**实验七 应用层协议Ⅰ-HTTP协议分析**

**班 级：网络工程班**

**姓 名：沈 梓 建**

**学 号：2015118066**

目录

[一、实验目的 1](#_Toc499840170)

[二、实验环境 1](#_Toc499840171)

[三、实验步骤 1](#_Toc499840172)

[1．基础知识 1](#_Toc499840173)

[2．DNS 域名解析过程 2](#_Toc499840174)

[3．实验复现 5](#_Toc499840175)

[（1）HTTP GET请求 6](#_Toc499840176)

[（2）HTTP响应 7](#_Toc499840177)

[（3）对每个URL的多重GET请求 8](#_Toc499840178)

[（4）明文数据 9](#_Toc499840179)

[4．作业 11](#_Toc499840180)

[四、实验心得 14](#_Toc499840181)

**实验七 应用层协议Ⅰ-HTTP协议分析**

**一、实验目的**

1．掌握HTTP协议的实现细节；

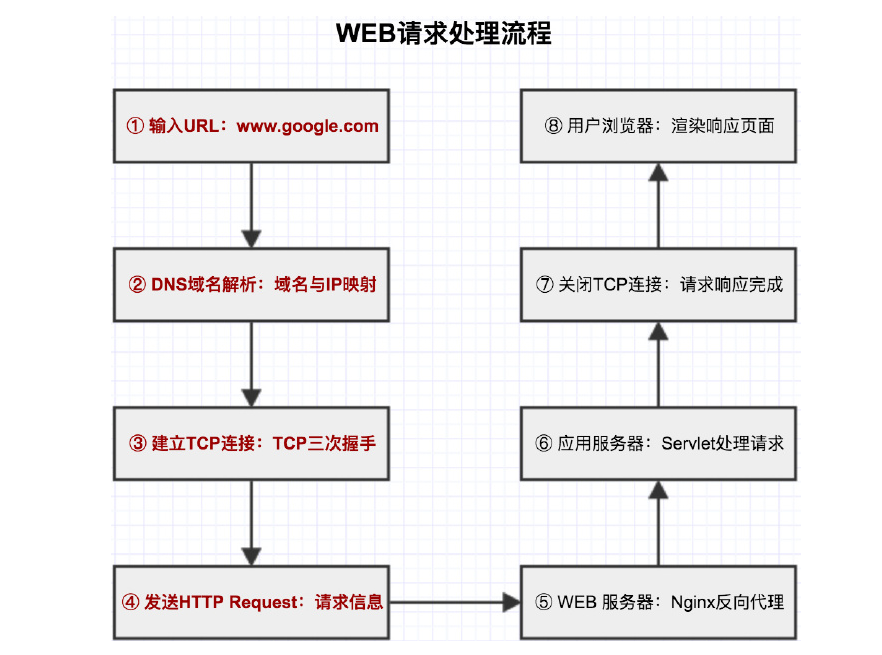
2．了解HTTP高速缓存、授权和cookie。

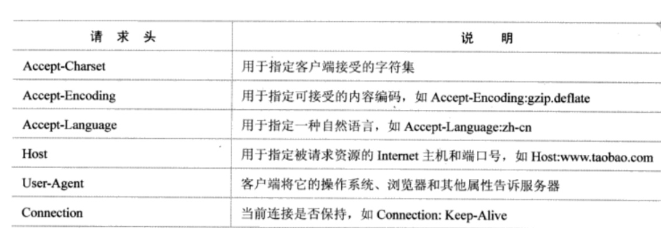
**二、实验环境**

一台联网的PC机。

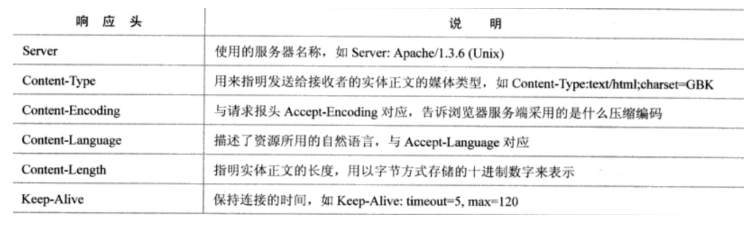
**三、实验步骤**

1．基础知识

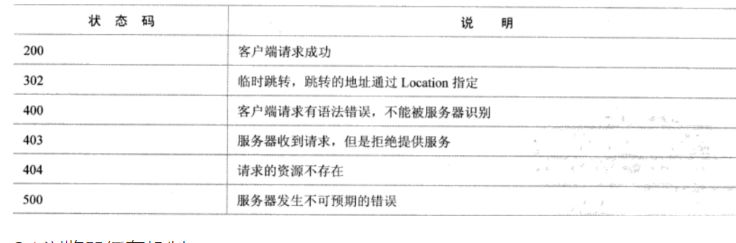




【常见的HTTP请求头】

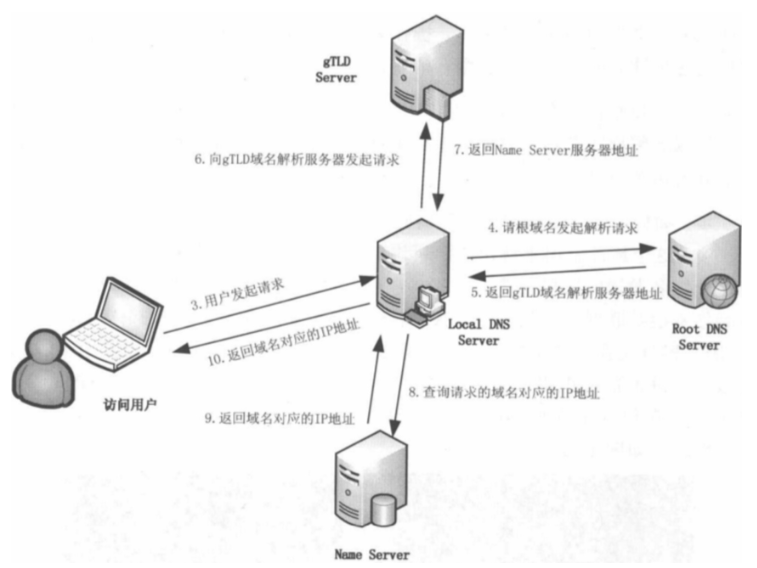


【常见的HTTP响应头】



【常见的HTTP状态码】

2．DNS 域名解析过程



（举例访问谷歌：）

a.浏览器缓存检查（本机）浏览器会首先搜索浏览器自身的DNS缓存（缓存时间比较短，大概只有1分钟，且只能容纳1000条缓存），看自身的缓存中是否有www.google.com对应的条目，而且没有过期，如果有且没有过期则解析到此结束。浏览器缓存域名也是有限制的，不仅浏览器缓存大小有限制，而且缓存的时间也有限制，通常情况下为几分钟到几小时不等，域名被缓存的时间限制可以通过TTL属性来设置。这个缓存时间太长和太短都不好，如果缓存时间太长，一旦域名被解析到的IP有变化，会导致被客户端缓存的域名无法解析到变化后的IP地址，以致该域名不能正常解析，这段时间内有可能会有一部分用户无法访问网站。如果时间设置太短，会导致用户每次访问网站都要重新解析一次域名。

b. 操作系统缓存检查（本机）+hosts解析（本机）如果浏览器自身的缓存里面没有找到对应的条目，其实操作系统也会有一个域名解析的过程，那么Chrome会首先搜索操作系统自身的DNS缓存中是否有这个域名对应的DNS解析结果，如果找到且没有过期则停止搜索解析到此结束。其次在Linux中可以通过/etc/hosts文件来设置，你可以将任何域名解析到任何能够访问的IP地址。如果你在这里指定了一个域名对应的IP地址，那么浏览器会首先使用这个IP地址。当解析到这个配置文件中的某个域名时，操作系统会在缓存中缓存这个解析结果，缓存的时间同样是受这个域名的失效时间和缓存的空间大小控制的。

c. 本地区域名服务器解析（LDNS）如果在hosts文件中也没有找到对应的条目，浏览器就会发起一个DNS的系统调用，就会向本地配置的首选DNS服务器（LDNS一般是电信运营商提供的，也可以使用像Google提供的DNS服务器）发起域名解析请求（通过的是UDP协议向DNS的53端口发起请求，这个请求是递归的请求，也就是运营商的DNS服务器必须得提供给我们该域名的IP地址）。在我们的网络配置中都会有“DNS服务器地址”这一项，这个地址就用于解决前面所说的如果两个过程无法解析时要怎么办，操作系统会把这个域名发送给这里设置的LDNS，也就是本地区的域名服务器。这个DNS通常都提供给你本地互联网接入的一个DNS解析服务，例如你是在学校接入互联网，那么你的DNS服务器肯定在你的学校，如果你是在一个小区接入互联网的，那这个DNS就是提供给你接入互联网的应用提供商，即电信或者联通，也就是通常所说的SPA，那么这个DNS通常也会在你所在城市的某个角落，通常不会很远。这个专门的域名解析服务器性能都会很好，它们一般都会缓存域名解析结果，当然缓存时间是受域名的失效时间控制的，一般缓存空间不是影响域名失效的主要因素。大约80%的域名解析都到这里就已经完成了，所以LDNS主要承担了域名的解析工作。

运营商的DNS服务器首先查找自身的缓存，找到对应的条目，且没有过期，则解析成功。

d. 根域名服务器解析（Root Server）如果LDNS没有找到对应的条目，则有运营商的DNS代我们的浏览器发起迭代DNS解析请求。它首先是会找根域的DNS的IP地址（这个DNS服务器都内置13台根域的DNS的IP地址），找到根域的DNS地址，就会向其发起请求（请问www.google.com这个域名的IP地址是多少啊？）。

e. 根域名服务器返回给本地域名服务器一个所查询域的主域名服务器(gTLD Server)地址，gTLD是国际顶级域名服务器，如.com、.cn、.org等，全球只有13台左右。根域发现这是一个顶级域com域的一个域名，于是就告诉运营商的DNS我不知道这个域名的IP地址，但是我知道com域的IP地址，你去找它去。

f. 本地域名服务器(Local DNS Server)再向上一步返回的gTLD服务器发送请求。于是运营商的DNS就得到了com域的IP地址，又向com域的IP地址发起了请求（请问www.google.com这个域名的IP地址是多少?），com域这台服务器告诉运营商的DNS我不知道www.google.com这个域名的IP地址，但是我知道google.com这个域的DNS地址，你去找它去。

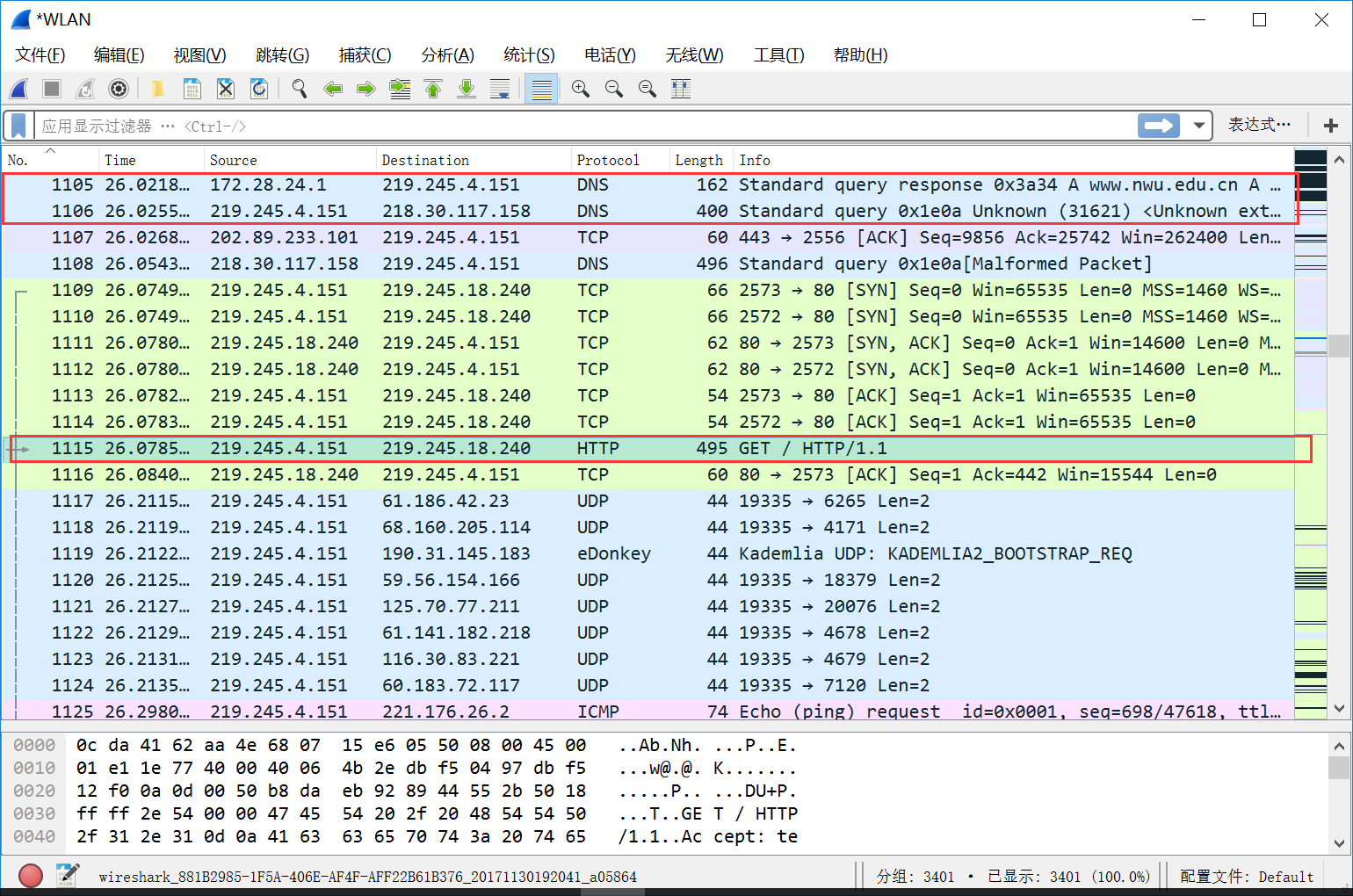
g, 接受请求的gTLD服务器查找并返回此域名对应的Name Server域名服务器的地址，这个Name Server通常就是你注册的域名服务器，例如你在某个域名服务提供商申请的域名，那么这个域名解析任务就由这个域名提供商的服务器来完成。于是运营商的DNS又向google.com这个域名的DNS地址（这个一般就是由域名注册商提供的，像万网，新网等）发起请求（请问www.google.com这个域名的IP地址是多少？），这个时候google.com域的DNS服务器一查，果真在我这里，于是就把找到的结果发送给运营商的DNS服务器，这个时候运营商的DNS服务器就拿到了www.google.com这个域名对应的IP地址。

h. Name Server域名服务器会查询存储的域名和IP的映射关系表，正常情况下都根据域名得到目标IP记录，连同一个TTL值返回给DNS Server域名服务器。

返回该域名对应的IP和TTL值，Local DNS Server会缓存这个域名和IP的对应关系，缓存的时间由TTL值控制。

把解析的结果返回给用户，用户根据TTL值缓存在本地系统缓存中，域名解析过程结束。

3．实验复现



首先看一下分组1105和分组1106：

分组1105是一个将域名[www.nwu.edu.cn转换成对应的IP](http://www.nwu.edu.cn转换成对应的IP)地址的请求；

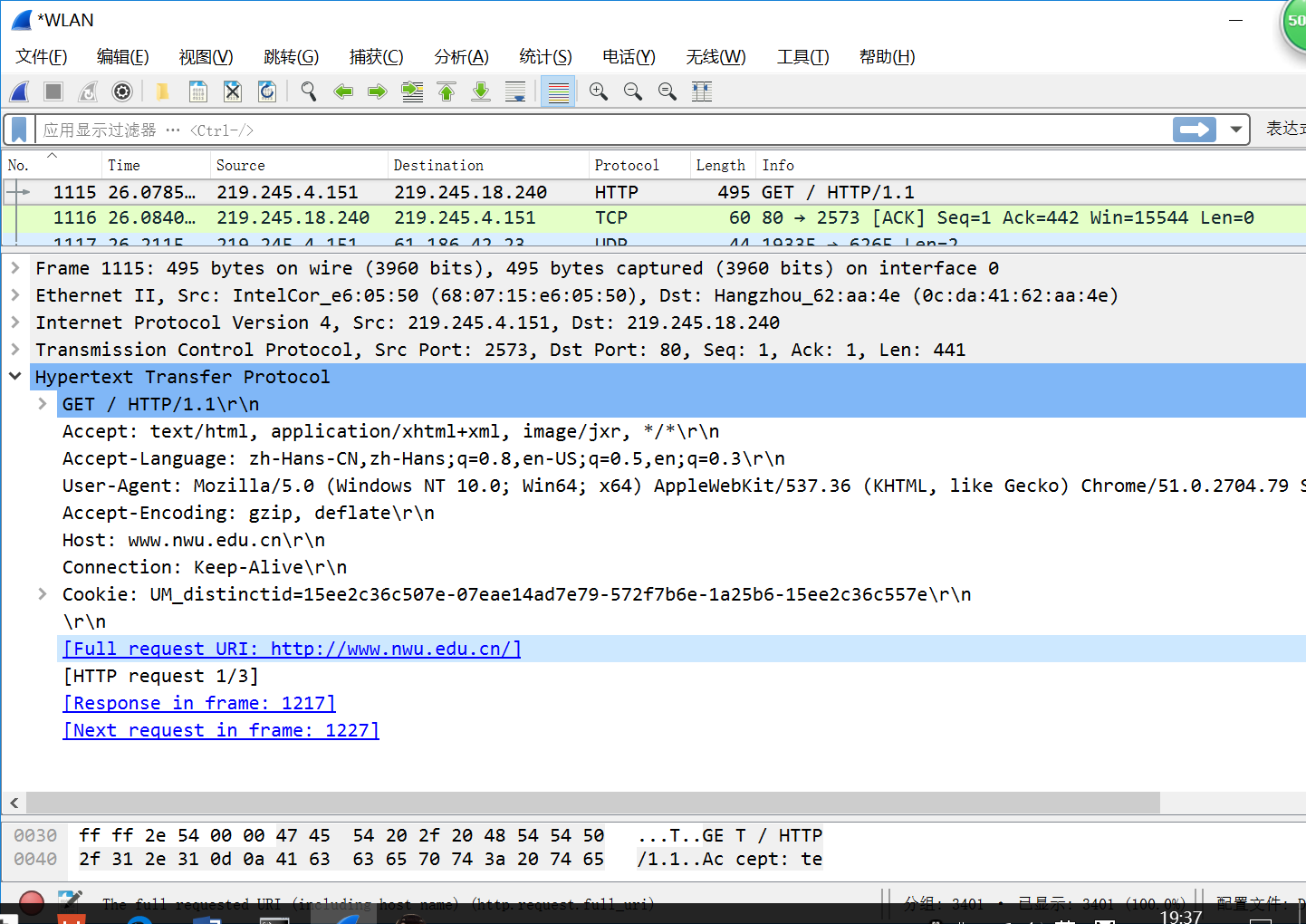
分组1106包含了转换的结果219.245.18.240

这两个分组使用的应用层协议是DNS。

分组1108也是一个DNS的应用层协议，注意：malformed packet是一个数据包，可以用来检测数据在转发传输的过程中是否某些字节发生错误。

分组1109到1114在Web浏览器与Web服务器间建立了一个TCP连接。

最后，Web浏览器使用已建立好的TCP连接来发送请求“GET/HTTP/1.1”，如分组1115所示。



【HTTP GET请求】

（1）HTTP GET请求

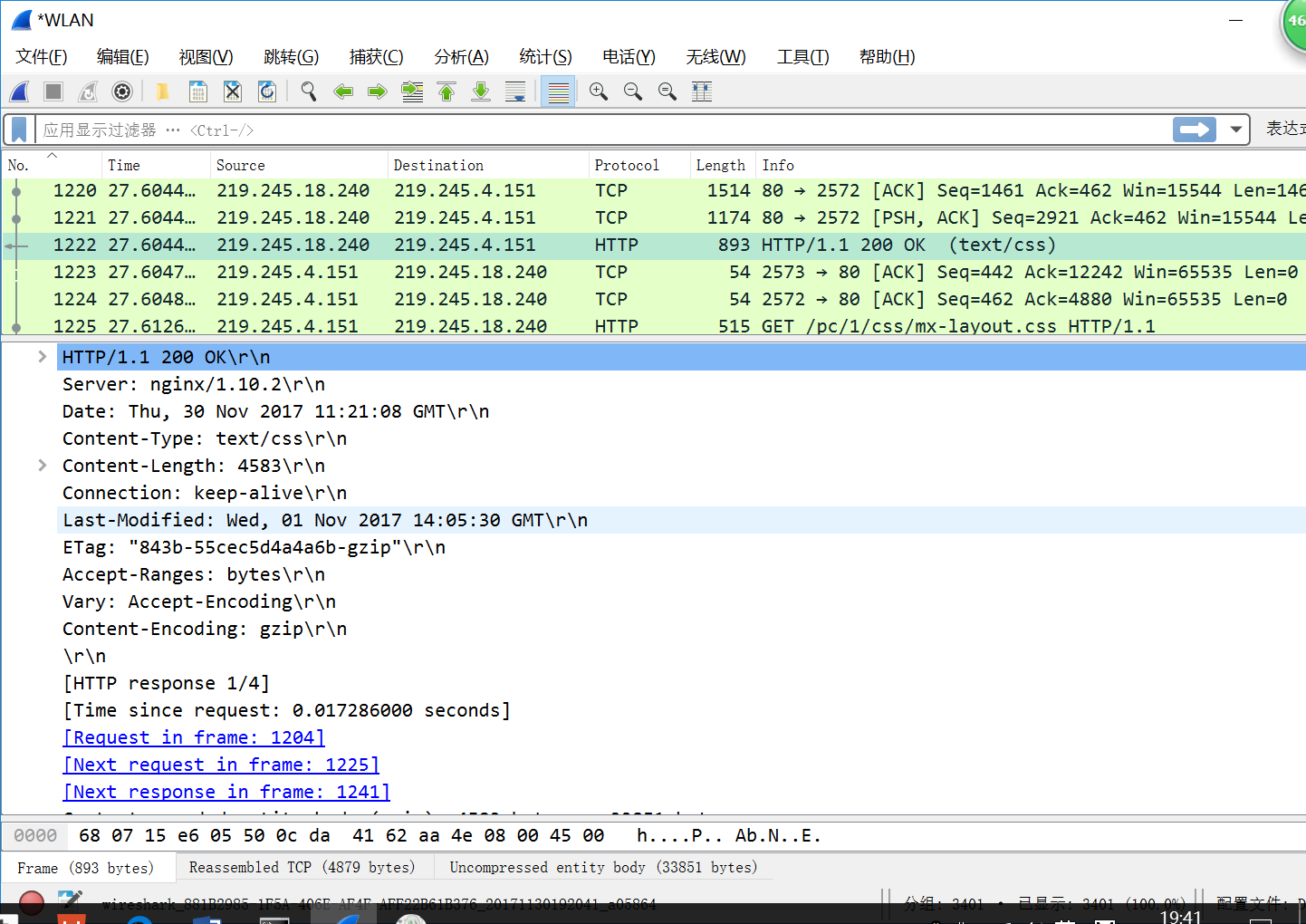
这个基本请求行后跟随着一系列额外的请求首部。在首部后的“\r\n”表示一个回车和换行，以此将该首部与下一个首部隔开。

“Host”首部在HTTP1.1中是必需的，它描述了URL中机器中的域名，本实验中是[www.nwu.edu.cn](http://www.nwu.edu.cn)。

User-Agent首部描述了提出请求的Web浏览器及客户机器。

一系列Accept首部：包括Accept（接受）、Accept-Language（接受语言）、 Accept-Encoding（接受编码）、Accept-Charset（接受字符集）。它们告诉Web服务器客户Web浏览器准备处理的数据类型。Web服务器可以将数据转变为不同的语言和格式。这些首部表明了客户的能力及偏好。

Keep-Alive及Connection首部描述了有关TCP连接的信息，通过此连接发送HTTP请求和响应。它表明在发送请求之后连接是否保持活动状态及保持多久。



【HTTP响应】

（2）HTTP响应

响应首先发送“HTTP/1.1 200 OK”，指明由它开始使用HTTP1.1版本来发送网页。同样，在响应分组中，它后面也跟随着一些首部，最后，被请求的实际数据被发送。

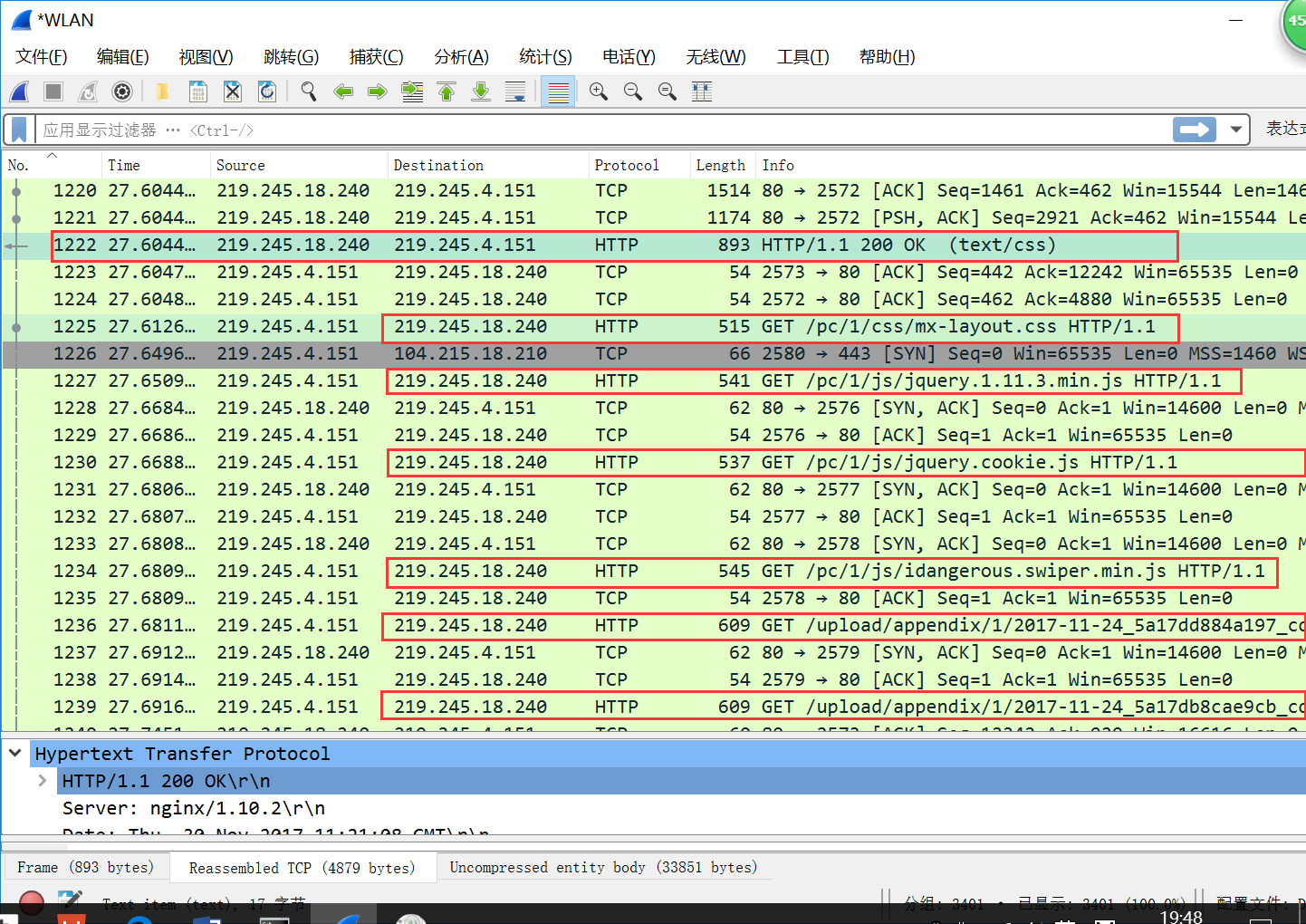
Date首部：列出数据发送的日期和时间。

Content-Length首部：描述数据的长度。

Content-Type首部：描述内容的类型。

Cache-Control首部：描述是否将数据的副本存储或告诉缓存起来，以便将来引用。

Content-Encoding首部：描述内容的编码方式。为“gzip”表明数据部分是压缩了的HTML。



【对每个URL的多重GET请求】

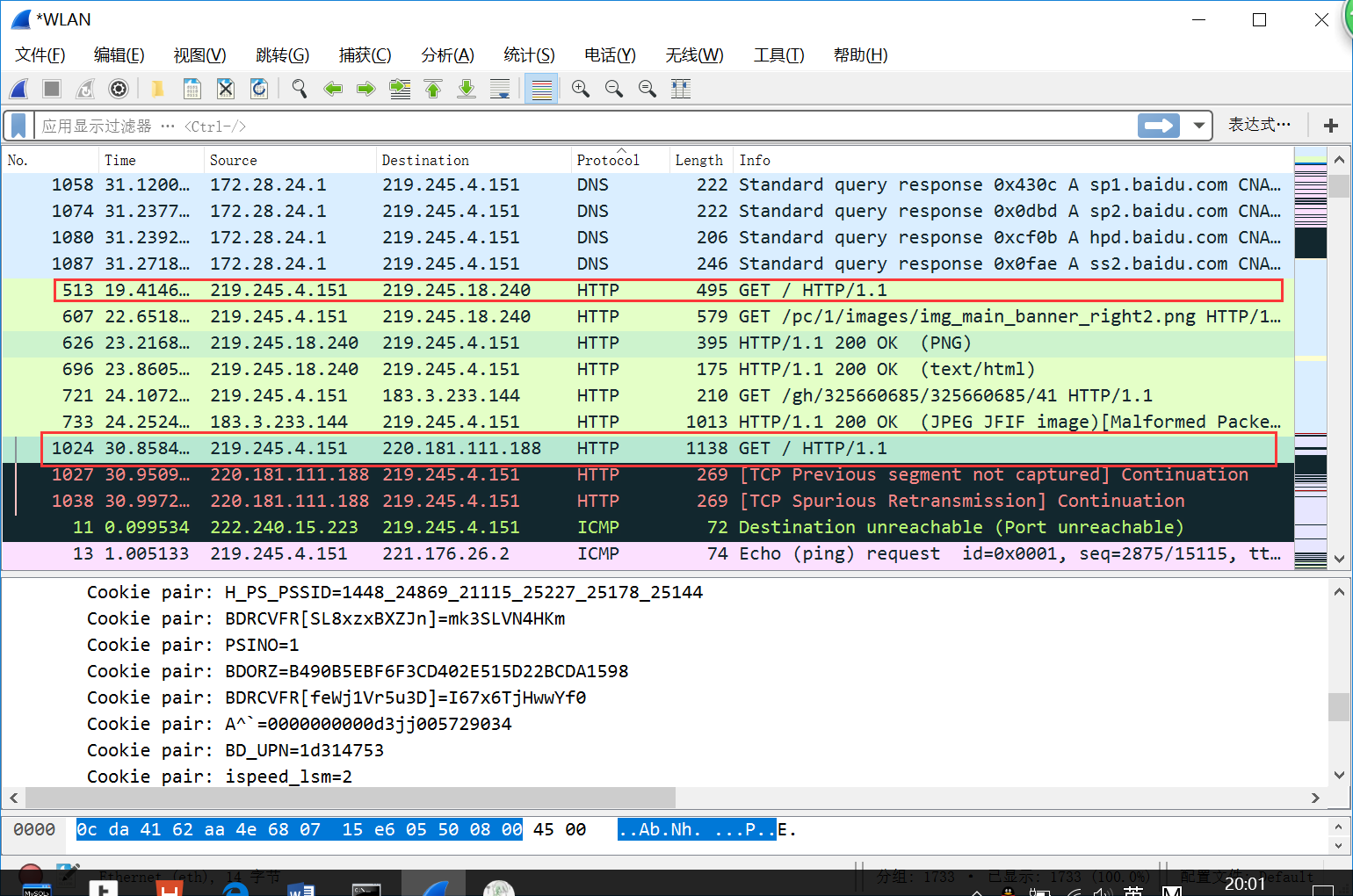
（3）对每个URL的多重GET请求

虽然我们只输入了一个URL，但在分组1225、1227、1230、1234、1236、1239中却出现了另一个到[www.nwu.edu.cn](http://www.nwu.edu.cn)的连接请求“GET /pc/1/css/mx-layout.css HTTP/1.1”等等。

我们在第一个到[www.nwu.edu.cn](http://www.nwu.edu.cn)的连接请求中已经获得了它的主页。第二请求是为了获得学校官网主页上显示的图标。这个图标并不与HTML的源文件存储于同一个文件内，而是被HTML源文件的一个图像标签所引用。

所以其他请求由Web浏览器本身发出。在浏览器接收到Web服务器的响应和没有压缩的HTML时，它必须解释HTML文件来知道如何向用户显示这个页面。它告诉Web浏览器为了显示这个页面，就必须从这个服务器取得响应的图片，这样才能让页面和图像一起显示给用户。

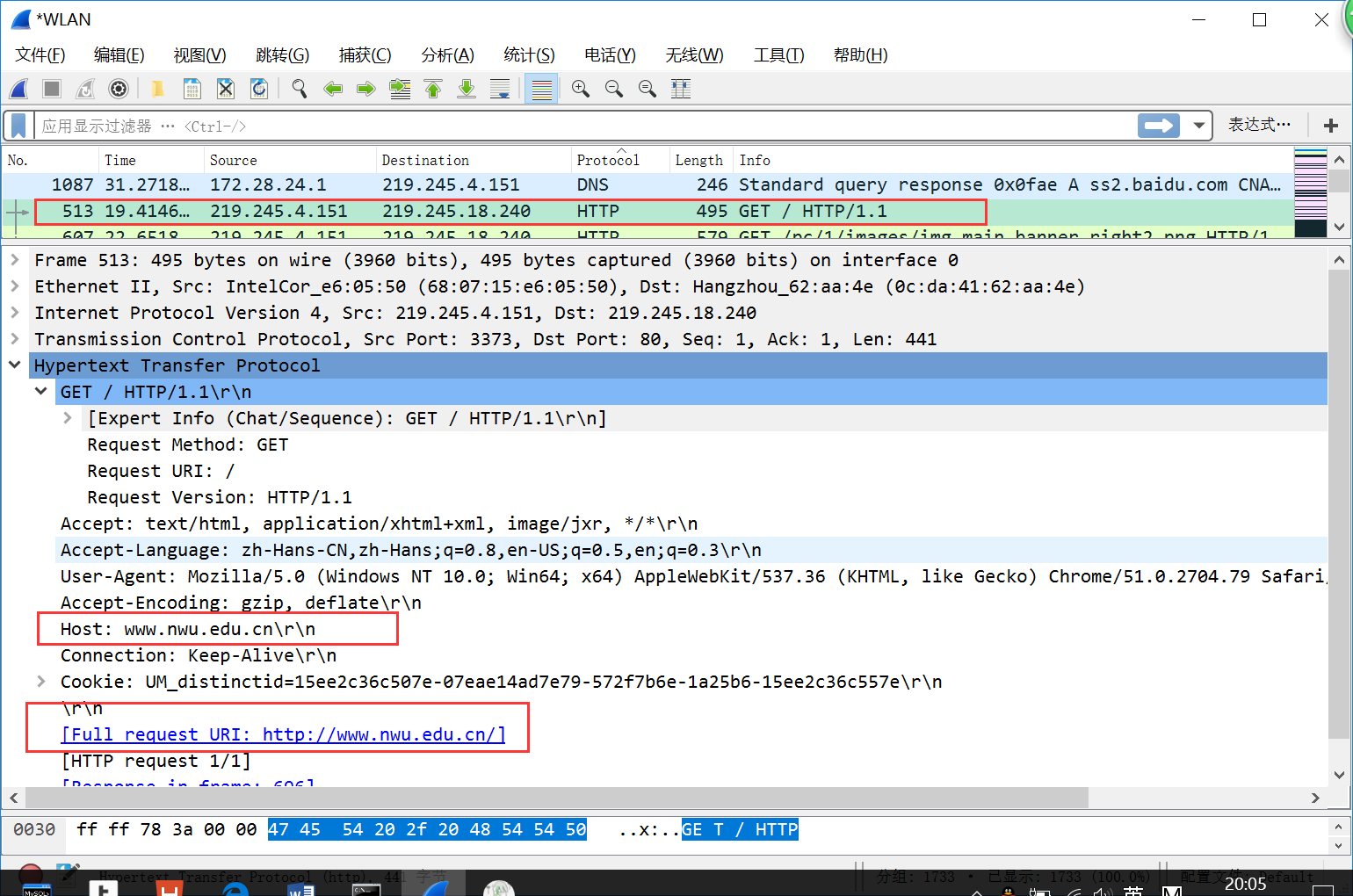
相比之下，百度的主页相对比较简单（因为它需要经常被访问）。而校园网等网页还会涉及到很多图像、样式表及其他对象。



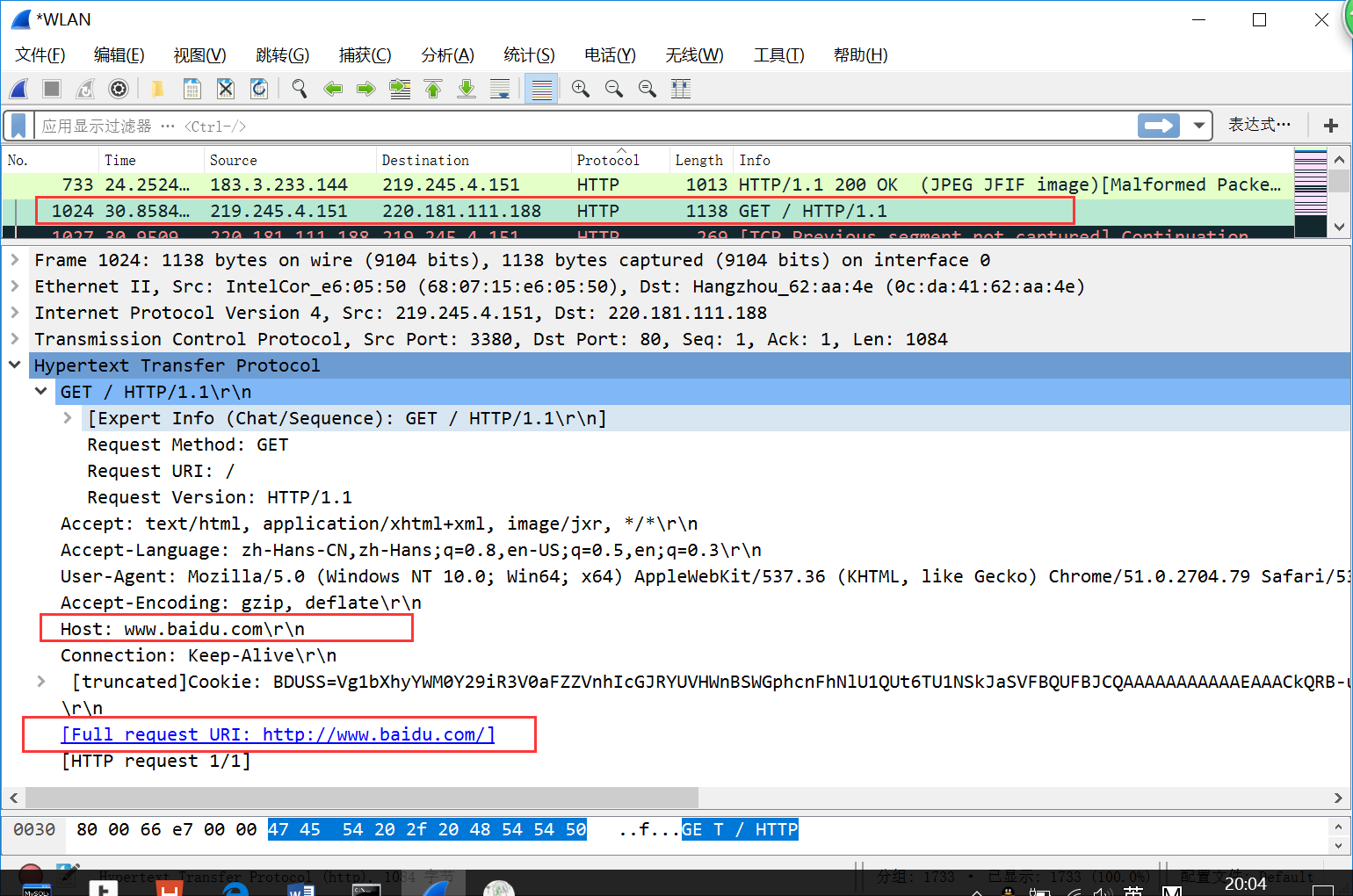
【明文数据】

（4）明文数据

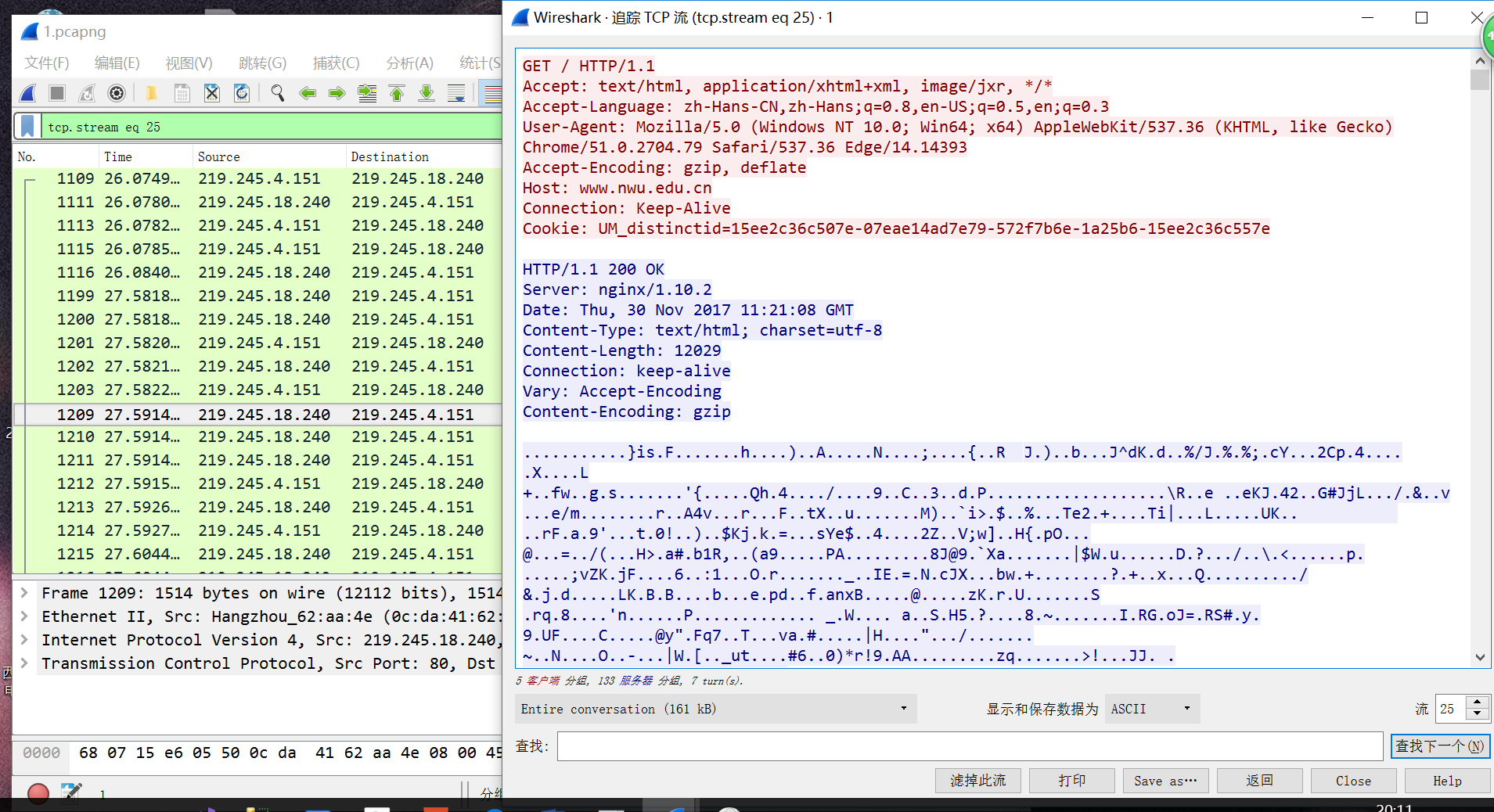
我们的Web浏览器和http:// www.nwu.edu.cn的Web服务器之间的数据交换也是同样的模式。在分组513和1024中，我们的Web浏览器发送的请求与它发送到www.baidu.com的请求相同（“GET /HTTP/1.1”），除了Host首部被设置为www.nwu.edu.cn，其他所有的首部都一样。响应中仍然还有一些有趣的不同之处。



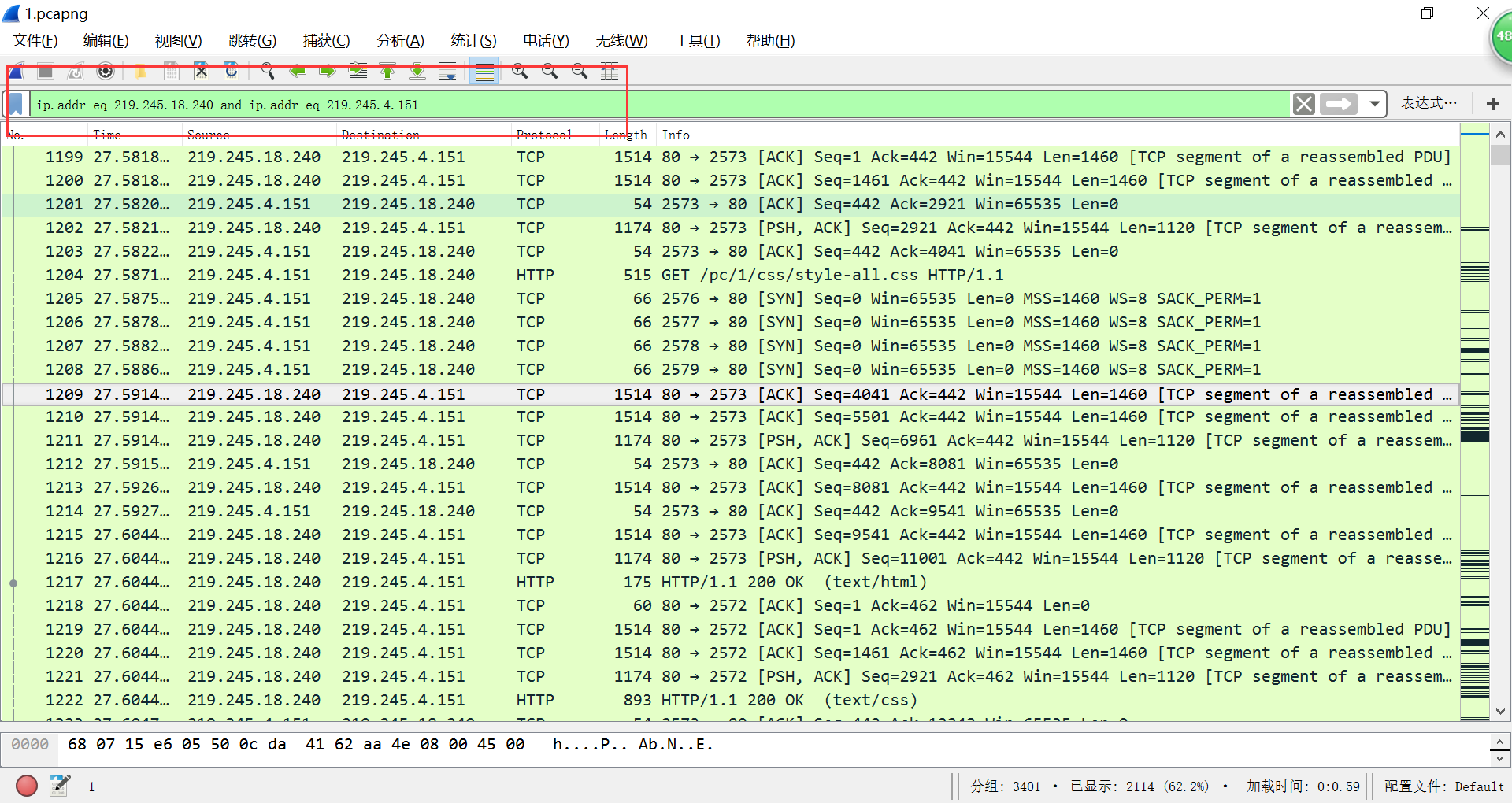
【对学校官网的请求】



【对百度页面的请求】



【HTTP数据流跟踪】



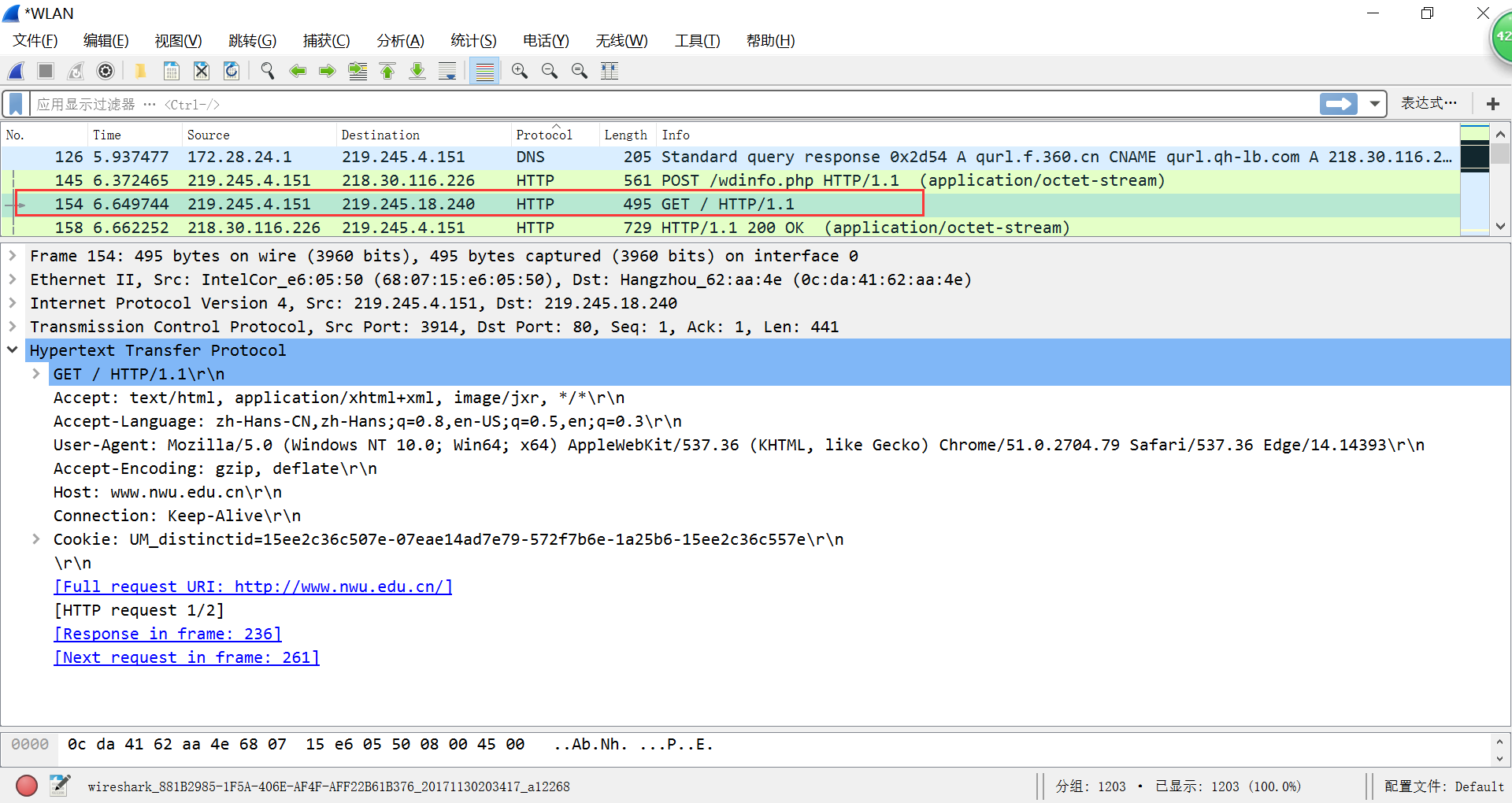
【ip.addr eq 219.245.18.240 and ip.addr eq 219.245.4.151】

4．作业

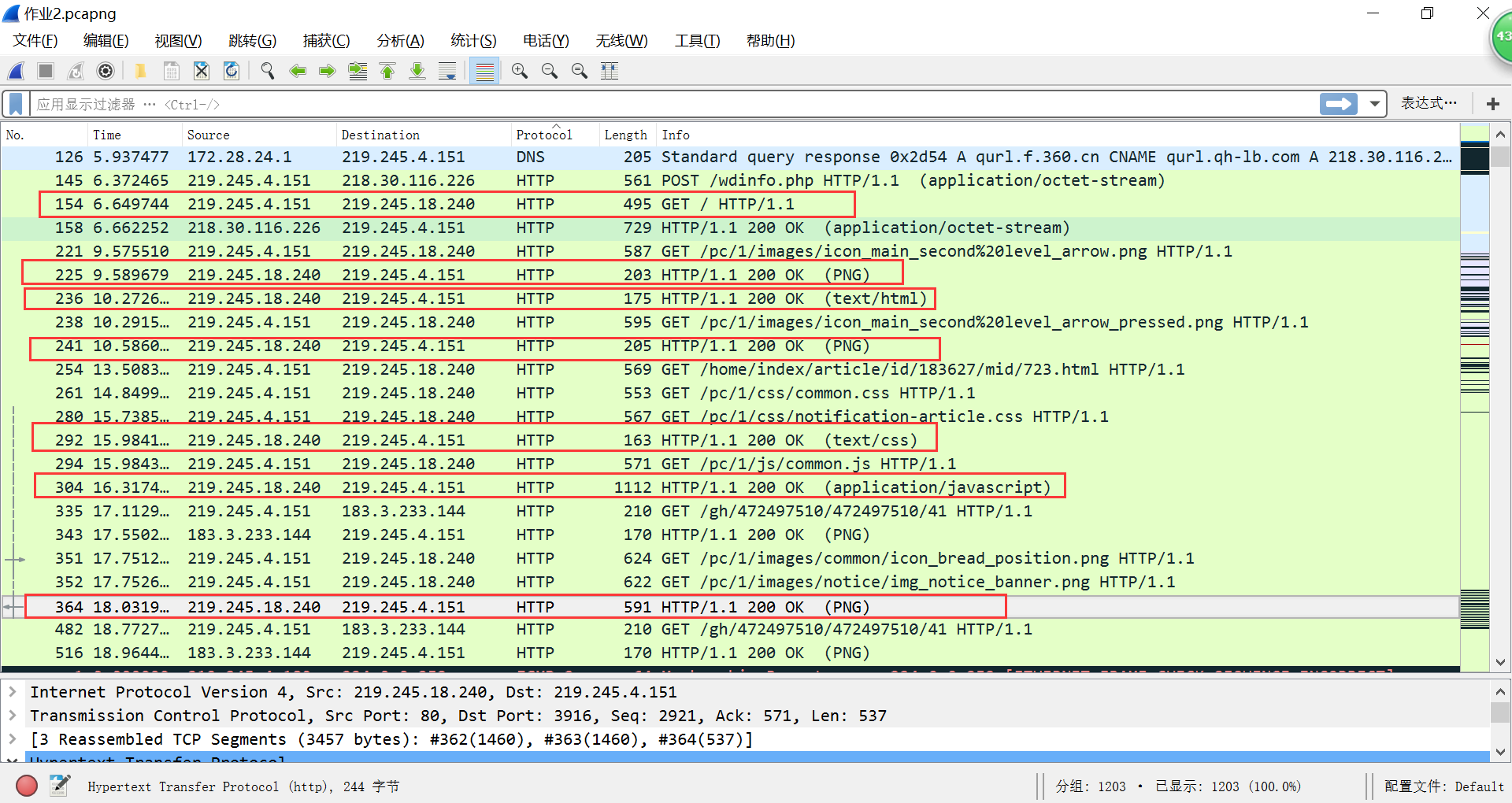
（1）跟踪自己主机的Web浏览器和www.baidu.com Web服务器之间的TCP数据流，分离出浏览器发出的请求，并将文本复制下来。



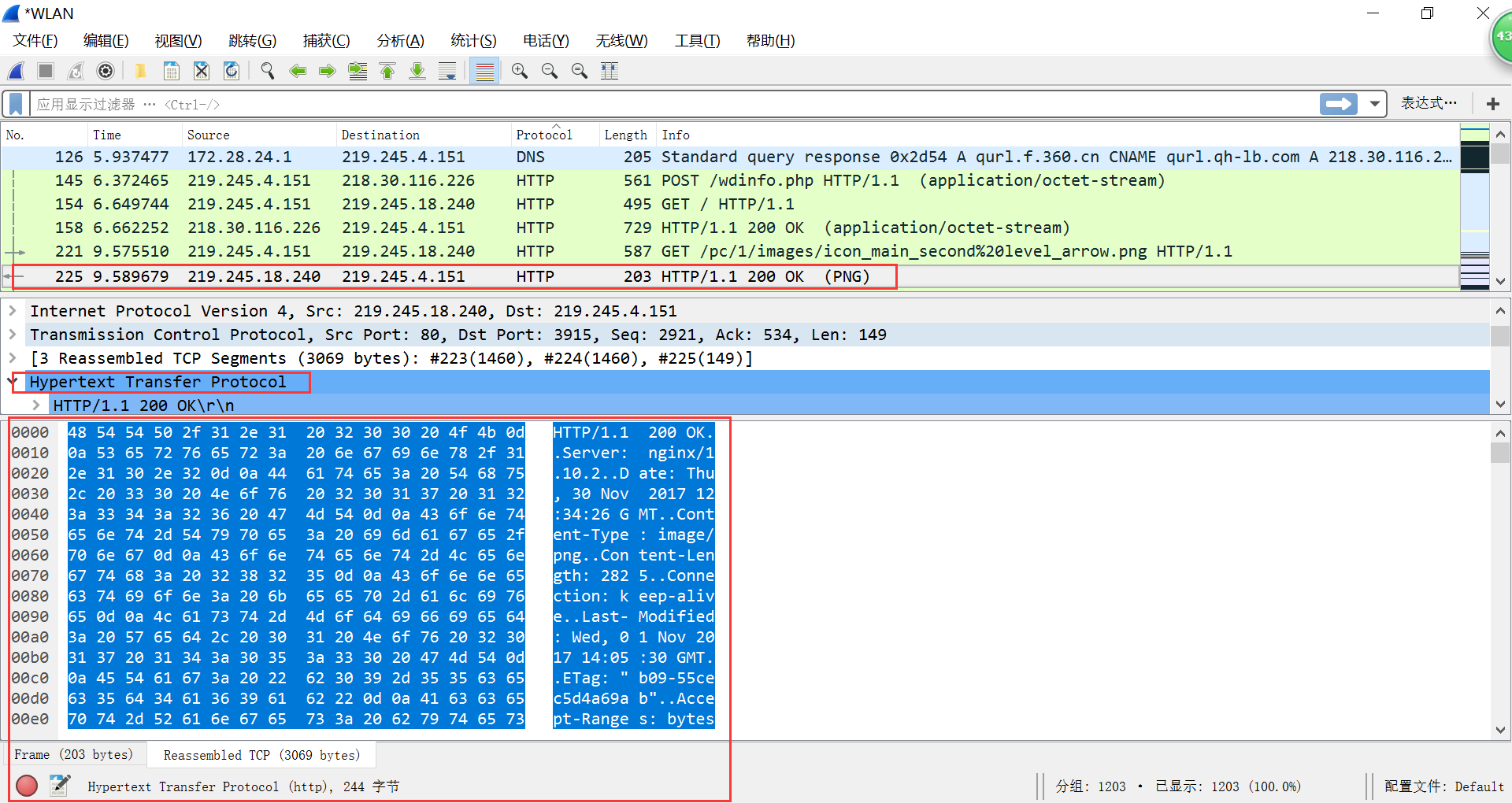
（2）跟踪自己主机的Web浏览器和www.nwu.edu.cn Web服务器之间的TCP数据流，写出第一个GET请求和www.nwu.edu.cn Web服务器响应中各个首部对应的值。



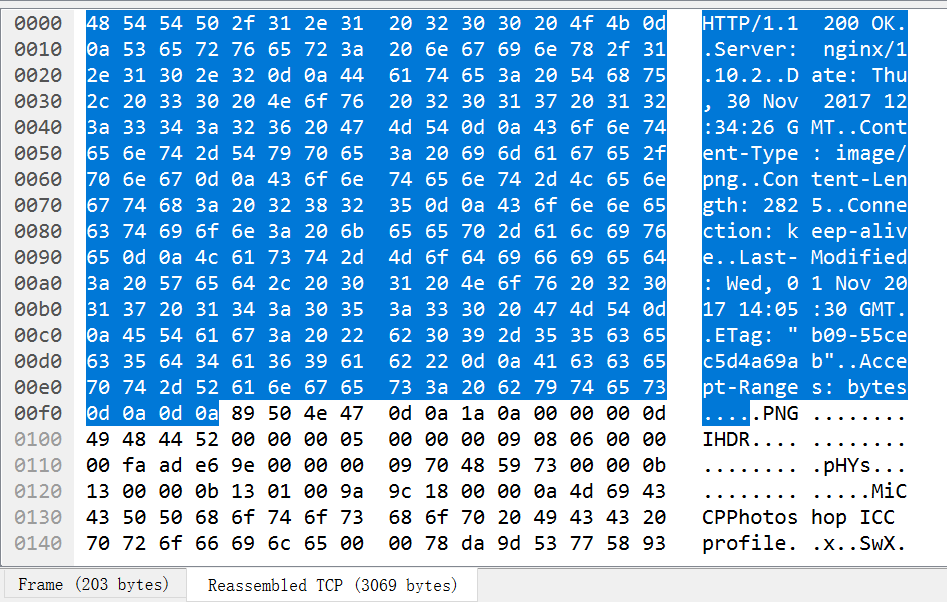
【第一个GET请求】



【web服务器的所有响应】



【web服务器第一次响应】



【Web服务器第一次响应中首部对应的值】

**四、实验心得**

经过该次实验，我对http协议的理解有了进一步的加深。尤其是对协议的发起和回应的包进行了详细的分析，因此对其细节方面有了自己的认识。

本次实验的过程中我学会了怎么使用Ethereal进行基本的数据的抓取，以及对数据的基本的一个分析，已经基本掌握了对数据包结构的了解。同时也更好的理解了数据报的内容以及Ethereal的使用方法。