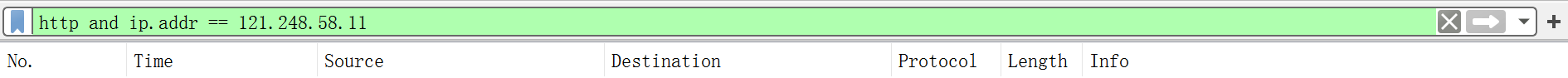
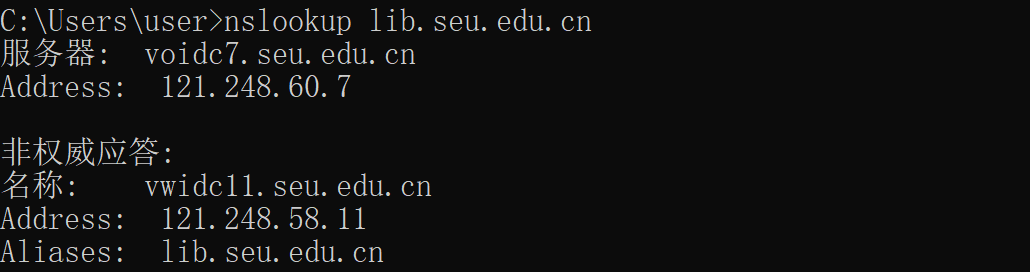
**实验一 应用层**

09019204 曹邹颖

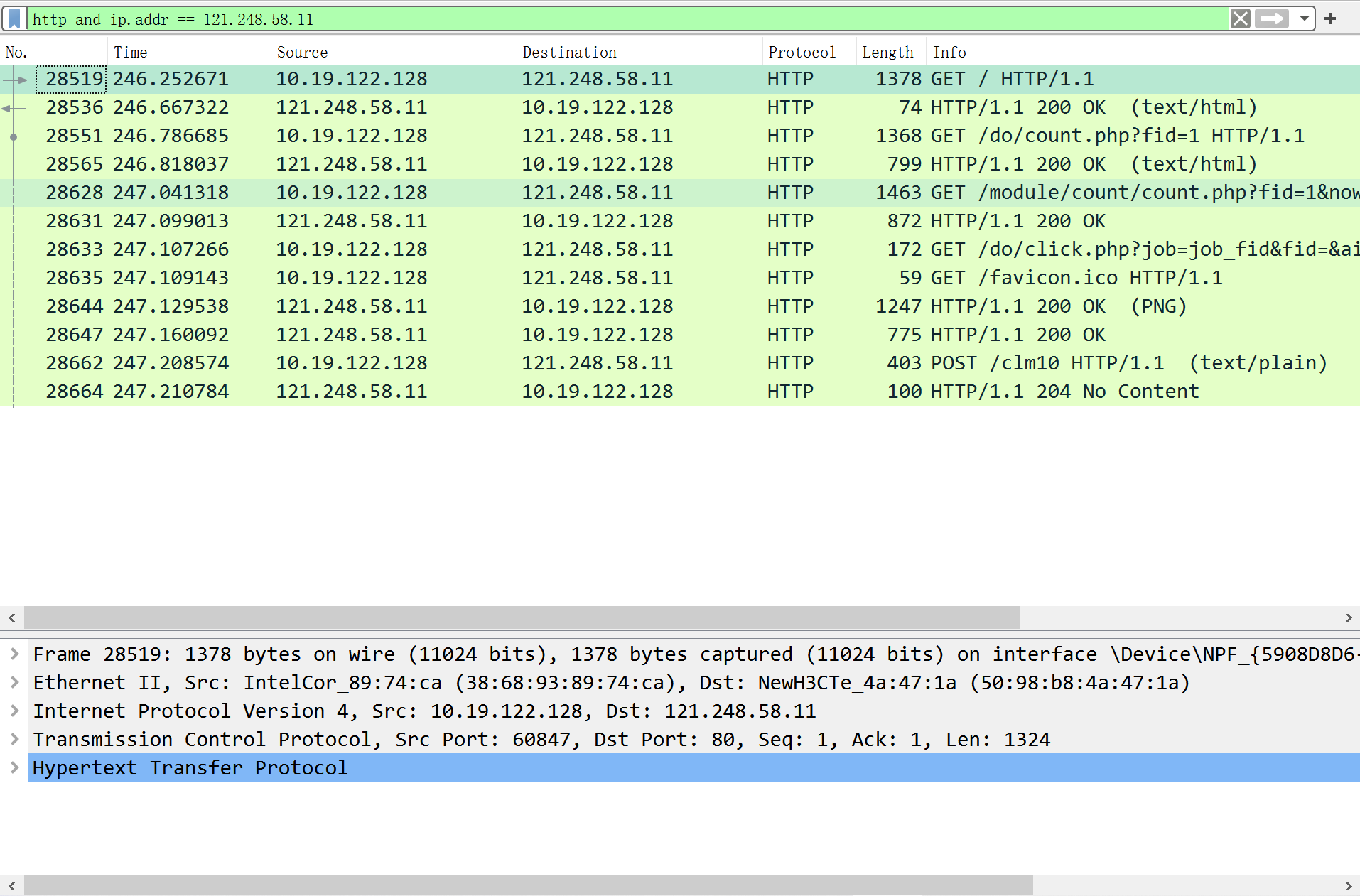
1. **实验内容**
2. 学会使用Wireshark抓包软件，会使用过滤器
3. 学习Wireshark基本操作：重点掌握捕获过滤器和显示过滤器。分析HTTP和DNS协议
4. 测试curl命令，访问一个web页面。（选做）
5. 利用telnet命令测试get命令，访问www.baidu.com。（选做）
6. 利用telnet命令测试SMTP服务，解析其过程。（选做）
7. 测试tracert命令，并解析其过程。
8. 使用nslookup查询域名信息，简要分析。
9. **实验步骤与分析**
10. **Wireshark实验**
11. HTTP协议分析
12. 打开Wireshark设置过滤器：http and ip.addr == 121.248.58.11



121.248.58.11是使用nslookup命令查询的东南大学图书馆网站的IP地址。

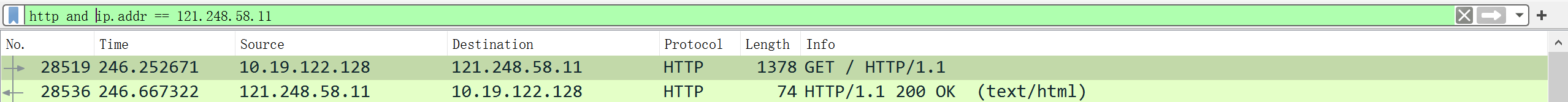


1. 打开浏览器访问东南大学图书馆网站lib.seu.edu.cn, 浏览器打开网址开始抓包

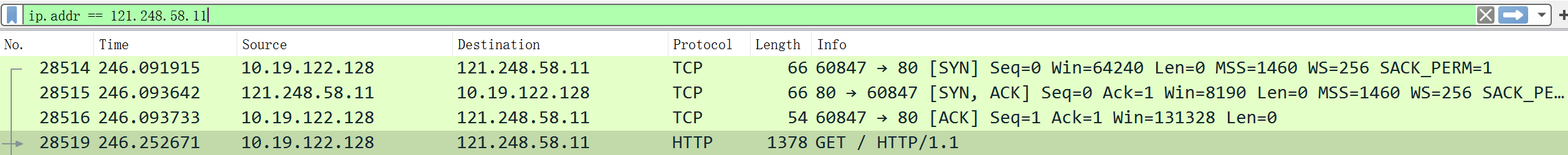


1. 这里，我以HTTP-GET数据包为例分析HTTP协议。

选取第一个客户端向服务器发送的HTTP请求包以及其对应的HTTP响应包分析。

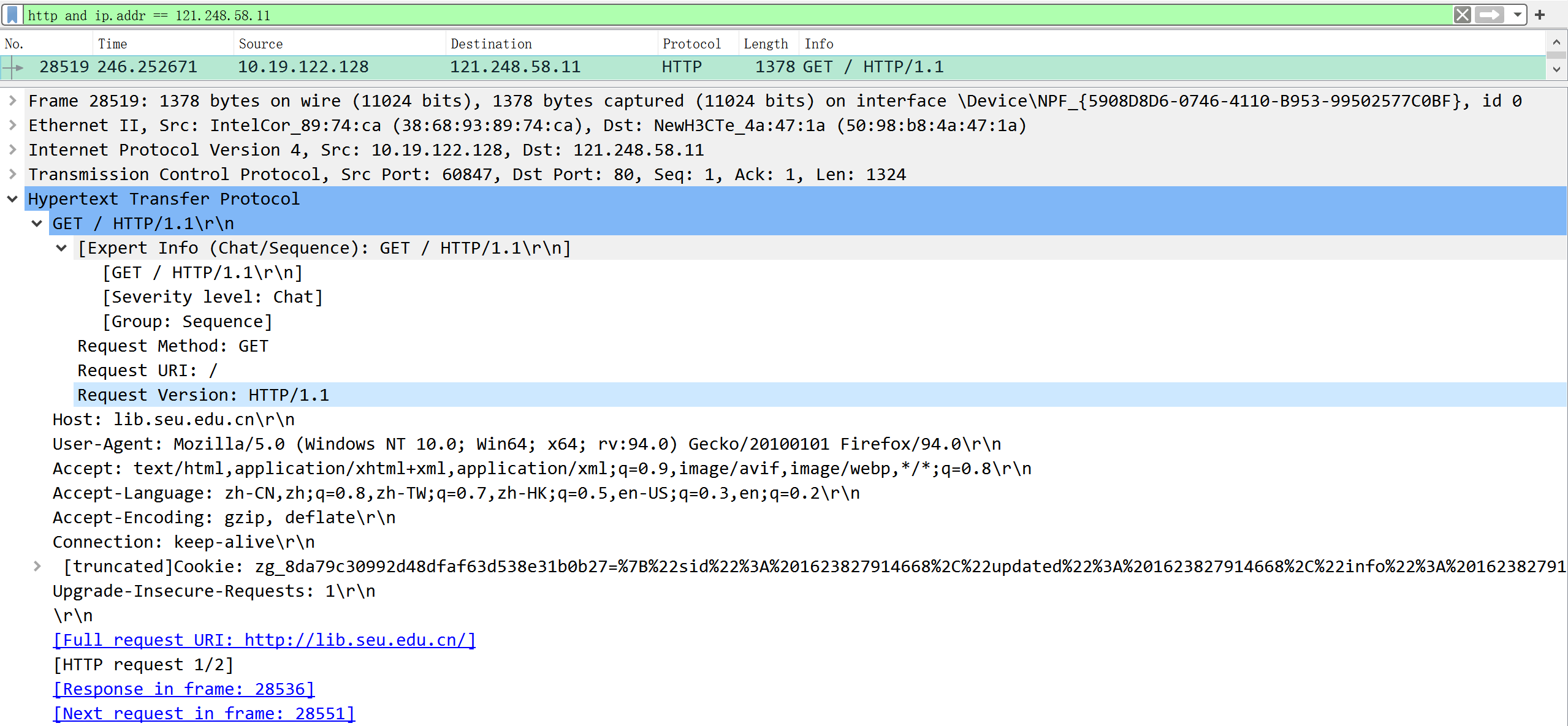


在开始分析之前，我修改显示过滤器为：ip.addr == 121.248.58.11

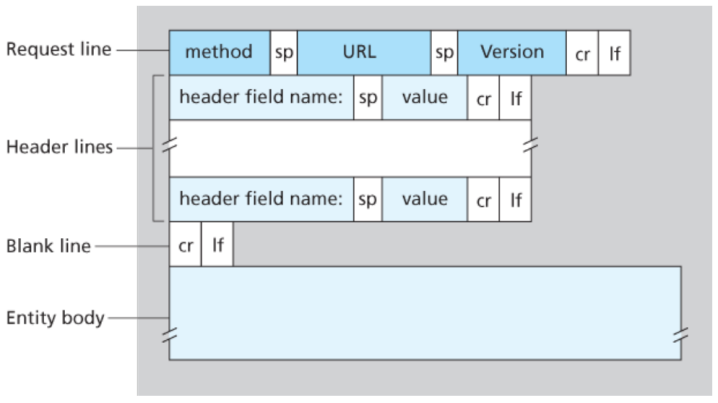


由此可见，HTTP工作流程为客户端通过TCP 三次握手与服务器建立连接，TCP 建立连接成功后，开始向服务器发送HTTP 请求。

开始分析第一个HTTP请求包：



对照HTTP请求报文的格式，逐句分析：



GET / HTTP/1.1\r\n 为请求行信息

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

[GET / HTTP/1.1\r\n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET 表示请求方法为GET

Request URI: / 表示请求的URI

Request Version: HTTP/1.1 表示请求的版本为HTTP/1.1

Host: 1ib.seu.edu.cn\r\n 表示请求的主机为东南大学图书馆网站

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0\r\n 为浏览器类型

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp, \*/\*;q=0.8\r\n 表示请求的类型

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 表示请求语言

Accept-Encoding: gzip, deflate\r\n 表示请求的编码格式

Connection: keep-alive\r\n 表示使用持久连接

\r\n 表示空行

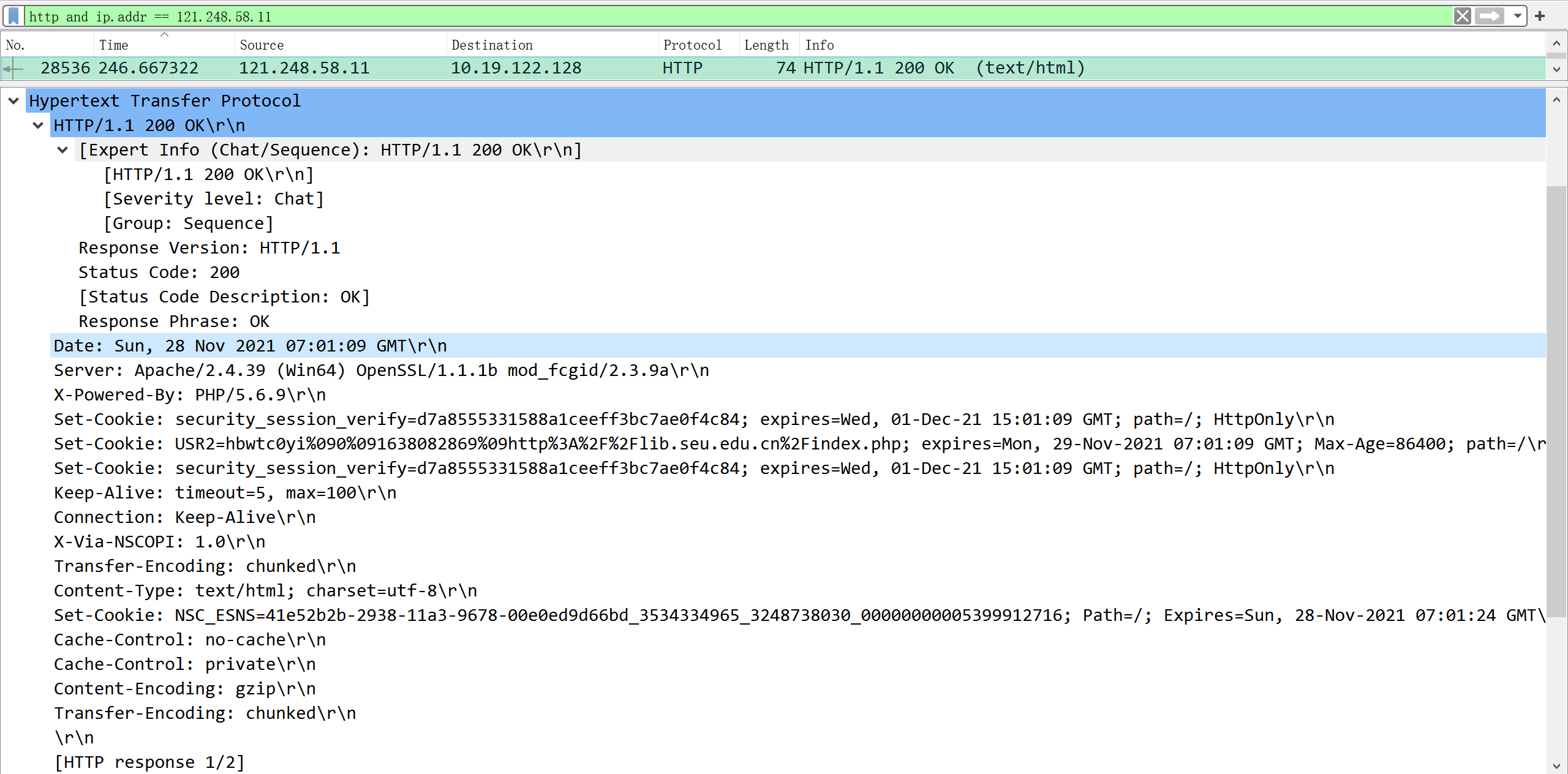
[Full request URI: http://1ib.seu.edu.cn/] 表示请求的URI为http://1ib.seu.edu.cn/

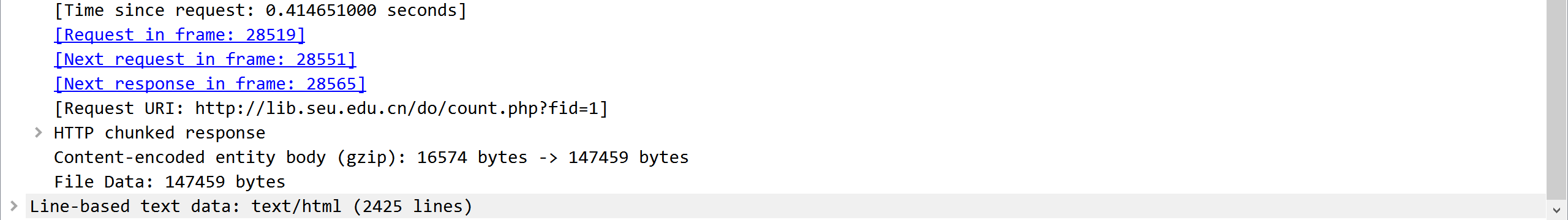
HTTP request 1/2

[Response in frame: 28536] 表示应答是第28536帧

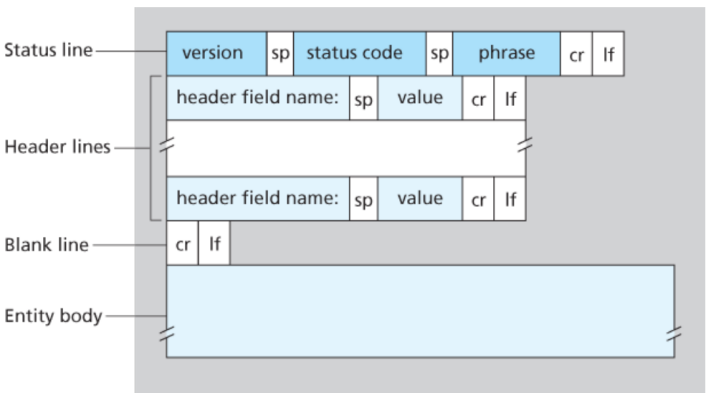
[Next request in frame: 28551] 表示下一个请求是第28551帧

开始分析第一个HTTP响应包：





对照HTTP响应报文的格式，逐句分析：



HTTP/1.1 200 OK\r\n 为响应行信息

[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n] 为专家信息

[HTTP/1.1 200 OK\r\n] 为HTTP响应信息，响应码为200

[Severity level: Chat]

[Group: Sequence]

Request Version: HTTP/1.1 为请求版本

Status Code: 200 为状态码

Response Phrase: OK 为响应短语

Date: Sun, 28 Nov 2021 07:01:09 GMT\r\n表示消息发送的时间  
 Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod\_fcgid/2.3.9a\r\n表示服务器类型

X-Powered-By: PHP/5.6.9\r\n表示网站是用PHP开发的

Set-Cookie: security\_session\_verify=d7a8555331588a1ceeff3bc7ae0f4c84; expires= Wed, 01-Dec-21 15:01:09 GMT; path=/; HttpOnly\r\n设置HTTP Cookie  
 Set-Cookie: USR2=hbwtc0yi%090%091638082869%09http%3A%2F%2Flib.seu.edu. cn%2Findex.php; expires=Mon, 29-Nov-2021 07:01:09 GMT; Max-Age= 86400; path=/\r\n  
 Set-Cookie: security\_session\_verify=d7a8555331588a1ceeff3bc7ae0f4c84; expires= Wed, 01-Dec-21 15:01:09 GMT; path=/; HttpOnly\r\n  
     Keep-Alive: timeout=5, max=100\r\n过期时间5秒，最多一百次请求强制断掉连接  
     Connection: Keep-Alive\r\n表示使用持久连接  
 X-Via-NSCOPI: 1.0\r\n  
 Transfer-Encoding: chunked\r\n 使用分块编码的编码传输  
 Content-Type: text/html; charset=utf-8\r\n 表示响应的内容类型  
 Set-Cookie: NSC\_ESNS=41e52b2b-2938-11a3-9678-00e0ed9d66bd\_3534334965\_ 3248738030\_00000000005399912716; Path=/; Expires=Sun,

28-Nov-2021 07:01:24 GMT\r\n  
 Cache-Control: no-cache\r\n 缓存控制  
 Cache-Control: private\r\n  
 Content-Encoding: gzip\r\n表示实体数据的压缩格式  
 Transfer-Encoding: chunked\r\n

\r\n表示空行

[HTTP response 1/2] 为HTTP响应

[Time since request: 0.414651000 seconds] 为响应使用的时间

[Request in frame: 28519] 表示请求的帧号为28519

[Next request in frame: 28551] 表示下一个请求的帧号28551

[Next response in frame: 28565] 表示下一个响应的帧号是28565

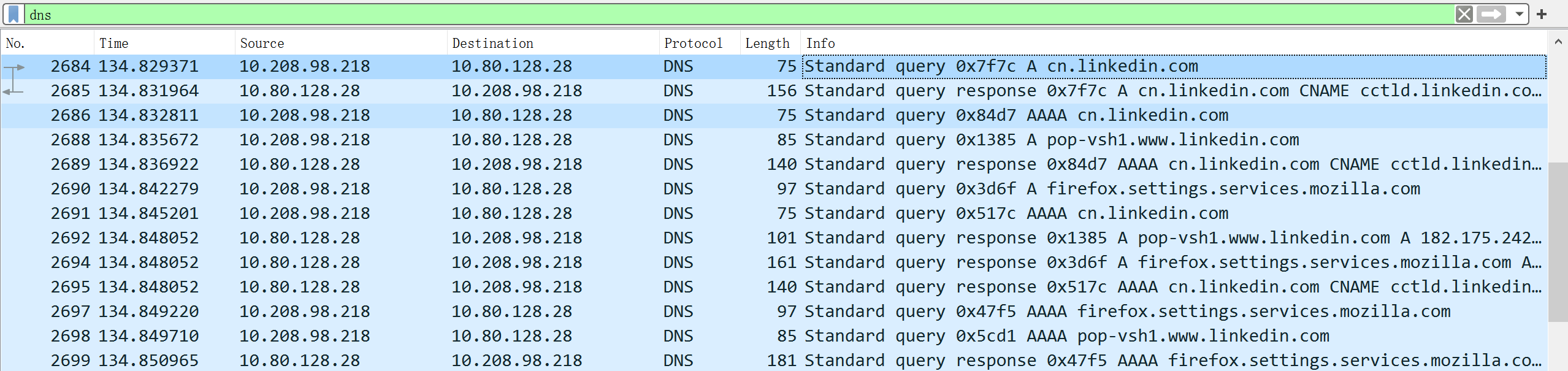
[Request URI: http://lib.seu.edu.cn/do/count.php?fid=1] 表示请求的URI

HTTP chunked response  
 Data chunk (6357 octets)  
 Data chunk (7709 octets)  
 Data chunk (2508 octets)  
 End of chunked encoding  
 \r\n  
 Content-encoded entity body (gzip): 16574 bytes -> 147459 bytes内容编码（gzip）  
 File Data: 147459 bytes  
 Line-based text data: text/html (2425 lines) 表示基于行的文本数据

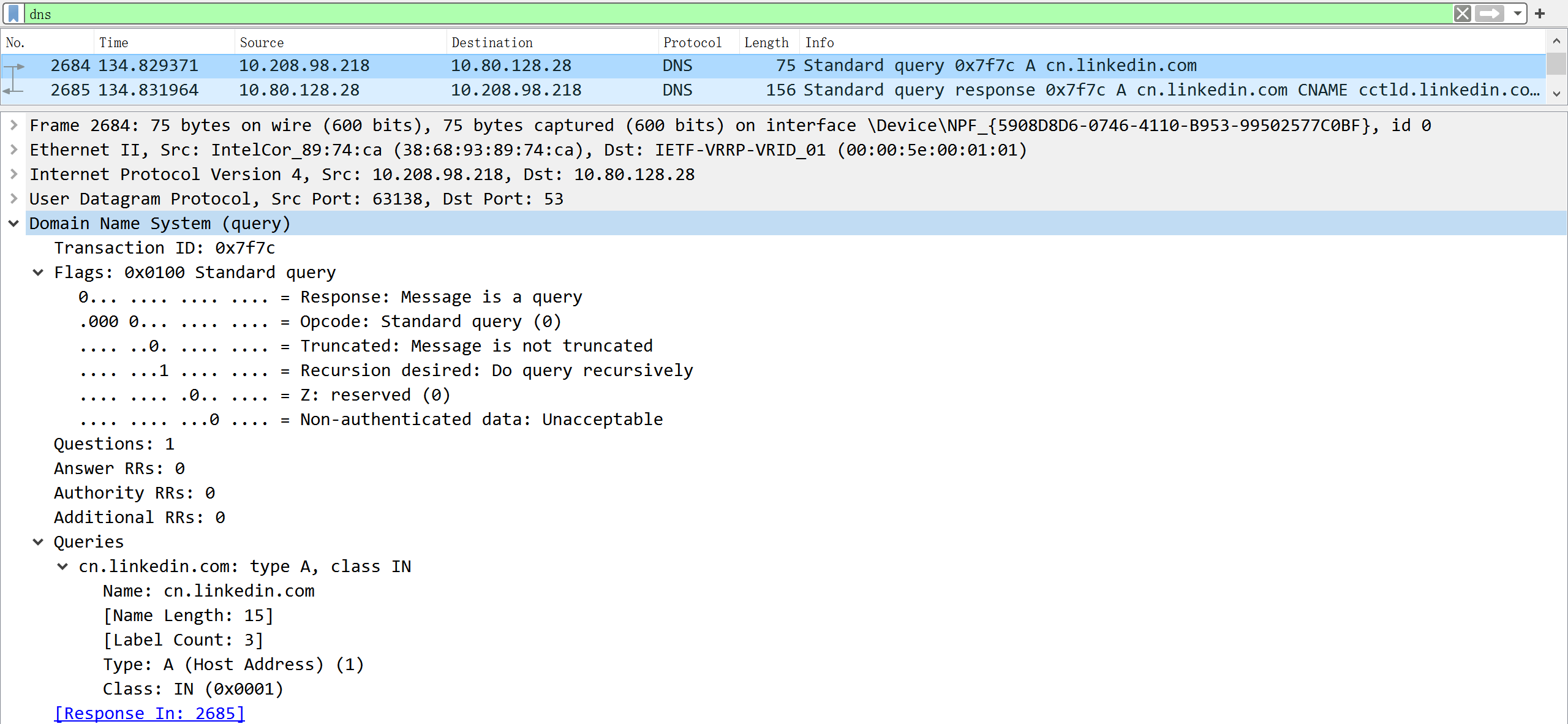
1. DNS协议分析
2. 打开Wireshark设置过滤器：dns



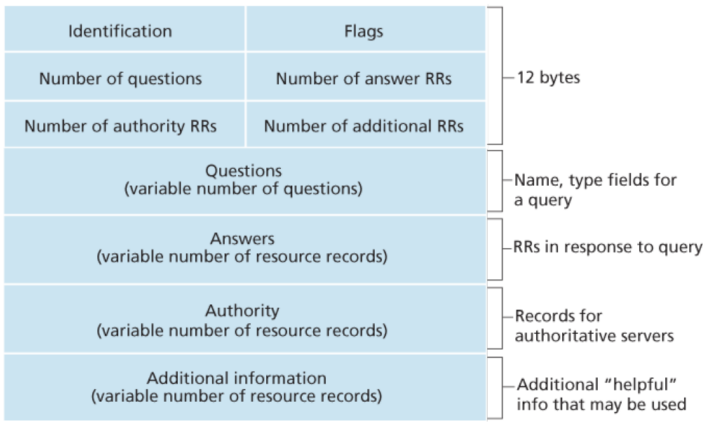
1. 打开浏览器访问领英www.linkedin.com, 浏览器打开网址开始抓包



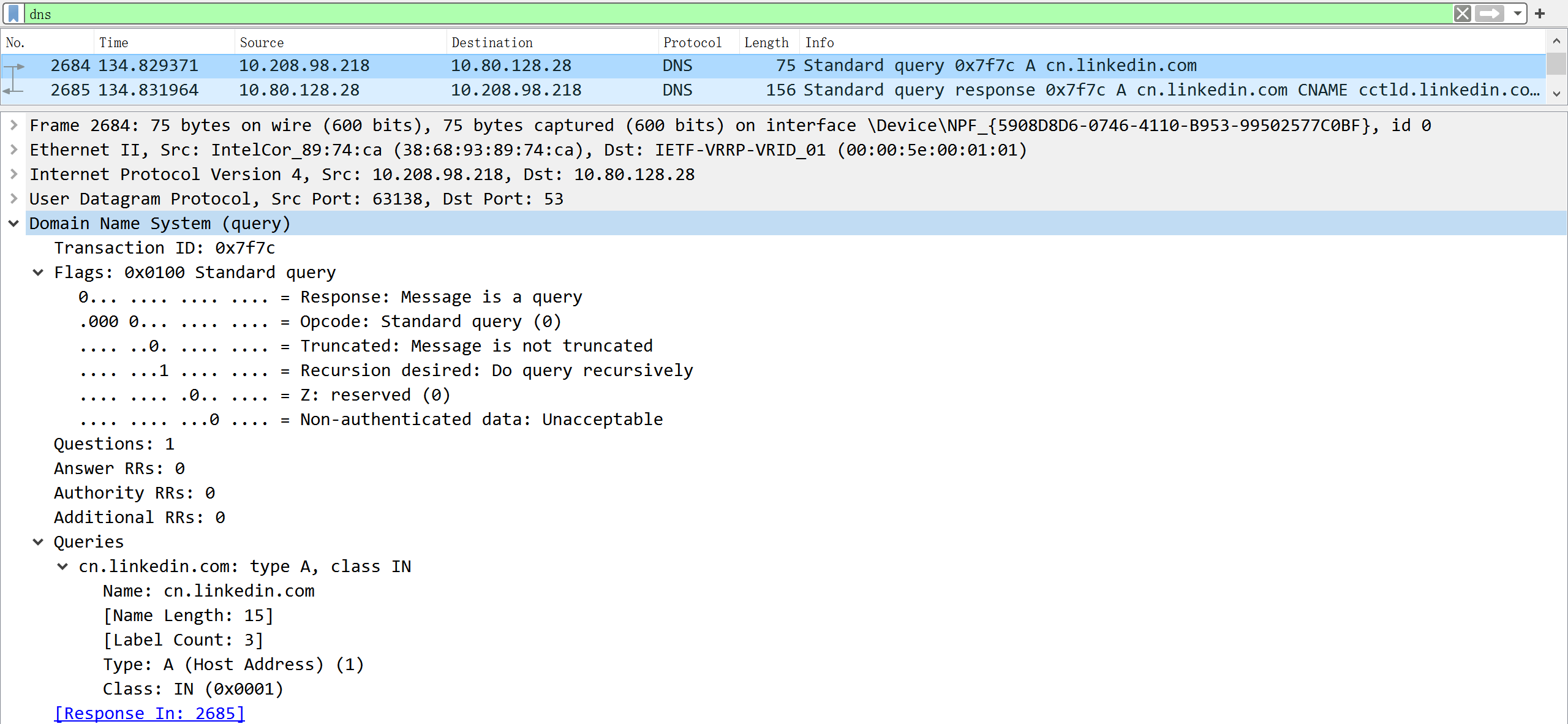
1. 这里，我选取客户端向本地DNS服务器发送的第一个查询报文以及其对应的回答报文分析。



首先，DNS只有两种报文：查询报文、回答报文，两者有着相同格式，如下：



开始分析DNS查询报文：



Transaction ID: 0x7f7c 标识字段，用于辨别DNS回答报文是哪个查询报文的响应  
     Flags: 0x0100 Standard query　递归请求  
 0... .... .... .... = Response: Message is a query 0表示查询

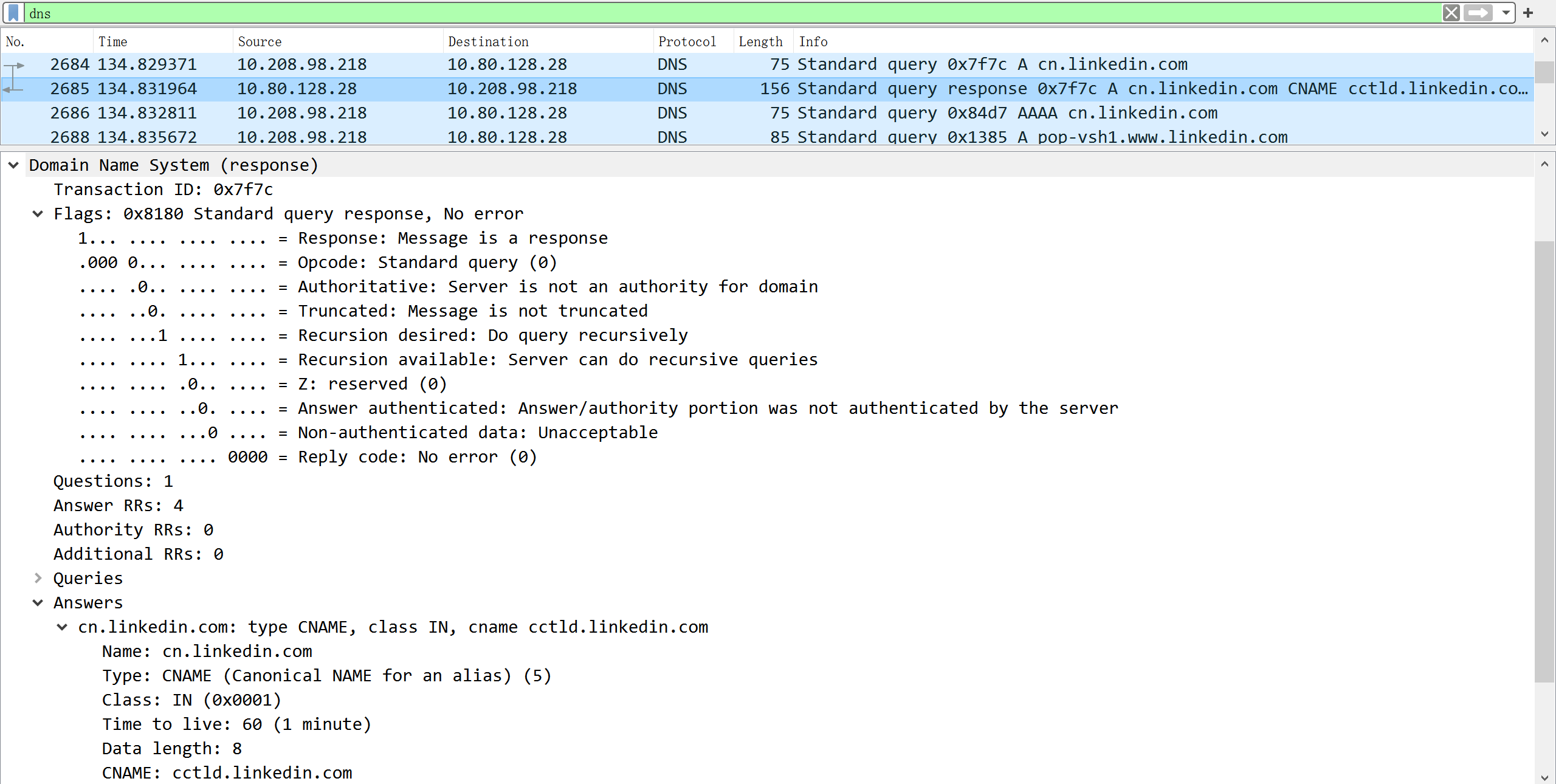
.000 0... .... .... = Opcode: Standard query (0) 表示查询或响应类型，0表示标准  
 .... ..0. .... .... = Truncated: Message is not truncated 截断，0表示没有发生截断  
 .... ...1 .... .... = Recursion desired: Do query recursively 是否希望得到递归回答  
 .... .... .0.. .... = Z: reserved (0) 保留字段  
 .... .... ...0 .... = Non-authenticated data: Unacceptable保留字段  
Questions: 1 问题数  
Answer RRs: 0 资源记录数

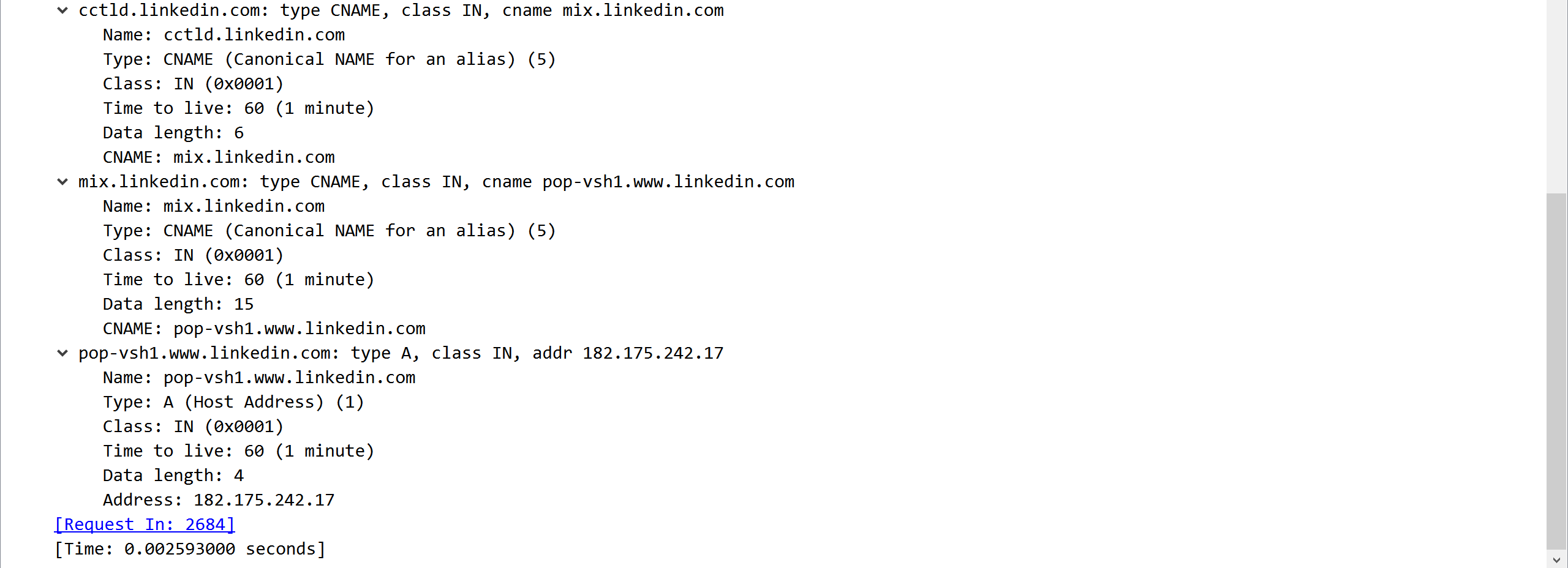
Authority RRs: 0 授权资源记录数

Additional RRs: 0 额外资源记录数  
 Queries 查询或者响应的正文部分  
         cn.linkedin.com: type A, class IN  
 Name: cn.linkedin.com 查询名称  
 [Name Length: 15]  
 [Label Count: 3]  
 Type: A (Host Address) (1) 查询类型，这里是主机A记录  
 Class: IN (0x0001) 类，IN表示Internet数据,通常为1  
 [Response In: 2685]

接着，分析DNS回答报文：

可以看到和查询报文相比，回答报文多出了一个Answers字段，同时Flags字段每一位都有定义。可见，Flags中Answer RRs为4说明对应的Answers字段中将会出现4项解析结果。





Transaction ID: 0x7f7c标识字段，可见该回答报文是上面查询报文的响应  
    Flags: 0x8180 Standard query response, No error

1... .... .... .... = Response: Message is a response 1 表示回答

.000 0... .... .... = Opcode: Standard query (0) 表示查询或响应类型，0表示标准

.... .0.. .... .... = Authoritative: Server is not an authority for domain 表示服务器不 是所请求名字的权威DNS服务器

.... ..0. .... .... = Truncated: Message is not truncated截断，0表示没有发生截断

.... ...1 .... .... = Recursion desired: Do query recursively 表示对应的查询报文是 递归请求

.... .... 1... .... = Recursion available: Server can do recursive queries表示递归可用

.... .... .0.. .... = Z: reserved (0) 保留字段

.... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server保留字段

.... .... ...0 .... = Non-authenticated data: Unacceptable 保留字段

.... .... .... 0000 = Reply code: No error (0) 返回码表示响应的差错状态

Questions: 1 问题数  
 Answer RRs: 4 回答数  
 Authority RRs: 0 授权资源记录数  
 Additional RRs: 0额外资源记录数  
 Queries 同查询报文处，故折叠  
 Answers 回答问题区域字段  
 cn.linkedin.com: type CNAME, class IN, cname cctld.linkedin.com资源记录，

类型CNAME表示能够向请求主机提供一个主机名对应的规范主机名  
 Name: cn.linkedin.com 主机别名  
 Type: CNAME (Canonical NAME for an alias) (5)   
 Class: IN (0x0001)  
 Time to live: 60 (1 minute) 表示该资源记录的生命周期 Data length: 8资源数据长度  
 CNAME: cctld.linkedin.com  
 cctld.linkedin.com: type CNAME, class IN, cname mix.linkedin.com  
 Name: cctld.linkedin.com  
 Type: CNAME (Canonical NAME for an alias) (5)  
 Class: IN (0x0001)  
 Time to live: 60 (1 minute)  
 Data length: 6  
 CNAME: mix.linkedin.com

mix.linkedin.com: type CNAME, class IN, cname pop-vsh1.www.linkedin.com  
 Name: mix.linkedin.com  
 Type: CNAME (Canonical NAME for an alias) (5)  
 Class: IN (0x0001)  
 Time to live: 60 (1 minute)  
 Data length: 15  
 CNAME: pop-vsh1.www.linkedin.com

　　pop-vsh1.www.linkedin.com: type A, class IN, addr 182.175.242.17 资源记录，

类型A表示提供了标准的主机名到IP地址的映射

Addr(资源数据)：返回的IP地址

Name: pop-vsh1.www.linkedin.com  
 Type: A (Host Address) (1)

Class: IN (0x0001)

Time to live: 60 (1 minute)

Data length: 4

Address: 182.175.242.17

[Request In: 2684]

[Time: 0.002593000 seconds]

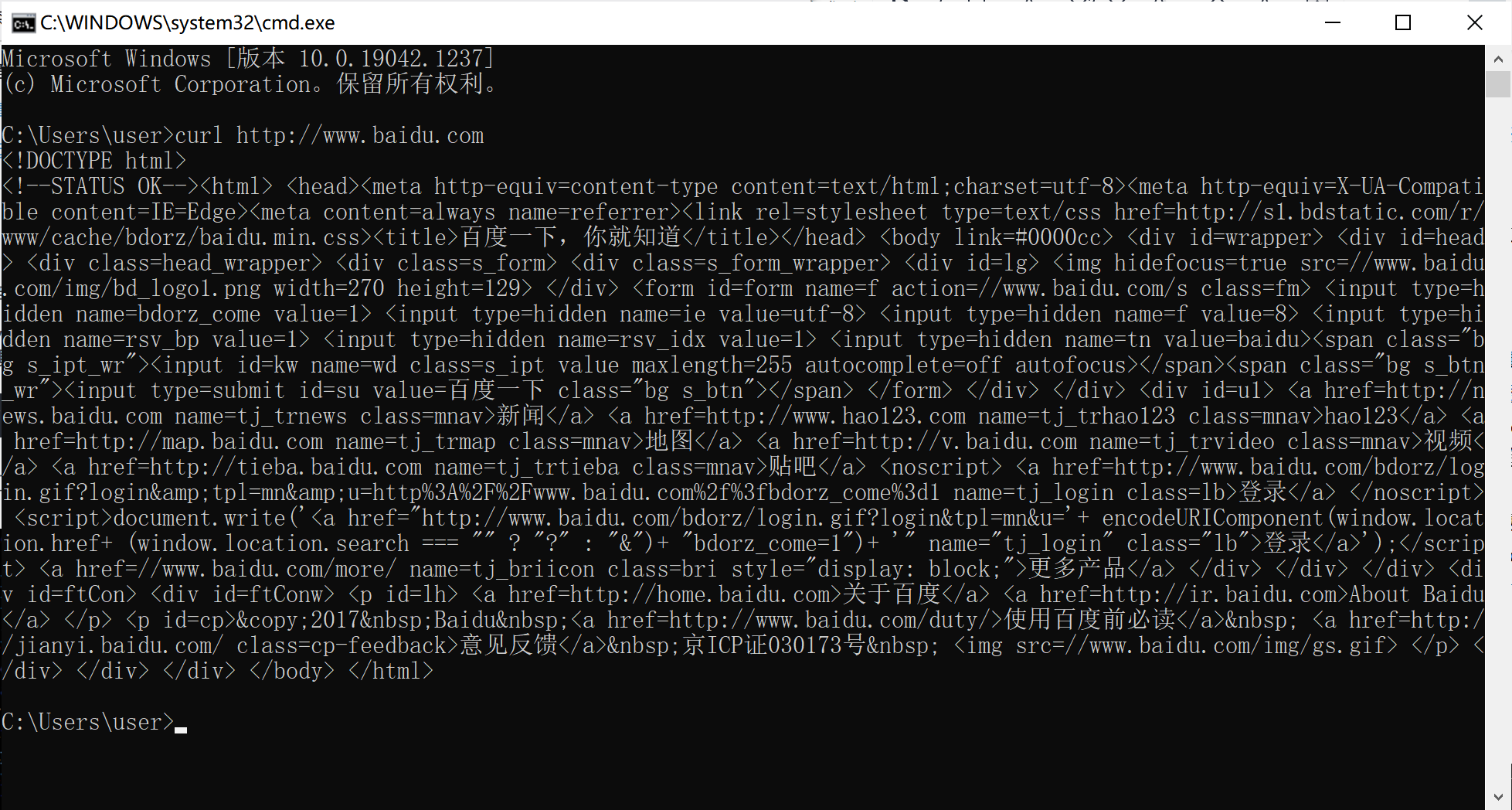
1. **curl命令测试**

curl全写是：CommandLine Uniform Resource Locator， 命令行统一资源定位器，是使用命令行访问网页URL的工具。

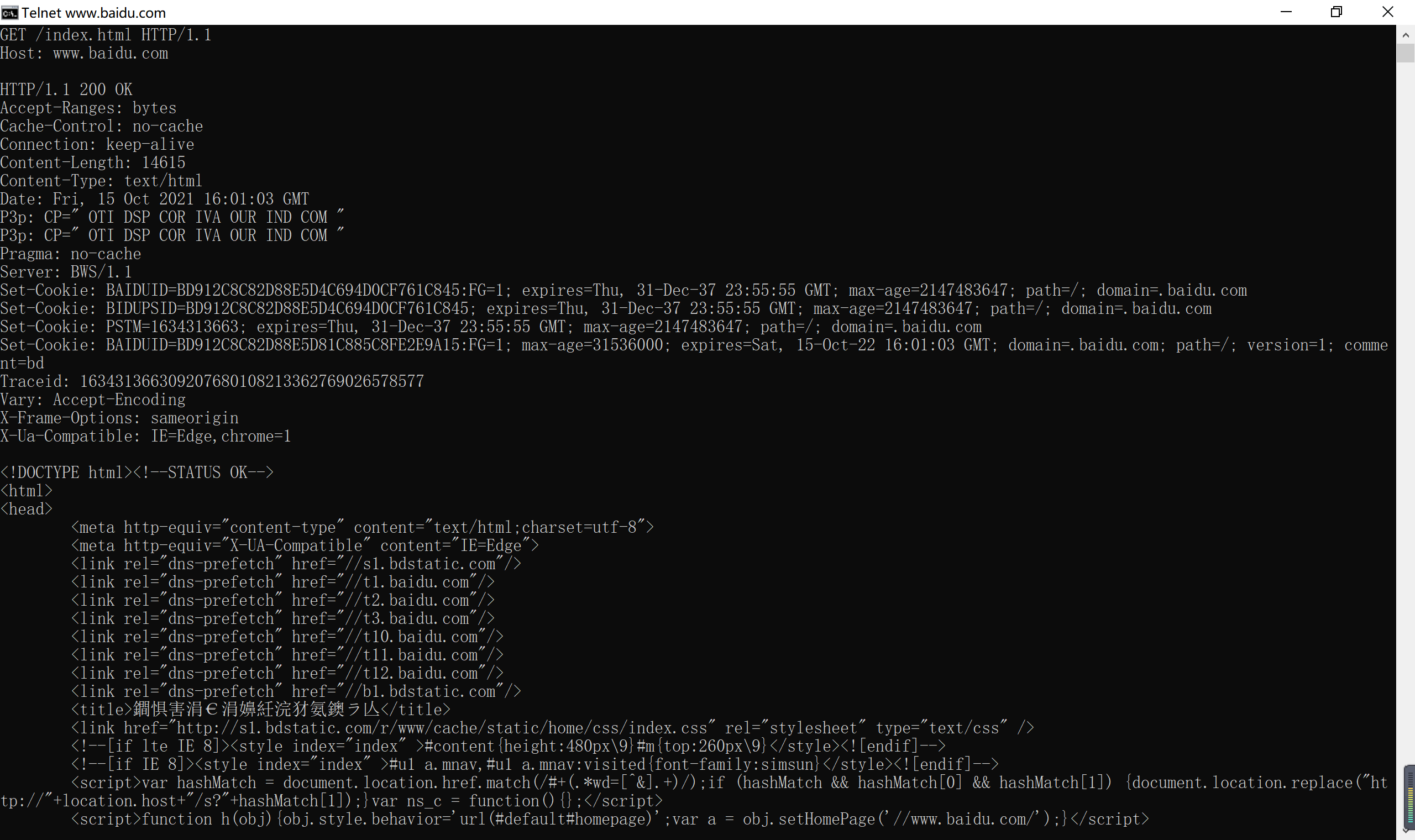
**测试：**

cmd中输入：curl [www.baidu.com](http://www.baidu.com)

稍好片刻，终端会返回Web网站的响应源代码，效果如下:

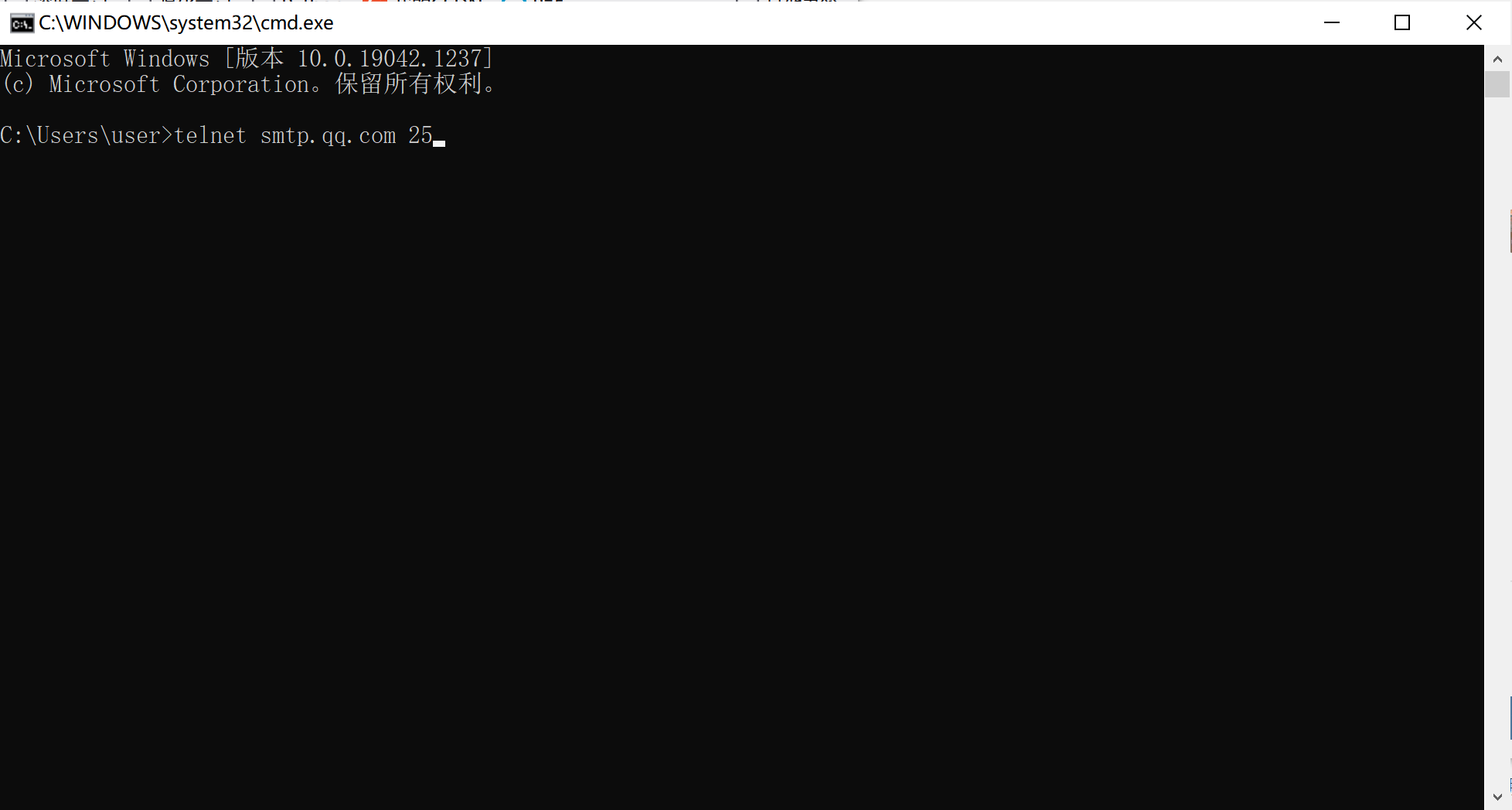


1. **telnet命令测试get命令**
2. 用win+r打开cmd
3. 在cmd中执行telnet www.baidu.com 80， 然后可以看到一个黑色的框框
4. 按 ctrl + ] 退出，结果为：  
                 欢迎使用 Microsoft Telnet Client  
                     Escape 字符是 'CTRL+]'  
                     Microsoft Telnet>
5. 按enter, 进入到输入框
6. 输入如下内容（有时间限制，因此先写好然后整体拷贝进去）  
   GET /index.html HTTP/1.1  
   Host: [www.baidu.com](http://www.baidu.com)
7. 连续按两下enter键盘， 得到结果如下：

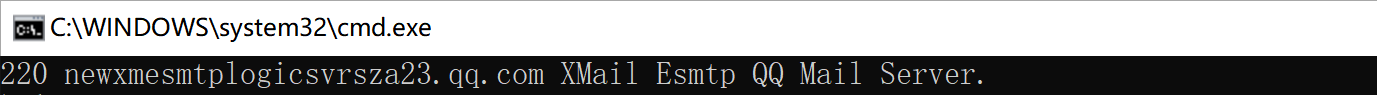


1. **telnet命令测试SMTP服务**
2. cmd中输入：telnet smtp.qq.com 25

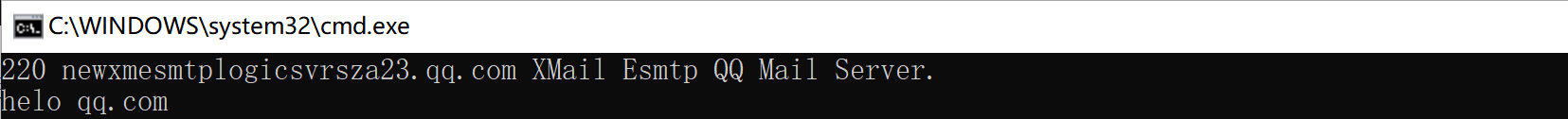
客户端TCP连接邮件服务器25 端口，如下图：



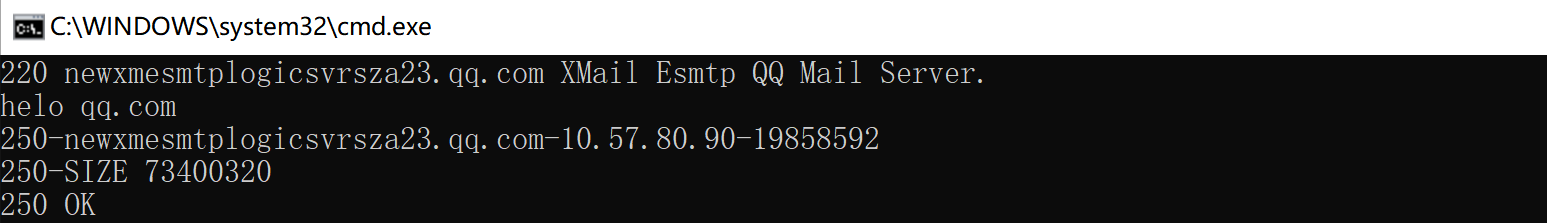
1. 三次握手以后，连接建立成功，SMTP服务器（邮件服务器S）发送服务就绪信息，这里220代表服务就绪，后接服务器的主机名；



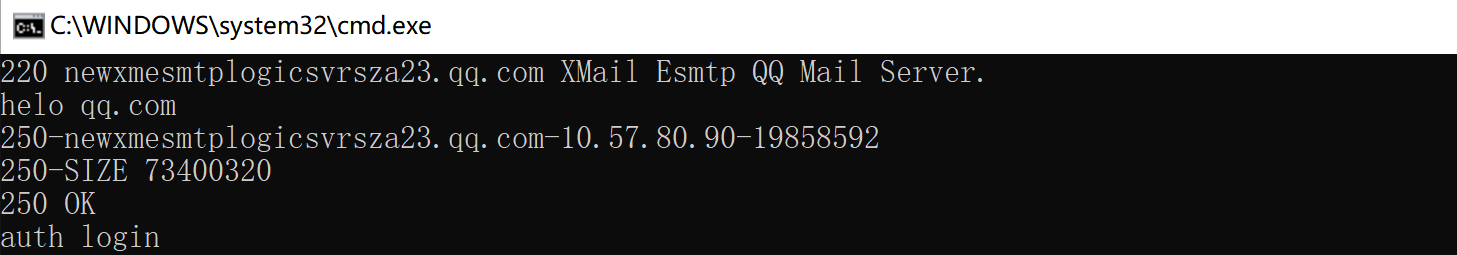
1. 客户端通过helo命令向服务器表明身份，交代自己认证SMTP服务器的域名，这里采用我自己的QQ邮箱；



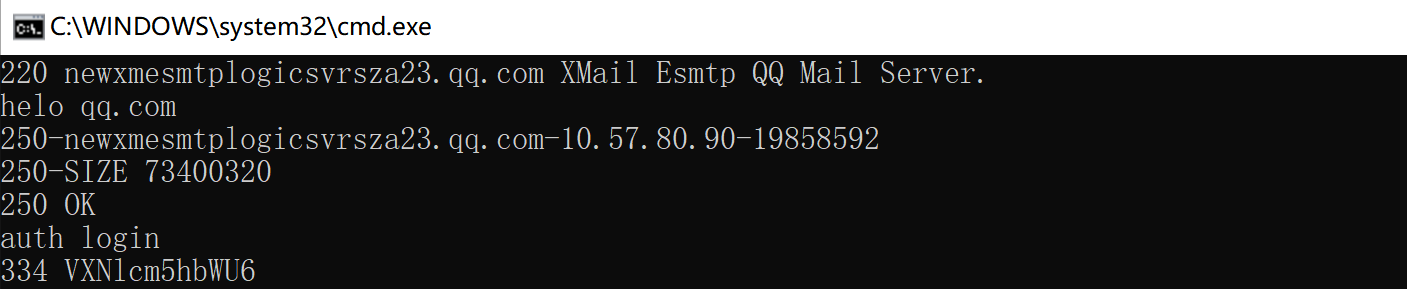
1. 如果身份有效，则服务器进入等待认证状态，下面三行是QQ邮箱发送的内容；



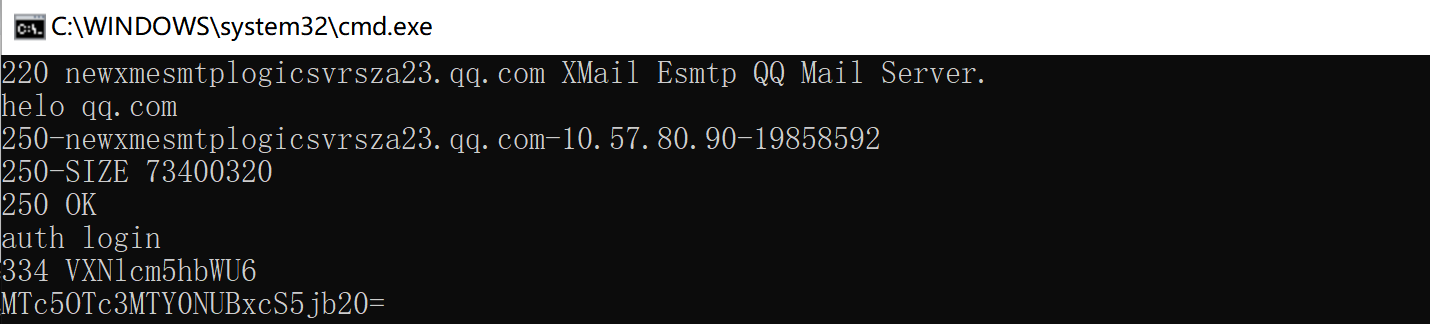
1. 客户端发送auth login，向服务器请求认证；



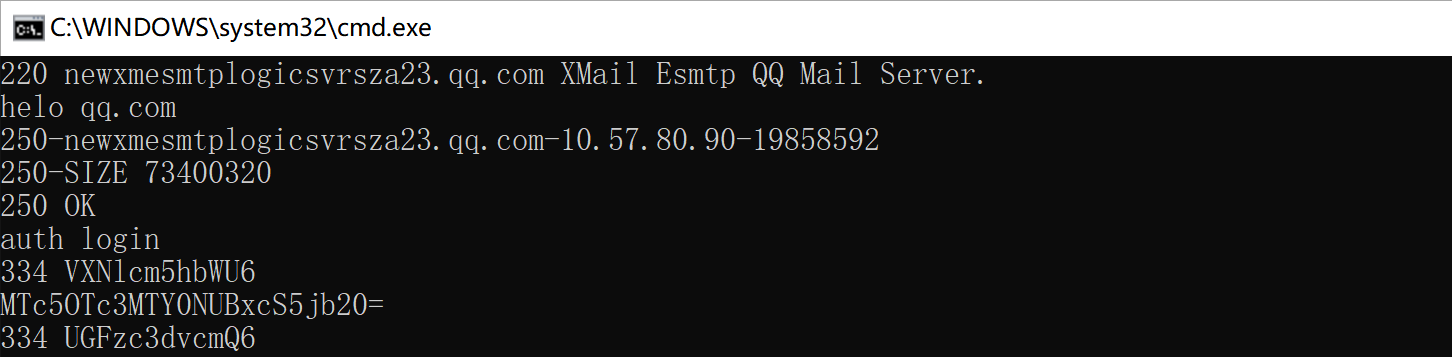
1. 如果认证请求合理，服务器将进入等待用户输入状态，这里334表示等待客户端输入，VXNlcm5hbWU6表示等待输入用户名；



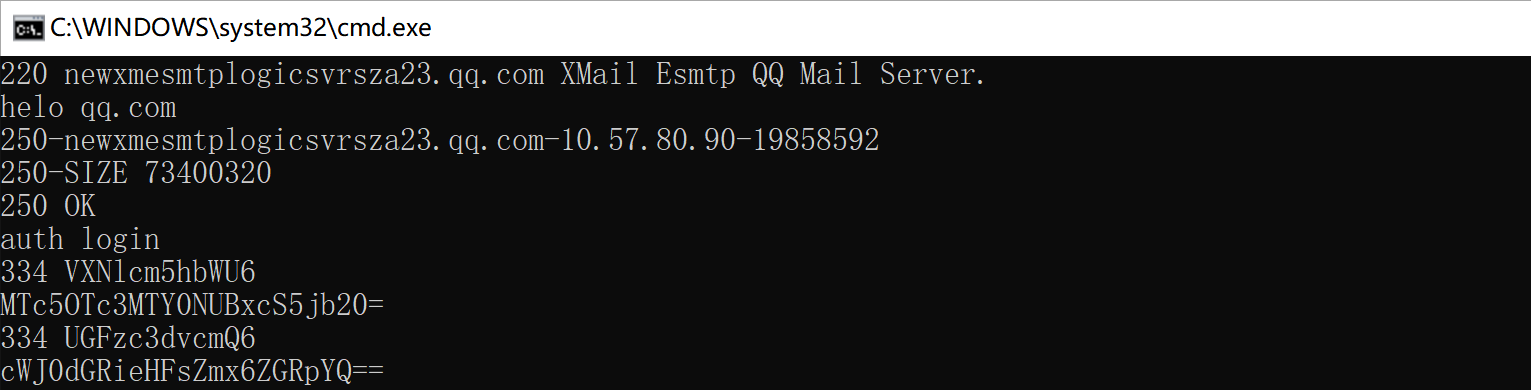
1. 客户端向服务器发送Base64编码后的QQ邮箱用户名(caozouying@qq.com)；



1. 服务器再次进入等待用户输入状态,这里334表示等待客户端输入，UGFzc3dvcmQ6表示等待输入[密码](http://www.cpplive.com/html/tag/password" \o "查看 密码 中的全部文章" \t "https://blog.csdn.net/woshinia/article/details/_blank)；



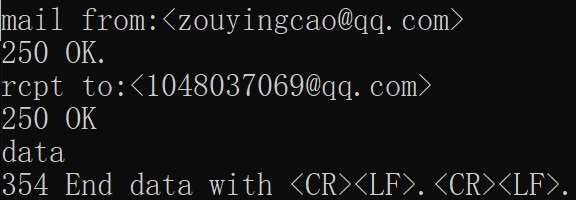
1. 客户端向服务器发送Base64编码后的密码(开启IMAP/SMPT时授权码的base64编码)；



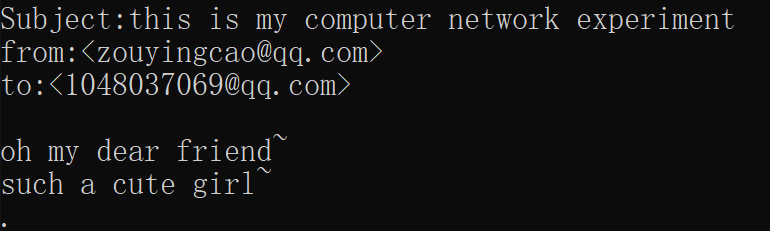
1. 服务器返回235 Authentication successful，表示认证成功；



1. 客户端通过mail from命令告诉服务器邮件来自何方；
2. 服务端返回250 OK表示成功；
3. 客户端通过rcpt to命令告诉服务器邮件去往何地；
4. 服务端返回250 OK表示成功；
5. 客户端通过data命令告诉服务器自己准备发送邮件正文；
6. 服务器返回354表示准备接受邮件并提醒客户端开始发送邮件并以“.”结束；



1. 客户端发送邮件正文；客户端发送完正文以后，紧接着发送结束符“.”；



1. 如果合理，服务端返回“250 OK:queued as.\r\n”表示发送成功；
2. 客户端通过quit命令表示邮件发送结束，客户端请求断开连接；
3. 服务器返回“211 Bye.”表示断开申请被采纳并主动断开连接，邮件发送过程结束。



查看邮件服务结果：



1. **tracert命令测试**
2. **cmd中输入**：tracert [www.baidu.com](http://www.baidu.com)

表示追踪当前IP到www.baidu.com域名指向的IP地址所经过的路由地址列表。



首先，会自动将www.baidu.com域名找到其对应的ip地址——112.80.248.76，并提示到达目的地址的路由跃点估算——最多30个；

接下来，每一行为所经过的一个路由地址，包括序号、3次实验的往返时延、路由IP；其中出现序号后面是\*号，且有请求超时的提示，可能原因是路由跃点禁PING或者路由跃点不对TTL超时做响应处理，直接丢弃；

最后提示追踪完成表示命令执行完毕。

1. **解析原理过程：**

tracert 命令用 IP生存时间 (TTL) 字段和ICMP错误消息来确定从一个主机到网络上其他主机的路由。

首先，tracert发出一个TTL=1的数据包到目的地，当路径上的第一个路由器收到这个数据包时，它将其TTL减1。此时，TTL变为0，所以该路由器会将此数据包丢掉，并送回一个「ICMP time exceeded」消息（包括发IP包的源地址，IP包的所有内容及路由器的IP地址），tracert 便知道这个路由器为追踪路径上的第一个路由，接着tracert 再发出一个TTL=2的数据包，发现第2 个路由器......

就这样，tracert 每次将发出的数据包TTL加1来发现路径中下一个路由器，这个重复的动作一直持续到某个数据包抵达目的地。当数据包到达目的地后，该主机则不会返回「ICMP time exceeded」消息，一旦到达目的地，由于tracert通过UDP数据包向不常见端口(30000以上)发送数据包，因此会收到「ICMP port unreachable」消息，故可判断到达目的地。

1. **nslookup查询**

Nslookup全称为name server lookup(域名查询)，

**测试：**

1. 直接查询：cmd中输入：nslookup www.baidu.com



1. 查询其它记录：

nslookup -qt = type domain [dns-server]

选择type：MX→邮件服务器记录

cmd中输入：nslookup -qt=mx www.baidu.com



1. **实验体会**

在Wireshark实验中，选择好过滤器，不然会有很多妨碍观察HTTP/DNS协议的数据包，增加无谓的寻找时间。分析HTTP协议时，还可以将访问的IP地址作为过滤条件。在使用Wireshark抓取HTTP和DNS数据包时，对HTTP与DNS报文内容有了更全面的学习与了解。

同时，学习了多种在命令行中经常使用的命令，例：curl与服务器之间传输数据，telnet测试get命令以及远程登录SMTP服务器发邮件，tracert追踪到达某个网站的路由信息，nslookup从域名中解析IP地址等。