**暨南大学本科实验报告专用纸**

课程名称 计算机网络 指导教师 潘冰 成绩

实验项目名称 Internet应用与应用层协议分析 实验项目编号

实验项目类型 验证 实验地点 计算机网络实验教室 学院 专业 智科院19级信息安全

学生姓名 李媛 学号 2019050385

实验时间 2021 年 9 月 30 日

**1.【实验目的】**

* + 理解WWW 、 DNS服务、FTP服务、SMTP的作用和原理；
  + **学会使用wireshark分析HTTP、FTP、SMTP和DNS协议的工作过程，加深对协议格式和工作原理的理解。**

**2.【实验内容】**

* + 通过域名访问WWW、FTP服务器，分析DNS、WWW、FTP工作过程，并使用WireShark分析相关协议格式；
  + 在客户端访问SMTP服务器，使用wireshark分析SMTP、POP3协议的工作过程。（可以在客户端安装outlook或使用QQ邮件服务器或自己编程）

**3.【实验设备】**

局部网环境，计算机若干台。本实验不分组，独立完成。

**4.【实验步骤】**

**一、HTTP协议分析**

访问任意web站点，用wireshark分析HTTP协议的工作过程和HTTP协议格式。（协议首部含义需要查询了解）

**注意：**

如果访问的网站信息量太大，可以利用windows自带组件（如IIS）或自行下载WWW服务器软件，建立并配置WWW站点，主页内容需要包含姓名、学号等个人信息，然后访问自己建立的网站。

1. **DNS协议分析**

通过域名访问某网站，用wireshark捕获DNS数据包，并分析DNS工作过程和DNS格式。

**注意：**

1. 有时候可能捕获不到DNS解析过程，为什么？要仔细分析。
2. 可以自己建立DNS服务器并使用它进行域名解析。

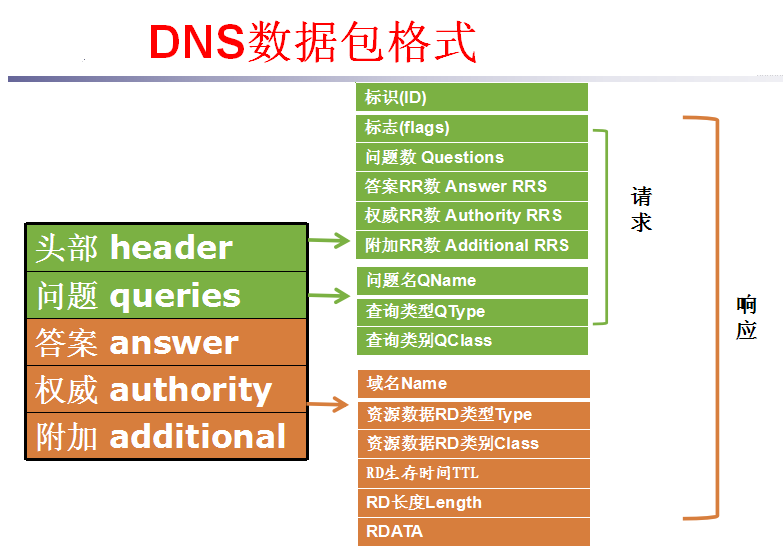
1）建立DNS服务器。

利用windows自带组件或下载DNS软件（如**ntbind**），建立DNS服务器，并对DNS服务器的配置（建立域名-IP地址的解析关系）。

2）使用DNS

在任意一台计算机上配置TCP/IP协议，将DNS服务器配置成上述DNS服务器（注意该DNS服务器的IP地址是静态的），用该DNS服务器中设置的器域名访问该WWW服务，并与用IP地址访问的结果进行比较。（注意该客户机的IP地址最好与DNS服务器地址同一个网段，即网络号相同）

3、附DNS数据包格式（细节见课件）



**三、FTP协议分析**

**1.**访问FTP服务器。如ftp://ftp.jnu.edu.cn

2.用wireshark分析FTP的工作过程。注意观察FTP的工作模式，用于控制连接的端口和数据连接的端口。

**注意：**可以利用Windows自带的FTP组件或下载FTP软件，建立FTP服务器，并进行配置，并访问分析FTP协议

**四、 SMTP和POP协议分析**

基于Web的邮件或客户端的邮件软件（如outlook）收发邮件，捕获数据报分析邮件收发过程和SMTP、POP3等协议格式和工作过程。

**注意：**

1、可以利用windows自带组件安装SMTP服务器，并配置邮件服务器，并访问。

2、 什么情况下可以捕获POP或SMTP数据包

**五.【实验过程】**

**5.1.http协议分析**

**5.1.1 解析URL**

分析网址URL：[http://www.chinanews.com](http://www.chinanews.com/shipin/cns/2021/10-14/news903896.shtml)[/shipin/cns/2021/10-14/news903896.shtml](http://www.chinanews.com/shipin/cns/2021/10-14/news903896.shtml)

HTTP 协议使用 URI 定位互联网上的资源。在互联网上任意位置的资源都能访问到。URL 带有请求对象的标识符。

第一步，观察URL的组成，尝试从URL了解一些访问对象的一些信息：

1. http://：告诉我的web浏览器使用的是http协议。对于大部分 Web 资源，通常使用 HTTP 协议或其安全版本，如HTTPS 协议。

2. [www.chinanews.com](http://www.chinanews.com)：这是是一个域名，也代表管理该域名的机构，它指示了需要向网络上的哪一台主机发起请求。这里说明我要访问的网页属于中国新闻网，这是中国新闻网的域名。

3.端口号: 80（已省略）：两个主机之间要发起 TCP 连接需要两个条件，主机 + 端口。它表示用于访问 Web 服务器上资源的入口。

Web 服务器使用HTTP协议的标准端口（HTTP为80，HTTPS为443）授予对其资源的访问权限，通常省略此部分。否则端口就是 URI 必须的部分。

4. /shipin/cns/2021/10-14/news903896.shtml：这是我即将要访问的资源在相应的Web 服务器上的路径。

以端口后面的第一个 / 开始，到 ? 号之前结束，中间的 每一个/ 都代表了层级（上下级）关系。这个 URL 的请求资源是一个 shtml 页面。

5. 紧跟着路径后面的是提供给 Web 服务器的额外 查询参数和锚点，在这条网址中没有特殊的查询参数和锚点。

于是，能否尝试去找几个拥有额外的查询参数和锚点的链接？例如：<https://so.csdn.net/so/search?q=http&t=&u=，q=http&t=&u=> 就是额外的查询参数，这些参数是用 & 符号分隔的键/值对列表。再例如：<http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument>中的#SomewhereInTheDocument 是资源本身的某一部分的一个锚点。锚点代表资源内的一种“书签”，它给予浏览器显示位于该“加书签”点的内容的指示。

**5.1.2 思考http请求响应过程**

了解了URL的组成，我就开始回顾应用层的知识，思考当输入网址按下回车会有的操作，只有对整个过程有一个通透的了解才是实验的基础，这也是接下来的DNS的实验的一个预备：

1. DNS服务器会首先进行域名的映射，找到www.chinanews.com所在的ip地址，然后HTTP 客户端进程在 80 端口发起一个到服务器 www.chinanews.com 的 TCP 连接（80 端口是 HTTP 的默认端口）。连接成功后，在客户和服务器进程中都会有一个套接字与其相连。

补充知识：TCP进行三次握手，建立一个TCP连接时，需要客户端和服务器总共发送3个包。 三次握手的目的是连接服务器指定端口，建立TCP连接,并同步连接双方的序列号和确认号并交换 TCP 窗口大小信息.

2. HTTP 客户端通过它的套接字向服务器发送一个 HTTP 请求报文。该报文中包含了请求资源news903896.shtml的资源路径，而 HTTP 请求报文就需要接下来使用wireshark进行捕获并详细分析，这部分是实验一的重点。

3. HTTP 服务器通过它的套接字接受该报文，进行请求的解析工作，并从其存储器(RAM 或磁盘)中根据资源路径检索出对象 /shipin/cns/2021/10-14/news903896.shtml然后把检索出来的对象news903896.shtml进行封装，封装到 HTTP 响应报文中，并通过套接字向客户进行发送。同样，HTTP 响应报文就需要使用wireshark进行捕获和分析。

4. HTTP 服务器随即通知 TCP 断开 TCP 连接，实际上是需要等到客户接受完响应报文后才会断开 TCP 连接。

补充知识：TCP进行四次挥手。TCP的连接的拆除需要发送四个包。

5. HTTP 客户端接受完响应报文后，TCP 连接会关闭。HTTP 客户端从响应中提取出报文中是一个 HTML 响应文件，并检查该 HTML 文件，然后循环检查报文中其他内部对象。

6. 检查完成后，HTTP 客户端会把对应的资源通过显示器呈现给用户。

**5.1.3 查找资料详细了解http请求和响应的报文格式**

查找课外资料详细了解http的请求响应格式，了解了课堂上很多没有介绍的请求和响应方法的细致内容，参考博客<https://cxuan.blog.csdn.net/article/details/103930478>

**总结大致如下：**

**1.http请求报文格式**

开始行：区分请求(请求行)和响应（状态行）。

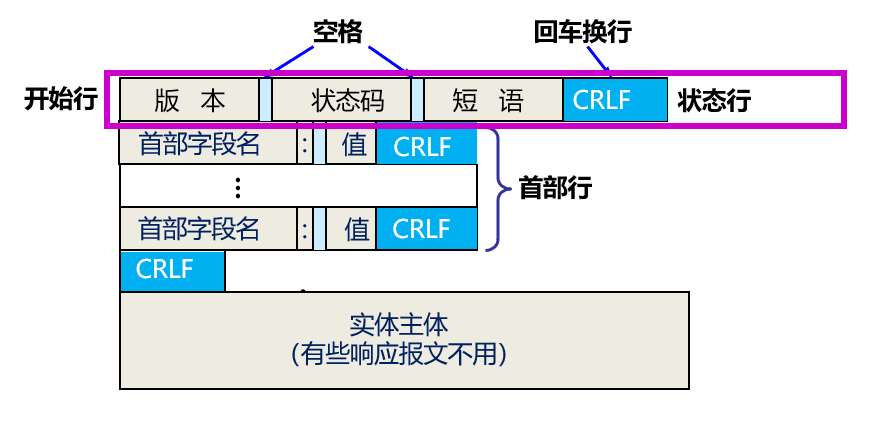
首部行：说明浏览器、服务器或报文主体的一些信息。

实体主体：请求报文中一般不用，响应报文也可不用。

2. **HTTP响应报文格式**

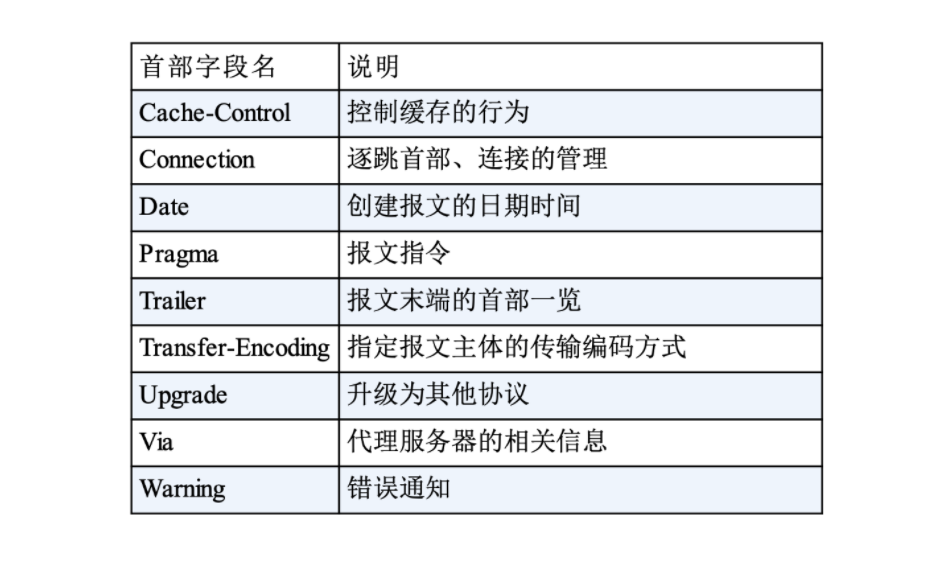
响应报文的开始行是状态行。

状态行包括三项内容，即 HTTP 的版本，状态码，以及解释状态码的简单短语。

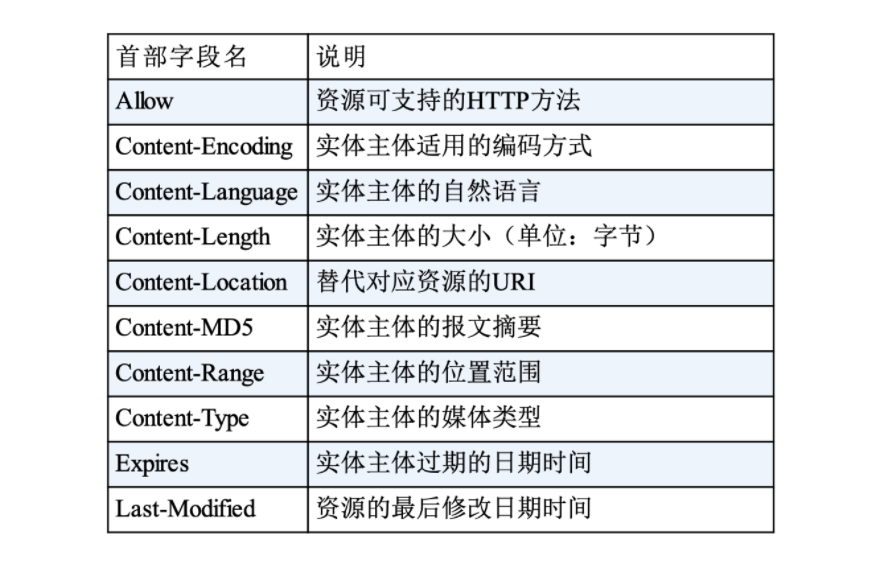


**3.** **HTTP 的请求标头分为四种： 通用标头、请求标头、响应标头 和 实体标头**

**通用标头：它可以出现在请求标头和响应标头中：Date、Cache-Control 和 Connection，及下图**



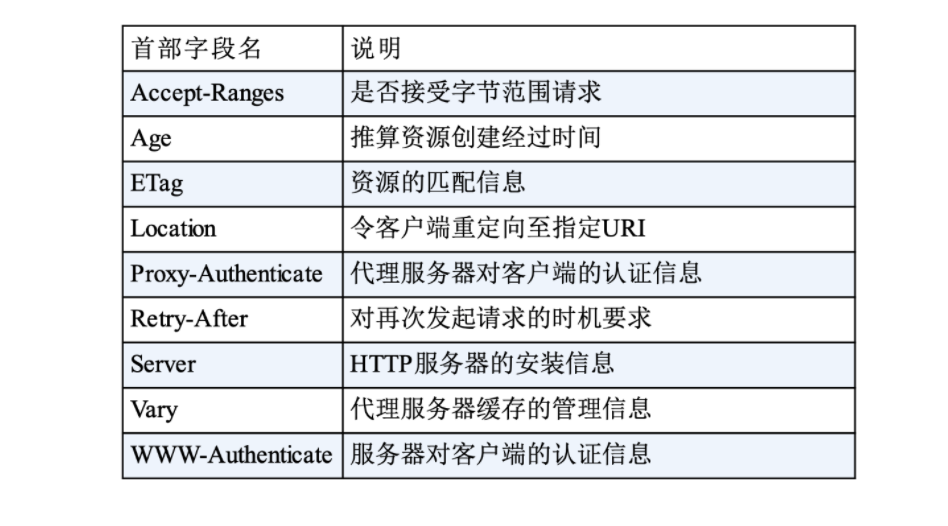
**实体标头：实体标头是描述消息正文内容的 HTTP 标头。实体标头用于 HTTP 请求和响应中。头部Content-Length、 Content-Language、 Content-Encoding 是实体头，以及：**

****

**请求标头：出现在请求报文中，如Host，Referer，Upgrade-Insecure-Requests，If-Modified-Since，If-None-Match，ETag，Accept，如下：**

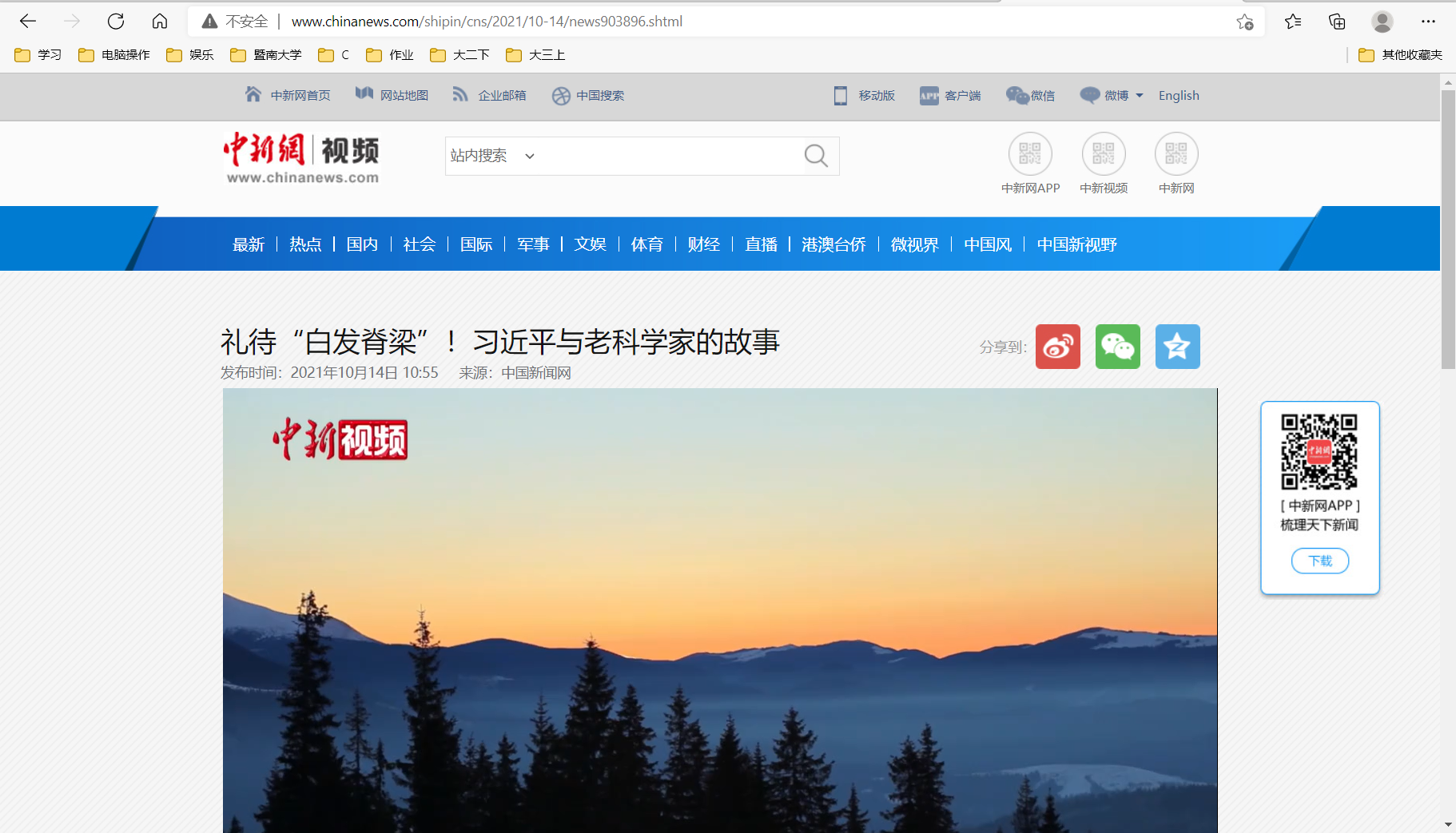
****

**响应标头：响应标头是可以在 HTTP 响应种使用的 HTTP 标头，如Access-Control-Allow-Origin，Server，Set-Cookie，Transfer-Encoding等等，如下：**

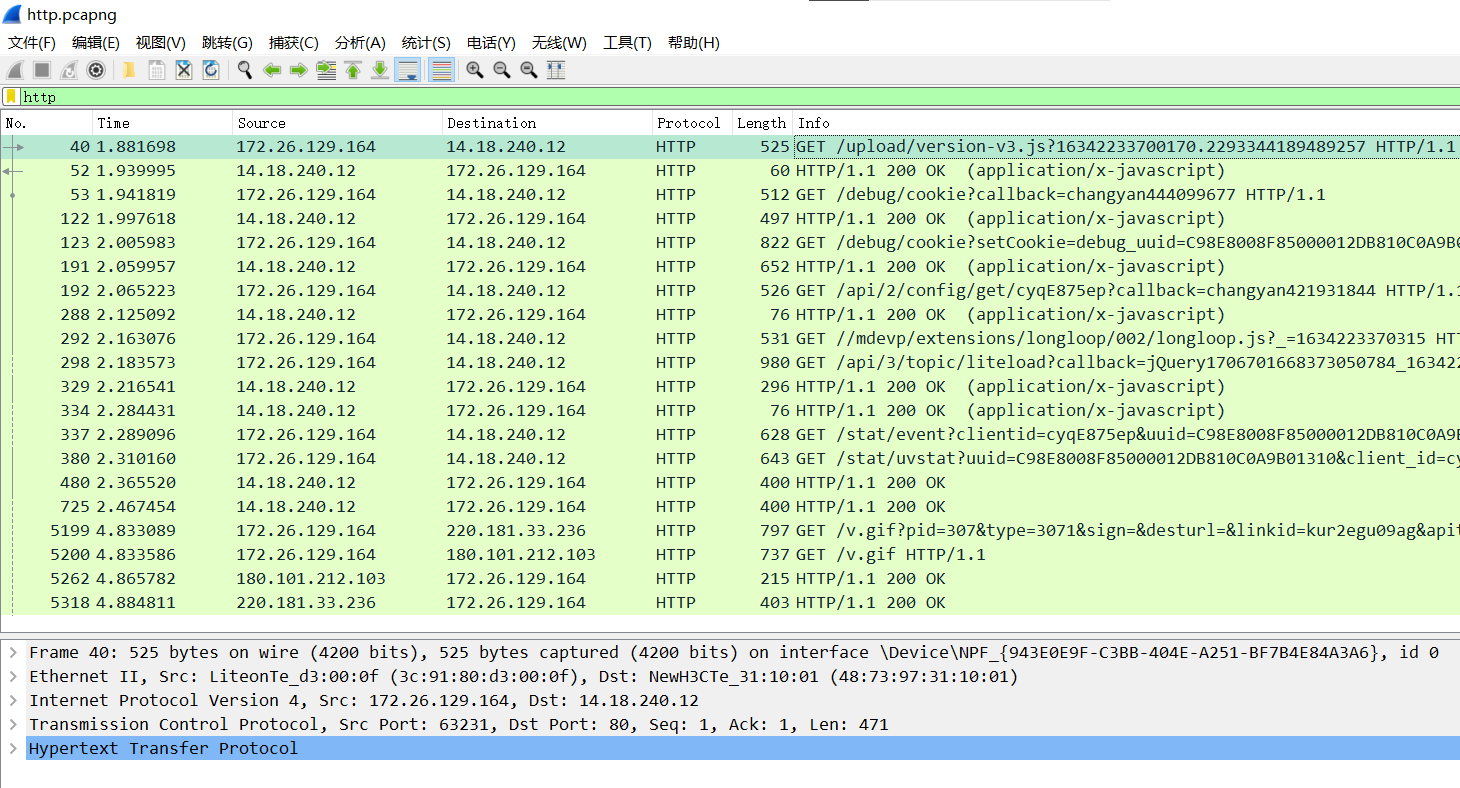


**5.1.4 点击网址并用wireshark进行捕获**

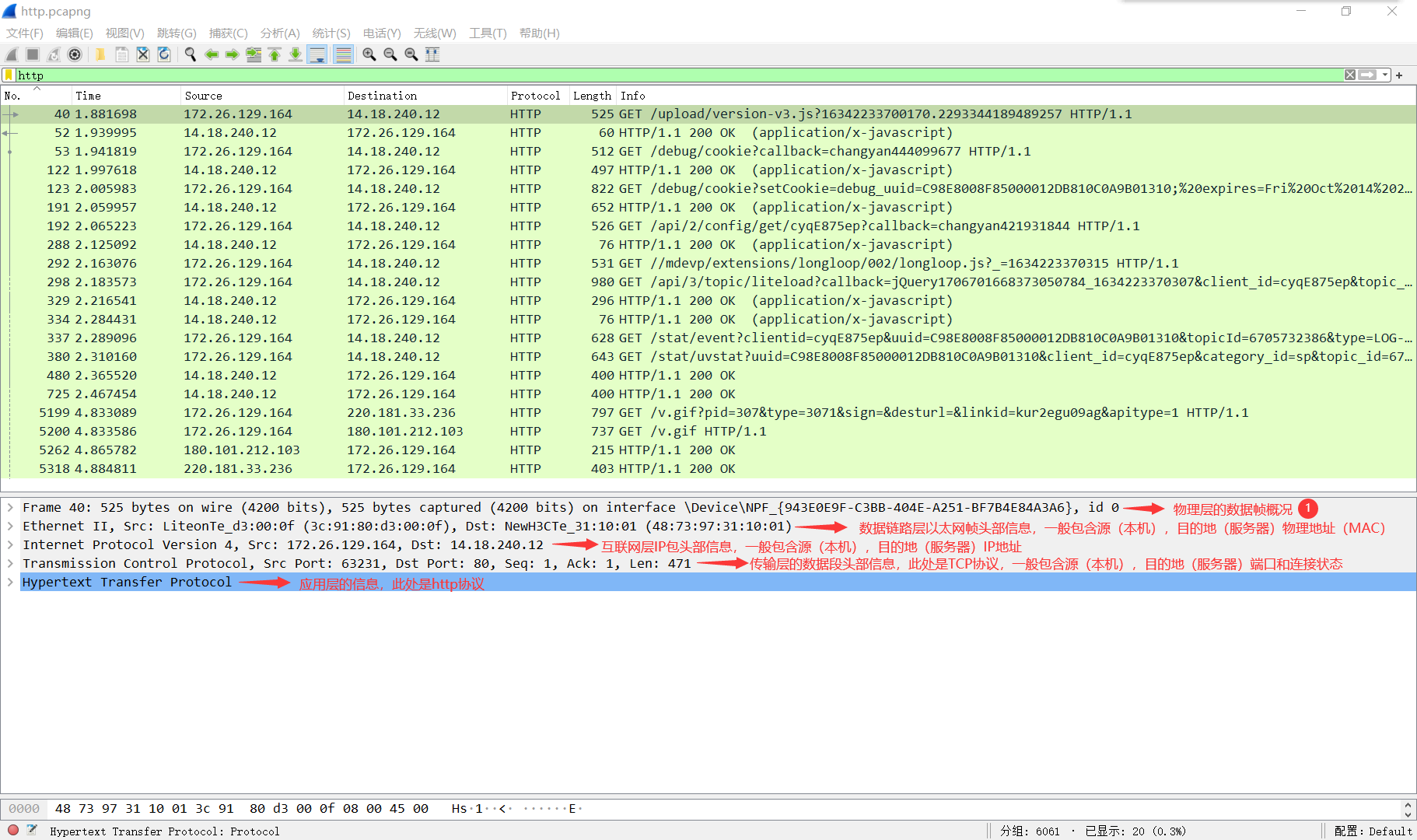
**网址页面：**

****

**Wireshark捕获（选择过滤http进行分析）：**

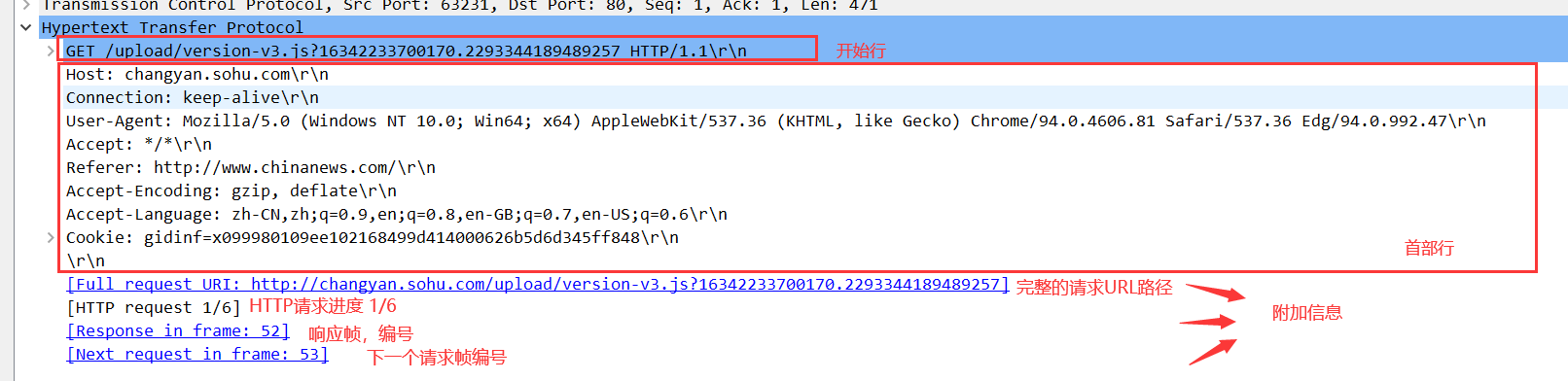
****

一共抓取了20个http协议数据包，客户端一共向服务器发送10次请求，服务器回应了10次，接下来从第一个数据包分析：



**分析请求报文**：

着重对http协议进行解读，对三个组成组成分析（附加消息即为实体消息）：



**1.开始行：**

GET /upload/version-v3.js?16342233700170.2293344189489257 HTTP/1.1\r\n

**请求方法：**GET：GET 方法用来请求访问已被 URI 识别的资源。指定的资源经服务器端解析后返回响应内容。

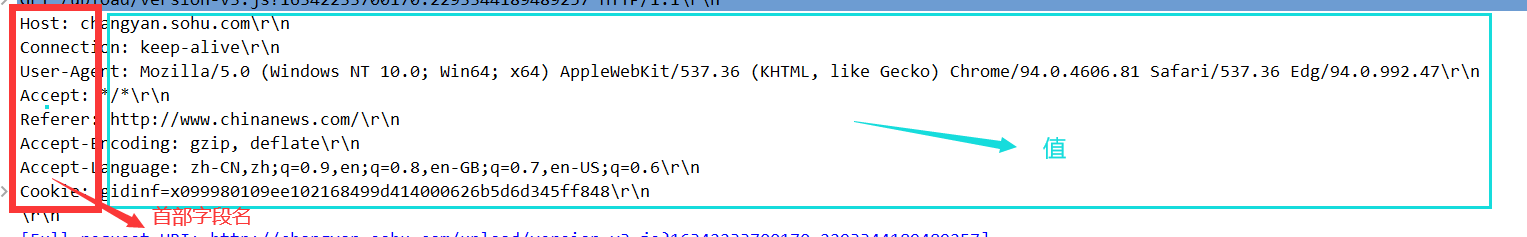
**URL：**/upload/version-v3.js?16342233700170.2293344189489257 浏览器请求一个对象upload/version-v3.js?16342233700170.2293344189489257的资源

**http版本：**http1.1版本，表示报文使用的是http1.1版本，HTTP/1.1 协议使用持续连接 万维网服务器在发送响应后仍然在一段时间内保持这条连接，使同一个客户（浏览器）和该服务器可以继续在这条连接上传送后续的 HTTP 请求报文和响应报文。

如下图：



**2.首部行：**



2.1 Host: changyan.sohu.com\r\n：

Host 请求头指明了服务器的域名（对于虚拟主机来说），以及（可选的）服务器监听的TCP端口号。如果没有给定端口号，会自动使用被请求服务的默认端口（比如请求一个 HTTP 的 URL 会自动使用80作为端口）。

Host 表示的是此时请求的对象所在的主机为changyan.sohu.com，端口号为80

2.2 Connection: keep-alive\r\n：

Connection 决定当前事务（一次三次握手和四次挥手）完成后，是否会关闭网络连接。此条Connection表示的是浏览器需要告诉服务器使用的是持久连接。它要求服务器在发送完响应的对象后不需要关闭连接，这是http1.1版本的特点，使用持续连接

2.3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.47\r\n

User-agent: 这是请求头用来告诉 Web 服务器，客户端浏览器使用的类型和版本信息是 Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Safari/537.36 Edg/94.0.992.47

2.4 Accept: \*/\*\r\n

用于表示浏览器想要的资源类型。根据请求的上下文不同，所设置的 Accept 请求头会相应变化。此标头会通告客户端其能够理解的 MIME 类型，\*/\* 代表的是任意 MIME 类型。

【什么是 MIME？】MIME (Multipurpose Internet Mail Extensions) 是描述消息内容类型的因特网标准。MIME 消息能包含文本、图像、音频、视频以及其他应用程序专用的数据。也就是说，MIME 类型其实就是一系列消息内容类型的集合。

【MIME 类型】

文本文件： text/html、text/plain、text/css、application/xhtml+xml、application/xml 图片文件： image/jpeg、image/gif、image/png

视频文件： video/mpeg、video/quicktime

应用程序二进制文件： application/octet-stream、application/zip

Accep相关知识：<https://www.cnblogs.com/Wayou/p/http_content_negotiation.html>

2.5 Referer: <http://www.chinanews.com/\r\n>

HTTP Referer 属性是请求标头的一部分，当浏览器向 web 服务器发送请求的时候，一般会带上 Referer，告诉服务器该网页是从哪个页面链接过来的，服务器因此可以获得一些信息用于处理。

此条referer说明我是从[www.chinanews.com](http://www.chinanews.com)（即中新网）页面链接过来的

2.6 Accept-Encoding: gzip, deflate\r\n

请求的 HTTP 标头通告其内容编码，通常是一个压缩算法中，客户端是能够理解的。使用内容协商，服务器选择其中一个提议，使用它并通过

2.7 Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n

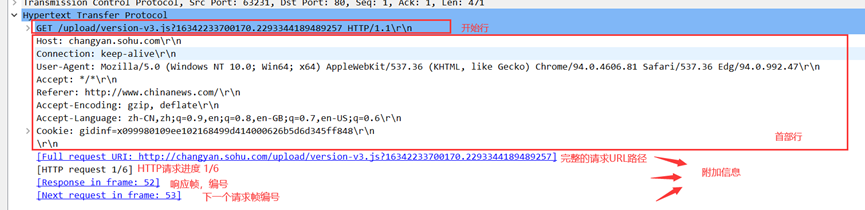
请求的语言，CN代表中文

2.8 Cookie: gidinf=x099980109ee102168499d414000626b5d6d345ff848\r\n

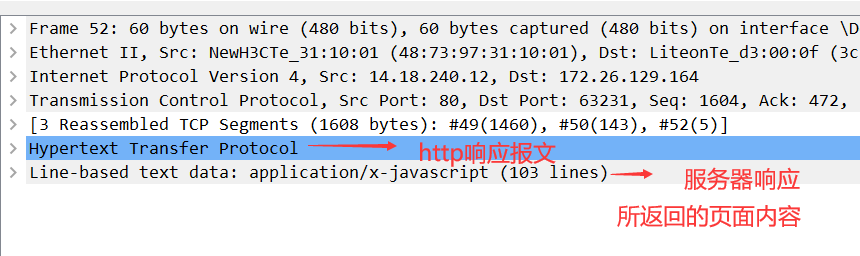
这是客户的cookie信息，因为此前我已经访问过百度新闻网站，而且百度一直登陆着我自己的百度帐号，这说明此前这个web站点为我产生一个唯一的识别码,以此作为索引存储在后台数据库中，并且此cookie已经保存在我的的cookie文件中。

所以说，当我再向百度新闻网站发送HTTP请求时，将会读取本地cookie，并写入HTTP首部发送给服务器，使得网站就能根据我的cookie来跟踪我的客户端了

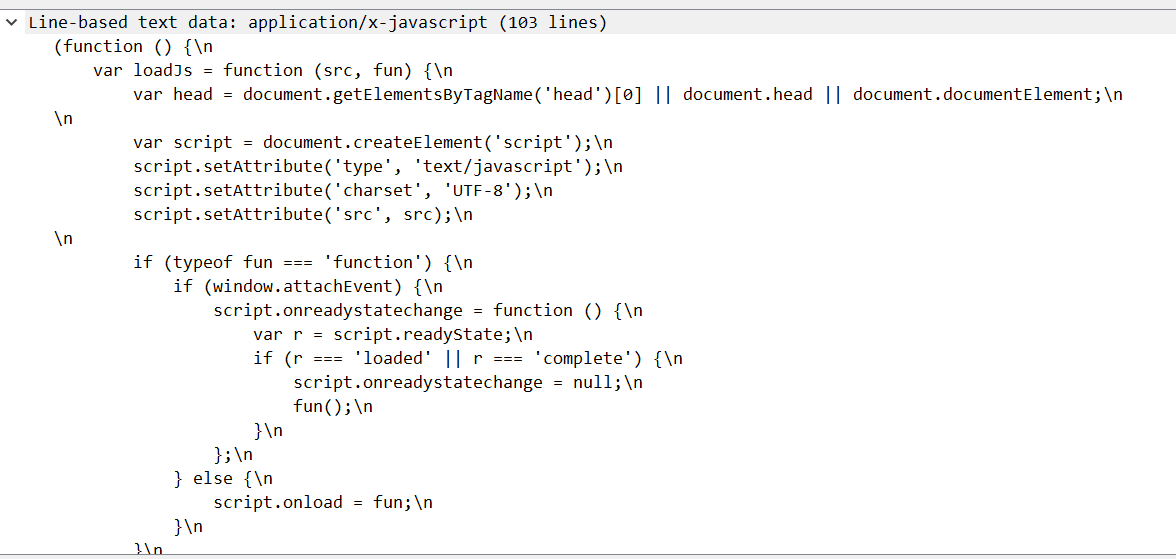
**3. 实体正文**



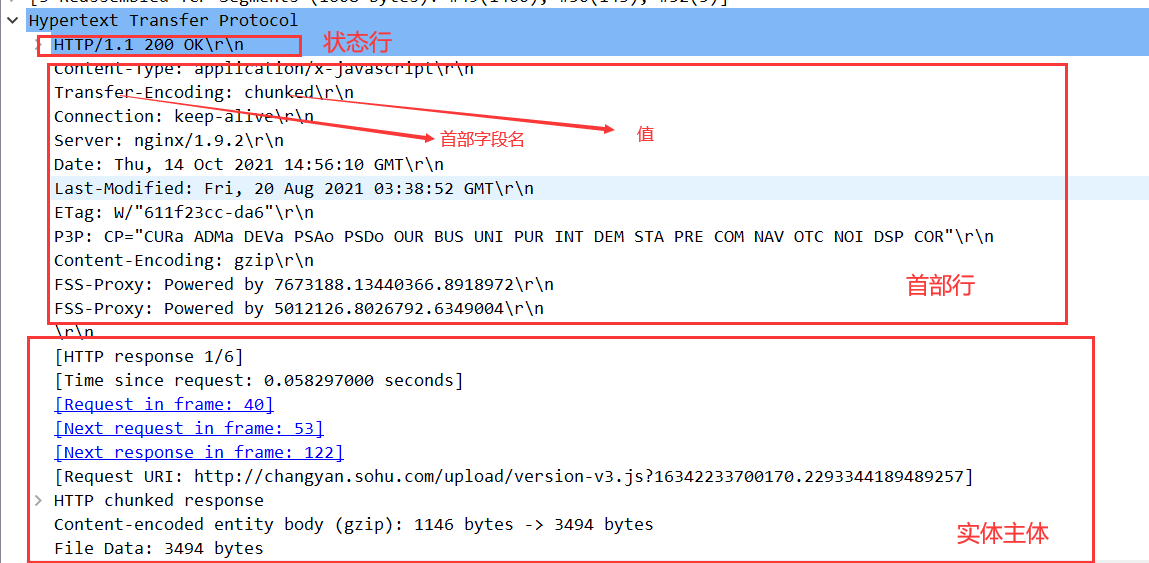
接下来看第二个数据包，此为服务器的http响应报文：



返回的页面内容：

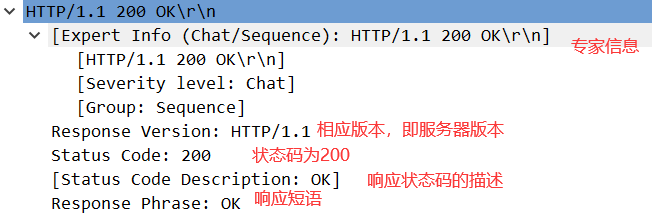


**分析响应报文：**



**1.状态行：**

HTTP/1.1 200 OK\r\n



版本：HTTP 状态码：200 状态码短语：OK

200表示着响应成功完成，以 2xx 为开头的都表示请求成功响应。

【以 3xx 为开头的都表示需要进行附加操作以完成请求，以 4xx 的响应结果表明客户端是发生错误的原因所在。以 5xx 为开头的响应标头都表示服务器本身发生错误】

**2.首部行：**

2.1Content-Type: application/x-javascript\r\n

表示的是实体主体的实体类型，此处为javascript文件

2.2Transfer-Encoding: chunked\r\n

首部字段 Transfer-Encoding 规定了传输报文主体时采用的编码方式。

2.3 Connection: keep-alive\r\n

表示使用的是持久连接

2.4 Server: nginx/1.9.2\r\n

服务器标头包含有关原始服务器用来处理请求的软件的信息。

应该避免使用过于冗长和详细的 Server 值，因为它们可能会泄露内部实施细节，这可能会使攻击者容易地发现并利用已知的安全漏洞。

2.5 Date: Thu, 14 Oct 2021 14:56:10 GMT\r\n

Date 是一个通用标头，它可以出现在请求标头和响应标头中，表示的是格林威治标准时间，这个时间要比北京时间慢八个小时

2.6 Last-Modified: Fri, 20 Aug 2021 03:38:52 GMT\r\n

在浏览器第一次请求某一个URL时，服务器端的返回状态会是200，内容是客户端请求的资源，同时有一个Last-Modified的属性标记此文件在服务器端最后被修改的时间。

客户端第二次请求此URL时，根据[HTTP协议](https://baike.baidu.com/item/HTTP%E5%8D%8F%E8%AE%AE/1276942)的规定，浏览器会向服务器传送If-Modified-Since报头，询问服务器该时间之后文件是否有被修改过。

如果服务器端的资源没有变化，则自动返回 HTTP 304（Not Changed.）状态码，内容为空，这样就节省了传输数据量。当服务器端代码发生改变或者重启服务器时，则重新发出资源，返回和第一次请求时类似。从而保证不向客户端重复发出资源，也保证当服务器有变化时，客户端能够得到最新的资源。

2.7 ETag: W/"611f23cc-da6"\r\n

ETag为“被请求变量的实体值”。ETag是一个可以与Web资源关联的记号。它标志着上一次修改的标志

Etag 是URL的Entity Tag，用于标示URL对象是否改变，区分不同语言和Session等等。具体内部含义是使服务器控制的，就像Cookie那样。

客户端请求一个页面（A）。 服务器返回页面A，并在给A加上一个ETag。 客户端展现该页面，并将页面连同ETag一起缓存。 客户再次请求页面A，并将上次请求时服务器返回的ETag一起传递给服务器。 服务器检查该ETag，并判断出该页面自上次客户端请求之后还未被修改，直接返回响应304（未修改——Not Modified）和一个空的响应体。

2.8 P3P: CP="CURa ADMa DEVa PSAo PSDo OUR BUS UNI PUR INT DEM STA PRE COM NAV OTC NOI DSP COR"\r\n

P3P是一种被称为个人隐私安全平台项目的标准能够保护在线隐私权，使Internet冲浪者可以选择在浏览网页时，是否被第三方收集并利用自己的个人信息。如果一个站点不遵守P3P标准的话，那么有关它的Cookies将被自动拒绝，并且P3P还能够自动识破多种Cookies的嵌入方式。p3p是由全球资讯联盟网所开发的。

2.9 Content-Encoding: gzip\r\n

Content-Encoding这个实体报头用来压缩媒体类型。Content-Encoding 指示对实体应用了何种编码。

【Accept-Encoding: gzip, deflate //请求头

Content-Encoding: gzip //响应头 】

3.0 FSS-Proxy: Powered by 7673188.13440366.8918972\r\n

FSS-Proxy: Powered by 5012126.8026792.6349004\r\n

FSS指的是频率选择表面(Frequency Selective Surface ) ，就其本质而言是一个空间滤波器,与电磁波相互作用表现出明显的带通或带阻的滤波特性。FSS 具有特定的频率选择作用而被广泛地应用于微波、红外至可见光波段。

**3.实体正文**

[HTTP response 1/6]：http响应进度 1/6，还需要6次响应，对应上文的请求报文中的请求进度

[Time since request: 0.058297000 seconds]：响应时间

[Request in frame: 40]：请求帧，编号

[Next request in frame: 53]：下一次请求帧

[Next response in frame: 122]：下一次响应帧

[RequestURI:http://changyan.sohu.com/upload/versionv3.js?16342233700170.2293344189489257]：这是完整的URL路径

HTTP chunked response：实体正文分块编码

Content-encoded entity body (gzip): 1146 bytes -> 3494 bytes：内容编码长度

File Data: 3494 bytes：文件长度

4.Line-based text data: application/x-javascript (103 lines) 响应数据的具体内容

至此，一份http请求和响应报文的分析结束，接下来又进行了类似的9次请求对象和响应，成功显示了网页

实验至此，我对http协议的工作流程还有报文格式有了很深刻的认识

**5.1.5 查找资料总结http的优缺点**

回顾课堂讲过的http优缺点，在实验里我也有了一些体会：

**优点：简单灵活易扩展，**HTTP 的协议比较简单，它的主要组成就是 header + body，头部信息也是简单的文本格式，而且 HTTP 的请求报文根据英文也能猜出来个大概的意思，降低学习门槛，能够让更多的人研究和开发 HTTP 应用。并且，HTTP 协议里的请求方法、URI、状态码、原因短语、头字段等每一个核心组成要素都没有被制定死，允许开发者任意定制、扩充或解释，给予了浏览器和服务器最大程度的信任和自由。

这一点在实验里很能体会到，所有的信息都是明文，在报文里很容易看出来大致意思，很简单，报文的具体内容没有硬性规定，每一个数据包的要素都不一定相同，体现了灵活的特点

其次，http**无状态**的特点，因为没有记忆能力，所以就不需要额外的资源来记录状态信息，不仅实现上会简单一些，而且还能减轻服务器的负担，能够把更多的 CPU 和内存用来对外提供服务。

实验中我们可以在报文里面看到cookie和set-cookie字段，弥补了无状态的缺点，充分发挥了优点

**缺点：明文传输不能保证安全。**明文意思就是协议里的报文（准确地说是 header 部分）不使用二进制数据，而是用简单可阅读的文本形式。

在实验中，对比 TCP、UDP 这样的二进制协议，它的优点显而易见，我不需要借助任何外部工具，用Wireshark抓包后，直接用肉眼就可以很容易地查看或者修改，为我们的开发调试工作带来极大的便利。

当然缺点也是显而易见的，就是不安全，可以被监听和被窥探，很容易被修改。因为无法判断通信双方的身份，不能判断报文是否被更改过。

**5.2 DNS协议分析**

**5.2.1 思考DNS的工作流程**

**1.先思考什么是DNS？**

识别主机有两种方式：主机名、IP地址。前者便于记忆(如www.yahoo.com)，但路由器很难处理(主机名长度不定)；后者定长、有层次结构，便于路由器处理，但难以记忆。

建立IP地址与主机名间的映射，这就是域名系统DNS做的工作。DNS通常由其他应用层协议使用(如HTTP、SMTP、FTP)，将主机名解析为IP地址，其运行在UDP之上，使用53号端口。

DNS除了提供主机名到IP地址转换外，还提供如下服务：主机别名、邮件服务器别名、负载分配。

**2.输入网址后http是怎么使用DNS的？**

(1)一台用户主机上运行着DNS应用的客户机端(如浏览器)，输入URL

(2)从上述URL抽取主机名，即域名，传给DNS应用的客户机端(浏览器)

(3)该DNS客户机向DNS服务器发送一个包含主机名的请求(DNS查询报文)

(4)该DNS客户机收到一份回答报文(即DNS回答报文)，该报文包含该主机名对应的IP地址119.75.218.70

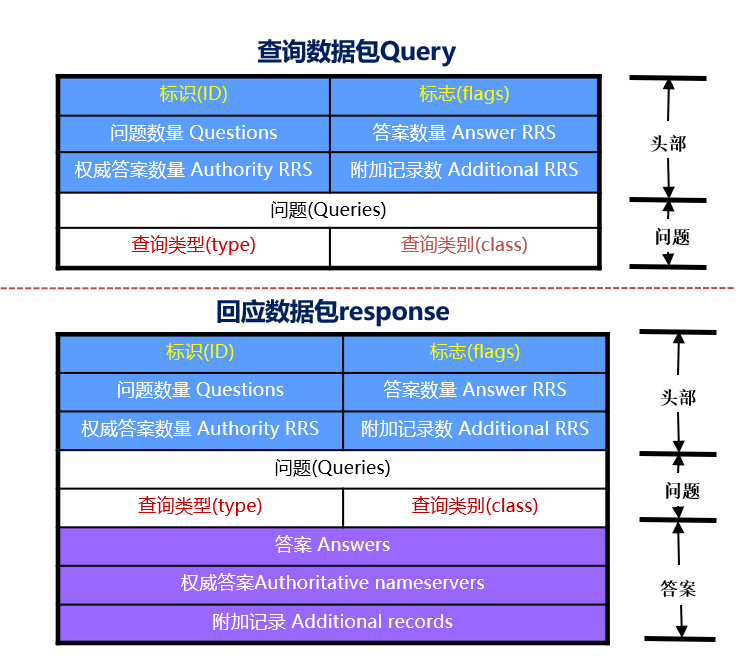
(5)浏览器由该IP地址定位的HTTP服务器发送一个TCP链接，接下来就是进行TCP三次握手以及http的传输等

**5.2.2 查找资料了解DNS报文格式**

**参考博客：**[**https://blog.csdn.net/zhaqiwen/article/details/18048791**](https://blog.csdn.net/zhaqiwen/article/details/18048791)

**1.DNS报文格式**

DNS只有两种报文：查询报文、回答报文，两者有着相同格式，如下图：



**2.头部区域**

**标识数：**对该查询进行标识，该标识会被复制到对应的回答报文中，客户机用它来匹配发送的请求与接收到的回答。

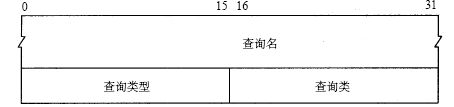
**标志：**

**问题数、回答RR数、权威RR数、附加RR数：**这四个字段都是两字节，分别对应下面的查询问题、回答、授权和附加信息部分的数量。

一般问题数都为1，DNS查询报文中，资源记录数、授权资源记录数和附加资源记录数都为0

**3.问题区域**

包含正在进行的查询信息。包含查询名(被查询主机名字的名字字段)、查询类型、查询类。



查询名：查询名部分长度不定，一般为要查询的域名(也会有IP的时候，即反向查询)。此部分**由一个或者多个标示符序列组成**，每个标示符以首字节数的计数值来说明该标示符长度，每个名字以0结束。

查询类型：通常查询类型为A(由名字获得IP地址)或者PTR(获得IP地址对应的域名)

**4.回答，权威，附加区域**

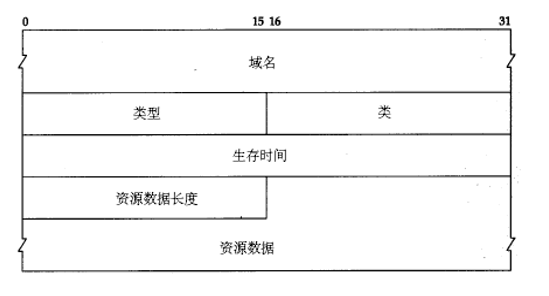
回答区域包含了最初请求名字的资源记录，一个回答报文的回答区域可以包含多条资料记录RR(因为一个主机名可以对应多个IP地址，冗余Web服务器)。

权威区域包含了其他权威DNS服务器的记录。

附加区域包含其他一些"有帮助"的记录，例如，对于一个MX(邮件交换)请求的回答报文中，回答区域包含一条资料记录(该记录提供邮件服务器的规范主机名)，附加区域可以包含一条类型A记录(该记录提供了该邮件服务器的规范主机名的IP地址)。

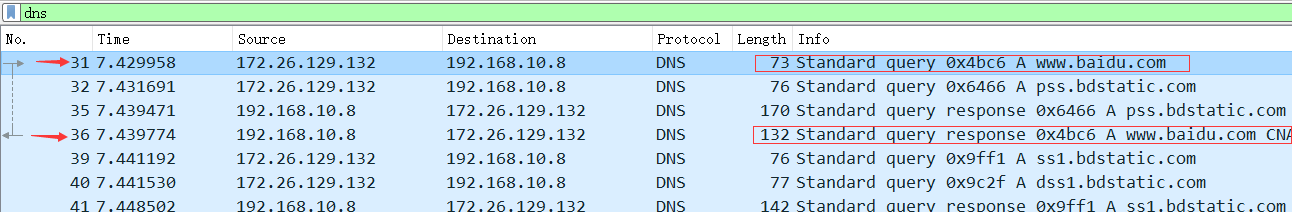
以上每条资料记录是一个五元组，如下：

(域名，生存期，类别，类型，值)



**5.2.3 用wireshark捕获DNS报文**

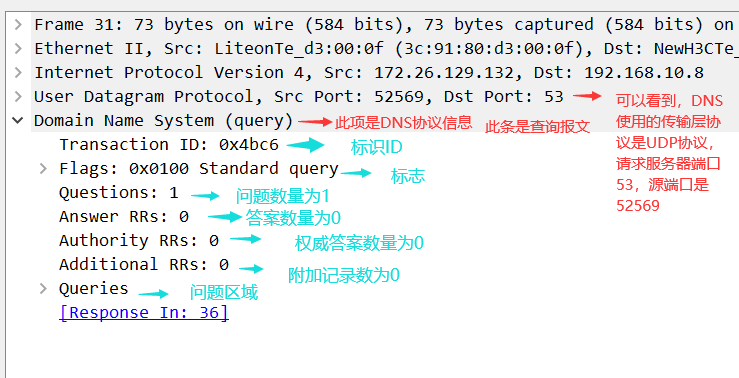
在浏览器里输入<https://www.baidu.com/>，如图：



用红线指出来的第一个以及第四个数据包是一对DNS查询/响应报文，显然第一个是DNS查询报文，第四个是DNS回答报文

分析DNS查询报文：

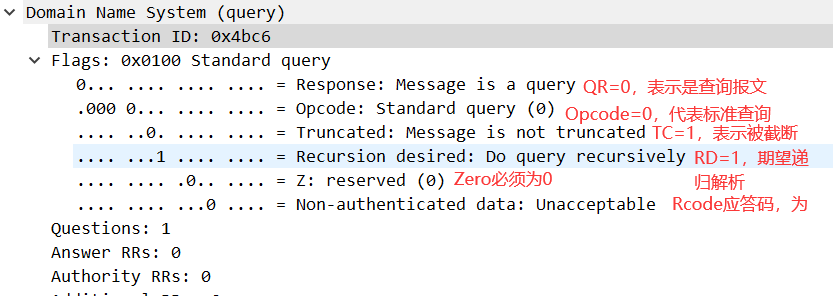
如图：

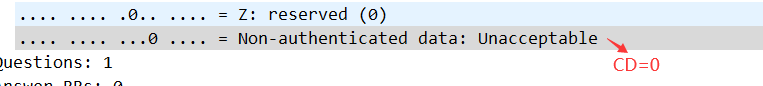


**1.头部区域：**

标识ID为：0x00004bc6客户端设置的16位标识，服务器使用标识字段来匹配DNS请求。只有使用相同的ID，才能证明是同一个会话。

标志：如图





**分析16bit标志的组成：**

QR(1比特)：查询/响应的标志位，1为响应，0为查询。此条报文为0，QR=0，代表查询报文

opcode(4比特)：定义查询或响应的类型(若为0则表示是标准的，若为1则是反向的（IP地址查域名，IP地址.in-addr.arpa.），若为2则是服务器状态查询，3~15保留)。此条报文里等于0，代表标准查询

AA(1比特)：表示授权应答 Authoritative Answer ，应答的时候有意义，指出给出应答的服务器是查询域名的授权解析服务器。注意因为别名的存在，应答可能存在多个主域名，这个AA位对应请求名，或者应答中的第一个主域名。1标识域名服务器是授权服务器。0 为应答服务器不是该域名的权威解析服务器

**这里是查询报文，AA并没有任何意义，我认为这是不显示出来的原因**

TC(1比特)：截断标志位，仅当DNS报文使用UDP服务时使用。因为UDP数据包有长度限制，当长度过长时会被截断。报文里面=1表示响应已超过512字节并已被截断

RD(1比特)：报文里该位为1表示客户端希望得到递归回答，也就是说，建议域名服务器进行递归解析，递归查询的支持是可选的。

1标识执行递归查询，即DNS服务器无法解析某一个主机名，则它向其它DNS服务器继续查询，如此反复，直到它解析到目标主机名并返回给客户端。

0标识迭代查询，即DNS服务器无法解析目标主机时，它将查询到的其它DNS服务器的IP地址返回给客户端，供客户端进行参考。

RA(1比特)：表示支持递归 Recursion Available，这个比特位在应答中设置或取消，用来代表服务器是否支持递归查询。允许递归标识。此字段在应答字段中使用，1标识DNS服务器支持递归查询。0 为应答服务器不支持递归查询

**这里是查询报文，RA不需要设置，所以不显示**

zero(3比特)：保留字段，暂时未使用。在所有的请求和应答报文中必须置为0。

AD(1bit): 应答认证：应答/权限部分未由服务器进行认证

0 为应答服务器未验证了该查询相关的 DNSSEC 数字签名

1 为应答服务器已经验证了该查询相关的 DNSSEC 数字签名

**这里是查询报文，AD不需要设置，所以不显示**

CD(1比特)：未经认证的数据：不可接受。0 为服务器已经进行了相关 DNSSEC 数字签名的验证，1 为服务器并未进行相关 DNSSEC 数字签名的验证

可以看到报文里面，CD=0，显示Non-authenticated data: Unacceptable，表示www.baidu.com的的服务器已经进行了相关 DNSSEC 数字签名的验证

rcode(4比特)：应答码，表示响应的差错状态，通常为0和3，各取值含义如下：

0 无差错

1 格式差错

2 问题在域名服务器上

3 域参照问题

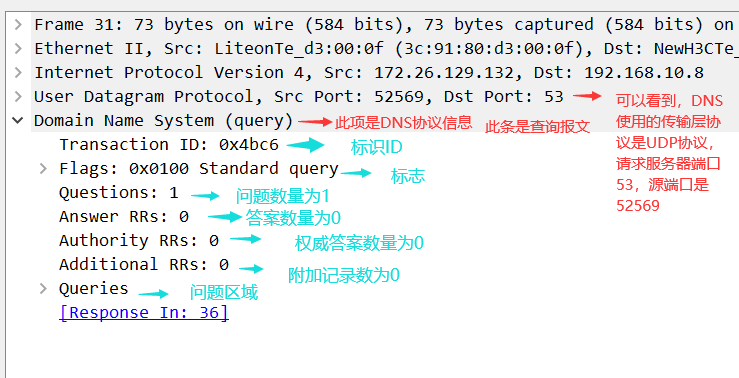
4 查询类型不支持

5 在管理上被禁止

6 -- 15 保留

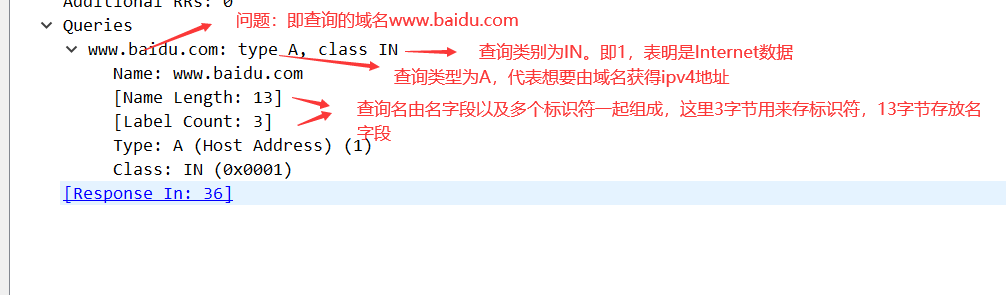
同样的，**这里是查询报文，应答码不需要设置，所以不显示**

问题数、回答RR数、权威RR数、附加RR数：分别为1，0，0，DNS查询报文中，资源记录数、授权资源记录数和附加资源记录数都为0



**2.问题区域**

如图所示：



**1）查询名 QNAME：**要查找的名字，是一个或多个标识符的序列，长度不固定，且不使用填充字节，一般该字段表示的就是需要查询的域名（如果是反向查询，则为IP，反向查询即由IP地址反查域名）。

每个标识符以首字节的计数值来说明随后标识符的字节长度，每个名字以最后字节为0结束，长度为0的标识符是根标识符。单个标识符最大长度为63字节。

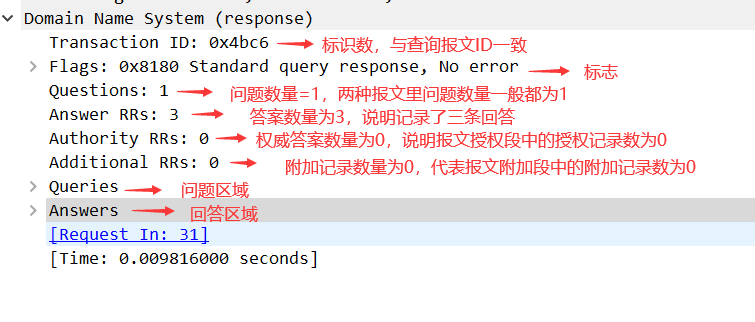
**2）查询类型 QTYPE：** 每个问题有一个查询类型。2个字节表示查询类型，取值可以为任何可用的类型值，以及通配码\*来表示所有的资源记录。

**3）查询类：**通常为1，表示IN，表明是Internet数据。

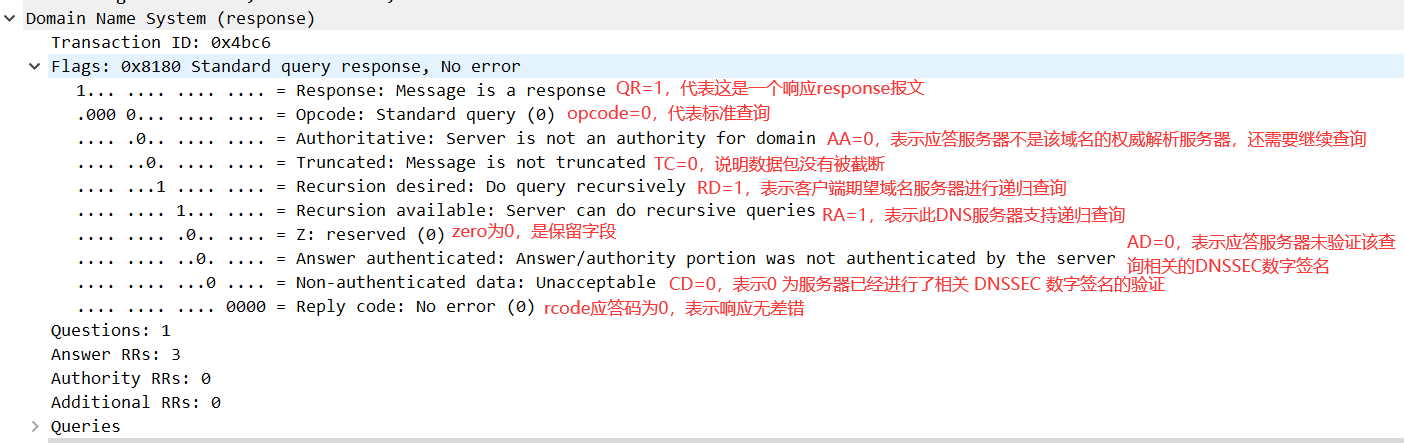
至此，查询报文分析完毕，接下来打开回答报文查看

分析回答报文：

如下图：

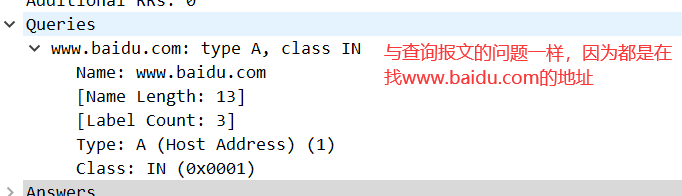


**1.头部标志：（具体解释已经在上文写出，不详细赘述了）**



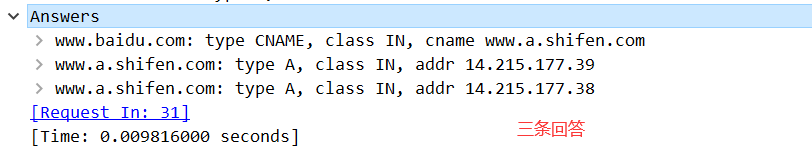
AA=0，说明给出应答的服务器不是查询域名的**权威解析服务器，权威服务器还需进行下一步查询**

**2.问题区域：**



**3.回答区域：**

回答区域包含了最初请求名字的资源记录，一个回答报文的回答区域可以包含多条资料记录RR**(因为一个主机名可以对应多个IP地址，冗余Web服务器)。**



资源记录(RR)区域（包括回答区域，授权区域和附加区域）：

**（1） 域名NAME**（2字节或不定长）：它的格式和Queries区域的查询名字字段是一样的。有一点不同就是，当报文中域名重复出现的时候，该字段使用2个字节的偏移指针来表示。比如，在资源记录中，域名通常是查询问题部分的域名的重复，因此用2字节的指针来表示，具体格式是最前面的两个高位是 11，用于识别指针。其余的14位从DNS报文的开始处计数（从0开始），指出该报文中的相应字节数。一个典型的例子，C00C(1100000000001100，12正好是头部的长度，其正好指向Queries区域的查询名字字段)。

**（2） 查询类型TYPE ：**表明资源纪录的类型

**（3）查询类CLASS ：**对于Internet信息，总是1，代表IN。表示RDATA的类

**（4）生存时间（TTL）：**4字节无符号整数表示资源记录可以缓存的时间。以秒为单位，表示的是资源记录的生命周期，一般用于当地址解析程序取出资源记录后决定保存及使用缓存数据的时间，它同时也可以表明该资源记录的稳定程度，极为稳定的信息会被分配一个很大的值（比如86400，这是一天的秒数）。0代表只能被传输，但是不能被缓存。

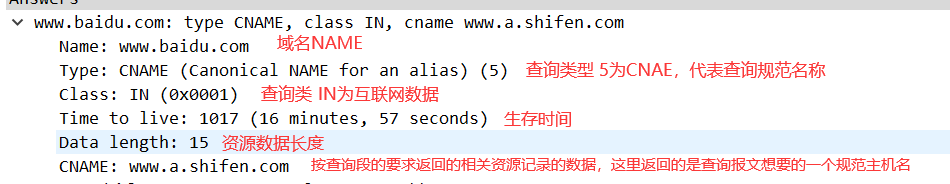
TTL 是 DNS 记录中的一个值，可决定对该记录所做的后续更改需要多少秒才会生效。网域的每条 DNS 记录（如 MX 记录、CNAME 记录等）都有一个 TTL 值。一条记录目前所设的 TTL 决定了您现在所做的任何更改需要多久才会生效。例如，如果一条记录的 TTL 为 86400 秒，则对该记录的更改最多需要 24 小时才会生效。

**（5）资源数据长度：**URDLENGT\*\*H 2个字节无符号整数表示RDATA的长度

**（6）资源数据 RDATA** 该字段是一个可变长字段，不定长字符串来表示记录，格式与TYPE和CLASS有关。比如，TYPE是A，CLASS 是 IN，那么RDATA就是一个4个字节的ARPA网络地址。表示按照查询段的要求返回的相关资源记录的数据。可以是Address（表明查询报文想要的回应是一个IP地址）或者CNAME（表明查询报文想要的回应是一个规范主机名）等。

**分析三条回答：**

**回答一：**如下图：



此条查询类型是CNAME，返回一个规范主机名

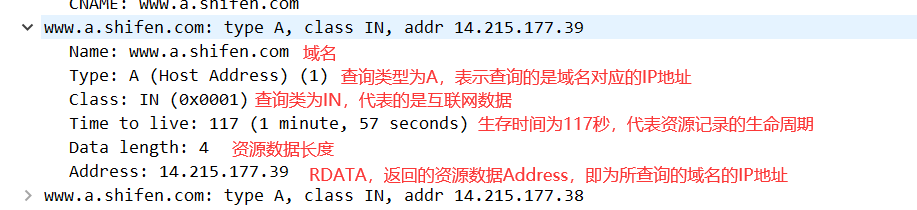
【CNAME：，CNAME是将域名解析成另外一个域名，即别名记录。这种记录允许将多个名字映射到另外一个域名。

A记录就是把一个域名解析到一个IP地址，而CNAME记录就是把域名解析到另外一个域名。

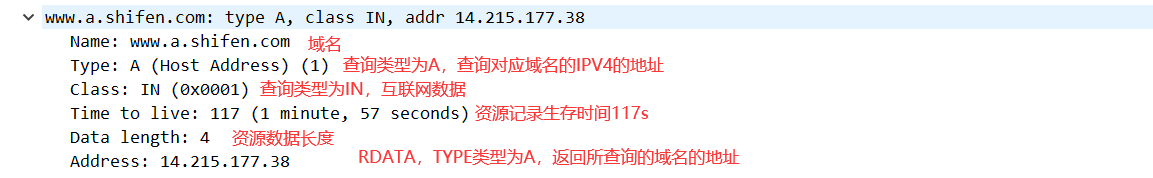
比如www.baidu.com，这个二级域名解析到的是一个服务器的IP地址，比如8.8.8.8，而CNAME是将新域名，比如：a.www.baidu.com解析到另外一个域名，比如：www.baidu.com。

当新域名产生后都指向CNAME域名后，在发生服务器需要更换或者迁移时，就只需要把www.baidu.com这个域名做重新设置就可以了，这样就减少了很多工作量。】

**回答二：**



**回答三：**



answers字段内容，翻译过来意思就是（从第一条开始按顺序翻译），

①'www.baidu.com'域名的别名有'www.a.shifen.com '；

②'www.a.shifen.com域名对应的IP4地址为'14.215.177.38'；

④'www.a.shifen.com域名对应的IP4地址为'14.215.177.38'；

**4.打开浏览器输入IP 14.215.177.38，打开了百度首页：**



通过以上报文结果数据分析，获取到的数据结果和我们准备工作所想要获取的结果是一致的。至此，访问[www.baidu.com的DNS](http://www.baidu.com的第一对DNS)数据包一个基本的初步分析已经完毕

**5.2.4 思考和总结**

**为什么有时候****wireshark可能捕获不到DNS解析过程?**

做实验的时候**，**实验查询百度的网址时，只需要一次DNS解析就可以找到百度的地址，我很疑惑为什么不如同预想的那样需要递归很多个DNS服务器呢？有的时候，wireshark可能捕获不到DNS解析过程，是为什么呢？

我认为原因就和DNS的解析过程有关了。首先，在浏览器中输入域名，操作系统会先检查自己本地的hosts文件是否有这个网址映射关系，如果有，就先调用这个IP地址映射，完成域名解析，**这种情况就不需要DNS服务器来解析**

其次，如果hosts里没有这个域名的映射，则查找本地DNS解析器缓存，是否有这个网址映射关系，**如果有，直接返回，完成域名解析。**

再者，如果hosts与本地DNS解析器缓存都没有相应的网址映射关系，首先会找TCP/IP参数中设置的首选DNS服务器，也就是本地DNS服务器，此服务器收到查询时，如果要查询的域名，包含在本地配置区域资源中，则返回解析结果给客户机，完成域名解析，此解析具有权威性。

如果本地DNS服务器本地区域文件与缓存解析都失效，则根据本地DNS服务器的设置（是否设置转发器）进行查询，如果未用转发模式，本地DNS就把请求发至13台根DNS，根DNS服务器收到请求后会判断这个域名(.com)是谁来授权管理，并会返回一个负责该顶级域名服务器的一个IP。本地DNS服务器收到IP信息后，将会联系负责.com域的这台服务器。这台负责.com域的服务器收到请求后，如果自己无法解析，它就会找一个管理.com域的下一级DNS服务器地址给本地DNS服务器，之后进行迭代或者递归查询。直至找到目标主机。

所以现在看到百度ip地址查询过程没有预想的那么复杂的过程，应该就是因为在本地DNS解析器缓存是存在百度的IP的域名解析，所以大大节省了查询IP的工作量

**5.3 FTP协议分析**

**5.3.1 了解什么是FTP**

**1.什么是FTP？**

FTP 是 TCP/IP 协议组中的协议之一，是英文File Transfer Protocol的缩写。

该协议是Internet文件传送的基础，它由一系列规格说明文档组成，目标是提高文件的共享性，提供非直接使用远程计算机，使存储介质对用户透明 和可靠高效地传送数据。简单的说，FTP就是完成两台计算机之间的拷贝，从远程计算机拷贝文件至自己的计算机上，称之为“下载 （download）”文件。若将文件从自己计算机中拷贝至远程计算机上，则称之为“上载（upload）”文件。

**2.什么是控制连接和数据连接？**

在TCP/IP协议中， 需要两个端口，一个是数据端口，一个是控制端口。

控制端口一般为21，而数据端口不一定是20，这和FTP的应用模式有关，如果是主动模式，应该为20，如果为被动模式，由服务器端和客户端协商而定。相比于HTTP，FTP协议要复杂得多。复杂的原因，是因为FTP协议要用到两个TCP连接，一个是控制链路，用来在FTP客户端与服务器之间传递命令；另一个是数据链路，用来上传或下载数据。

**3.FTP的传输方式**

FTP的传输有两种方式：ASCII传输模式和二进制数据传输模式。

**4.FTP的工作方式**

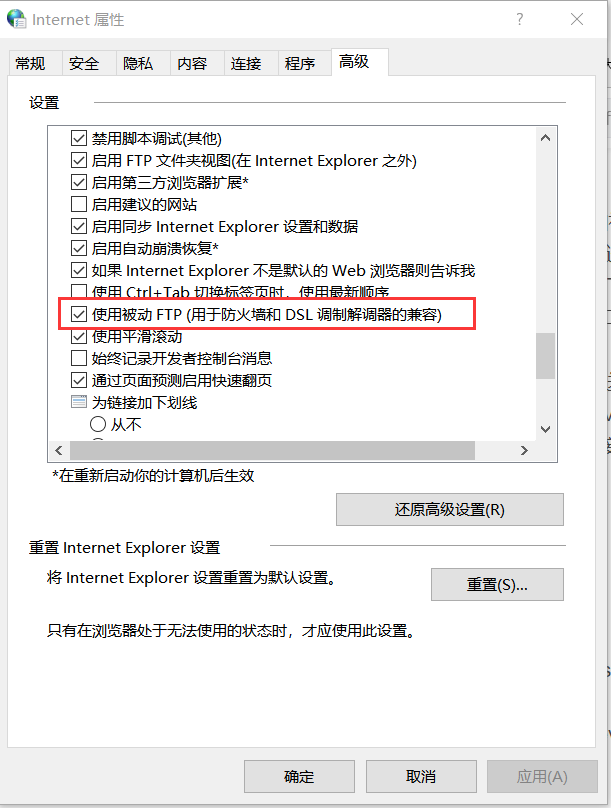
FTP协议有两种工作方式：**PORT方式**和**PASV方式，**中文意思为主动式和被动式。

**PORT(主动)方式**的连接过程是：客户端向服务器的FTP端口(默认是21)发送连接请求，服务器接受连接，建立一条命令链路。当需要传送数据时，客户端在命令链路上用PORT命令告诉服务器：“我打开了XXXX端口，你过来连接我”。于是服务器从20端口向客户端的XXXX端口发送连接请求，建立一条数据链路来传送数据。

**PASV(被动)方式**的连接过程是：客户端向服务器的FTP端口(默认是21)发送连接请求，服务器接受连接，建立一条命令链路。当需要传送数据时，服务器在命令链路上用PASV命令告诉客户端：“我打开了XXXX端口，你过来连接我”。于是客户端向服务器的XXXX端口发送连接请求，建立一条数据链路来传送数据。

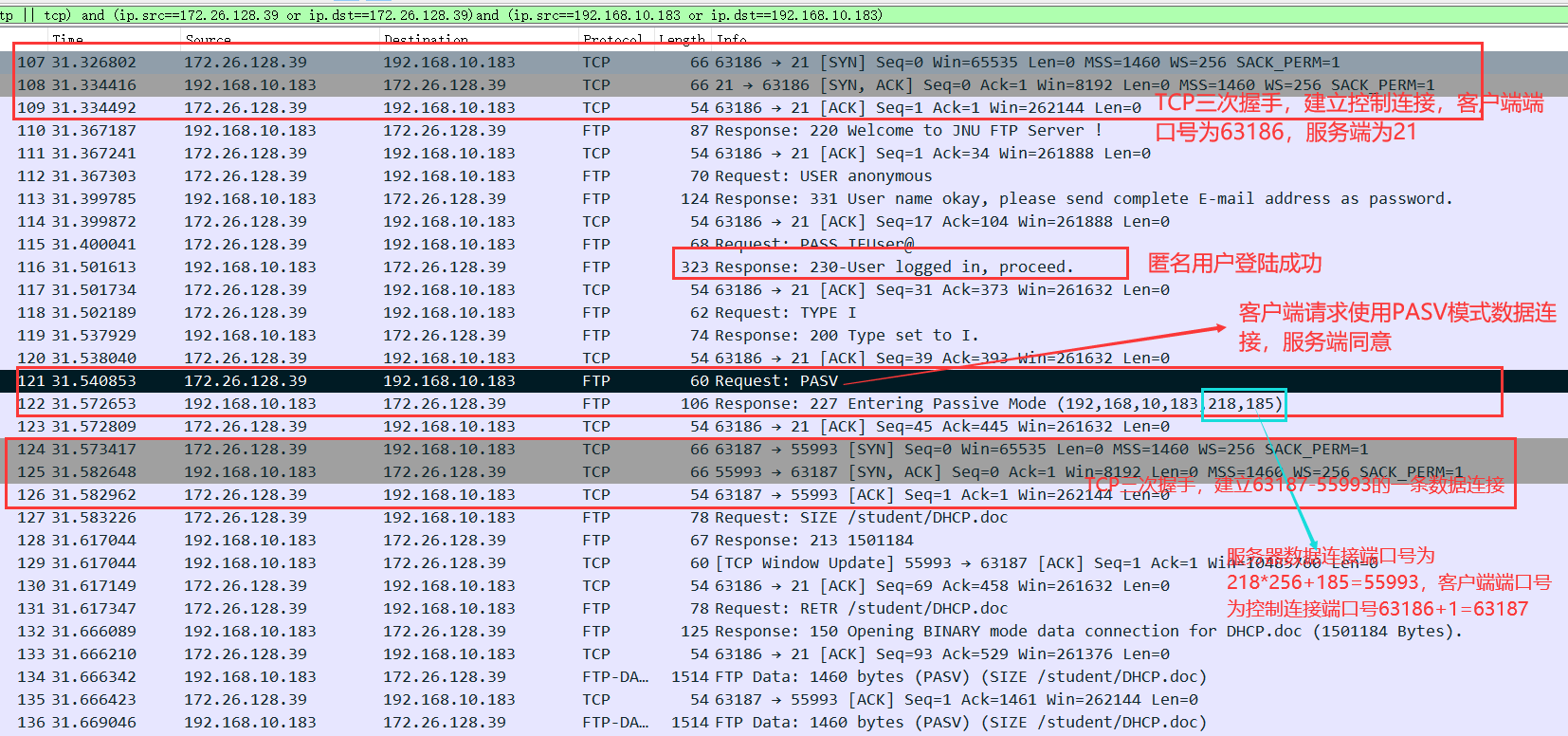
**5.3.2 使用被动模式访问暨南大学服务器**

**1.设置被动访问**



**2.访问<ftp://ftp.jnu.edu.cn>，并任意下载文件，整个过程用wireshark捕获结果如下：**

**如图:**



107~109号数据包：TCP三次握手，建立控制连接，可见ftp是基于tcp的，且服务器使用21端口连接客户端

110号数据包：服务器响应状态码220表示服务准备就绪，成功连接到服务器

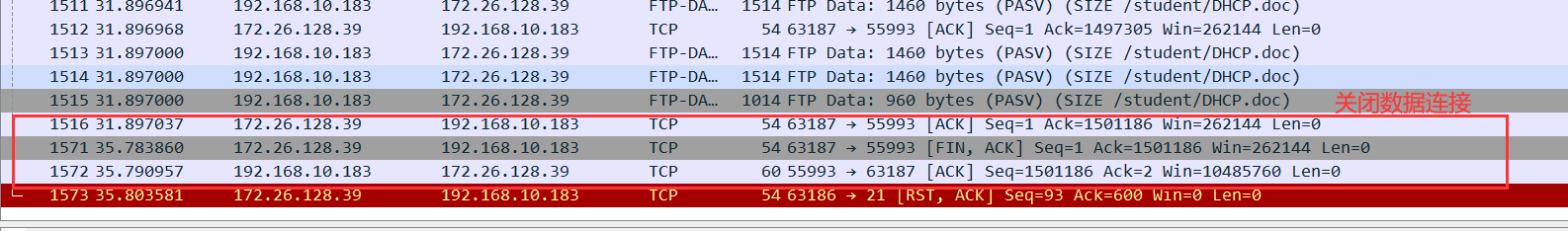
112-120号数据包：服务端响应码331，即要求密码，客户端使用匿名登录，服务端响应码230，表示登录到了因特网，匿名登陆成功

121-122号数据包：客户端请求协商工作模式为PASV模式（被动连接），服务器同意

124-126数据包：TCP三次握手，建立数据连接，客户端端口号为之前的控制连接端口63186+1=63187 服务器端口号为：218\*256+185=55993

127-128号数据包：客户端想知道文件的size大小，服务端回复

129-132号数据包：客户端请求RETR文件，即被动模式下从ftp下载文件，服务端响应码为150，代表已经打开文件数据连接 ，开始传输数据接下来开始传输数据



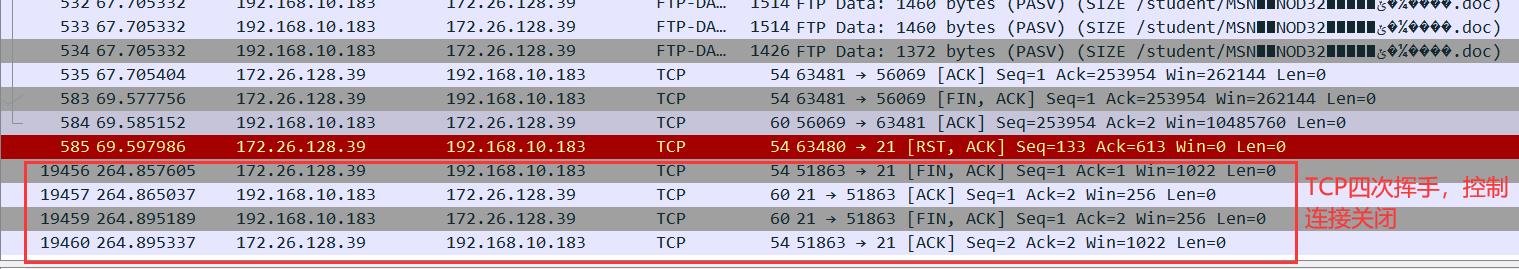
133-1515号数据包：传输文件，直到1515号，文件传输结束

1516-1572号数据包：文件传输结束，关闭数据连接

1571号数据包：连接到服务器21号端口的控制连接还在保持开启，并没有关闭

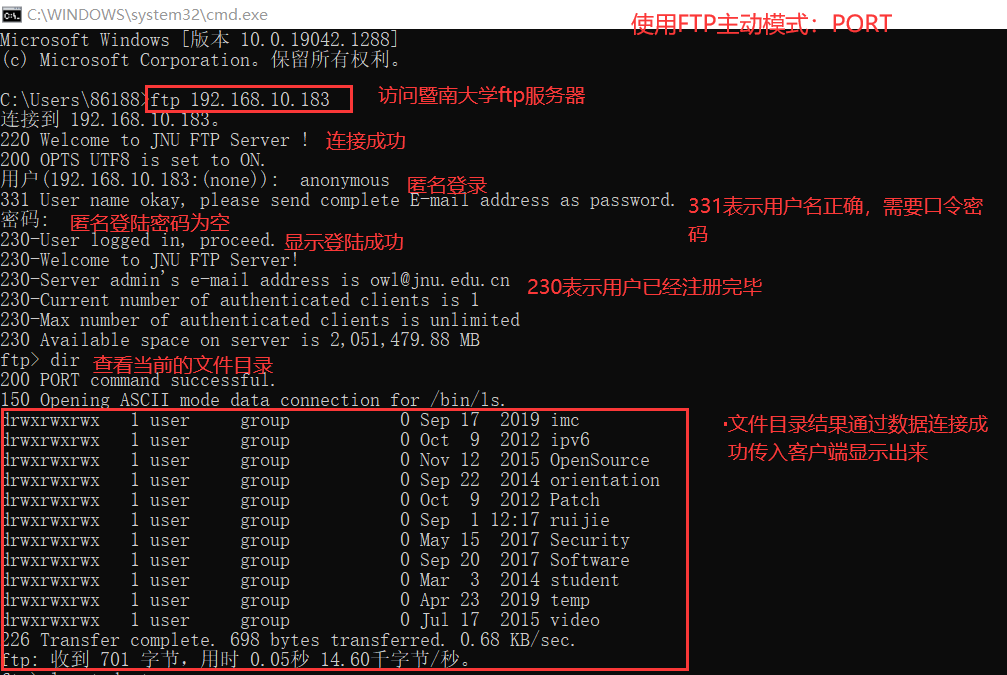
由于我是在文件资源管理器打开的ftp，在文件传输结束之后我立即关闭了wireshark，所以没有看到控制连接关闭的数据包，我猜想是不是控制连接还在等待客户下一步操作所以没有关闭，只有等待一段时间后，控制连接会自动关闭？

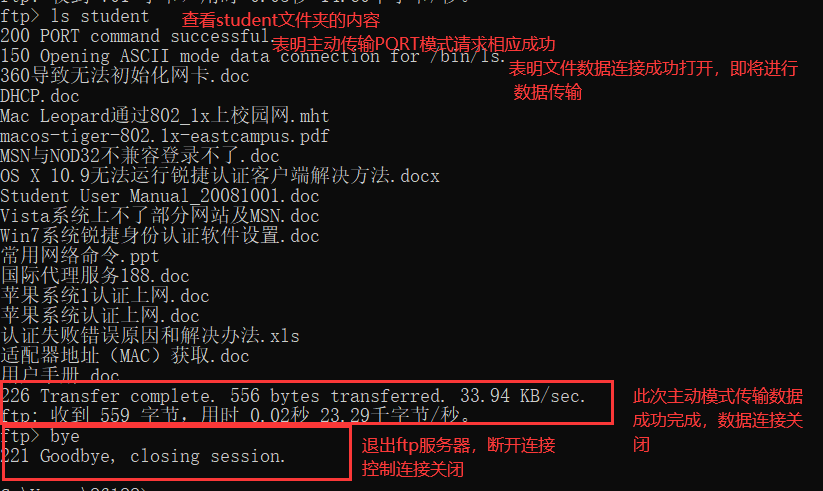
为了证实我的想法，我再次进行同样的操作，这次文件传输过后等待了大概5分钟，成功捕获到了控制连接关闭的数据包，如下图：



**5.3.3 使用主动模式访问ftp服务器**

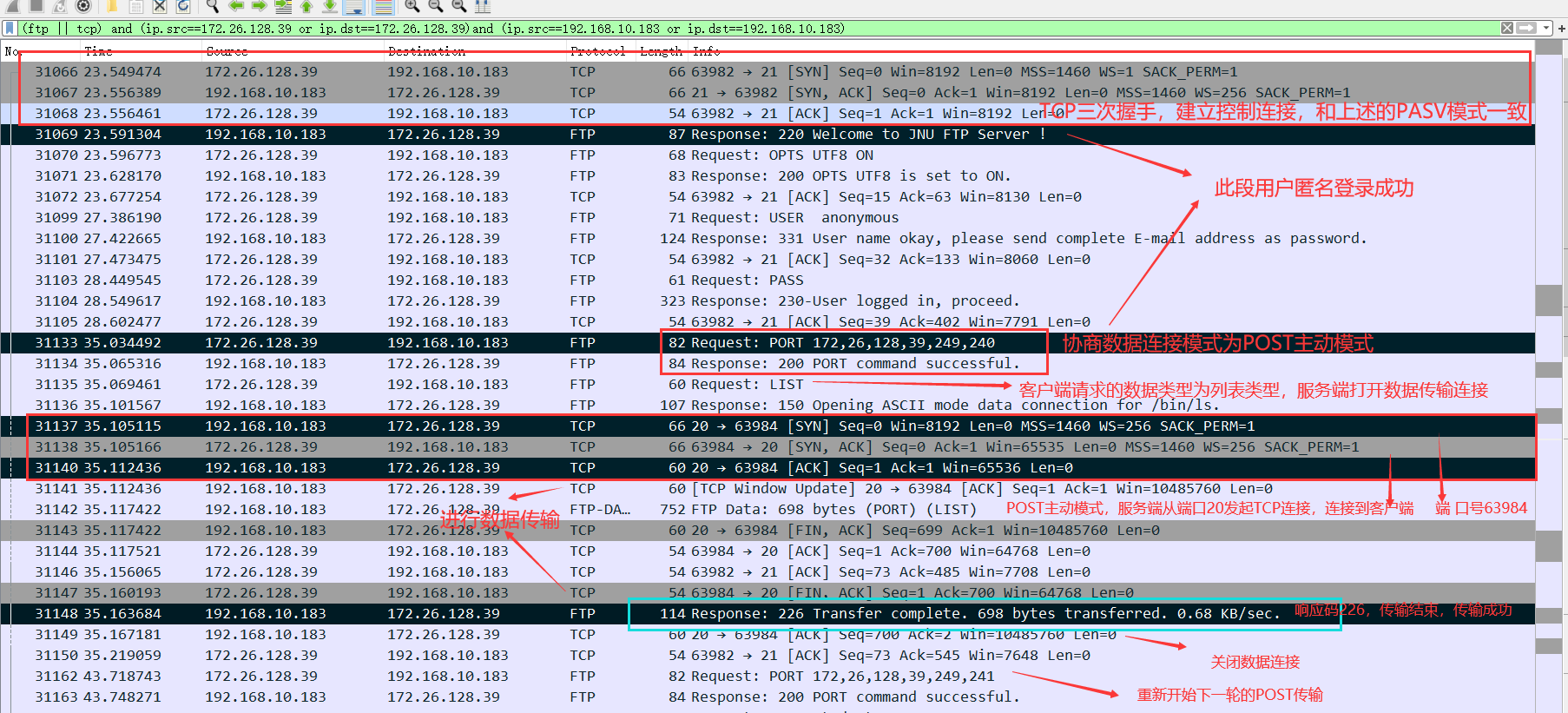
1.使用命令行主动访问ftp，如下图操作



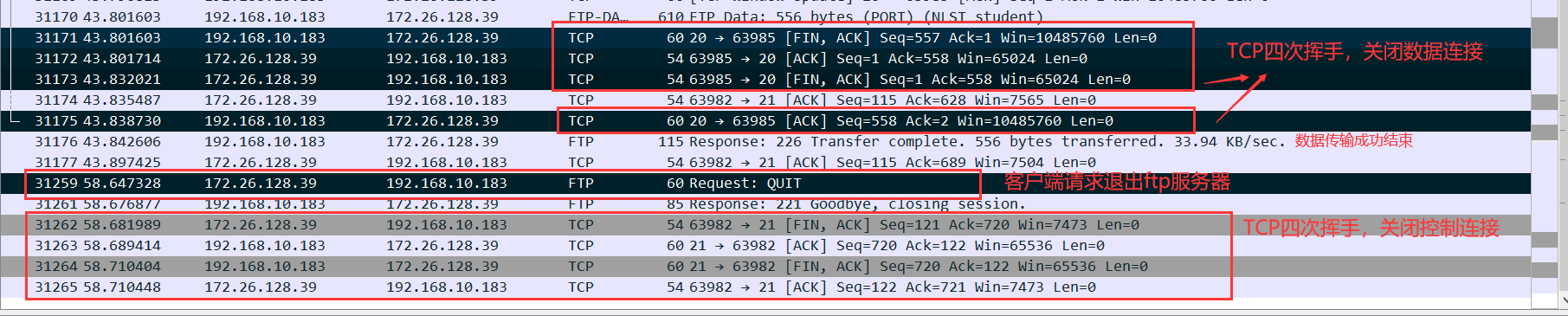


2.分析整个过程wireshark捕获到的报文

如图：



可以看到，主动模式固定使用的是服务器20号端口，在主动模式的 FTP 中，客户端从一个随机的非系统端口(N > 1023)连接到 FTP 服务器的命令端口端口 21。然后，客户端开始监听端口 N+1，并将 FTP 命令端口 N+1 告诉 FTP 服务器，“请把数据发送给我的 N+1 端口”。然后，服务器将从本地数据端口 (端口20) 连接回客户端的数据端口，也就是 N+1 端口。

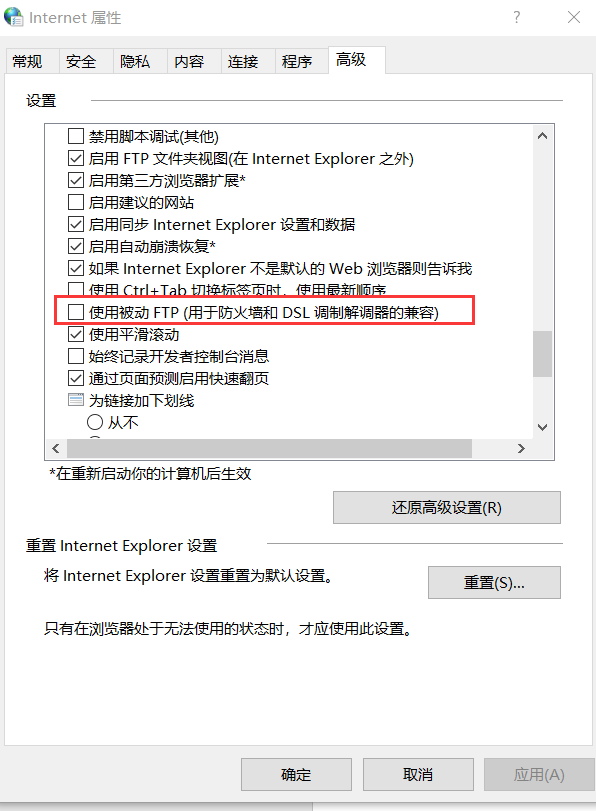


可以看到，主动模式下和被动模式一样，先关闭数据连接，只有等到退出命令的时候，控制连接才会关闭。

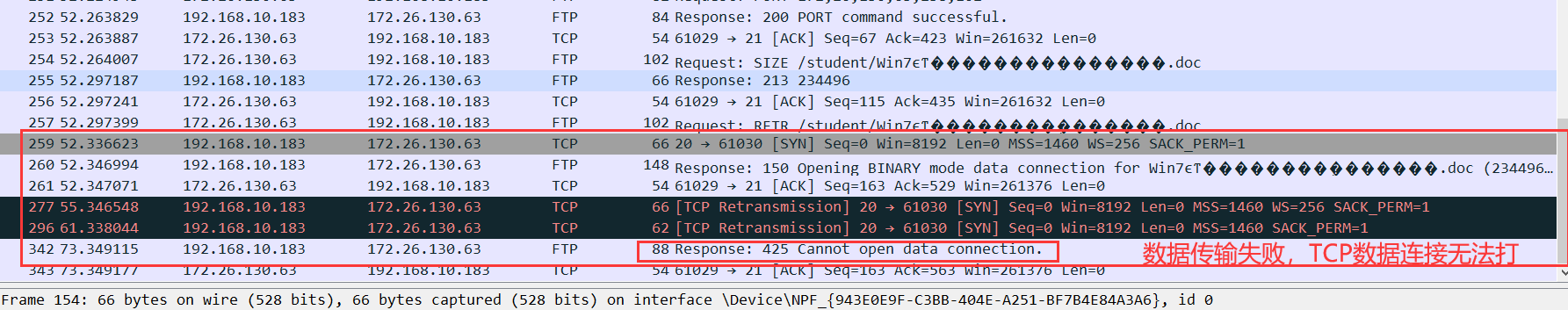
至此，FTP两种工作模式下，客户端下载文件的整个过程分析完毕。

**5.3.4 实验遇到的问题，思考和体会**

在实验中我想从被动模式切换成主动模式，一开始我不是从命令行输入命令，我是把Internet高级属性中的‘使用被动模式FTP’选项关闭，然后从资源管理器打开FTP服务，选择文件下载。



这时候就出问题了，无论我打开那个文件，都打开不成功，也无法下载保存，在wireshark里完全捕获不到FTP文件，打开任意一个文件后我看到wireshark里返回的TCP数据包写着数据连接失败。如图：



这时没过多久我就连续收到两条windows的防火墙警告。（未截图）

百度查找资料之后豁然大悟，原因就在于主动模式和被动模式的区别。

早先所有客户端都使用主动模式，而且工作的很好，而现在因为**客户端防火墙**的存在，**将会关闭一些端口**，这样主动模式将会失败。

主动模式传送数据时是“服务器”连接到“客户端”的端口；被动模式传送数据是“客户端”连接到“服务器”的端口。**主动模式需要客户端必须开放端口给服务器，很多客户端都是在防火墙内，开放端口给FTP服务器访问比较困难。**被动模式只需要服务器端开放端口给客户端连接就行了

所以这就是为什么我无法在FTP下载文件，因为为了电脑的安全，防火墙不允许随便开放端口给其他服务端使用，这样的操作很危险。因此FTP服务器企图与客户端的高位随机端口建立连接，这个端口很有可能被客户端的防火墙阻塞掉。阻塞后就无法下载文件。

在这种情况下就要使用被动模式，但是一些端口也可能被服务器的防火墙封掉。不过因为ftp服务器需要它的ftp服务连接到一定数量的客户端，所以他们总是支持被动模式的。这就是我们为什么要使用被动模式的原因，为了确保数据可以正确的传输，使用被动模式要明显优于主动模式。

（主动（PORT）模式建立数据传输通道是由服务器端发起的，**服务器使用20端口连接客户端的某一个大于1024的端口**；在被动（PASV）模式中，**数据传输的通道的建立是由FTP客户端发起的，他使用一个大于1024的端口连接服务器的1024以上的某一个端口**）

在明白这些原理后，我不仅明白了两种工作模式的区别，还知道了被动模式为什么会出现，以及它的优点，更加对ftp协议工作过程有了理性体会。

**5.4 SMOP和POP3协议分析**

**5.4.1 深入了解SMOP和POP3协议内容**

**SMTP协议：**

SMTP协议是邮件服务器之间传递消息所使用的协议

SMTP规定了在两个相互通信的SMTP进程之间应如何交换信息，负责发送邮件的SMTP进程是SMTP客户,负责接收邮件的进程是SMTP服务器。SMTP规定了14条命令和21种应答信息

(1)客户端：发送消息的服务器

(2)服务器：接收消息的服务器

(3)使用TCP进行email消息的可靠传输

(4)邮件对应的端口为25

(5)传输过程的三个阶段

1.握手(建立连接)

2.消息的传输(邮件传送)

3.关闭(释放连接)

(6)命令/响应交互模式

命令(command): ASCII文本

响应(response): 状态代码和语句

(7)Email消息只能包含7位ASCII码

**POP协议：**

POP3全称为Post Office Protocol version3，即邮局协议第3版。它被用户代理用来邮件服务器取得邮件，采用的也是C/S通信模型。

用户从邮件服务器上接收邮件的典型通信过程如下。

1.用户运行用户代理（如Foxmail, Outlook Express）。

2.用户代理（以下简称客户端）与邮件服务器（以下简称服务器端）的110端口建立TCP连接。

3.客户端向服务器端发出各种命令，来请求各种服务（如查询邮箱信息，下载某封邮件等）。

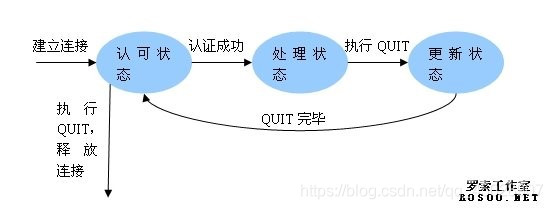
4.服务端解析用户的命令，做出相应动作并返回给客户端一个响应。

5.3和4交替进行，直到接收完所有邮件转到步骤6，或两者的连接被意外中断而直接退出。

用户代理解析从服务器端获得的邮件，以适当地形式（如可读）的形式呈现给用户。

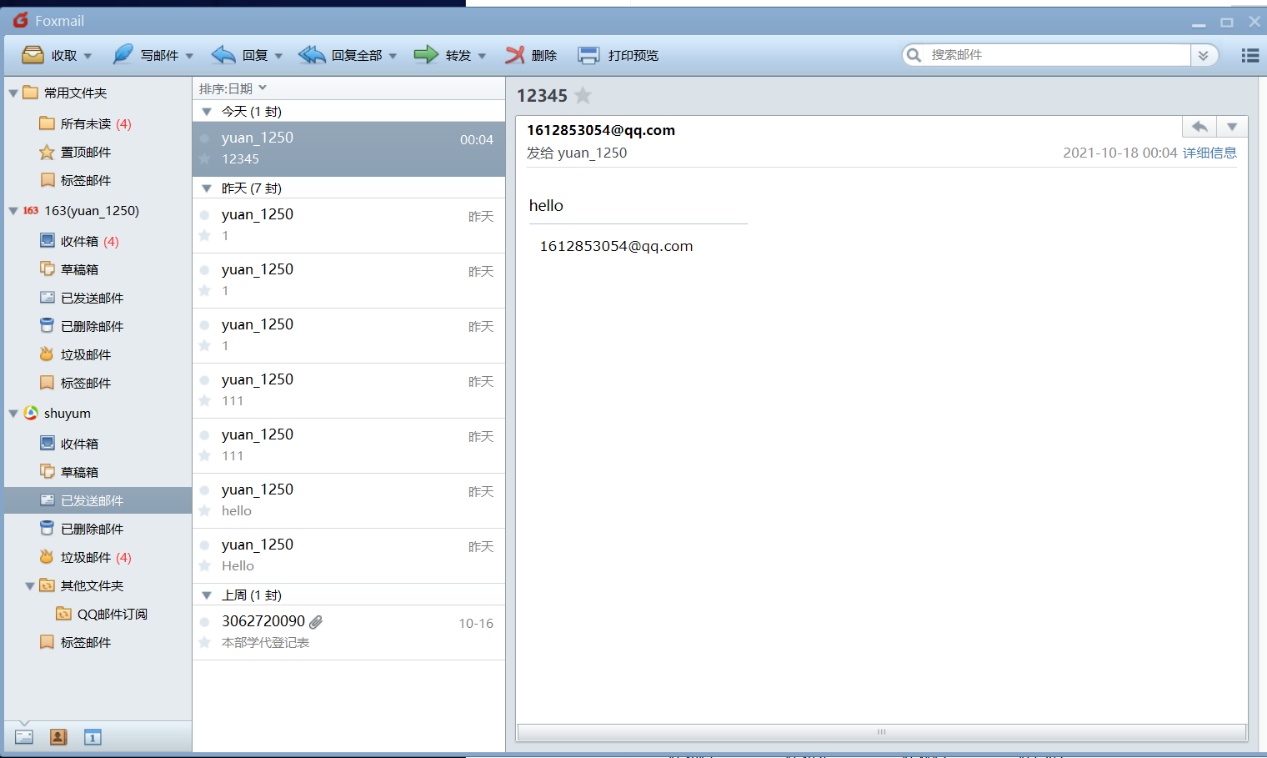
其中2、3和4用POP3协议通信。

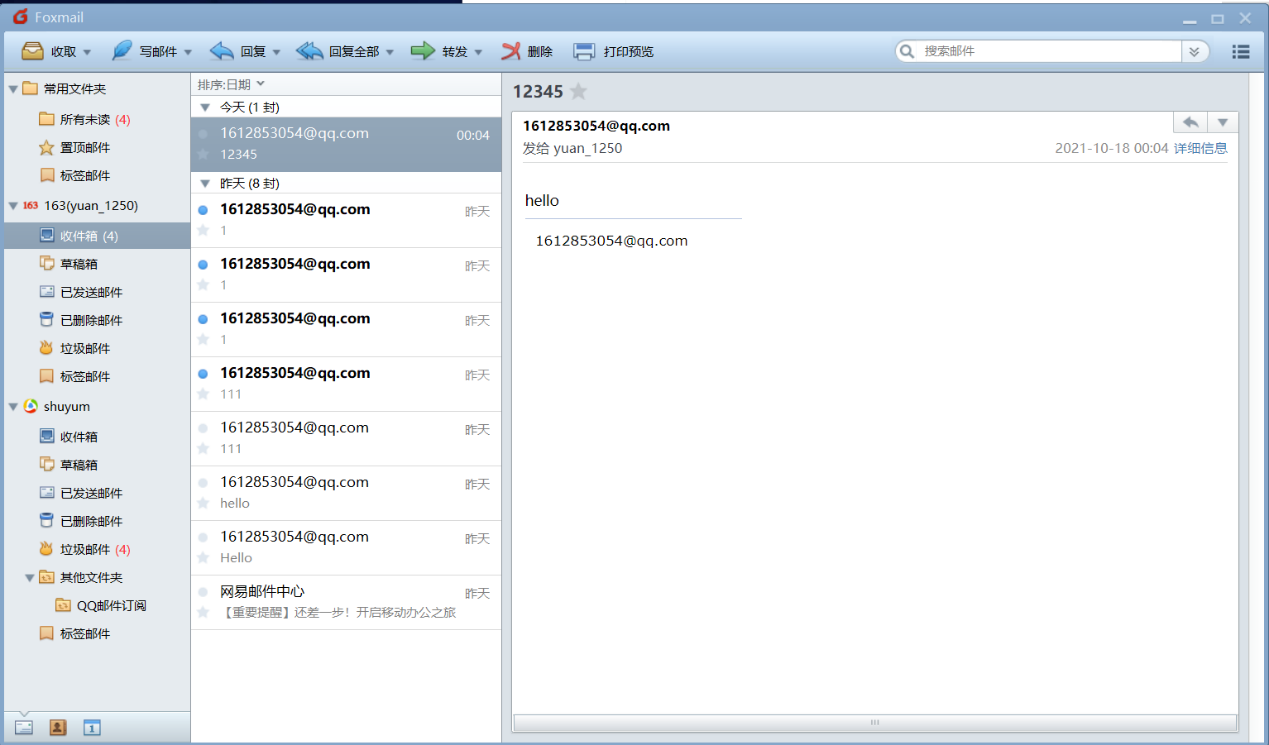
POP3协议中有三种状态，认证状态，处理状态，和更新状态。命令的执行可以改变协议的状态，而对于具体的某命令，它只能在具体的某状态下使用。客户机与服务器刚与服务器建立连接时，它的状态为认证状态；一旦客户机提供了自己身份并被成功地确认，即由认可状态转入处理状态； 在完成相应的操作后客户机发出QUIT命令，则进入更新状态，更新之后又重返认可状态；当然在认可状态下执行QUIT命令，可释放连接。状态间的转移如图所示。



**5.4.2 分析SMTP和POP协议报文**

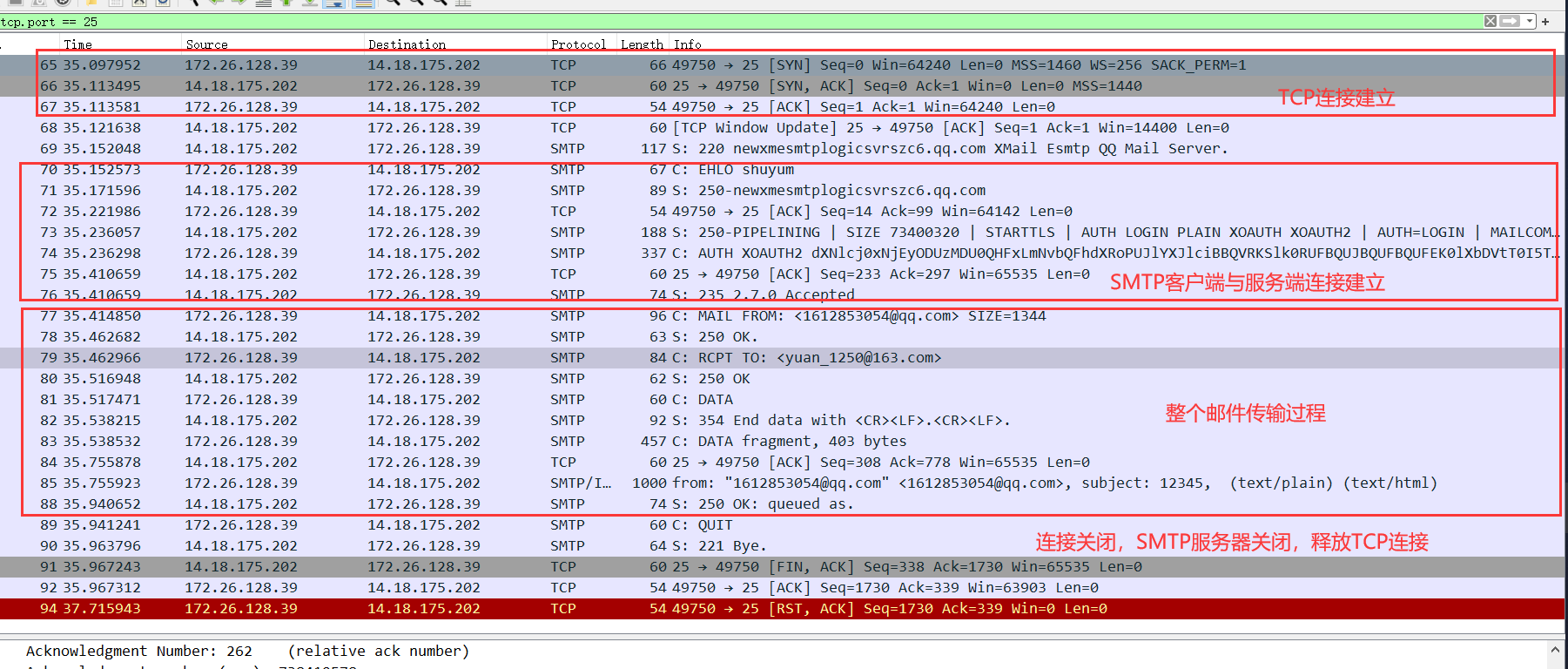
**1.使用Foxmail用我的qq邮箱向我的163邮箱发送一封邮件，并从163收取我的邮件，同时用wireshark捕获整个过程的报文如下：**





**2.分析SMTP协议**

**WIireshark截图：**



65-68号数据包：SMTP客户与目的主机的SMTP服务器建立TCP连接，可见SMTP是使用服务器端口25

69-76号数据包：连接建立后，接收方SMTP服务器发出220 Service ready (服务就绪)，然后SMTP客户向SMTP服务器发送HELO命令，附上发送方的主机名，这里发送方的主机名为shuyum，响应235表示用户验证成功 ，并且服务器响应已准备好接受

【HELO–发件方问候收件方，后面是发件人的服务器地址或标识。收件方回答OK时标识自己的身份。问候和确认过程表明两台机器可以进行通信，同时状态参量被复位，缓冲区被清空。】

77-78号数据包：使用MAIL命令：MAI这个命令用来开始传送邮件，它的后面跟随发件方邮件地址（我的qq邮箱）。

79-80号数据包：使用RCPT命令：RCPT–这个命令告诉收件方收件人的邮箱。它后面跟着的是收件箱163的邮箱名字

81-88号数据包：使用DATA：收件方把该命令之后的数据作为发送的数据。数据被加入数据缓冲区中，发给邮件服务器

重点分析第85号数据包，它带有IMF,里面写有邮件信息，如下图：



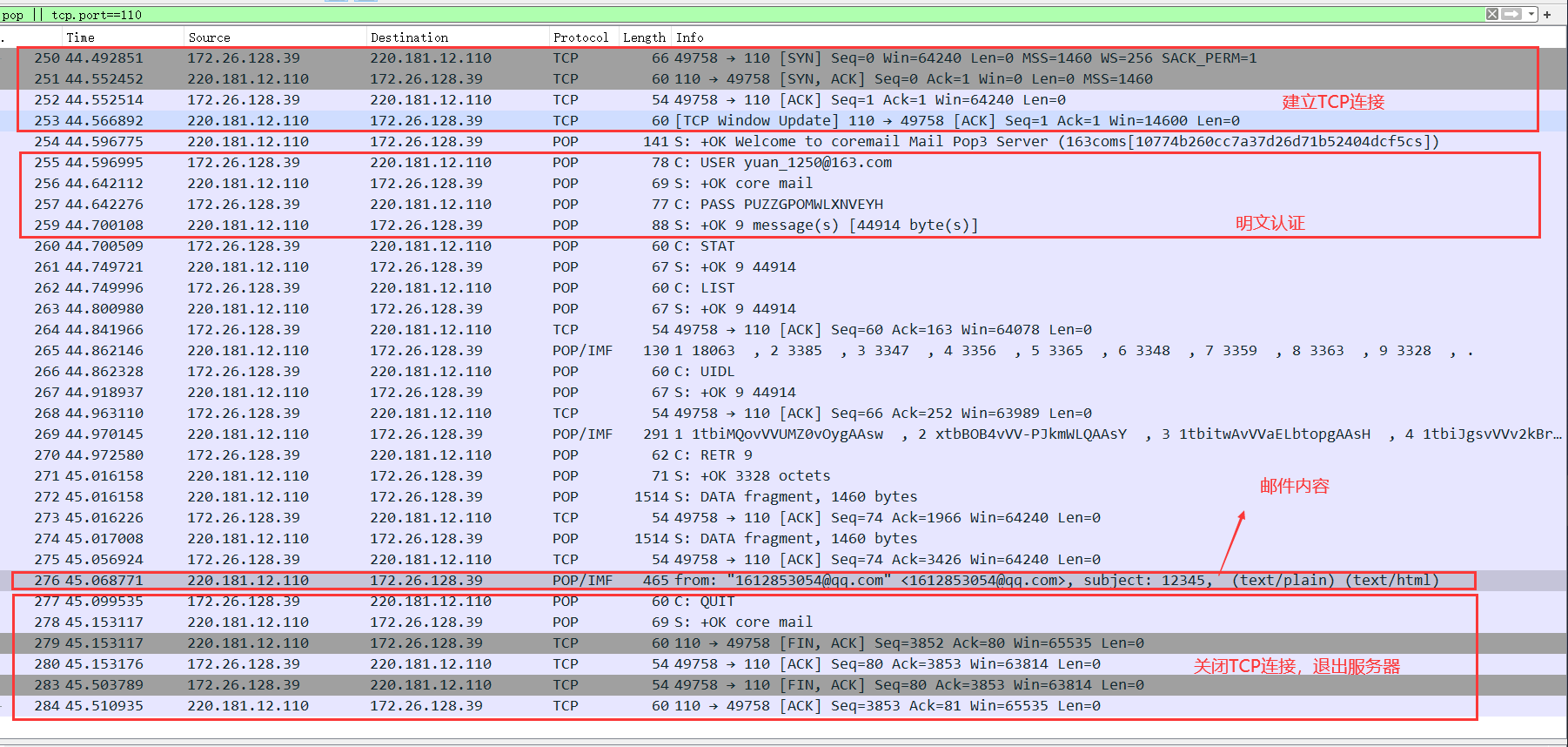
89-94数据包：邮件发送完毕后，SMTP客户应发送QUIT命令，使用quit命令：SMTP要求接收方必须回答OK，然后中断传输。

SMTP服务器返回的信息是221 (服务关闭)，表示SMTP同意释放TCP连接, 邮件传送的全部过程就此结束

至此SMTP的协议分析结束

**5.4.3 分析POP协议**

1.捕获POP数据包以及其TCP连接数据包如下图：



250-253号数据包：与POP3邮件处理器的110号端口建立TCP连接

254号数据包：POP3服务器响应成功，准备就绪

255-259号数据包：POP协议采用铭文认证，认证我的163邮箱账户，可以看到255和257号数据包分别显示了我的用户名和密码，259表示认证成功

260-267号：请求服务器发回关于邮箱的各类统计资料，返回邮件的唯一标识符

264-276号：向用户传输邮件内容，观察276号数据包：

276的IMF：



277-278号：在完成相应的下载读取操作后客户机发出QUIT命令，进入更新状态，更新之后又重返认可状态

此处在认可状态下执行QUIT命令，，服务器同意关闭，即将释放了TCP连接。

279-284：TCP四次挥手，关闭TCP连接

至此，POP协议分析完毕

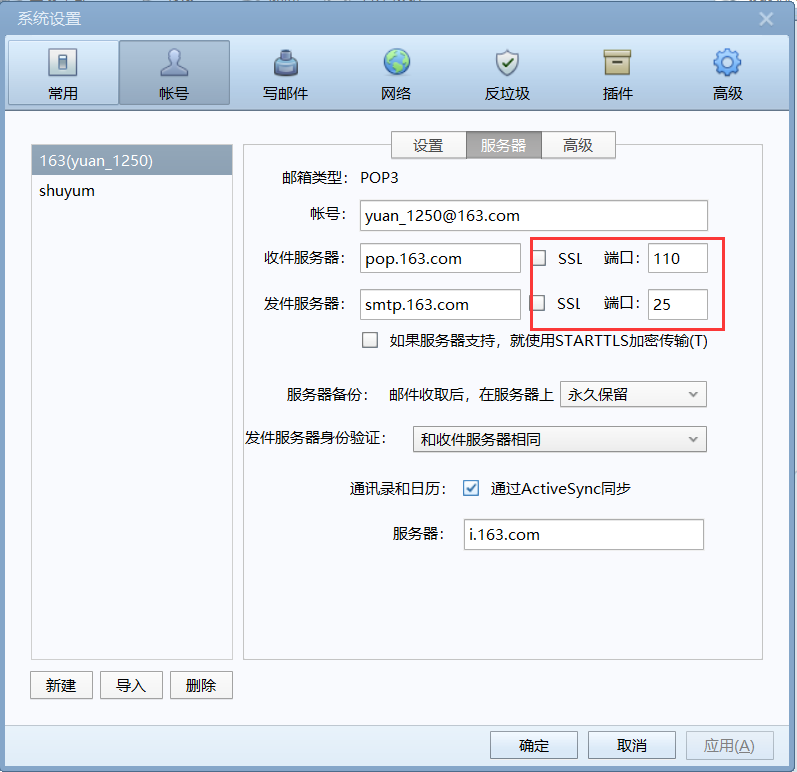
**5.4.4 实验的遇到的问题，思考和体会**

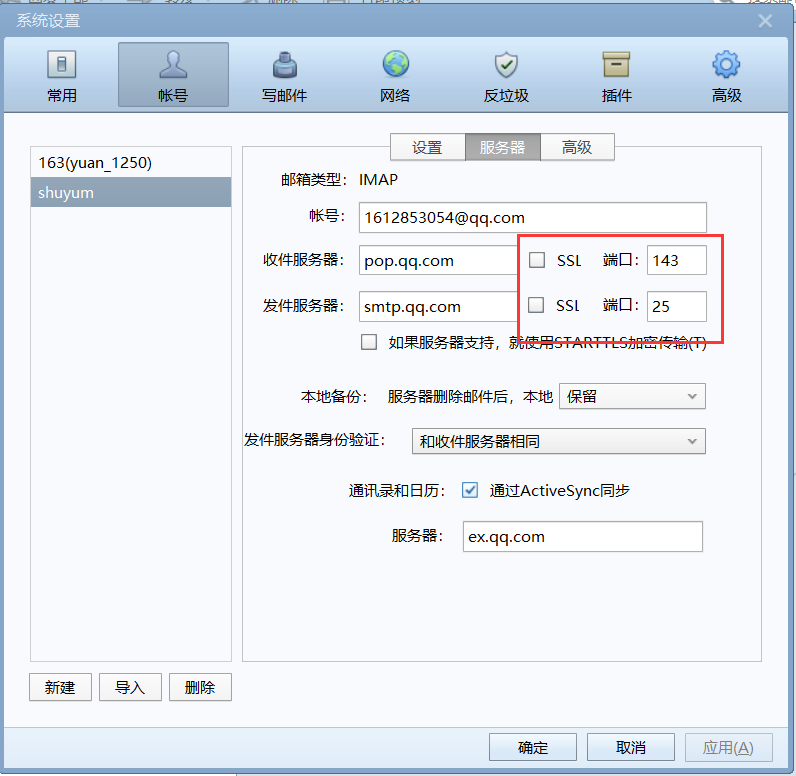
实验过程中一开始碰到了一些问题，当我下载完Foxmail并成功完成发送以及接受文件后，我的wireshark一直捕获不到SMTP和POP数据包，但是邮件确实发送成功。我检查配置后，发现都没有问题。

在经过不断查找资料，最后找到了捕获不到的原因：原因是我在配置的时候选择了SSL端口，即使用了ssl加密协议，在做实验的时候看到smtp和pop都是属于明文认证，在pop协议里面可以直接看到我的密码，这是非常危险的，所以现在普遍默认都是使用ssl加密协议，电子邮件服务器SSL安全证书确保客户端设备到邮件服务器端数据传输为加密方式。

除非获取SSL证书的私钥，否则目前第三方无法解密数据（不包括旧版本SHA1加密方式，因为其密钥太短）。因此使用ssl加密协议后不会看到SMTP和POP数据包

把ssl取消后，成功捕获到了数据包，这次的错误让给我对SMTP和POP的明文运输方式有了深刻的认识，也了解了什么是SSL协议，第一次初步了解加密协议的用处，了解它是如何工作的以及它的重要性，同时也是对SMTP和POP协议的工作方式有了更加深刻的体会和认知





**6.【实验总结】**

本次实验通过分析HTTP，DNS，FTP，SMTP, POP协议，分析5种的报文和格式，我了解了应用层这几种协议的工作流程和方式，明白了各协议在应用层的作用，这次通过wireshark对各个协议数据包工作流程的仔细研读，使得我对OSI模型中各个层次是怎么嵌套，怎么一层层向上提供服务有了更深理性的认识，不如之前云里雾里了。

其次，通过查阅资料和文献，之前在上课不明白的问题甚至是实验遇到的问题也都解决了，学会了很多知识，不仅不仅只是协议的相关知识，还拓展出很多知识，比如我不了解的SSL协议，还有防火墙的用处，CANAME给域名起别名的用处，以及TCP连接和UDP连接的差异，还有对TCP三次握手四次挥手有了进一步的了解等等。