**中国矿业大学计算机学院**

**2016级本科生计算机网络实验报告**

实验内容 协议报文分析

学生姓名 骆信智 08163337

专业班级 信息安全3班

学 院 计算机科学与技术学院

任课教师 高 璟

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程基础理论掌握程度** | 熟练 🞏 | 较熟练 🞏 | 一般 🞏 | 不熟练 🞏 |
| **综合知识应用能力** | 强 🞏 | 较强 🞏 | 一般 🞏 | 差 🞏 |
| **报告内容** | 完整 🞏 | 较完整 🞏 | 一般 🞏 | 不完整 🞏 |
| **报告格式** | 规范 🞏 | 较规范 🞏 | 一般 🞏 | 不规范 🞏 |
| **实验完成状况** | 好 🞏 | 较好 🞏 | 一般 🞏 | 差 🞏 |
| **工作量** | 饱满 🞏 | 适中 🞏 | 一般 🞏 | 欠缺 🞏 |
| **学习、工作态度** | 好 🞏 | 较好 🞏 | 一般 🞏 | 差 🞏 |
| **抄袭现象** | 无 🞏 | 有 🞏 姓名: | | |
| **存在问题** |  | | | |
| **总体评价** |  | | | |

综合成绩： 任课教师签字：

年 月 日

**实验编号：03**

**项目名称：网络应用程序设计**

**实验内容：**

1. 运用抓包工具，分别获取不同互联网访问情形下的本机网卡数据包；过滤捕获和过滤显示不同条件的数据包。
2. 分别对不同互联网访问情形下的数据包进行逐层分析，给出各层协议的主要参数及意义；要求分别获取WWW服务、Email服务、QQ通信和迅雷文件下载四种不同网络服务过程中的数据包。
3. 运用抓包工具，连续获取面向连接的互联网访问情形下的本机网卡数据包；对连续获取的数据包找到执行面向连接过程的报文，给出实现面向连接过程（TCP三次握手）的详细分析。

**实验要求：**

（1） 运用抓包工具，实时抓包，记录包状态变化；

（2）给出不同应用情境下的不同层次数据包的分析结果。

（3） 透过Web服务访问，分析HTTP协议工作过程，总结HTTP协议特点；透过HTTP工作过程分析，获取TCP协议的工作过程，验证连接建立的三次握手过程，以及滑动窗口工作机制

**预习要求：**

提前通过互联网或在实验室开始实验前登录实验管理服务器，点击预习链接，阅览或下载实验指导书——预习\网络协议\进阶-IP分组基本报文分析。

**操作与观察：**

正确按照实验指导书步骤操作，观察记录下操作结果。

**实验报告要求：**

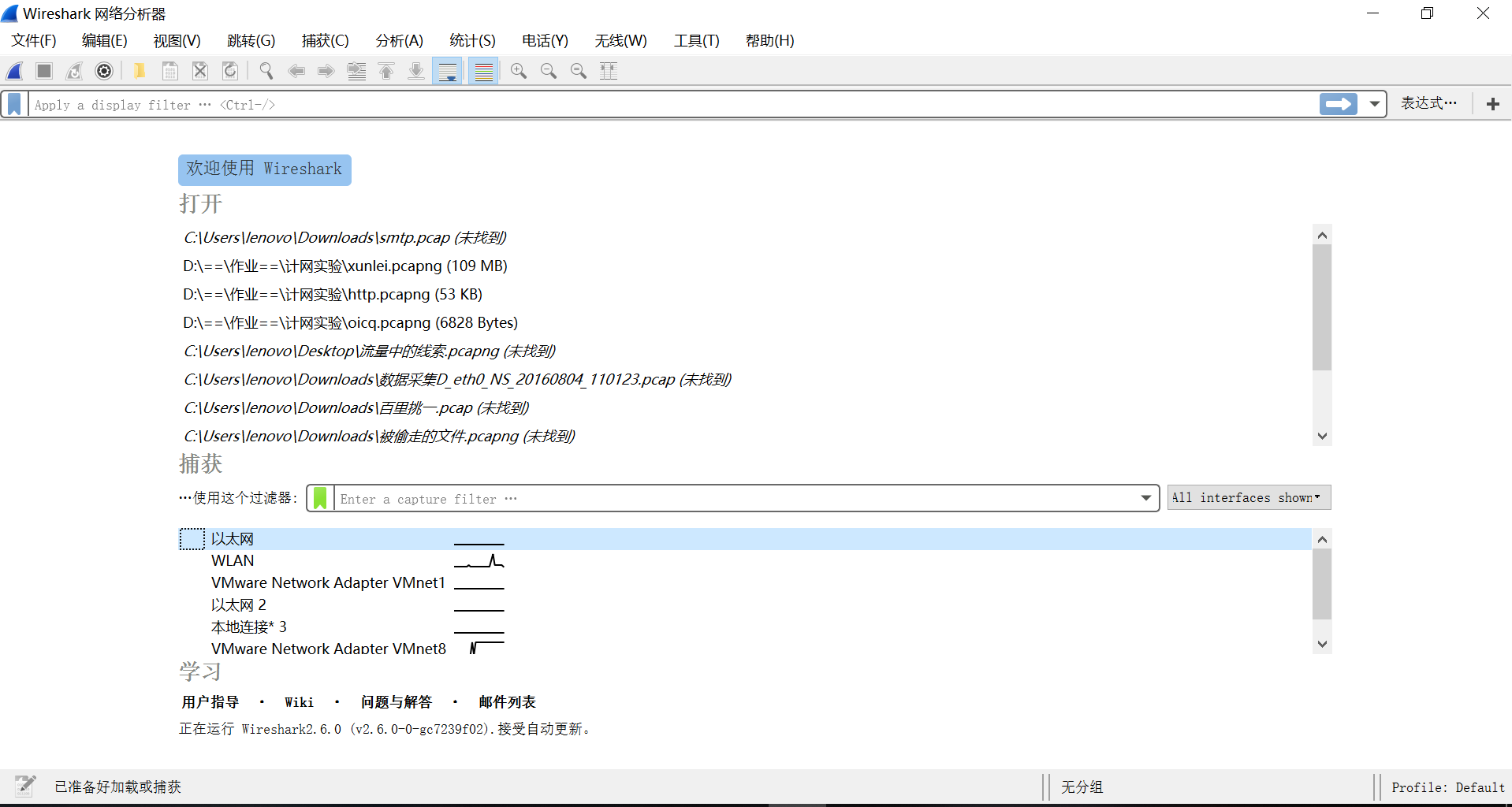
(1) 按照实验要求，完成全部实验内容

(2) 在标准实验报告书上填写全部实验操作记录和观察结果

(3) 登录实验管理服务器，提交实验报告电子档。

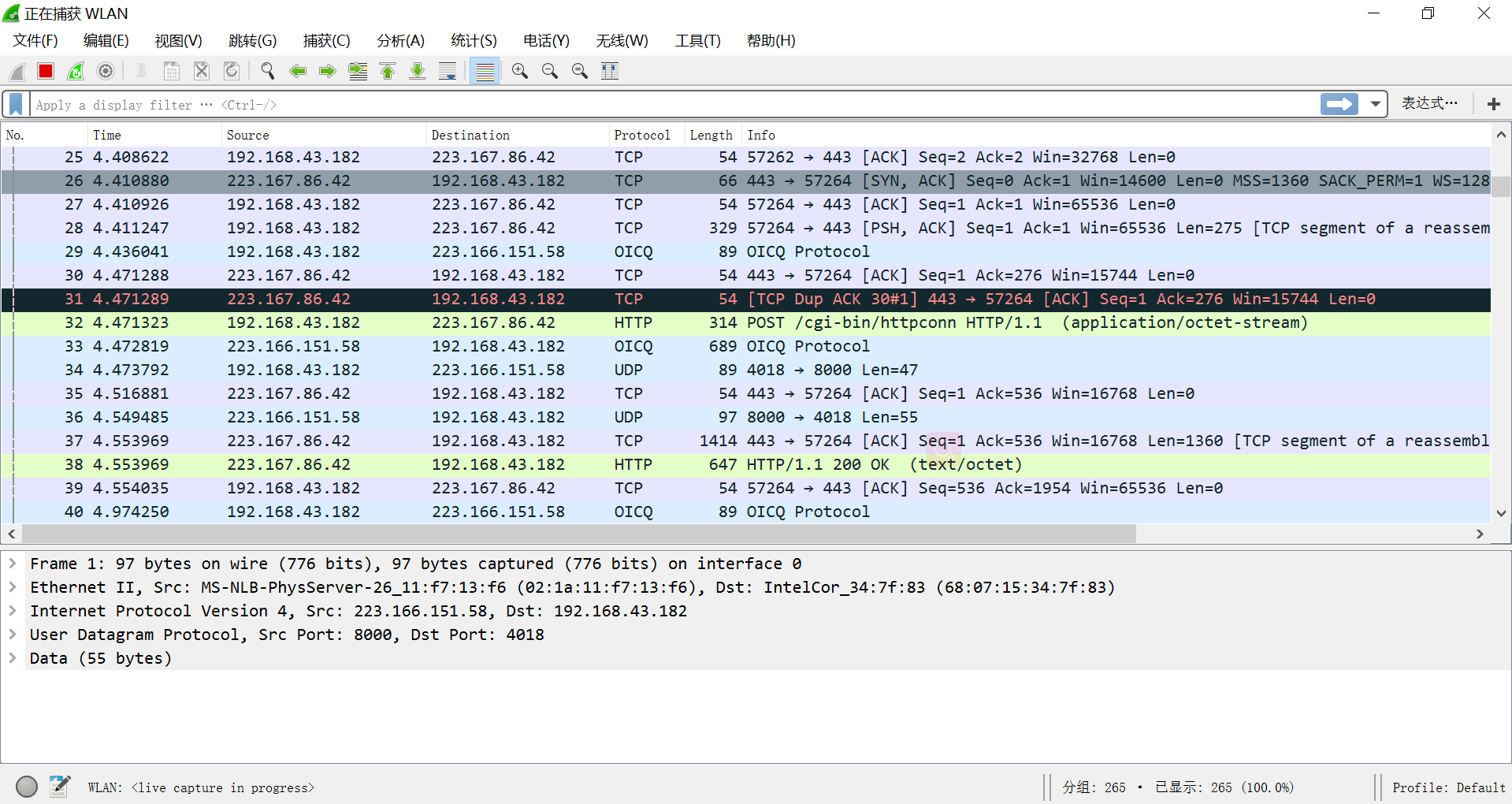
**实验报告内容：**

1. 运用抓包工具，分别获取不同互联网访问情形下的本机网卡数据包；过滤捕获和过滤显示不同条件的数据包。
2. 下载打开抓包分析工具wireshark

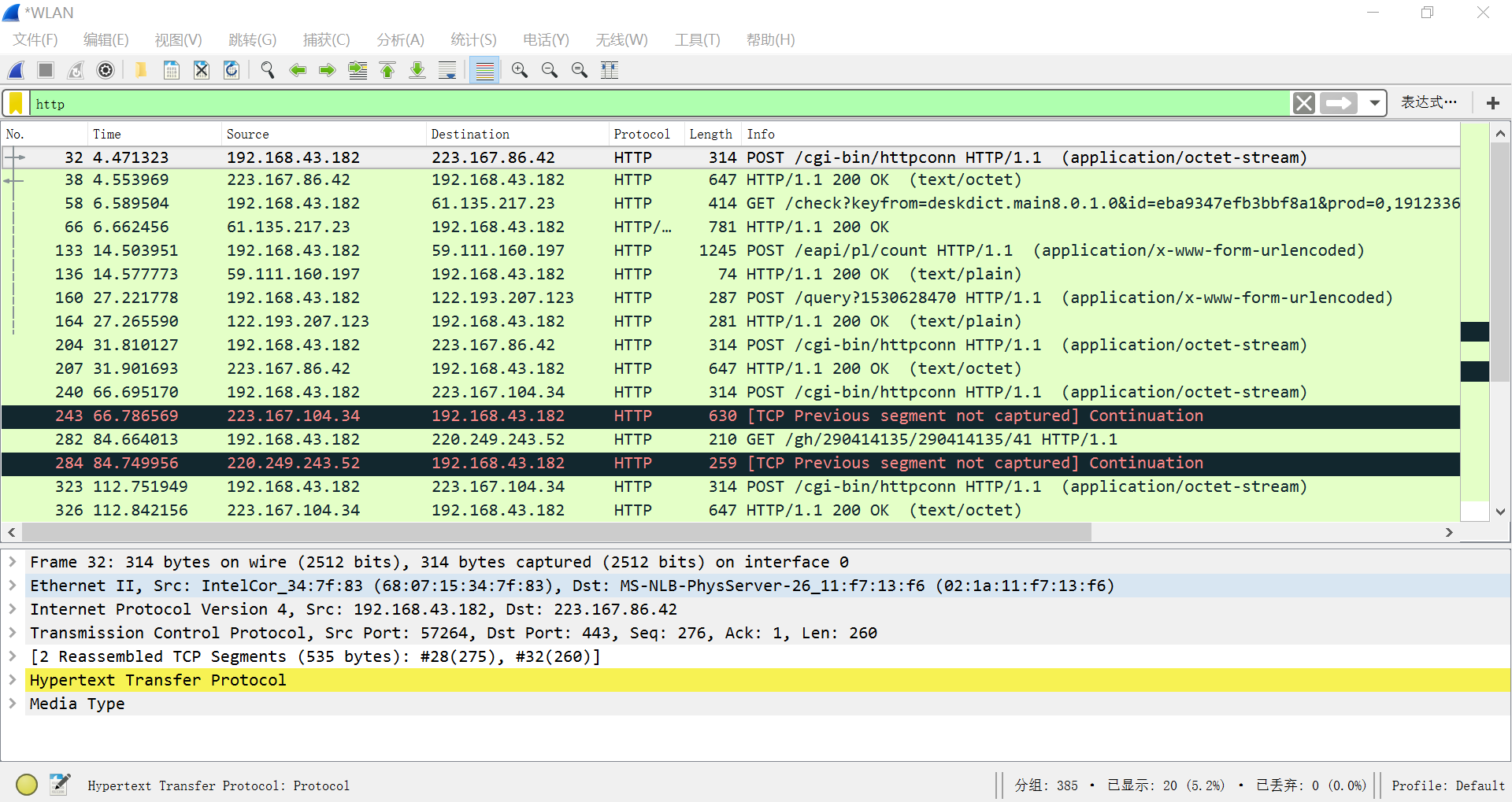


可以看到各网卡状态，可在过滤器中输入想捕获流量的过滤条件

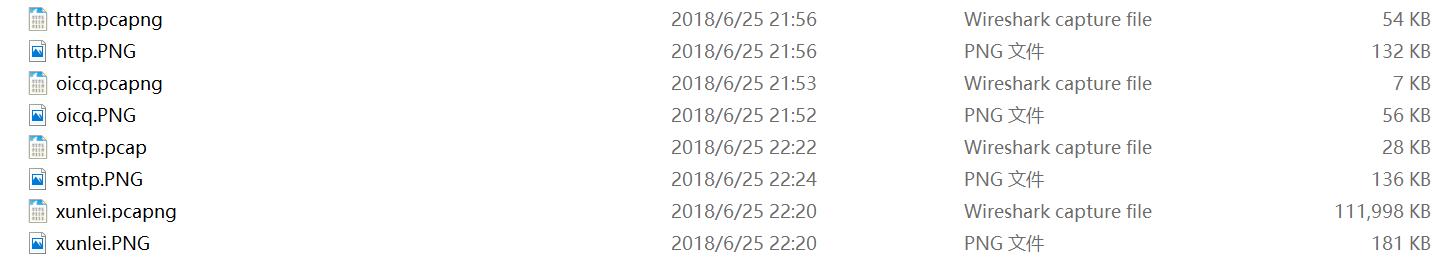
1. 选择wlan



可以看到实时抓包状态，可在顶端输入栏内输入过滤条件语句过滤出想查找的特定包

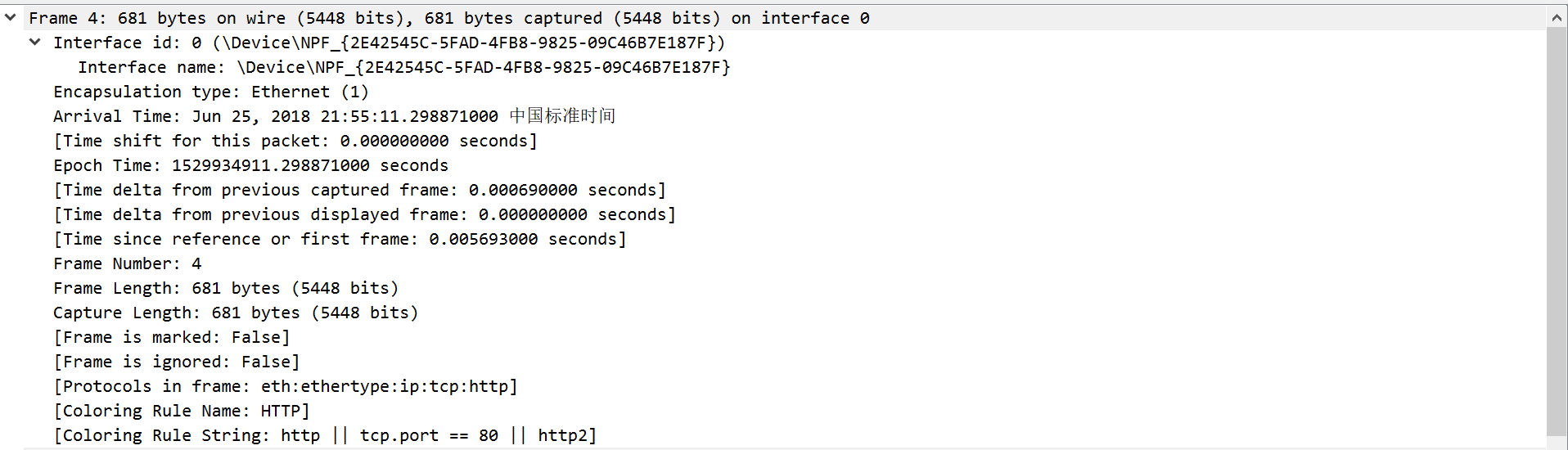


1. 在访问不同互联网情形下抓包分析



1. 给出不同应用情境下的不同层次数据包的分析结果。
2. 分别对不同互联网访问情形下的数据包进行逐层分析，给出各层协议的主要参数及意义

物理层数据帧参数



Frame 4: 681 bytes on wire (5448 bits), 681 bytes captured (5448 bits) on interface 0

Interface id: 0 (\Device\NPF\_{2E42545C-5FAD-4FB8-9825-09C46B7E187F}) #接口id

Interface name: \Device\NPF\_{2E42545C-5FAD-4FB8-9825-09C46B7E187F}

Encapsulation type: Ethernet (1) #封装类型

Arrival Time: Jun 25, 2018 21:55:11.298871000 中国标准时间 #捕获日期和时间

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1529934911.298871000 seconds

[Time delta from previous captured frame: 0.000690000 seconds] #此包与前一包的时间间隔

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 0.005693000 seconds] #此包与第一帧的时间间隔

Frame Number: 4 #帧序号

Frame Length: 681 bytes (5448 bits) #帧长度

Capture Length: 681 bytes (5448 bits) #捕获长度

[Frame is marked: False] #此帧是否做了标记：否

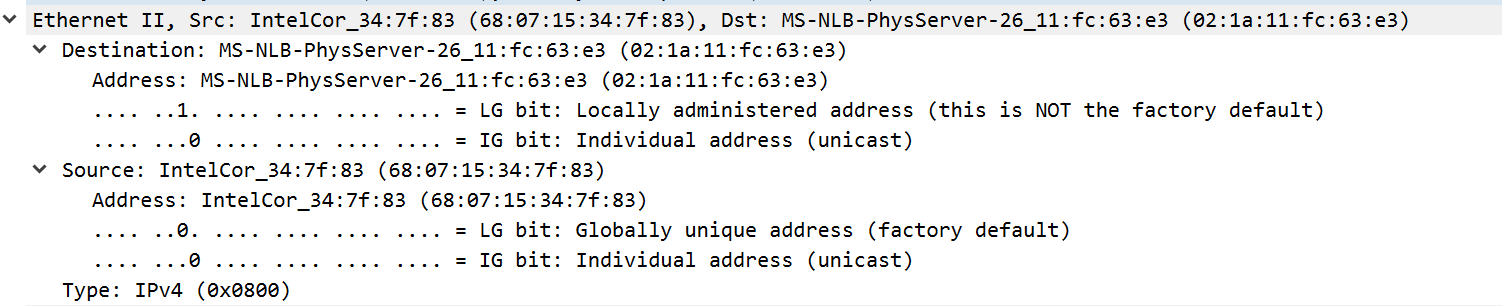
[Frame is ignored: False] #此帧是否被忽略：否

[Protocols in frame: eth:ethertype:ip:tcp:http] #帧内封装的协议层次结构

[Coloring Rule Name: HTTP] #着色标记的协议名称

[Coloring Rule String: http || tcp.port == 80 || http2] #着色规则显示的字符串

数据链路层参数：以太网帧头部信息



Ethernet II, Src: IntelCor\_34:7f:83 (68:07:15:34:7f:83), Dst: MS-NLB-PhysServer-26\_11:fc:63:e3 (02:1a:11:fc:63:e3)

Destination: MS-NLB-PhysServer-26\_11:fc:63:e3 (02:1a:11:fc:63:e3) #目标MAC地址

Address: MS-NLB-PhysServer-26\_11:fc:63:e3 (02:1a:11:fc:63:e3)

.... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Source: IntelCor\_34:7f:83 (68:07:15:34:7f:83) #源MAC地址

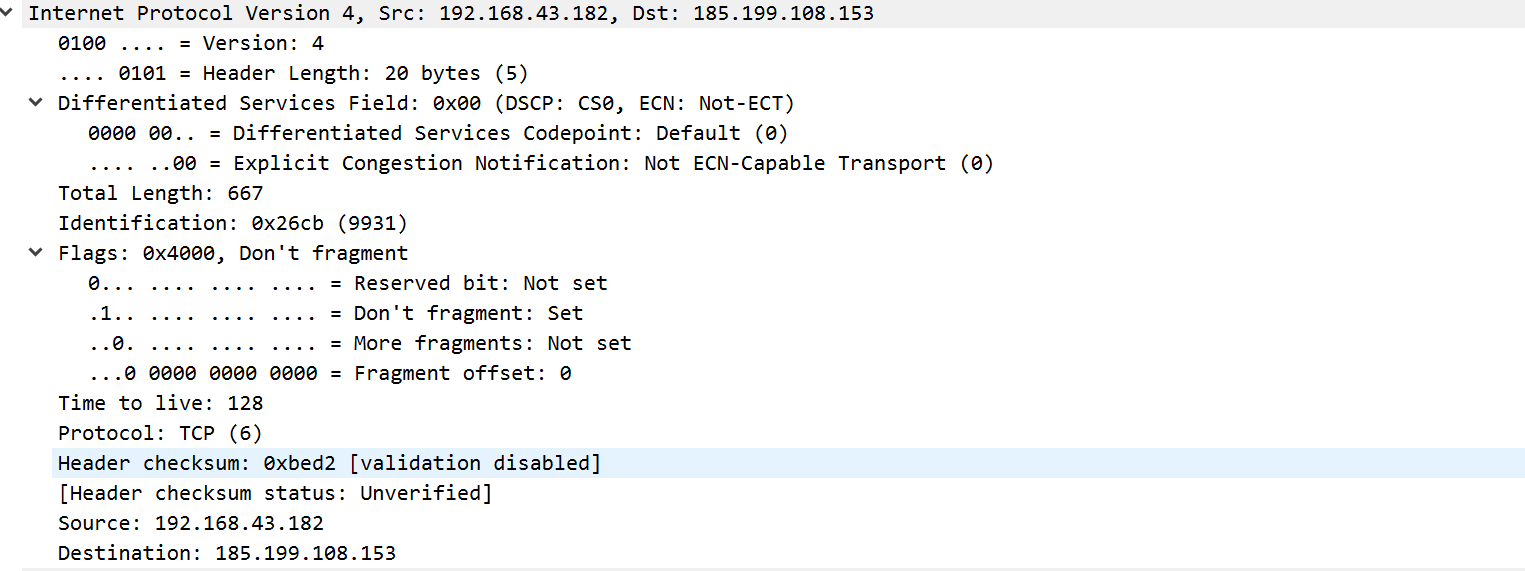
Address: IntelCor\_34:7f:83 (68:07:15:34:7f:83)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

网络层参数：IP包头部信息



Internet Protocol Version 4, Src: 192.168.43.182, Dst: 185.199.108.153

0100 .... = Version: 4 #互联网协议IPv4

.... 0101 = Header Length: 20 bytes (5) #IP包头部长度

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: 667 #IP包的总长度

Identification: 0x26cb (9931) #标志字段

Flags: 0x4000, Don't fragment #标记字段

0... .... .... .... = Reserved bit: Not set

.1.. .... .... .... = Don't fragment: Set

..0. .... .... .... = More fragments: Not set

...0 0000 0000 0000 = Fragment offset: 0 #分的偏移量

Time to live: 128 #生存期TTL

Protocol: TCP (6) #此包内封装的上层协议为TCP

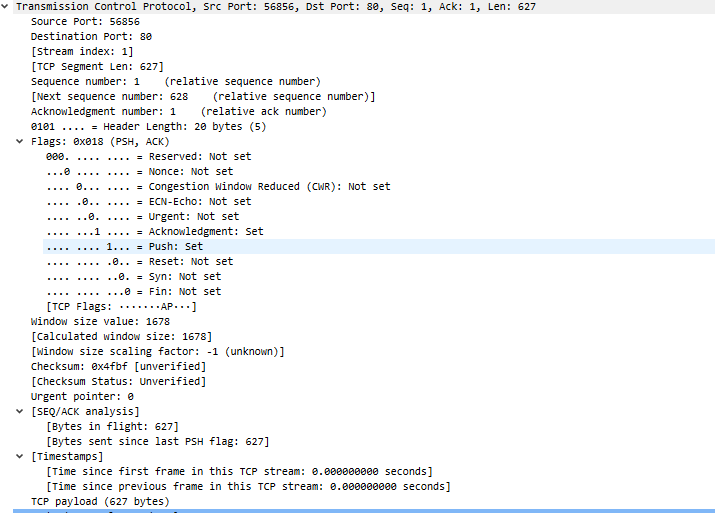
Header checksum: 0xbed2 [validation disabled] #头部数据的校验和

[Header checksum status: Unverified]

Source: 192.168.43.182 #源IP地址

Destination: 185.199.108.153 #目标IP地址

传输层参数：TCP数据段头部信息



Transmission Control Protocol, Src Port: 56856, Dst Port: 80, Seq: 1, Ack: 1, Len: 627

Source Port: 56856 #源端口号

Destination Port: 80 #目标端口号

[Stream index: 1]

[TCP Segment Len: 627]

Sequence number: 1 (relative sequence number) #序列号（相对序列号）

[Next sequence number: 628 (relative sequence number)] #下一个序列号

Acknowledgment number: 1 (relative ack number) #确认序列号

0101 .... = Header Length: 20 bytes (5) #头部长度

Flags: 0x018 (PSH, ACK) #TCP标记字段

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

.... .... 1... = Push: Set

.... .... .0.. = Reset: Not set

.... .... ..0. = Syn: Not set

.... .... ...0 = Fin: Not set

[TCP Flags: ·······AP···]

Window size value: 1678 #流量控制的窗口大小

[Calculated window size: 1678]

[Window size scaling factor: -1 (unknown)]

Checksum: 0x4fbf [unverified] #TCP数据段的校验和

[Checksum Status: Unverified]

Urgent pointer: 0

[SEQ/ACK analysis]

[Bytes in flight: 627]

[Bytes sent since last PSH flag: 627]

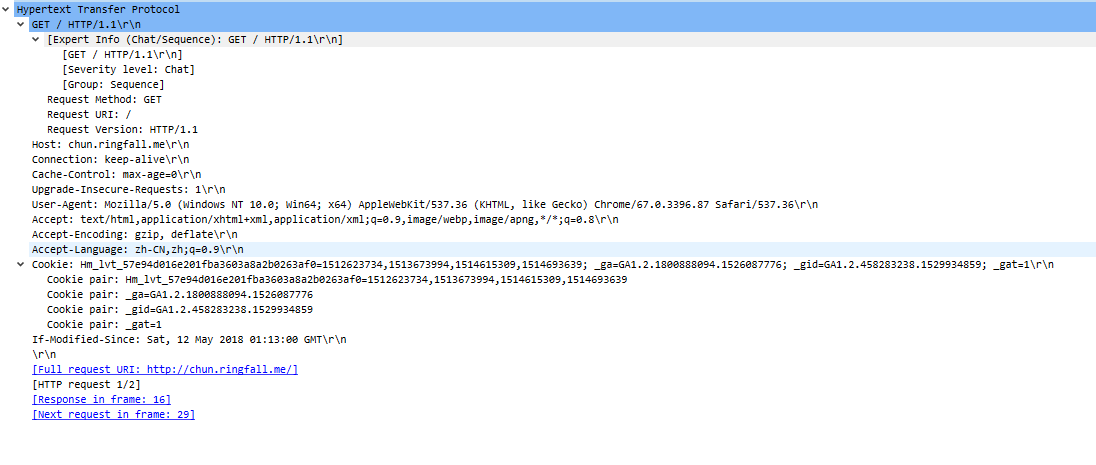
[Timestamps]

[Time since first frame in this TCP stream: 0.000000000 seconds]

[Time since previous frame in this TCP stream: 0.000000000 seconds]

TCP payload (627 bytes)

应用层参数：http请求包



Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]

[GET / HTTP/1.1\r\n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET #请求方法

Request URI: / #请求资源位置

Request Version: HTTP/1.1 #http协议1.1版

Host: chun.ringfall.me\r\n #请求主机

Connection: keep-alive\r\n #使用了持续连接

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/67.0.3396.87 Safari/537.36\r\n #请求的浏览器或其他客户端

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8\r\n

#客户端接受什么类型的响应

Accept-Encoding: gzip, deflate\r\n #编码类型

Accept-Language: zh-CN,zh;q=0.9\r\n #客户端的首选语言

Cookie: Hm\_lvt\_57e94d016e201fba3603a8a2b0263af0=1512623734,1513673994,1514615309,1514693639; \_ga=GA1.2.1800888094.1526087776; \_gid=GA1.2.458283238.1529934859; \_gat=1\r\n #cookie信息

Cookie pair: Hm\_lvt\_57e94d016e201fba3603a8a2b0263af0=1512623734,1513673994,1514615309,1514693639

Cookie pair: \_ga=GA1.2.1800888094.1526087776

Cookie pair: \_gid=GA1.2.458283238.1529934859

Cookie pair: \_gat=1

If-Modified-Since: Sat, 12 May 2018 01:13:00 GMT\r\n #只有当页面在指定的日期后已更改时更新页面

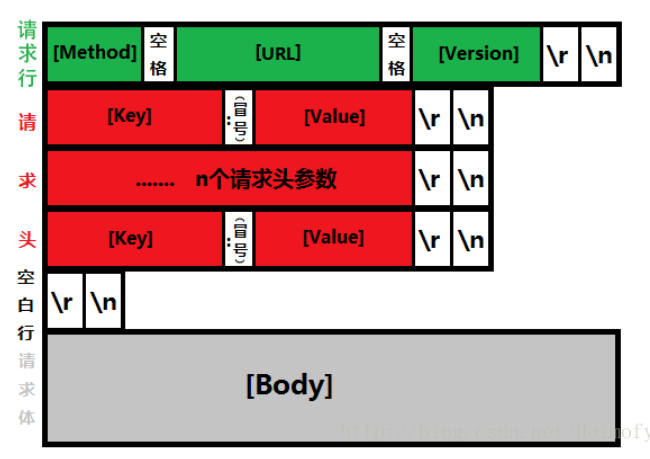
\r\n

[Full request URI: http://chun.ringfall.me/]

[HTTP request 1/2]

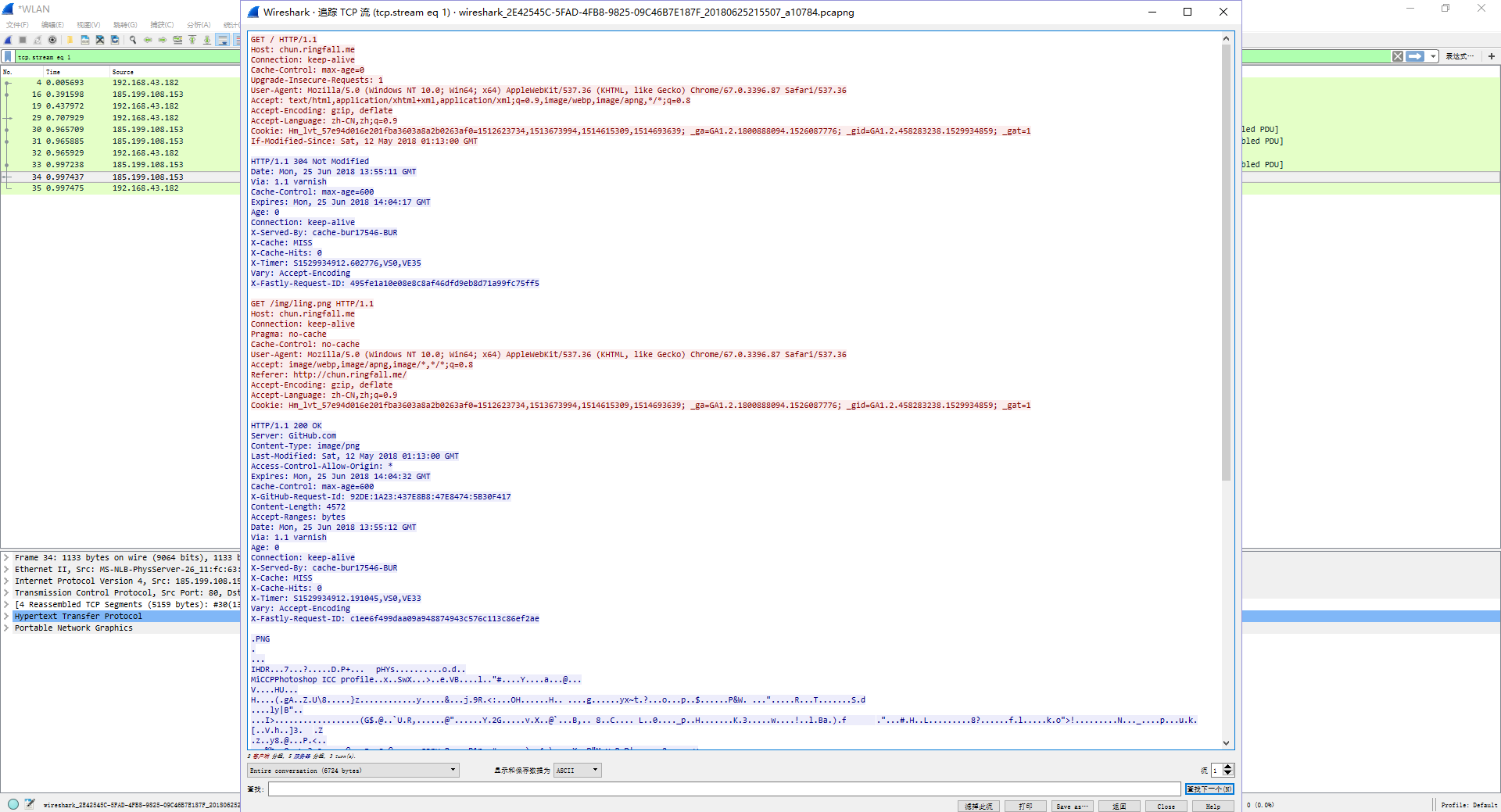
[Response in frame: 16]

[Next request in frame: 29]

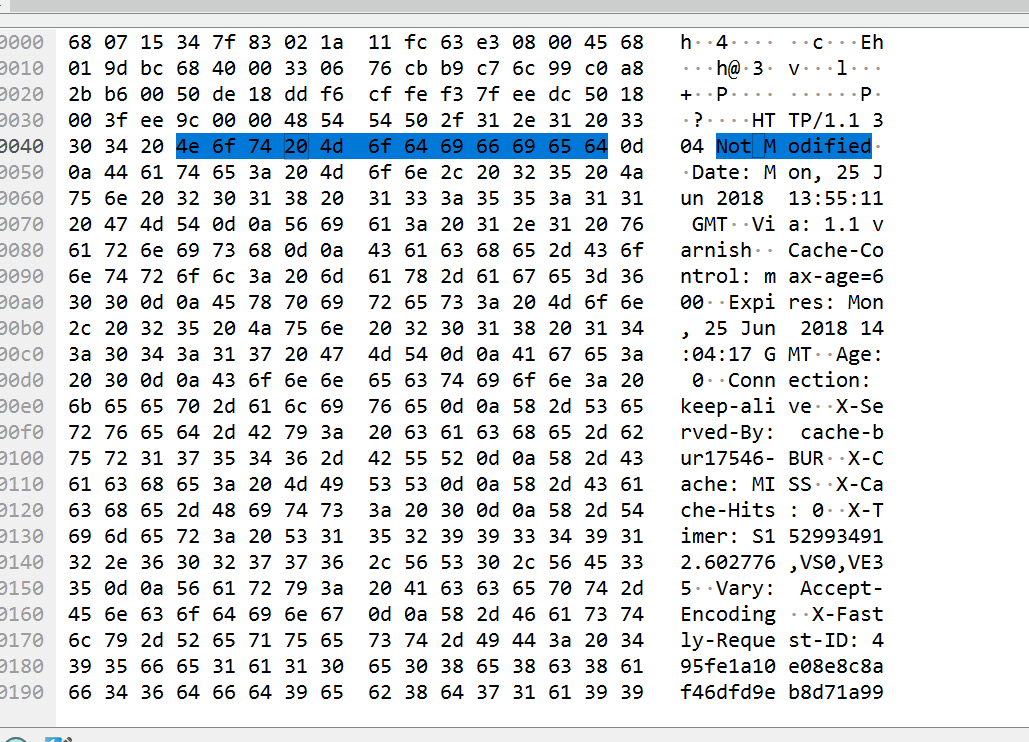
报文格式

1. 要求分别获取WWW服务、Email服务、QQ通信和迅雷文件下载四种不同网络服务过程中的数据包。

WWW服务：

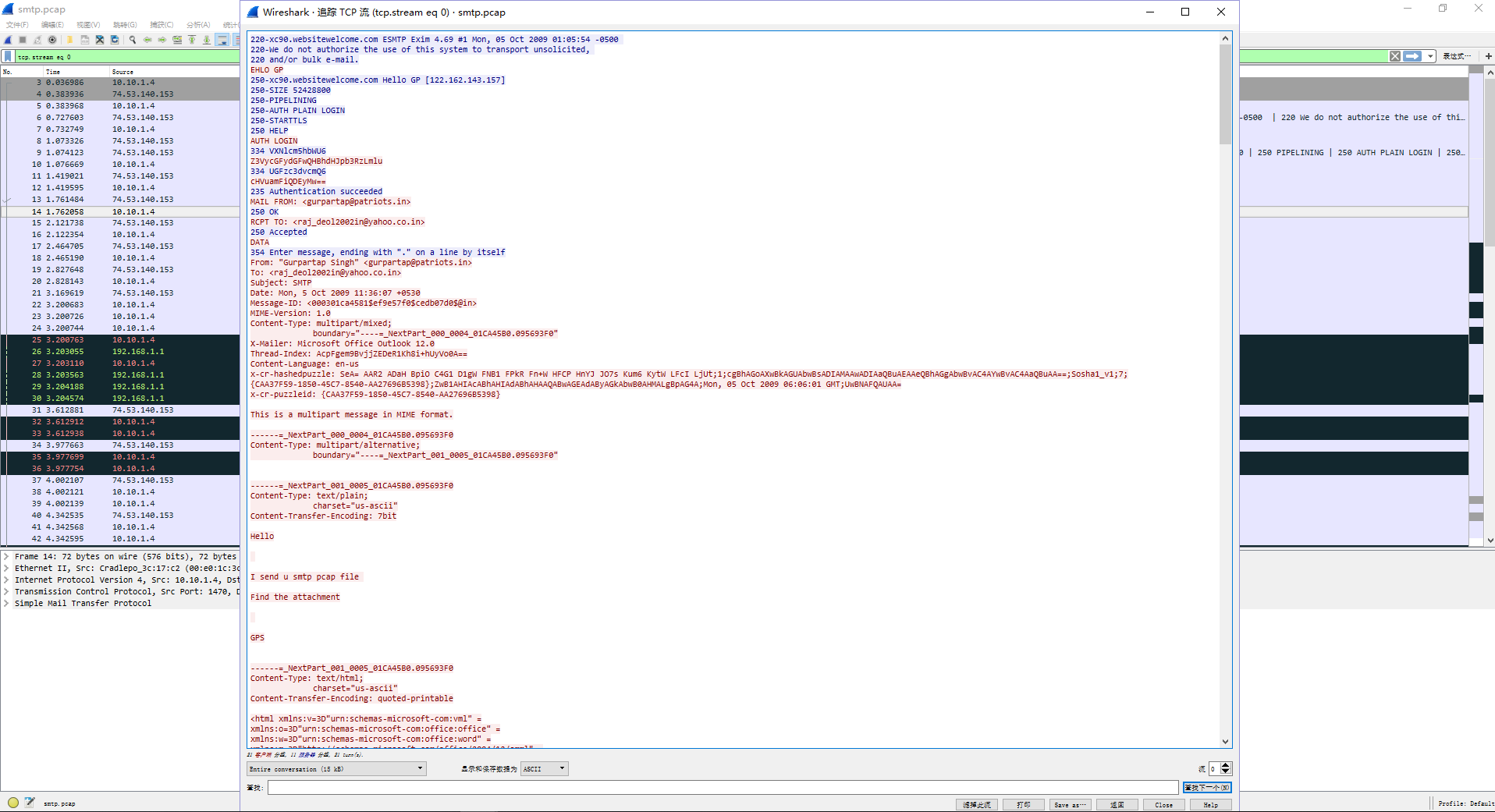


过滤http协议后选中一条追踪tcp流，可以看到完整的网络请求和相应包；第一次请求之后返回304 Not Modified，说明页面无改动，于是浏览器直接从缓存中读取数据呈现给用户；第二次请求了一张图片资源，服务器返回200 OK，然后把图片传输了过来。

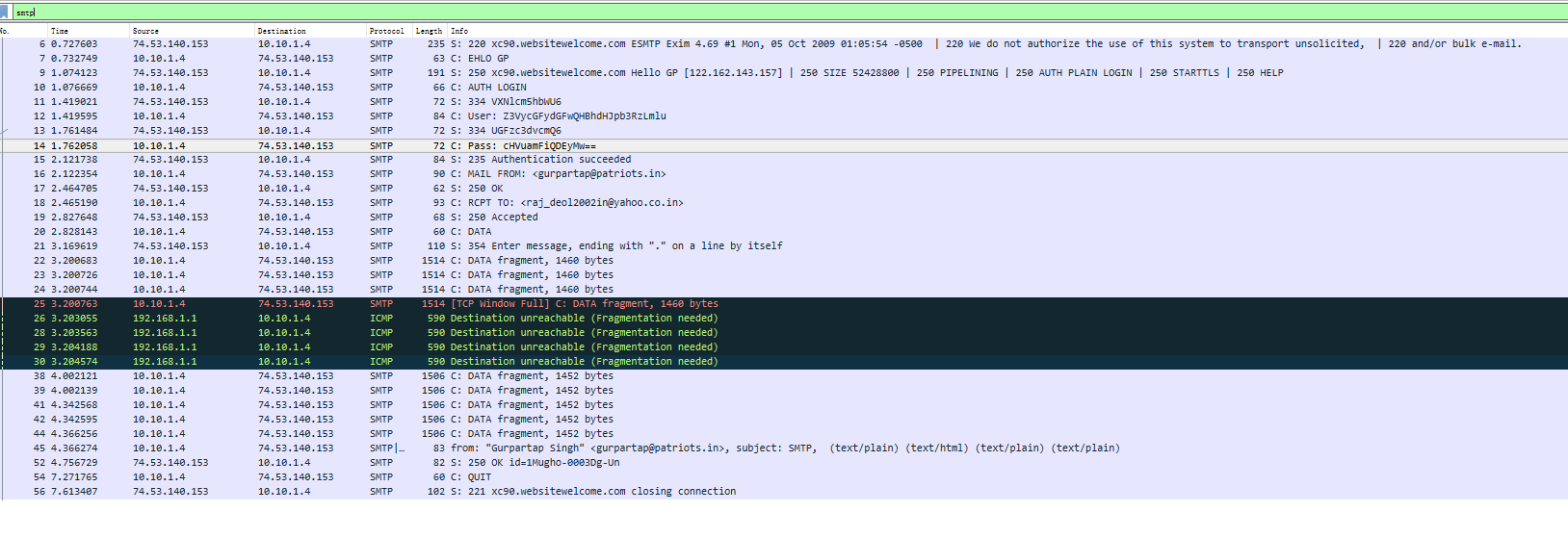


下方可以看到具体16进制信息和ASCII码。

Email服务：

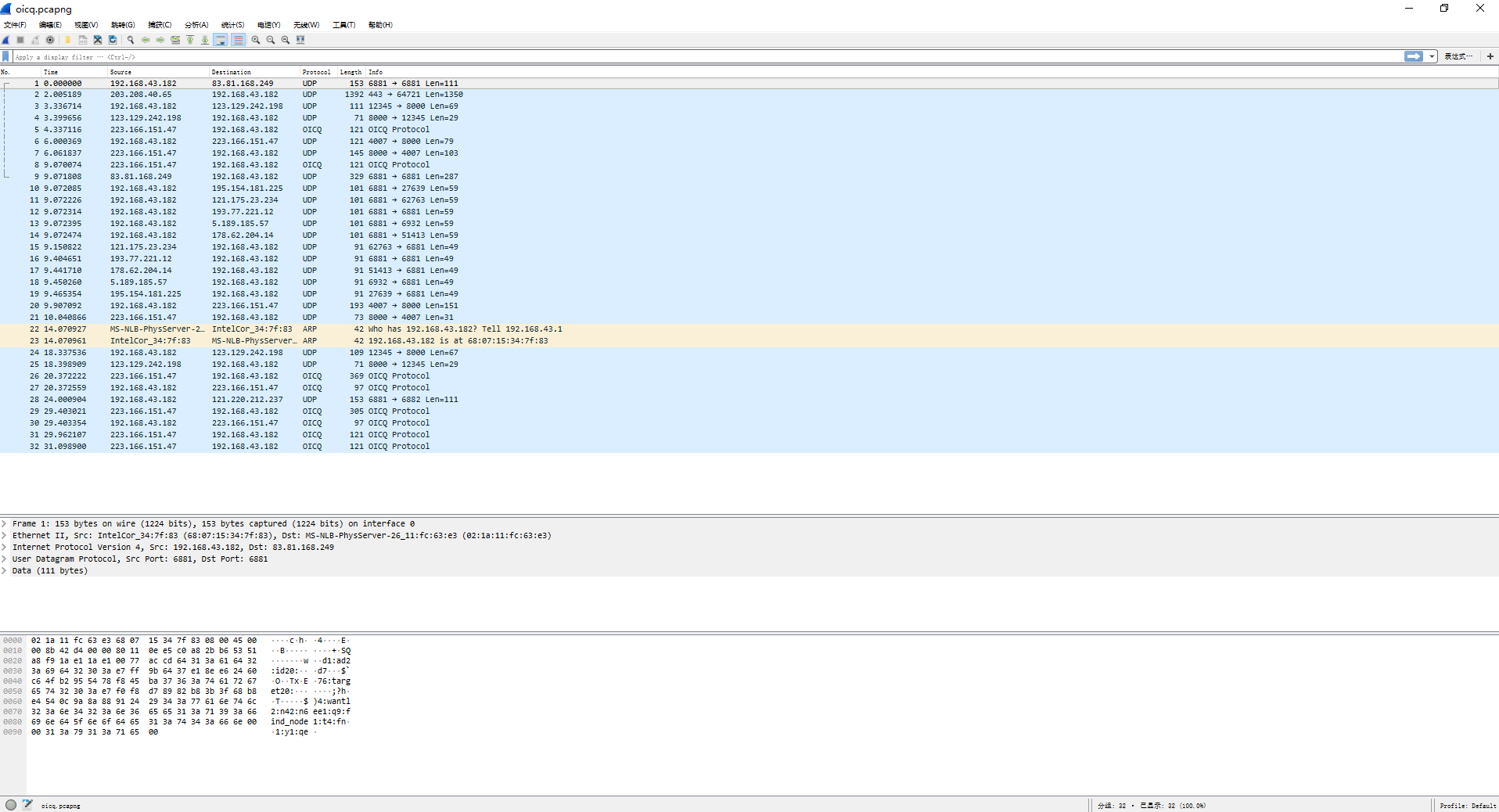


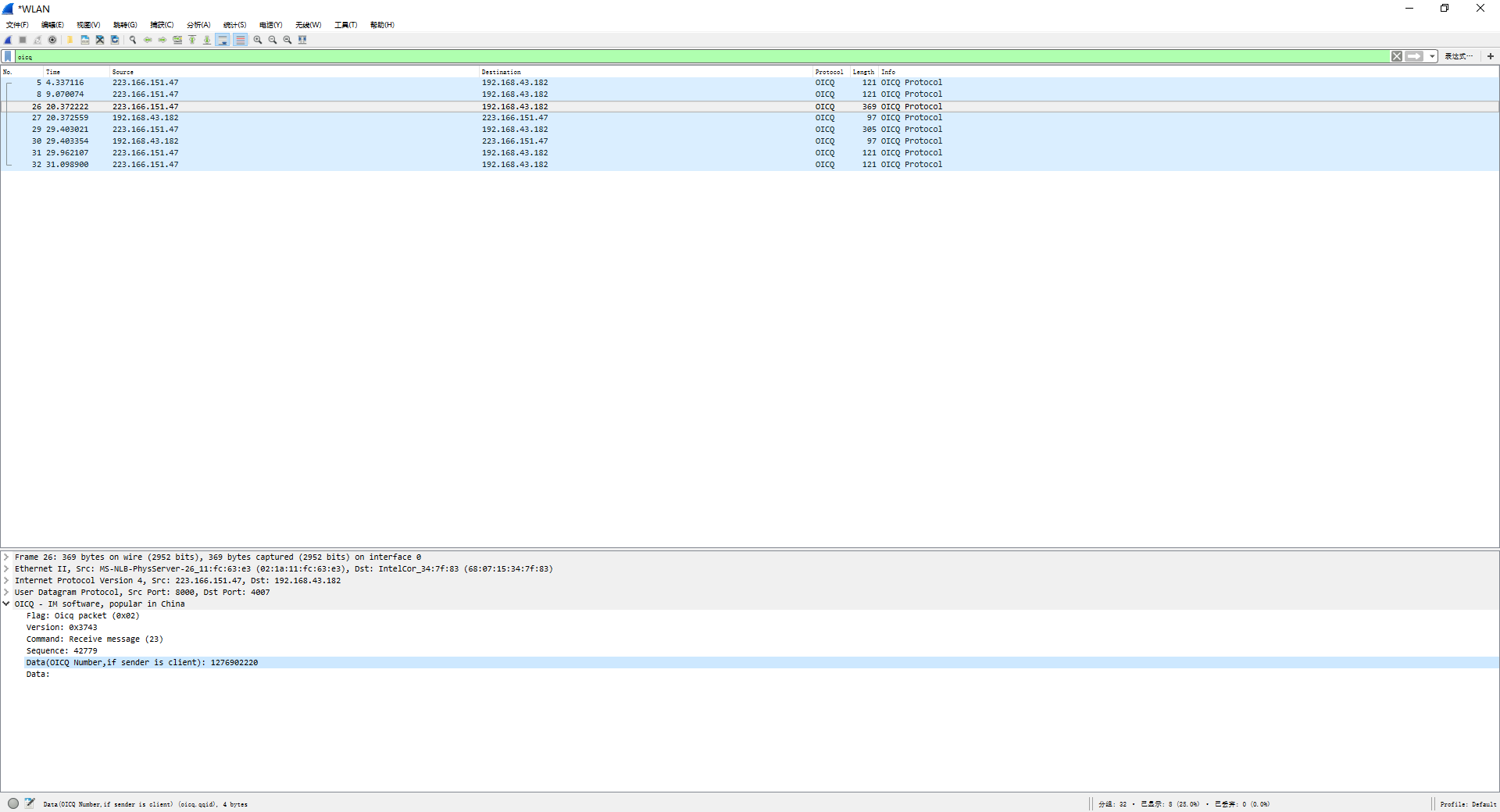
过滤smtp协议后选中一条追踪tcp流，可以看到完整的邮箱服务器登录、发送邮件到退出的过程。可以看到经过base64编码的用户名和密码以及完整邮件内容和一起发送的各种发送端及接收端信息。



直接观察info字段也可发现大致流程

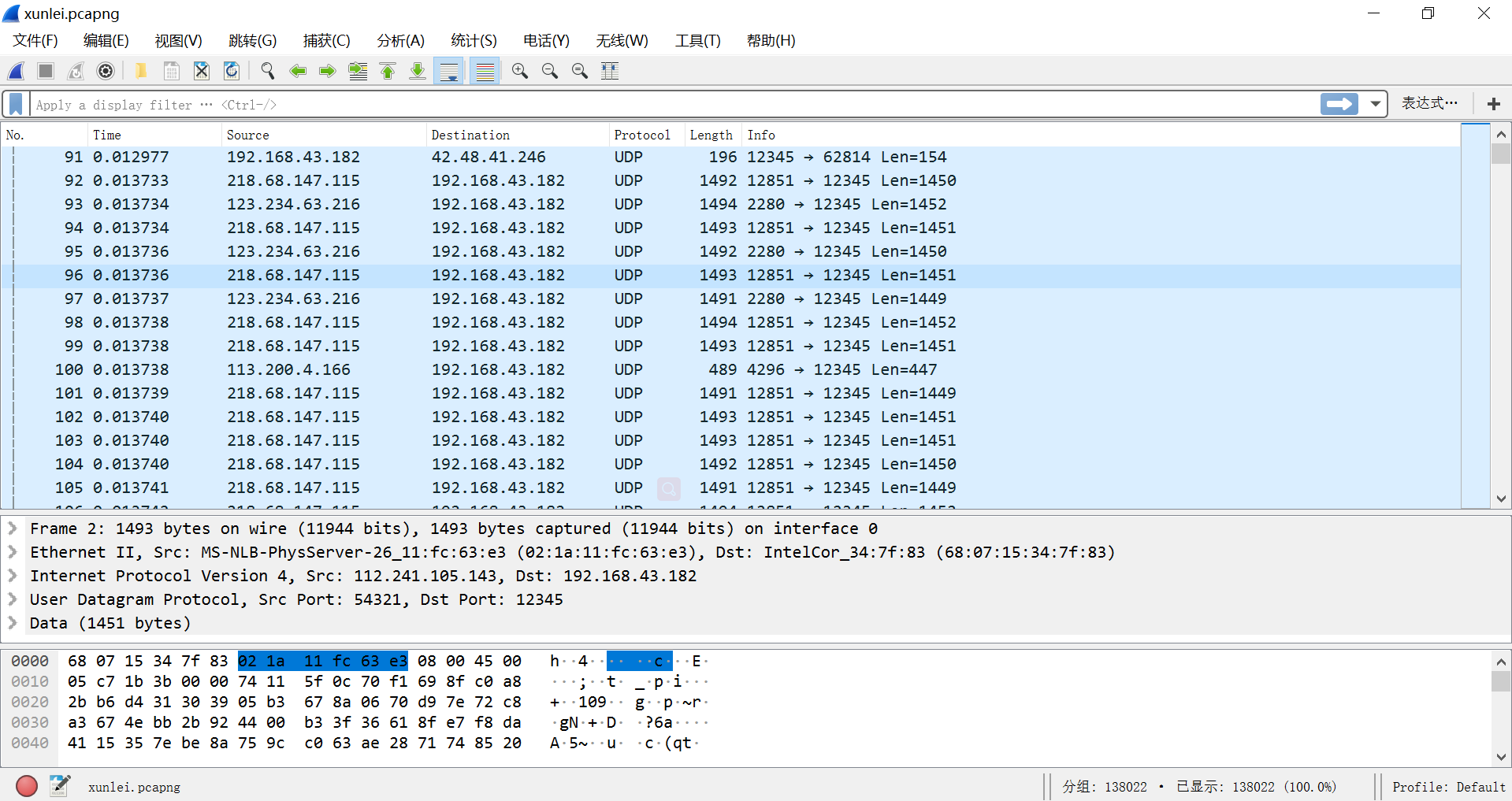
QQ通信：

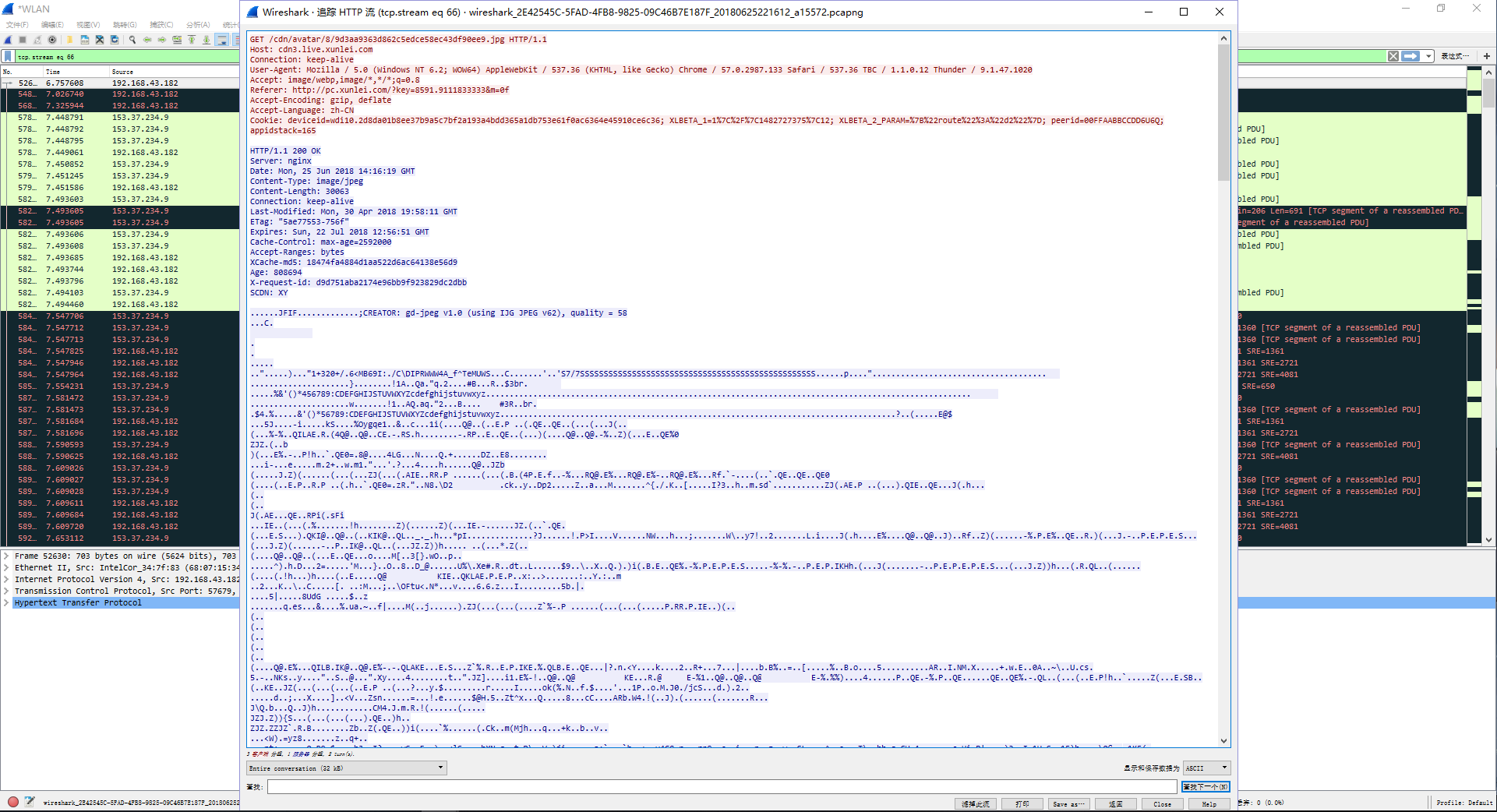




可以看到经过qq的通信内容都经过加密，肉眼无法识别，较为安全。另外可以发现qq使用oicq协议并用udp进行传输。

迅雷文件下载：





可以发现包很大而且大多经过加密，无法准确识别定位，使用的协议也很多，经过分析后等大致了解了迅雷的P2P下载方式优缺点。

1. 运用抓包工具，连续获取面向连接的互联网访问情形下的本机网卡数据包；对连续获取的数据包找到执行面向连接过程的报文，给出实现面向连接过程（TCP三次握手）的详细分析。

透过Web服务访问，分析HTTP协议工作过程，总结HTTP协议特点

工作过程：地址解析->封装HTTP请求数据包->封装成TCP包，建立TCP连接->客户机发送请求命令->服务器响应->服务器关闭TCP连接

特点：

1.支持客户/服务器模式。

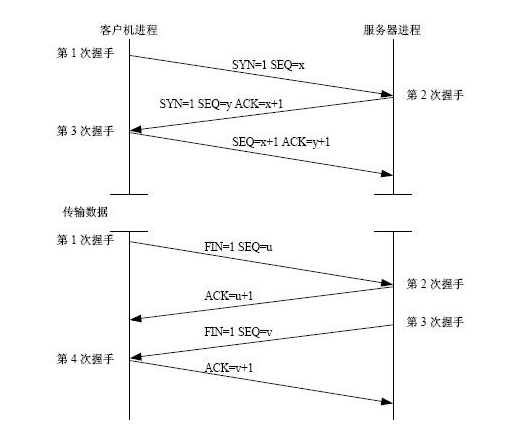
2.简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。

3.灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type（Content-Type是HTTP包中用来表示内容类型的标识）加以标记。

4.无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。

5.无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

通过HTTP工作过程分析，获取TCP协议的工作过程

第一阶段：连接建立（三次握手）  
第二阶段：数据传输  
第三阶段：连接拆除（四次握手）

验证连接建立的三次握手过程



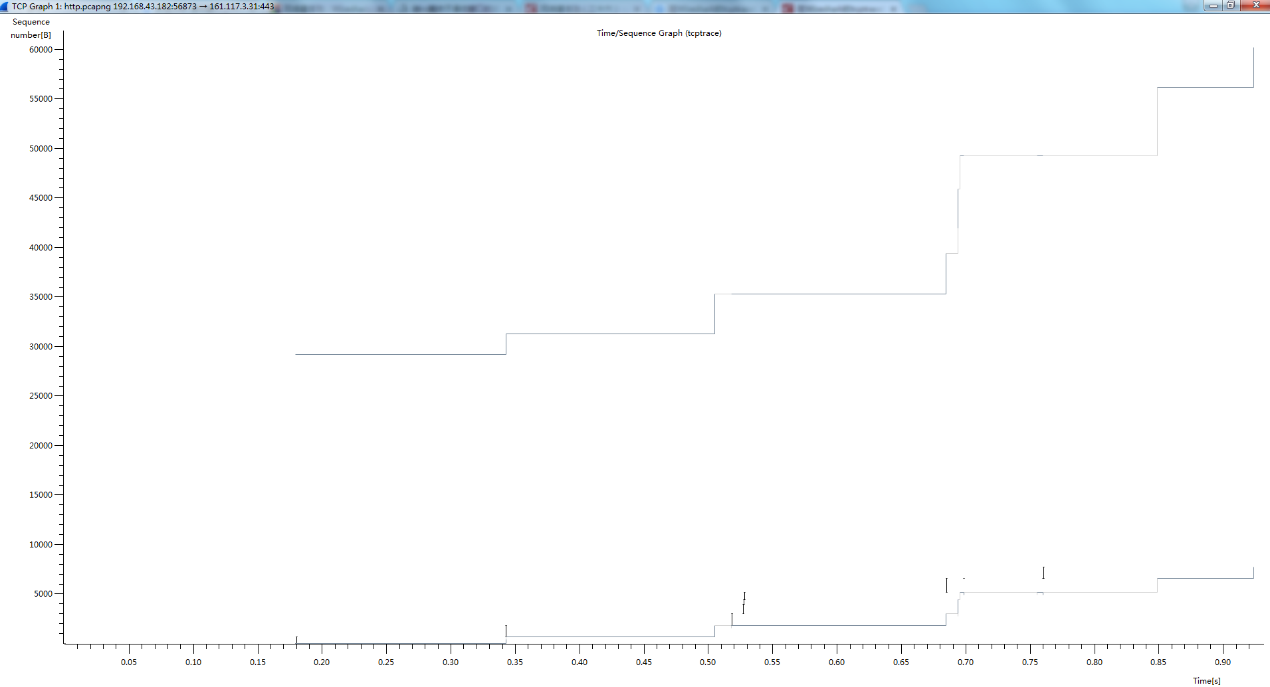
可以很清晰的看到TCP三次握手，客户端56856端口和服务器的80端口建立了连接



客户端56873端口和服务器的443端口建立了连接

TCP滑动窗口机制如下：

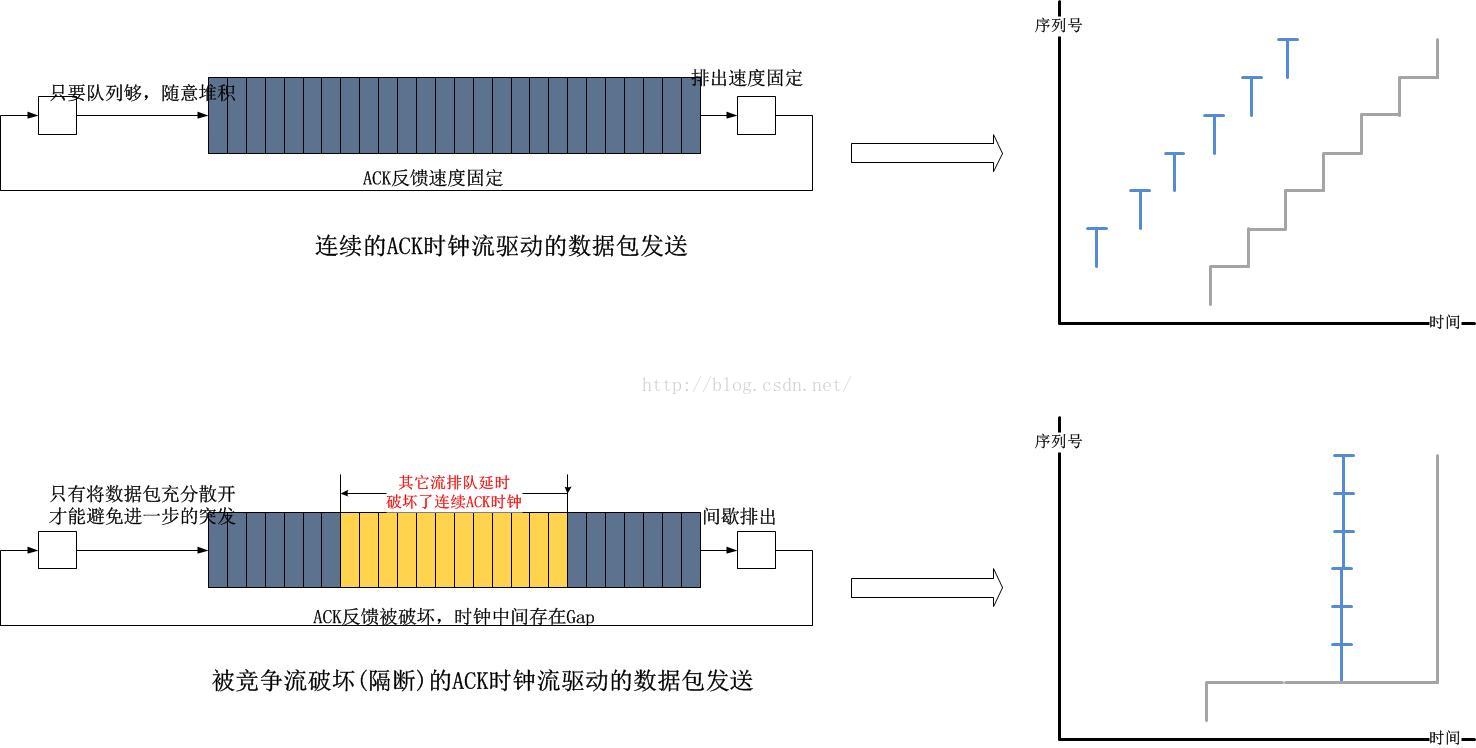
1. 连接建立之后，发送方将数据发送至接收方，填入接收窗口。
2. 若干报文之后，接收方发送ACK至发送方，确认接收到其发送的字节数。发送ACK将接收窗口清零。
3. 这一过程持续下去，发送方向窗口中填入数据，接收方清空并发送确认信息。
4. 扩大接收窗口大小告知发送方增加吞吐量，减小窗口告知对方减小吞吐量。这一机制按照WS/RTT规则（随着TCP版本不同而有所改变）

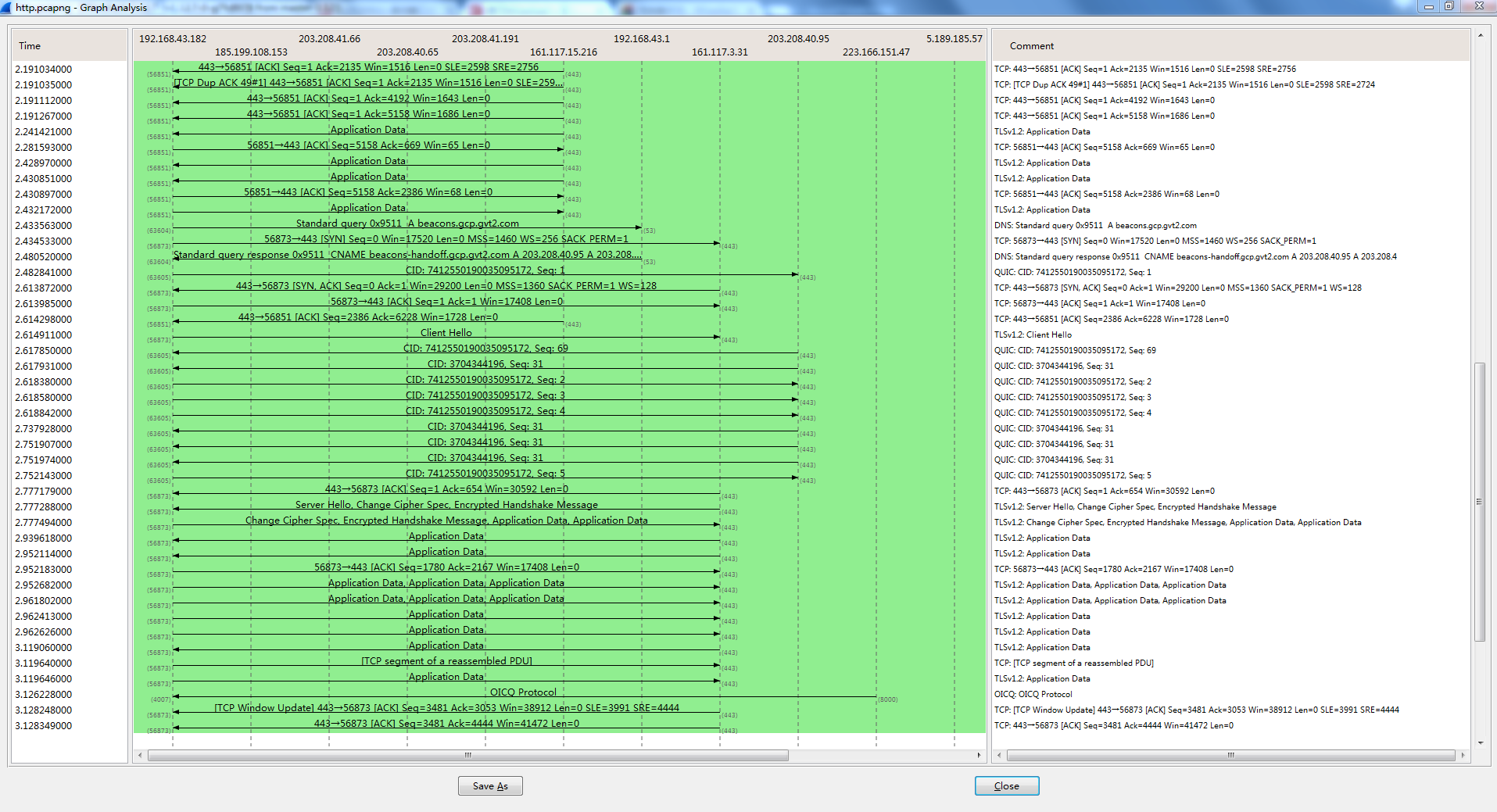
打开statistics-TCP StreamGraph-tcptrace

三条线分别为接收端窗口的序列号/时间线、发送端发送数据的序列号/时间线和接收端应答到达发送端的序列号/时间线

两行之间的固定距离表明接收方工作良好。当两行渐渐靠近，表明发送方速度高于接收方。只要这两行没有重叠，TCP就会继续发送数据。

我们知道，对于TCP而言，ACK时钟始终扮演了一个重要的角色，是它驱动了TCP发送端的行为，包括但不限于数据的发送，拥塞窗口的调整等，但是要想维持ACK时钟，就必须保证有源源不断的ACK回到TCP发送端，这个要求在单流场景下是必然被满足的，因为即便网络中存在队列，也是唯一的TCP流独占队列，该队列的处理速度决定了ACK时钟的频率。然而一旦有其它的流竞争队列，单一的TCP流在队列中就会被隔开，这个被隔开的间隔就是ACK时钟被中断的断层，这个断层造成了数据的突发：



或从flow graph每次的串口参数分析也可以

**实验体会：**

本次实验初步掌握了抓包分析工具wireshark的基本使用，通过各种功能系统且简单的学习到了报文交换的过程，以及报文所解析的具体格式。了解到了整个报文的传送以太网协议(物理链路) -> ip协议(网络) -> tcp/udp(传输层) -> 应用软件封装协议 (应用层)的方式进行。

通过实验，理解了各层包括以太网mac帧，ip数据报，tcp报文，udp报文，以及相应的http报文，oicq报文的相应的格式。

实验中对邮件进行stmp报文抓取的时候，在网页上发邮件只能抓取到http报文，最后运用客户端软件直接设置服务器抓到流量。

通过本次实验，对计算机网络整体情况有了细致且入微的认识，对各报文的格式有了基本的概念，提升了对网络探索的兴趣和能力，收获很多。