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	李旻翀		院系	计算机科学与技术	
班级	1903103		学号	1190200208	
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	2021.11.21	
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：

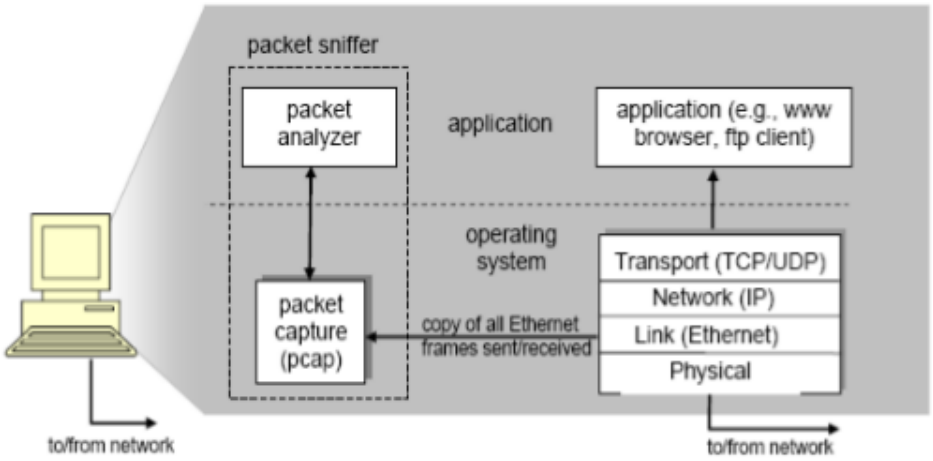
a) 利用 Wireshark 分析 DNS 协议
b) 利用 Wireshark 分析 UDP 协议
c) 利用 Wireshark 分析 ARP 协议

实验过程：

1. 了解实验相关基础知识

1) 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。



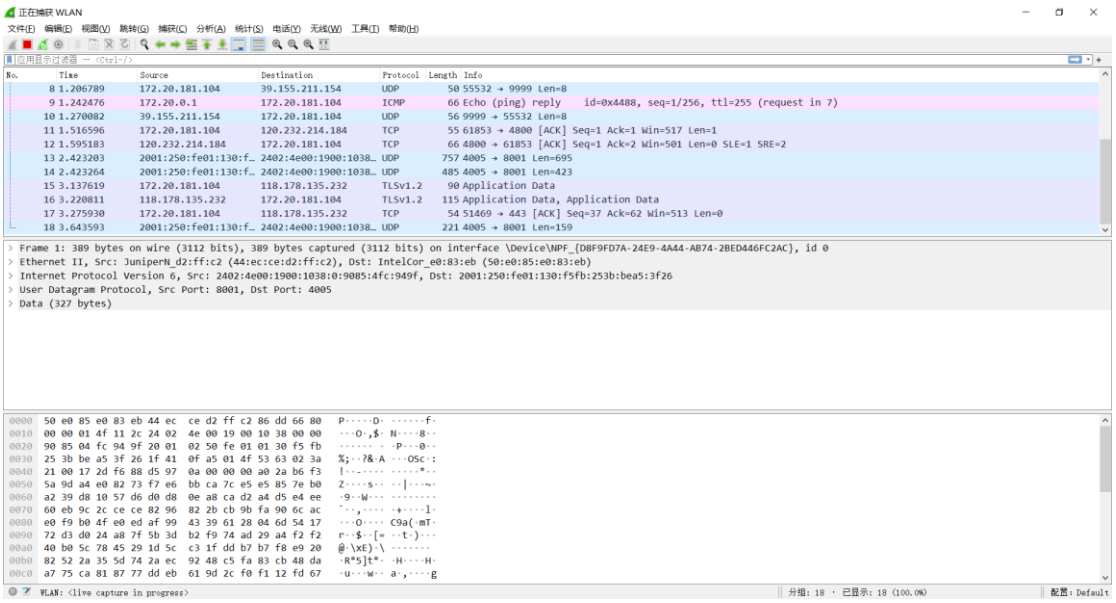
上图右边是计算机上正常运行的协议（在这里是因特网协议）和应用程序（如：web 浏览器和 ftp 客户端）。分组嗅探器（虚线框中的部分）是附加计算机普通软件上的，主要有两部分组成。分组俘获库（packetcapture library）接收计算机发送和接收的每一个链路层帧的拷贝。高层协议（如：HTTP、FTP、TCP、UDP、DNS、IP 等）交换的报文都被封装在链路层帧中，并沿着物理媒体（如以太网的电缆）传输。图 1 假设所使用的物理媒体是以太网，上层协议的报文最终封装在以太网帧中。

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 6-1 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

Wireshark 是一种可以运行在 Windows, UNIX, Linux 等操作系统上的分组分 析器。运行 Wireshark 程序时，其图形用户界面如图所示：



实验结果：

采用演示截图、文字说明等方式，给出本次实验的实验结果。

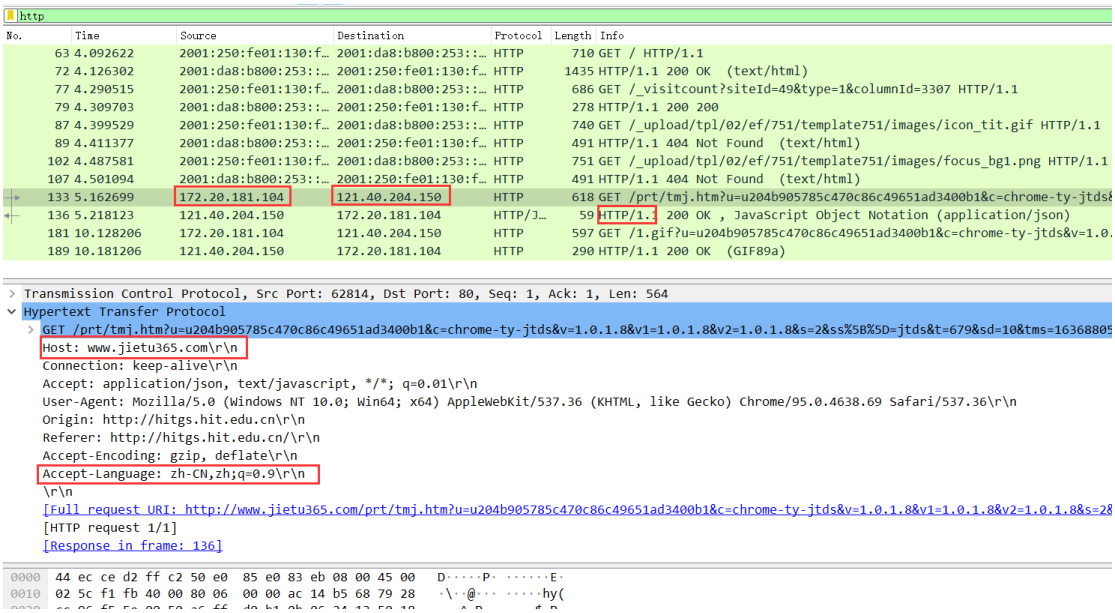
1. HTTP分析 - HTTP GET/response交互

由于实验指导书上的示例网站 http://hitgs.hit.edu.cn/news 无法打开，因此，在本报告中以 http://hitgs.hit.edu.cn/ 代替。

按照实验指导书上流程进行操作：

- ✧ 启动 Web browser，然后启动 Wireshark 分组嗅探器。在窗口的显示过滤说明处输入“http”，分组列表子窗口中将只显示所俘获到的HTTP 报文。
- ✧ 开始 Wireshark 分组俘获。
- ✧ 在打开的Web browser窗口中输入以下地址: http://hitgs.hit.edu.cn/
- ✧ 停止分组俘获

结果如下：



回答以下问题：

✧ 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？

答：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？

答：zh-CN, zh （即中文）

✧ 你的计算机的 IP 地址是多少？服务器 http://hitgs.hit.edu.cn/ 的 IP 地址是多少？

答：本机IP：172.20.181.104　　服务器IP：121.40.204.150

✧ 从服务器向你的浏览器返回的状态代码是多少？

答：200 OK

2. HTTP分析 - HTTP条件GET/response交互

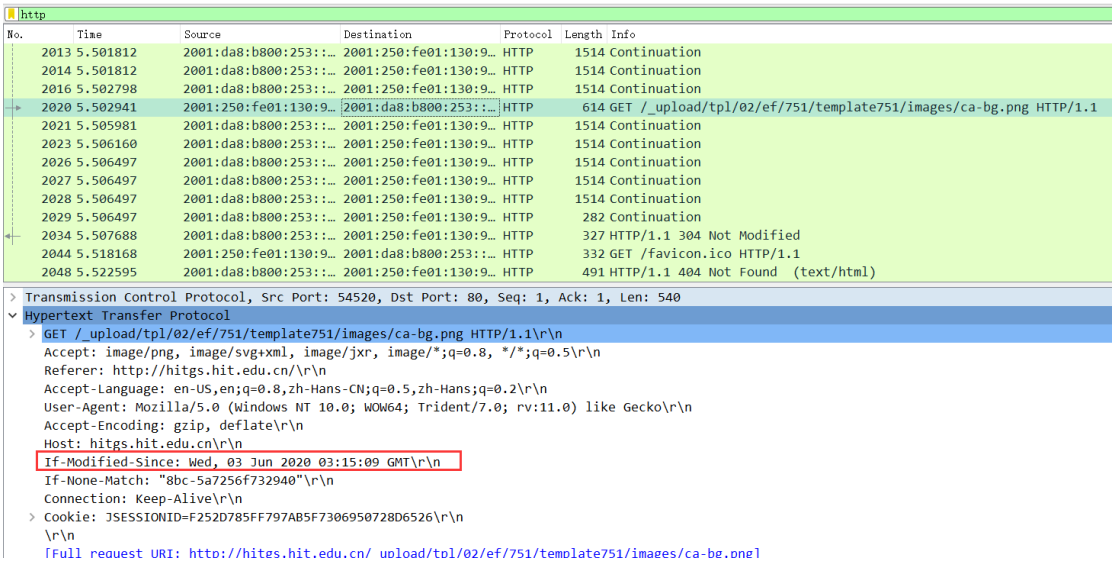
按照实验指导书上流程进行操作：

✧ 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”）。

✧ 启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。

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

✧ 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。



结果如下：

✧ 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：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？
答： 服务器明确返回了文件内容，若返回的状态码是200，代表明确返回了文件；若返回状态码为404，则不返回文件

✧ 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？
答：有。在该首部行后面跟着的信息是Wed, 03 Jun 2020 03:15:09 GMT\r\n，代表着缓存最后更新的时间。

✧ 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：服务器对较晚的 HTTP GET 请求的响应中的 HTTP 状态代码是 304 Not Modified。服务器不会明确返回文件内容，因为服务器判断的结果为 Not Modified，在此情况下，客户端可以使用本地仍为最新版本的缓存文件。

3. TCP分析

按照实验指导书上流程进行：

A. 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。

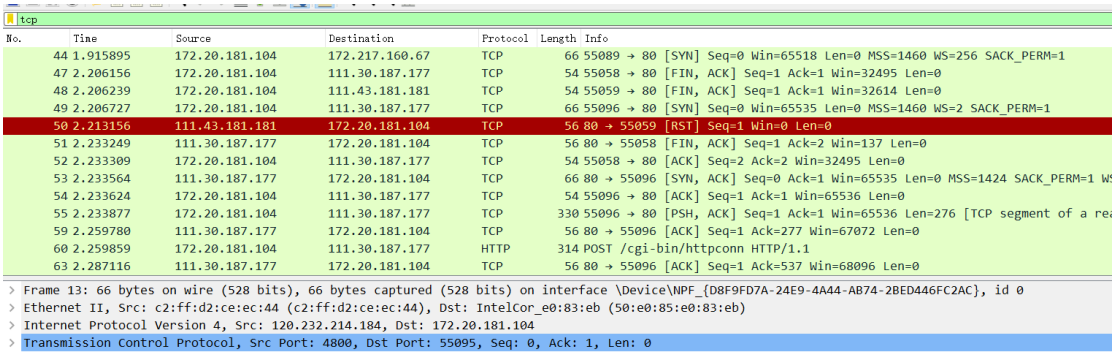
（2） 打开http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html ，在Browse按钮旁的文本框中输入保存在你的主机上的文件 ALICE'S ADVENTURES

INWONDERLAND的全名（含路径），此时不要按“Upload alice.txt file”按钮。

（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。



B. 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：

- 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：客户端主机的 IP 地址：172.20.181.104，TCP 端口号：55089

- 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？

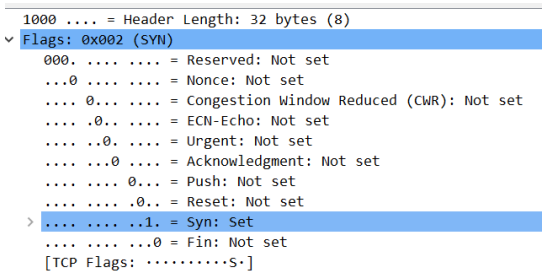
  答：服务器的 IP 地址：111.30.187.177，用来发送和接收 TCP 报文的端口号：80

C. TCP 基础

根据操作思考以下问题：

- 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？

  答：如图所示：



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位为1，来表示该报文段是 SYN 报文段。

- 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？

答：如图所示：

```
Sequence Number: 0    (relative sequence number)
Sequence Number (raw): 3347109514
[Next Sequence Number: 1    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 3504967002
1000 .... = Header Length: 32 bytes (8)
Flags: 0x012 (SYN, ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): 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: 65535
```

SYNACK 报文段序号是0；

Acknowledgement 字段的值是1；

Gaia.cs.umass.edu 服务器根据上一次客户端发给服务器的 seq+1 得到该字段；

在该报文段中，通过Flags位中SYN与ACK均为1来标示该报文段是SYNACK。

- 你能从捕获的数据包中分析出 tcp 三次握手过程吗？

答：如图所示：

44	1.915895	172.20.181.104	172.217.160.67	TCP	66 55089 → 80 [SYN] Seq=0 Win=65518 Len=0 MSS=1460 WS=256 SACK_PERM=1
47	2.206156	172.20.181.104	111.30.187.177	TCP	54 55058 → 80 [FIN, ACK] Seq=1 Ack=1 Win=32495 Len=0
48	2.206239	172.20.181.104	111.43.181.181	TCP	54 55059 → 80 [FIN, ACK] Seq=1 Ack=1 Win=32614 Len=0
49	2.206727	172.20.181.104	111.30.187.177	TCP	66 55096 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2 SACK_PERM=1
50	2.213156	111.43.181.181	172.20.181.104	TCP	56 80 → 55059 [RST] Seq=1 Win=0 Len=0
51	2.233249	111.30.187.177	172.20.181.104	TCP	56 80 → 55058 [FIN, ACK] Seq=1 Ack=2 Win=137 Len=0
52	2.233309	172.20.181.104	111.30.187.177	TCP	54 55058 → 80 [ACK] Seq=2 Ack=2 Win=32495 Len=0
53	2.233564	111.30.187.177	172.20.181.104	TCP	66 80 → 55096 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1424 SACK_PERM=1 WS=512
54	2.233624	172.20.181.104	111.30.187.177	TCP	54 55096 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0

客户端先向服务器发送一个 seq = 0 的建立连接请求，然后服务器向客户端返回 seq = 0, ack = 1 的响应。

- 包含 HTTP POST 命令的 TCP 报文段的序号是多少？

答：277，如图所示：

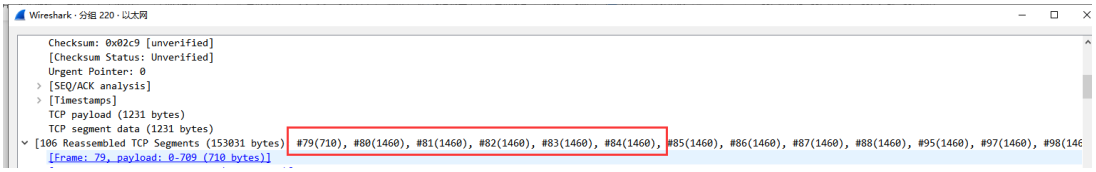
60	2.259859	172.20.181.104	111.30.187.177	HTTP	314 POST /cgi-bin/httpconn HTTP/1.1
63	2.287116	111.30.187.177	172.20.181.104	TCP	56 80 → 55096 [ACK] Seq=1 Ack=537

```
Destination Port: 80
[Stream index: 7]
[TCP Segment Len: 260]
Sequence Number: 277    (relative sequence number)
Sequence Number (raw): 3504967278
[Next Sequence Number: 537    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 3347109515
0101 .... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
```

- 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

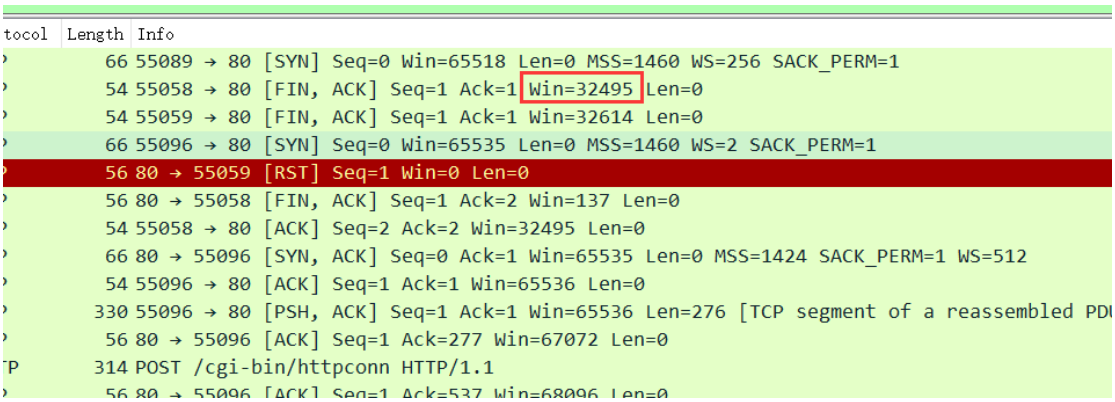
文段所对应的 ACK 是何时接收的？
答：第六个报文段为23，在 HTTP POST 发送之前，TCP连接建立之后发送。对应的 ACK 即为服务器返回的第六个 ACK。

- 前六个 TCP 报文段的长度各是多少？
答：如图：



长度分别为710,1460,1460,1460,1460,1460

- 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：接收端公示的最小的可用缓存空间是32495，且窗口大小整体递增，并未出现不够用的情况。



- 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：没有出现重传，因为客户端发送的报文序列号没有出现重复的情况。

- TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。
答： 发送数据总的长度为152741B + 108 x 54B = 158573B
发送时间间隔约为1.72s
因此吞吐量为158573B / 1.72s = 92193.6 Bps

4. IP分析

按照实验指导书上流程进行：

A. 通过执行 traceroute 执行捕获数据包

（1）启动 Wireshark 并开始数据包捕获

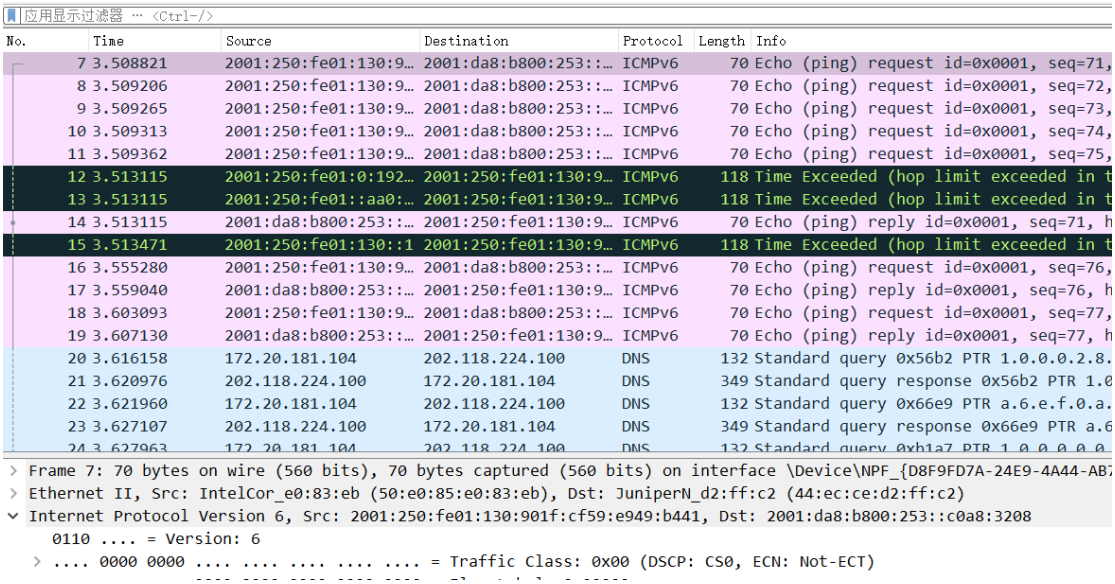
（2）启动pingplotter并“Address to Trace Window”域中输入目的地址。在“# of times to Trace”域中输入“3”，这样就不过采集过多的数据。

Edit->Options->Packet，将 Packet Size(in bytes,default=56)域设为 56，这样将发送一系列大小为 56 字节的包。然后按下“Trace”按钮。

（3）Edit->Options->Packet，然后将 Packet Size(in bytes,default=56)域改为 2000，这样将发送一系列大小为 2000 字节的包。然后按下“Resume”按钮。

（4）最后，将 Packet Size(in bytes,default=56)域改为 3500，发送一系列大小为 3500 字节的包。然后按下“Resume”按钮。

（5）停止 Wireshark 的分组捕获。



B. 对捕获的数据包进行分析

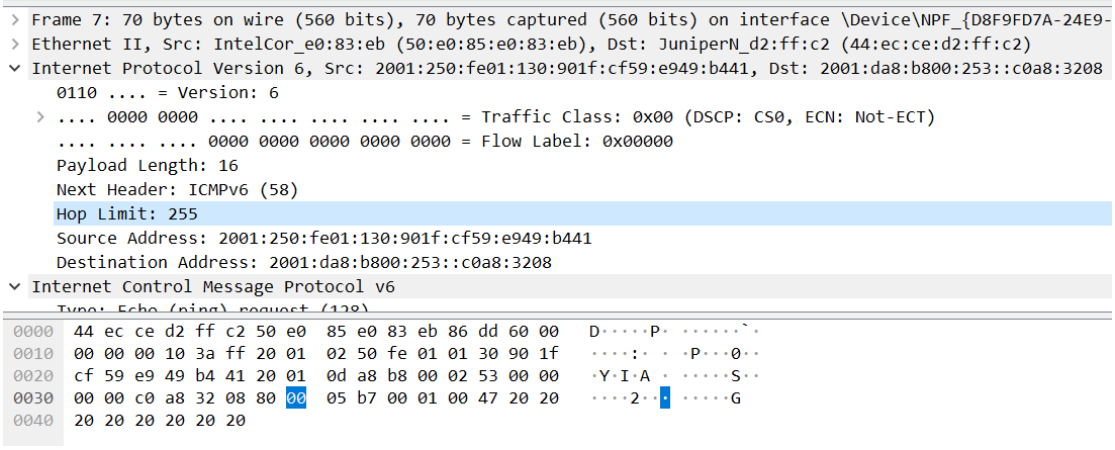
（1）在你的捕获窗口中，应该能看到由你的主机发出的一系列ICMPEcho Request包和中间路由器返回的一系列ICMP TTL-exceeded消息。选择第一个你的主机发出的ICMP Echo Request消息，在packet details窗口展开数据包的Internet Protocol部分，思考下列问题：

- 你主机的IP地址是什么？

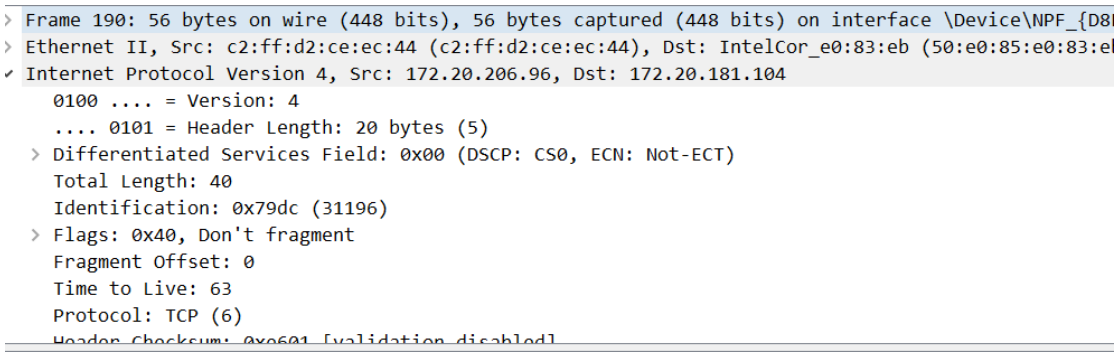
  答：172.20.181.104

- 在IP数据包头中，上层协议（upper layer）字段的值是什么？

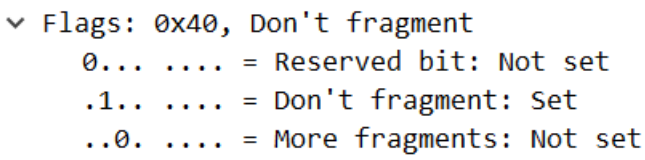
  答：01，如图所示



- 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：如图所示，IP头有20字节，数据报净载Total Length-Header Length=40B-20B = 20B



- 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？
  答：有分片，因为有分片的偏移量，如图所示：



（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）
思考下列问题：

- 你主机发出的一系列ICMP消息中IP数据报中哪些字段总是发生改变？
  答：Time to Live、Identification、Header Checksum

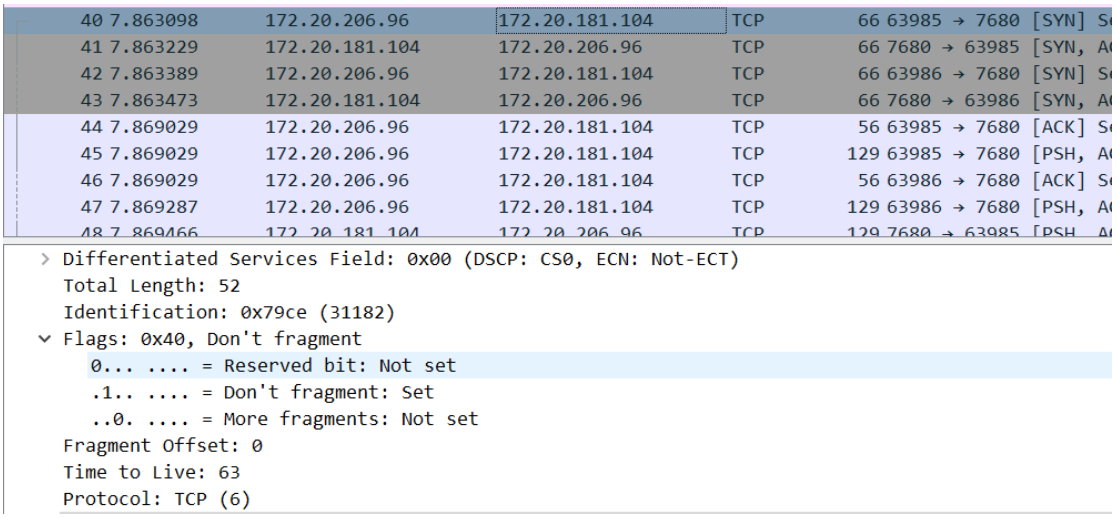
- 哪些字段必须保持常量？哪些字段必须改变？为什么？

  答：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。

- 描述你看到的IP数据包Identification字段值的形式。
  答：16位，且加一递增。

（3）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：

- Identification字段和TTL字段的值是什么？
  答：如图所示

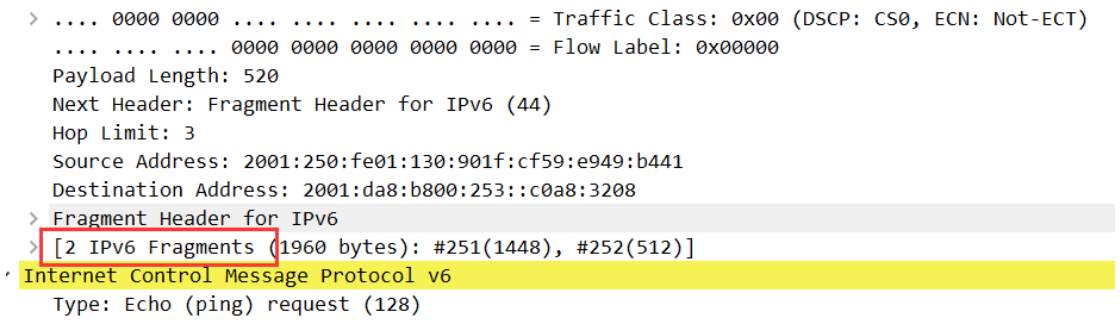


Identification：31182
TTL：63

- 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：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。

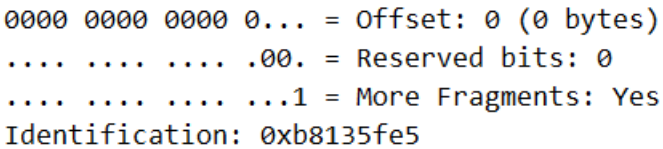
（4）单击Time列按钮，这样将对捕获的数据包按时间排序。找到在将包大小改为2000字节后你的主机发送的第一个ICMP Echo Request消息。
思考下列问题：

- 该消息是否被分解成不止一个IP数据报？
  答：如图所示。



被分成了两个数据报

- 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？
  答：如图所示。



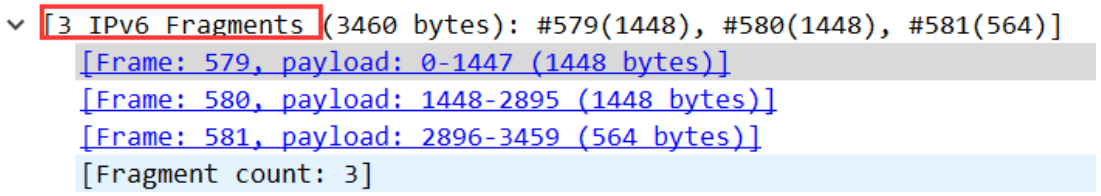
Flags标识位中，More fragments位被置为1，表示其被分片，但不是最后一个分片。

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？

  答：3片



- 这些分片中IP数据报头部哪些字段发生了变化？

  答：前两片的 More fragments 位均为1，而最后一片为0； 另外，第二片的分片的 offset 为1480，最后一片为2960。

5. 抓取 ARP 数据包

按照实验指导书上流程进行：

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

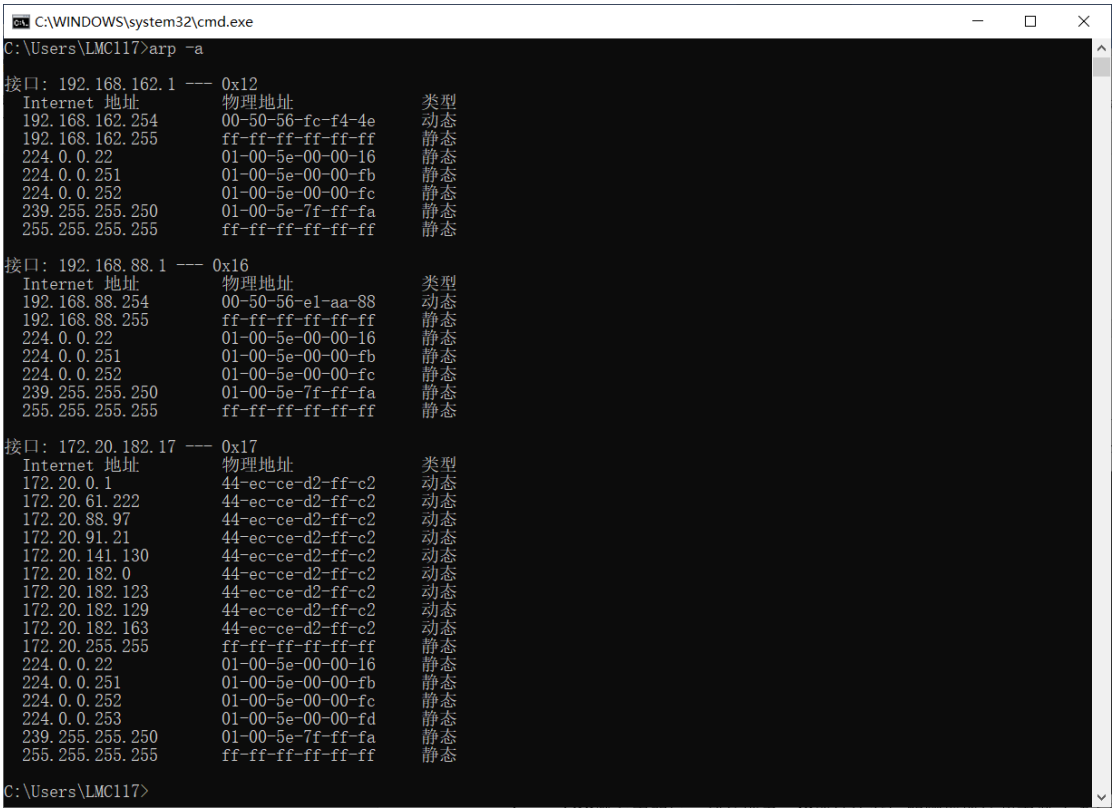
（2）在命令行模式下输入：ping 192.168.1.82（或其他 IP 地址）

（3）启动 Wireshark，开始分组俘获。

思考下面问题：

（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？

答：如图所示



ARP缓存中的每一列表示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：

- 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：数据包格式如图所示：

其由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）。

- 如何判断一个ARP数据是请求包还是应答包？
  答：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。

- 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：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响应报文明确知道查询主机的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。

6. 抓取 UDP 数据包

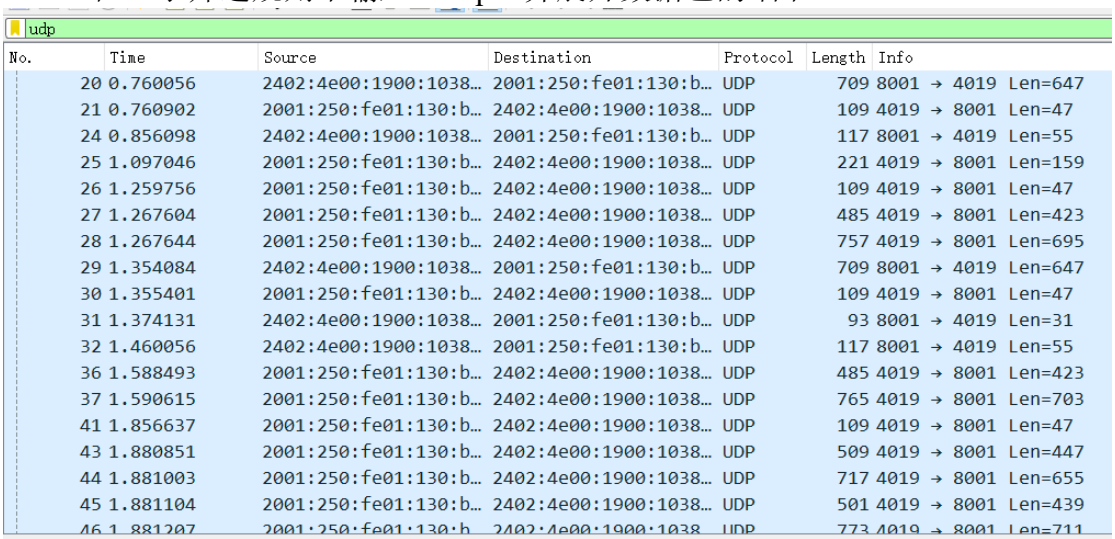
按照实验指导书上流程进行：

（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：

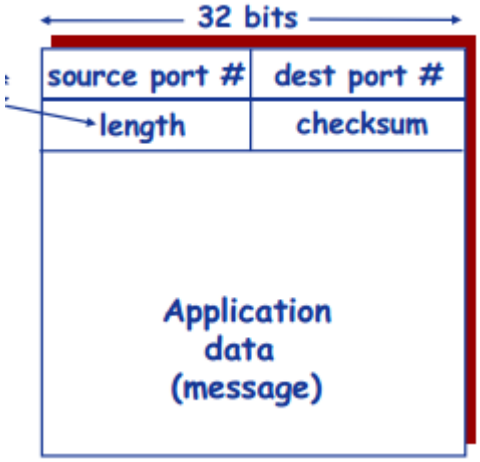
- 消息是基于UDP的还是TCP的？
  答：如上图所示，是基于UDP的

- 你的主机ip地址是什么？目的主机ip地址是什么？
  答：主机IP：172.20.182.17，目的主机IP：120.232.180.216

- 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QQ消息的端口号：64683，QQ服务器的端口号：9999

- 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？

答：UDP数据报格式如图所示：



UDP数据报由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）和其上附加的应用层数据。

- 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？

  答：服务器返回一个ICQ数据包，是因为服务器需要将接收到的结果返回给发送的客户端。

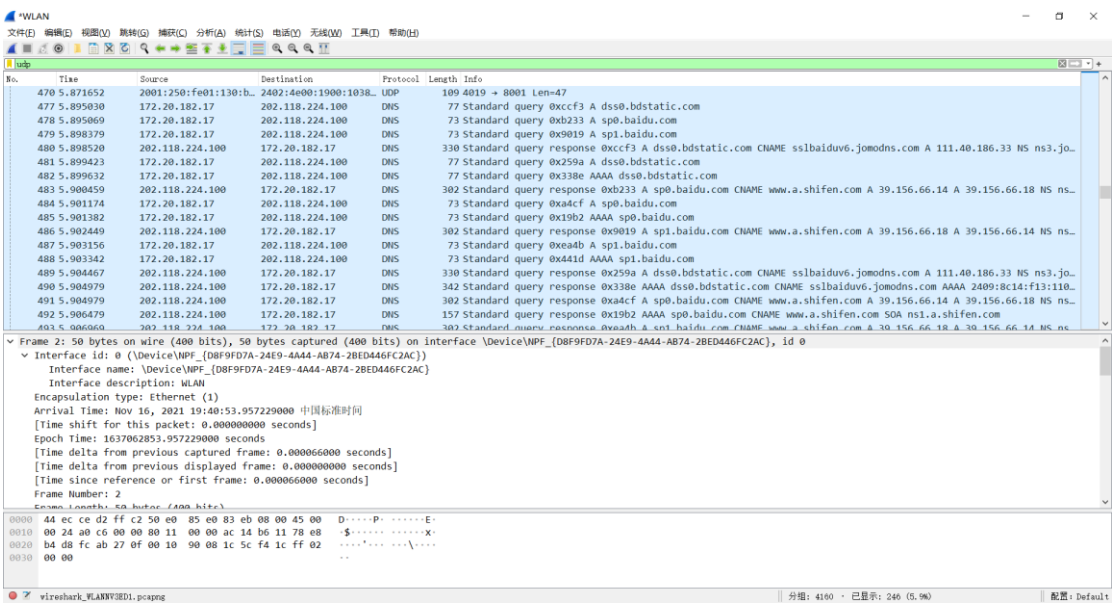
  这和UDP的不可靠数据传输的联系是：在UDP不可靠数据传输的机制下，服务器只提供一次返回的ACK，无法保证数据一定送达。

  能否看出UDP无连接：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。

7. 利用 WireShark 进行 DNS 协议分析

按照实验指导书上流程进行：

（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.
结果如下：



心得体会：

结合实验过程和结果给出实验的体会和收获。

- 熟悉了Wireshark工具的使用，能够熟练地进行数据包的抓取与分析
- 对计算机网络体系结构中不同的协议有了更清楚的认识
- 对数据包在网络传输中过程的认识更加深刻