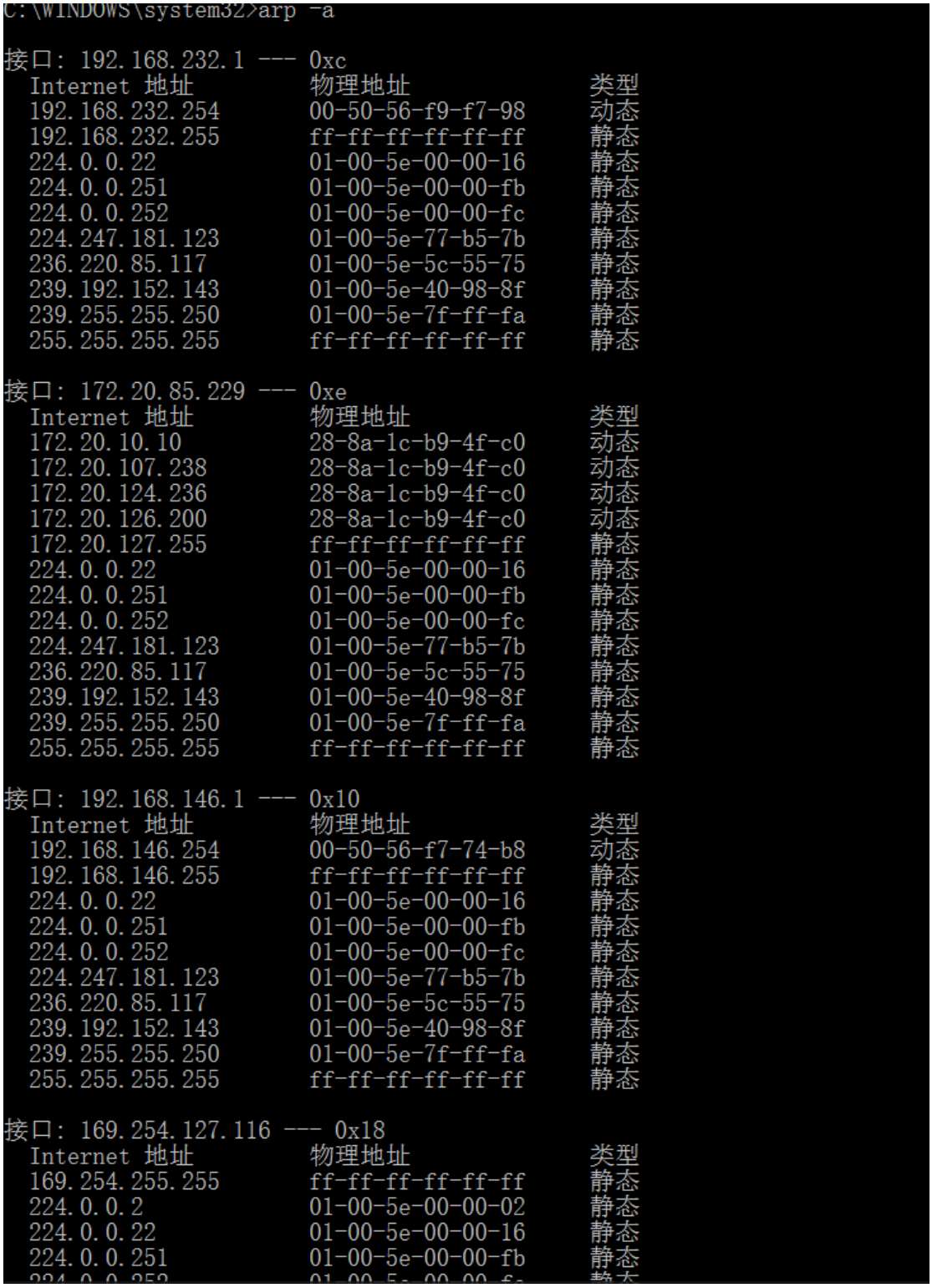
数据分为三片

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

（五）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：



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：

①ARP 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？

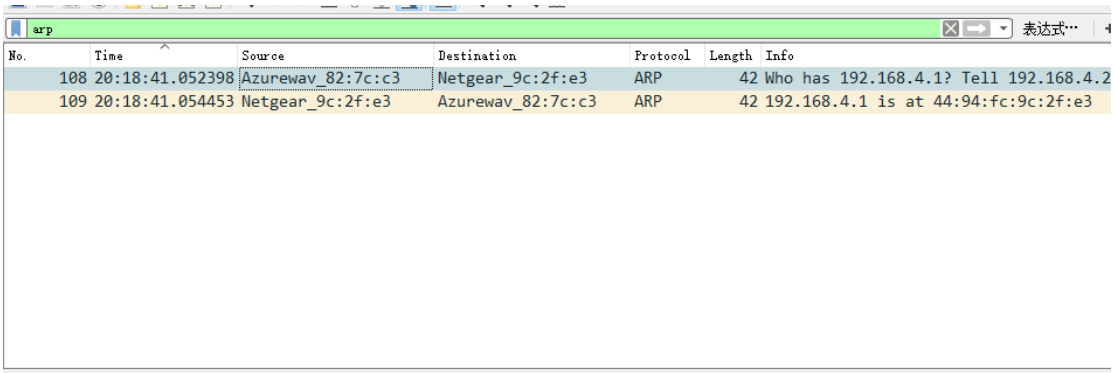
ARP 数据包格式如下图：

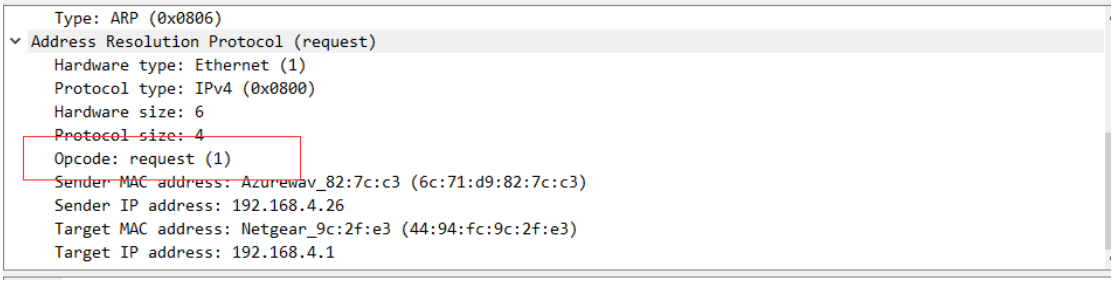
由 9 部分构成，分别是硬件类型（2 字节），协议类型（2 字节），硬件地址长度（1 字节），协议地址长度（1 字节），OP（2 字节），发送端 MAC 地址（6 字节），发送端 IP 地址（4 字节），目的 MAC 地址（6 字节），目的 IP 地址（4 字节）。

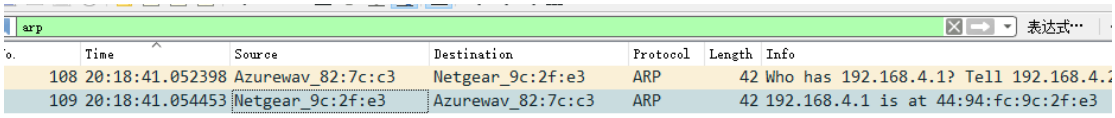
截取的一个 ARP 数据包如下：

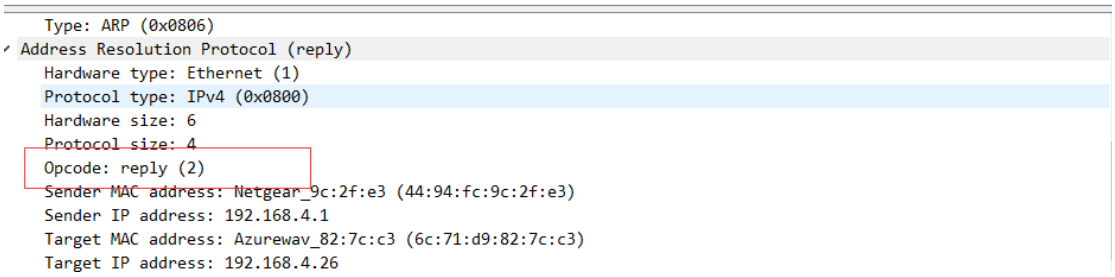
```
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 50:78:1c:14:8b:a6 (50:78:1c:14:8b:a6)
    Sender IP address: 192.168.1.254 (192.168.1.254)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.134 (192.168.1.134)
```

②如何判断一个 ARP 数据是请求包还是应答包？









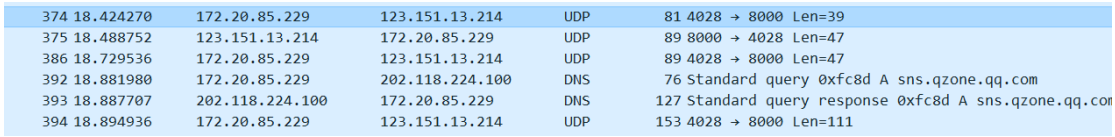
通过 OP 字段。当 OP 字段值为 0x0001 时是请求包，当 OP 字段值为 0x0002 时是应答包。

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？

因为进行 ARP 查询时并不知道目的 IP 地址对应的 MAC 地址，所以需要广播查询；而 ARP 响应报文知道查询主机的 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消息是基于 UDP 的还是 TCP 的？



消息是基于 UDP

②你的主机 IP 地址是什么？目的主机 IP 地址是什么？

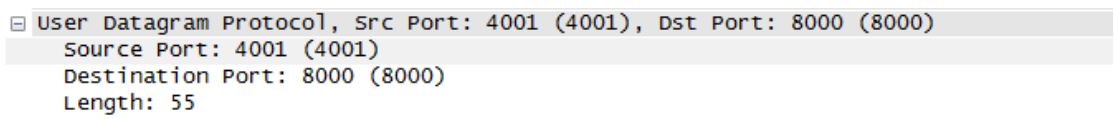
我的主机 IP 地址：172.20.85.229

目的主机 IP 地址：123.151.13.214

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？

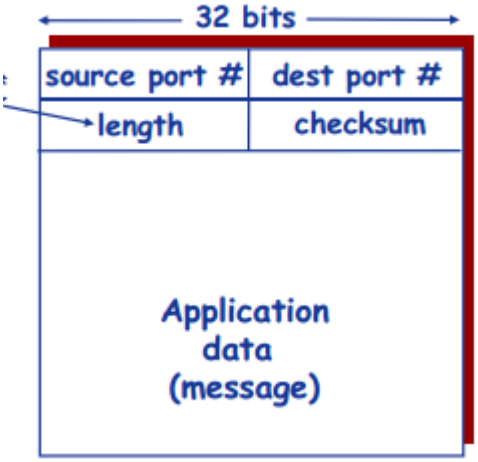
发送 QQ 消息端口号：4001

QQ 服务器端口号：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？

UDP 数据报格式如下图：



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 GeoIP: Unknown]
    [Destination GeoIP: Unknown]
v User Datagram Protocol, Src Port: 8000 (8000), Dst Port: 4010 (4010)
    Source Port: 8000
    Destination Port: 4010
    Length: 39
  > Checksum: 0x1b6e [validation disabled]
    [Stream index: 0]
> Data (31 bytes)
```

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？

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。

可以看出。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，因为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

① 打开浏览器，输入 www.baidu.com,DNS 查询消息如下图：

80	3.671599	172.20.85.229	202.118.224.100	DNS	73	Standard query 0xbc07 A www.baidu.com
81	3.671779	172.20.85.229	202.118.224.100	DNS	76	Standard query 0xa5c6 A ss1.bdstatic.com
82	3.672161	172.20.85.229	202.118.224.100	DNS	76	Standard query 0x4744 A ss3.bdstatic.com
83	3.692507	202.118.224.100	172.20.85.229	DNS	126	Standard query response 0x4744 A ss3.bdstatic.com CNAME sslbdstatic.jomodns.com A
84	3.693108	202.118.224.100	172.20.85.229	DNS	132	Standard query response 0xbc07 A www.baidu.com CNAME www.a.shifen.com A 119.75.217
85	3.693110	202.118.224.100	172.20.85.229	DNS	126	Standard query response 0xa5c6 A ss1.bdstatic.com CNAME sslbdstatic.jomodns.com A
166	5.421509	172.20.85.229	202.118.224.100	DNS	73	Standard query 0x8391 A ss0.baidu.com
167	5.421510	172.20.85.229	202.118.224.100	DNS	73	Standard query 0x07ff A sp1.baidu.com

② 我的电脑 IP 地址：172.20.85.229，本地域名服务器 IP 地址：202.118.224.100.如图：

```
⊞ Header checksum: 0x0000 [validation disabled]
  Source: 192.168.1.41 (192.168.1.41)
  Destination: 202.118.224.100 (202.118.224.100)
```

③ UDP 报文的源端口号 54061，目的端口号 53

```
> Internet Protocol Version 4, Src: 172.20.85.229, Dst: 202
v User Datagram Protocol, Src Port: 54061, Dst Port: 53
    Source Port: 54061
    Destination Port: 53
    Length: 39
    Checksum: 0x0760 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 9]
> Domain Name System (query)
```

④ DNS 查询报文内容如下图，表示查询主机域名为 www.baidu.com 的主机的 IP 地址

```
v 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: 59
  Identification: 0x0df8 (3576)
v Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 64
  Protocol: UDP (17)
  Header checksum: 0xbfe5 [validation disabled]
  [Header checksum status: Unverified]
  Source: 172.20.85.229
  Destination: 202.118.224.100
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
```

⑤ DNS 回复信息：

```
83 3.692507  202.118.224.100  172.20.85.229  DNS  126 Standard query response 0x4744 A ss3.bdstatic.com CNAME sslbdstatic.jomodns.com A 222.199.191.32
84 3.693108  202.118.224.100  172.20.85.229  DNS  132 Standard query response 0xbc07 A www.baidu.com CNAME www.a.shifen.com A 119.75.217.109 A 119.75.218.70
85 3.693110  202.118.224.100  172.20.85.229  DNS  126 Standard query response 0xa5c6 A ss1.bdstatic.com CNAME sslbdstatic.jomodns.com A 222.199.191.32
```

主机域名为 www.baidu.com 的主机 IP 地址为：222.118.224.100

```
> Frame 84: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface 0
> Ethernet II, Src: JuniperN_b9:4f:c0 (28:8a:1c:b9:4f:c0), Dst: LiteonTe_f7:95:78 (18:cf:5e:f7:95:
v Internet Protocol Version 4, Src: 202.118.224.100, Dst: 172.20.85.229
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
v   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: 118
    Identification: 0x3d76 (15734)
v   Flags: 0x00
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 61
    Protocol: UDP (17)
    Header checksum: 0x932c [validation disabled]
    [Header checksum status: Unverified]
    Source: 202.118.224.100
    Destination: 172.20.85.229
    [Source GeoIP: Unknown]
```

四、实验心得

通过本次实验，熟悉了 Wireshark 的基本操作，了解对 HTTP,TCP,IP,UDP,ARP,DNS 各种协议的分析，验证了课程中所教授的课程知识，对协议中的报文的格式有了更深的认识，对其各个部分的作用有了一定的了解。对网络中，各个协议的工作方式，能够通过 wirdshark 进行分析，并得到各种报文的格式。深入理解了报文在网络中发送接受及转发的过程。此次实验综合所学的协议，加深了对协议的认识