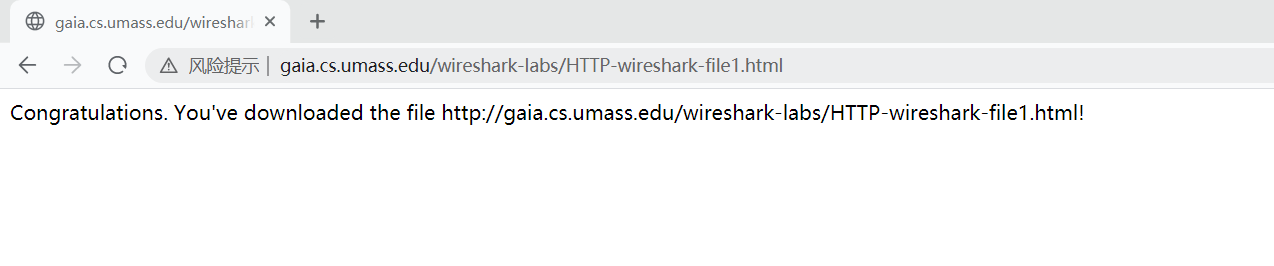
**计算机网络HTTP实验报告**

**姓名：陈鹤影 学号：PB21061287 日期：2022.9.18**

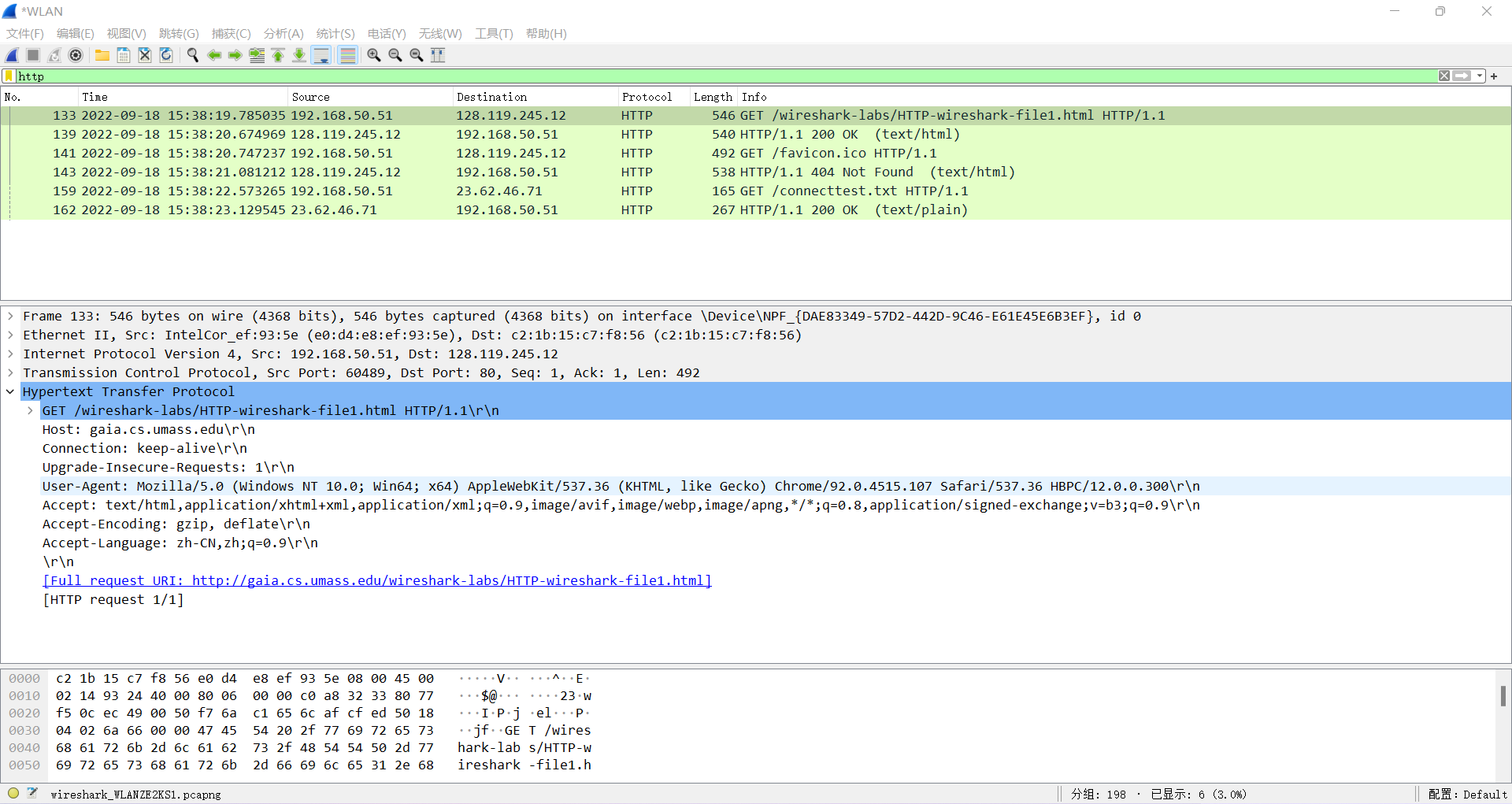
1. **实验目的：**
2. **巩固Wireshark的使用方法**
3. **通过嗅探实验了解条件GET/response交互，分析HTTP报文格式，检索大型HTML文件，检索嵌入网页对象以及了解HTTP身份验证和安全性。**
4. **实验流程及问题回答：**

**Step 1：The Basic HTTP GET/response interaction**

打开浏览器和Wireshark，并选择http过滤模式。等待一定时间，开启抓包。登入<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html> 网站，当HTML文件显示成功后，停止抓包。显示如下：



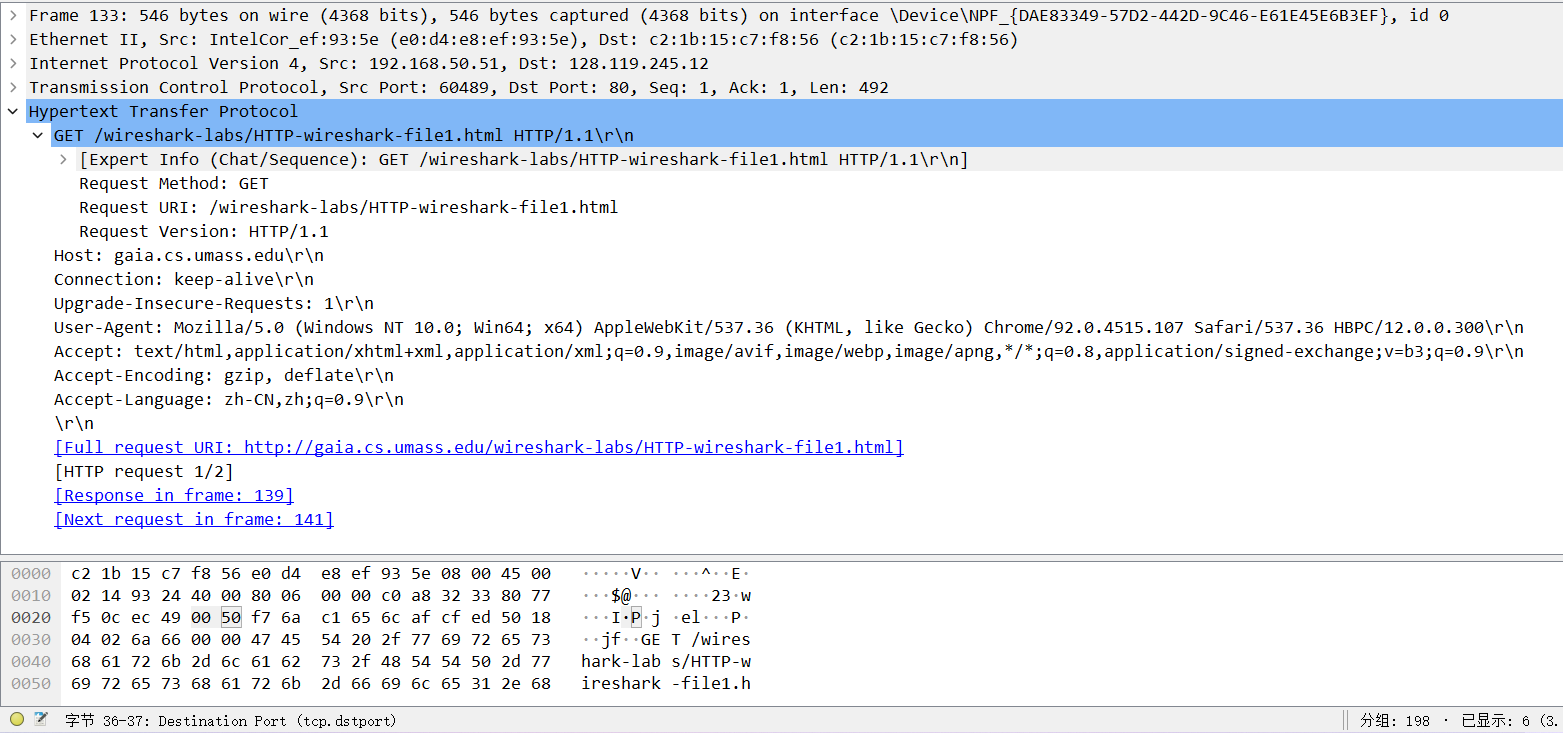
**（图1-1 HTML文件显示界面）**



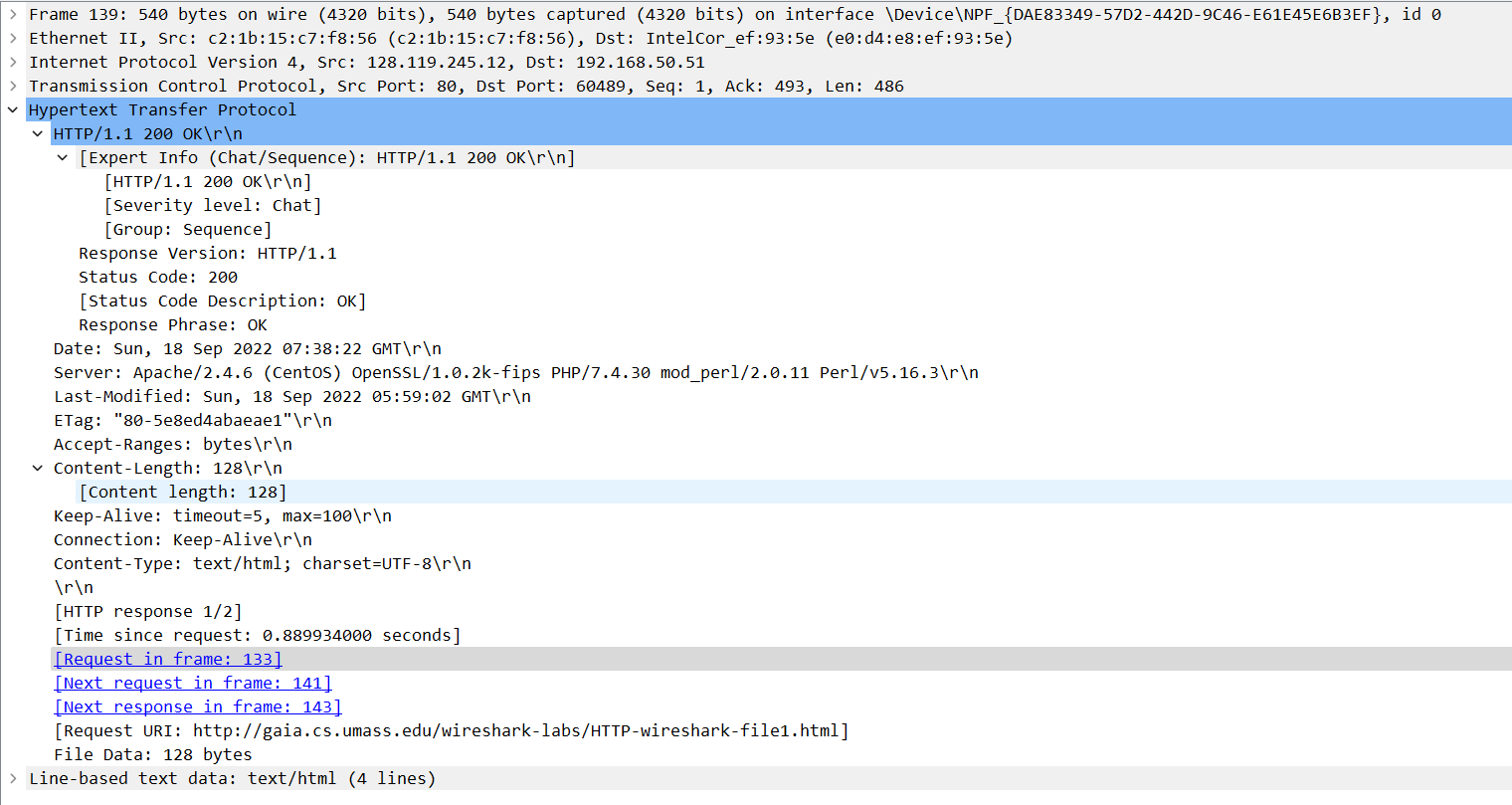
**（图1-2 Wireshark抓包显示界面）**

其中成功观察到浏览器向服务器请求对应HTML文件，服务器返回OK报文。

报文内容如下：



**（图1-3 GET请求报文）**



**（图1-4 OK响应报文）**

**Q&A：(具体位置见图1-5、1-6)**

**1. Is your** **browser running HTTP version 1.0 or 1.1? What version of HTTP is the**

**server running?**

**Ans：**browserHTTP version ：1.1 。

sever HTTP version ：1.1 。

**2. What languages (if any) does your browser indicate that it can accept to the**

**server?**

**Ans：**可以接受简体中文，即zh-CN,zh(华语区-中国大陆)。

**3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?**

**Ans：**my IP: 192.168.50.51。

sever IP:128.119.245.12。

**4. What is the status code returned from the server to your browser?**

**Ans：**200（代表OK）。

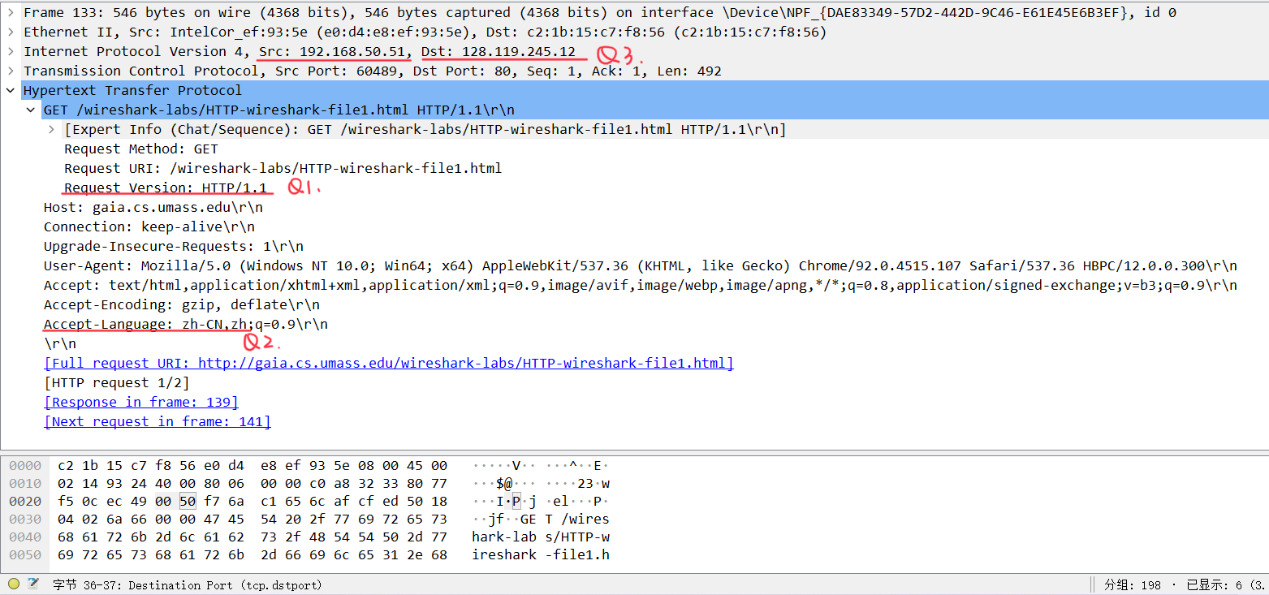
**5. When was the HTML file that you are retrieving last modified at the server? Ans：**Sun, 18 Sep 2022 05:59:02 GMT。

**6. How many bytes of content are being returned to your browser?**

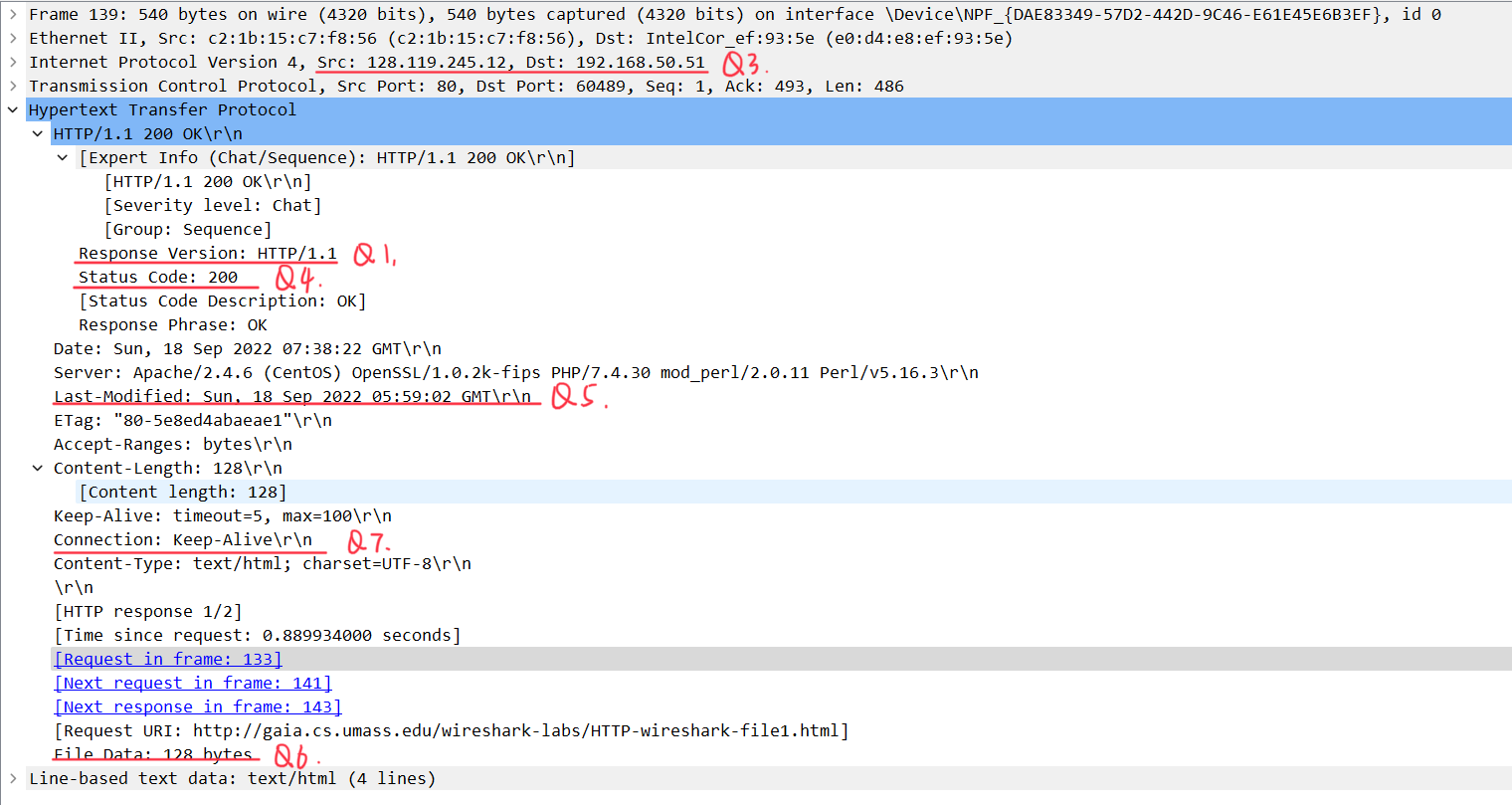
**Ans：**128 bytes。

**7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.**

**Ans：**例如Connection字段。



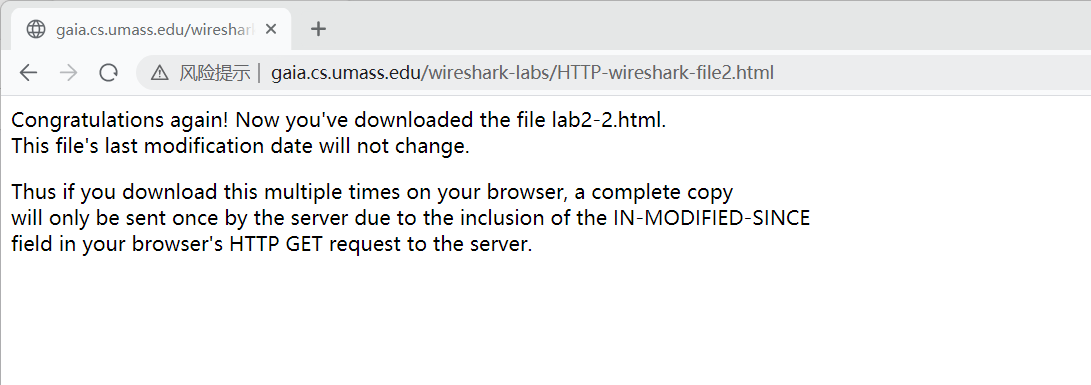
**（图1-5 GET请求报文标记）**



**（图1-6 OK响应报文标记）**

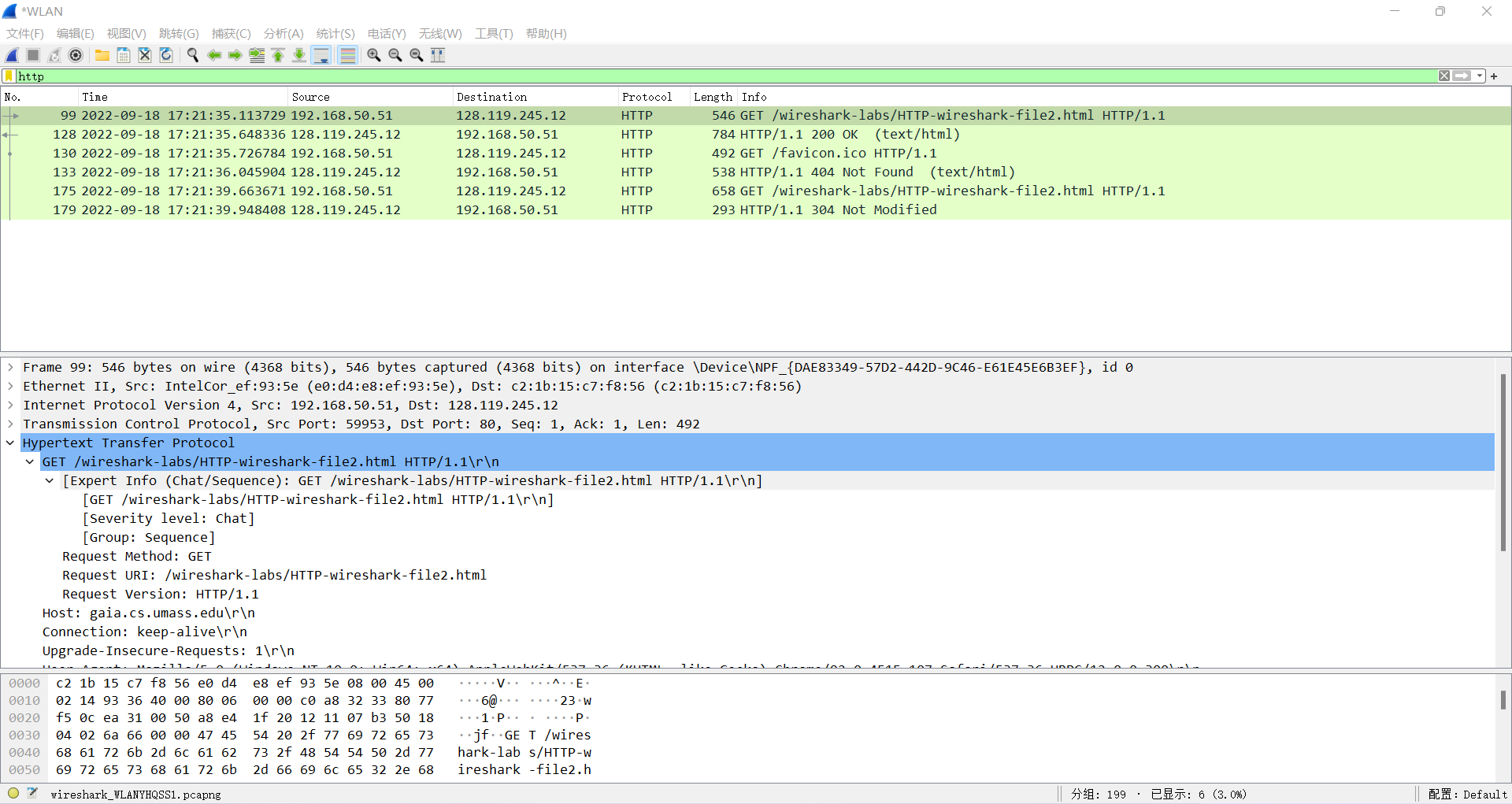
**Step 2：The HTTP CONDITIONAL GET/response interaction**

清除浏览器缓存（这里选用华为浏览器），操作步骤如下：选择历史记录->清除缓存。打开Wireshark抓包并登入以下URL：<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html> 。观察到如下界面：



**（图1-7登入界面）**

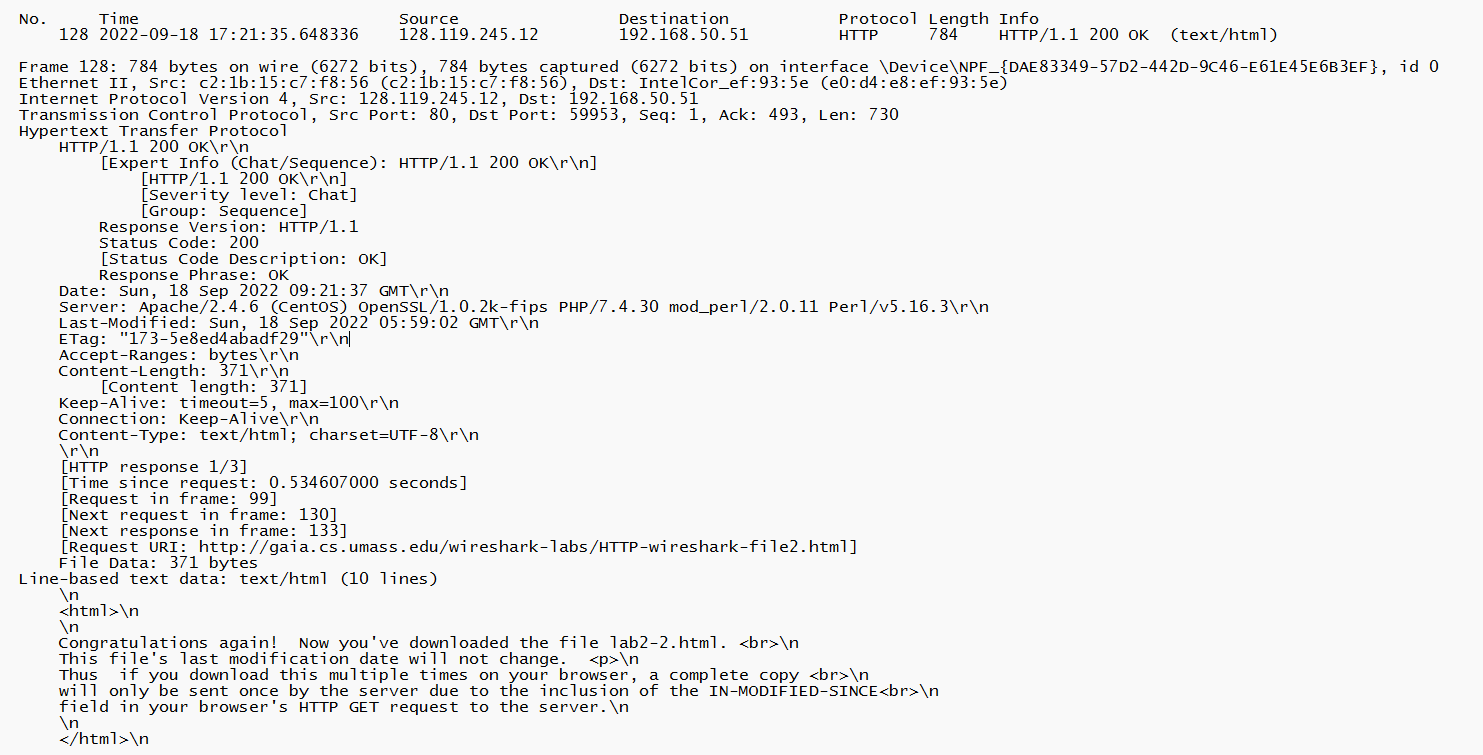
在短时间内再次登入上述URL（或直接刷新）（实验过程中选择http过滤），Wireshark抓包结果如图1-8。

**（图1-8 Wireshark抓包结果）**

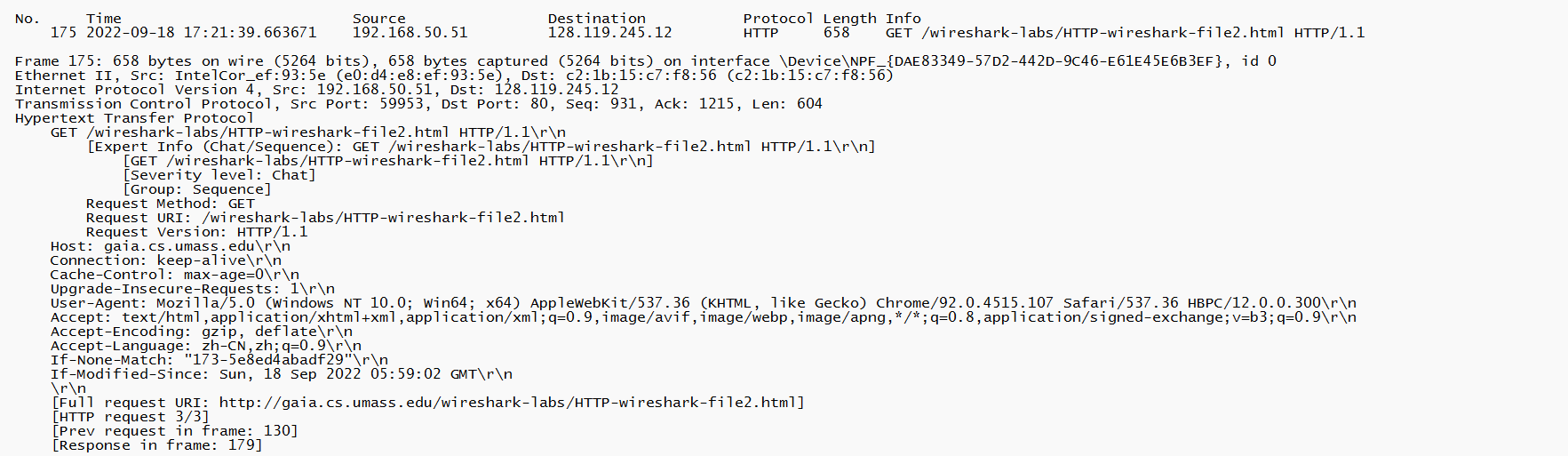
报文打印如下：



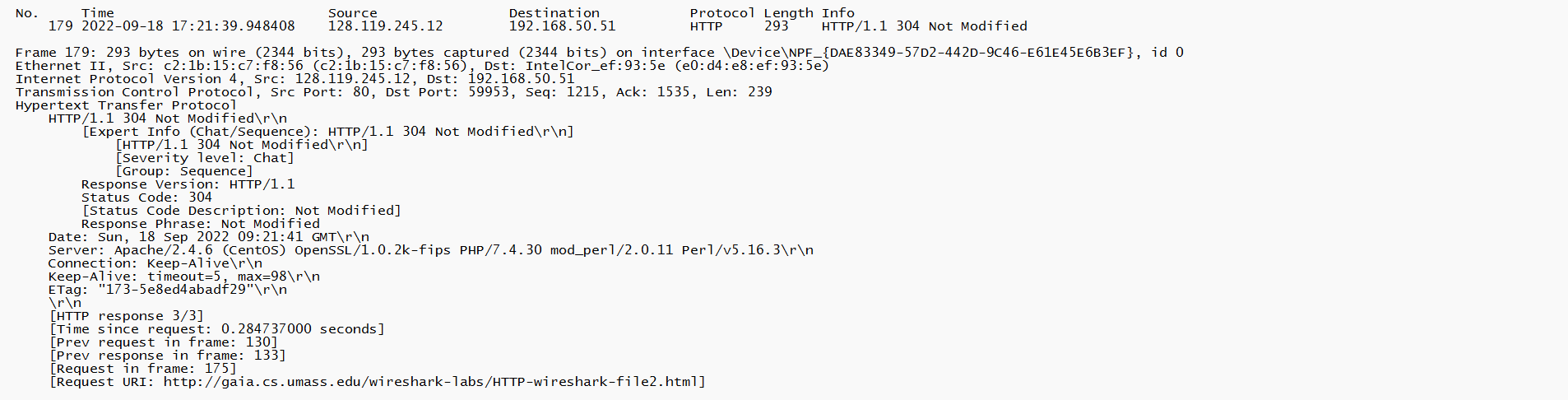
**（图1-9 第一次GET报文）**



**（图1-10 第一次OK报文）**



**（图1-11 第二次GET报文）**



**（图1-12 第二次OK报文）**

**Q&A：**

**8. Inspect the contents of the first HTTP GET request from your browser to the**

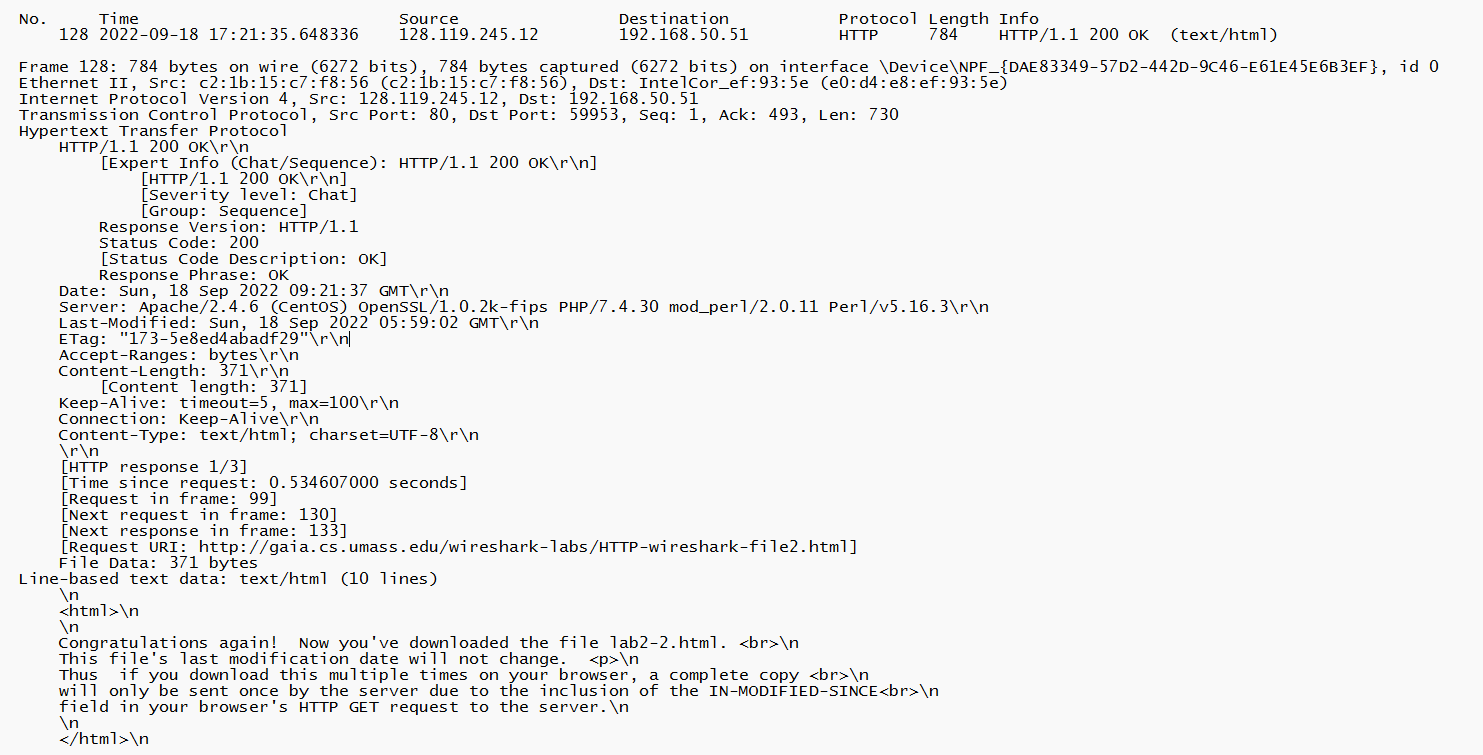
**server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?**

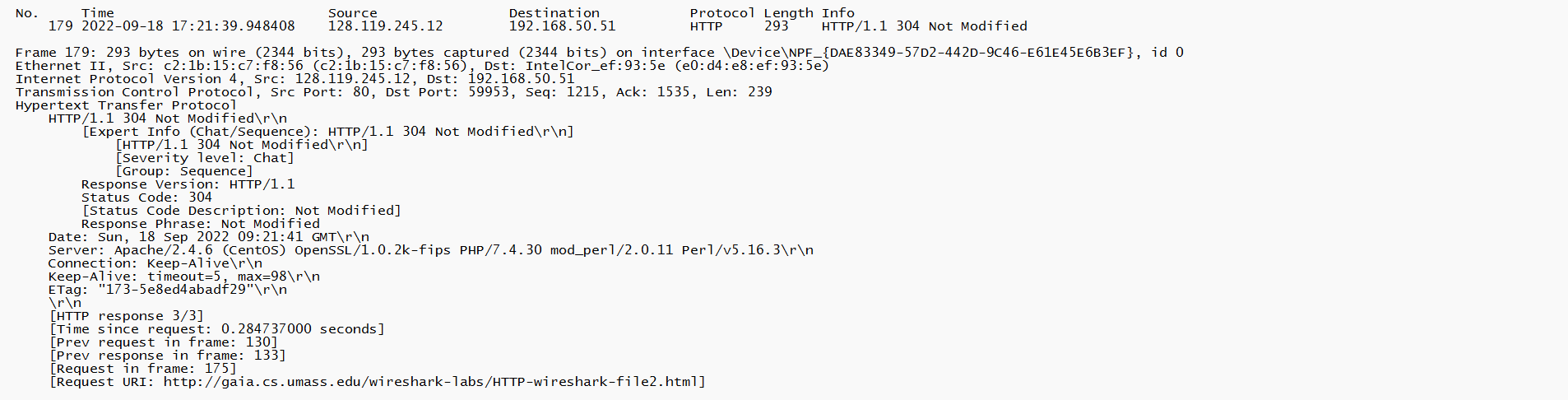
**Ans：**没有。



**9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?**

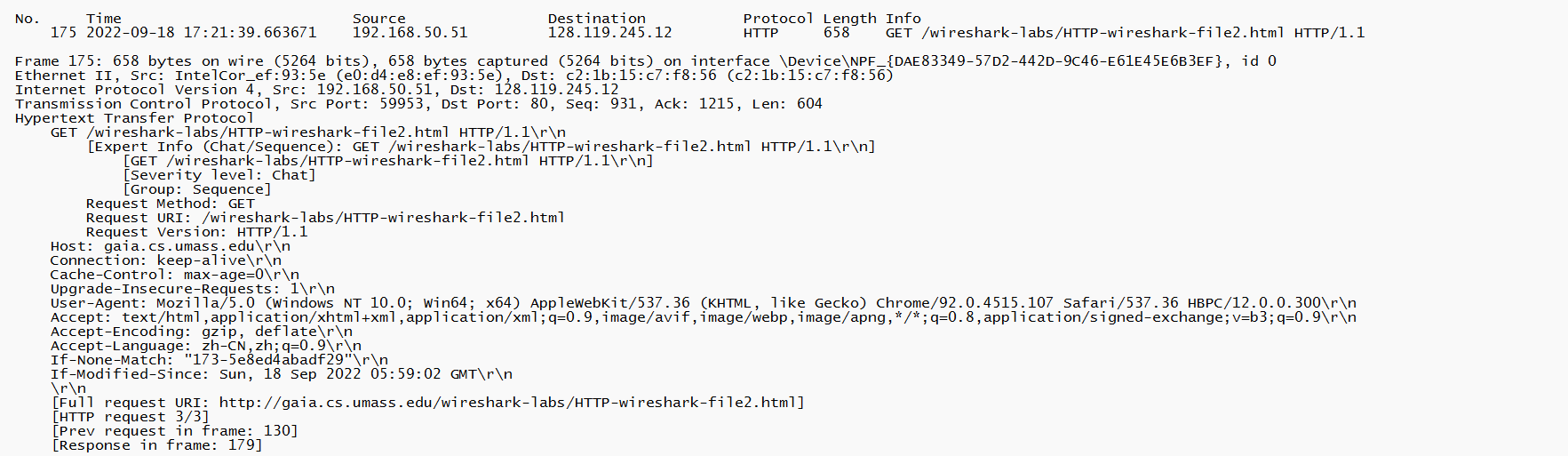
**Ans：**服务器在第一次响应时明确返回了文件内容，在第二次响应时没有。从是否有Content Length段和Line-based text data段可知。





**10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?**

**Ans：**Yes. Following information is ‘Sun， 18 Sep 2022 05：59：02 GMT\r\n’这是第一次响应报文返回的最后修改时间。

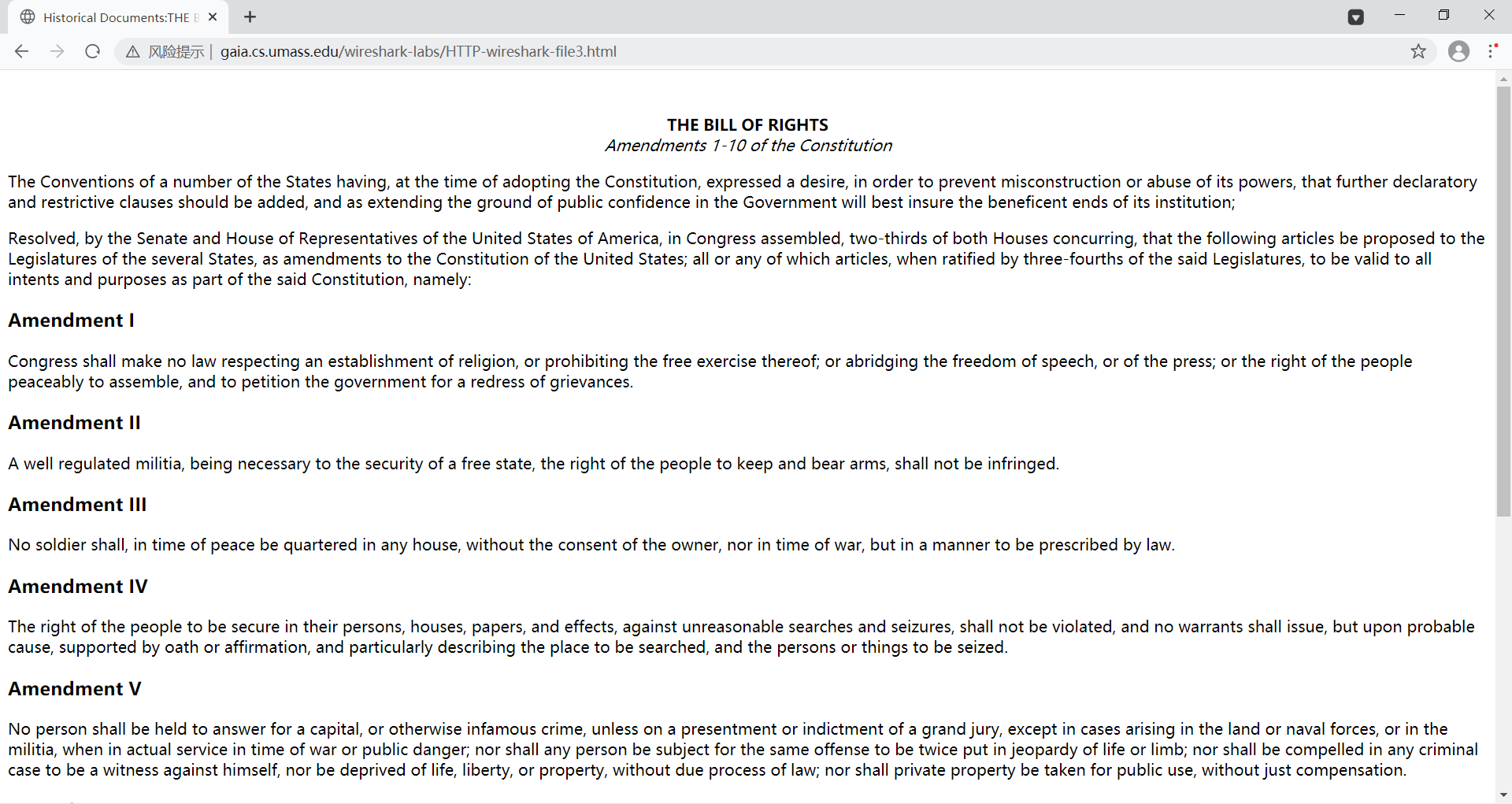


**11. What is the** **HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.**

**Ans：**HTTP status code and phrase：304 Not Modified。服务器没有明确地返回文件内容。因为报文显示从上一次获取文件，文件内容并未修改，由于缓存机制，浏览器可以直接从缓存区域读取文件内容，服务器不用再次发送文件。

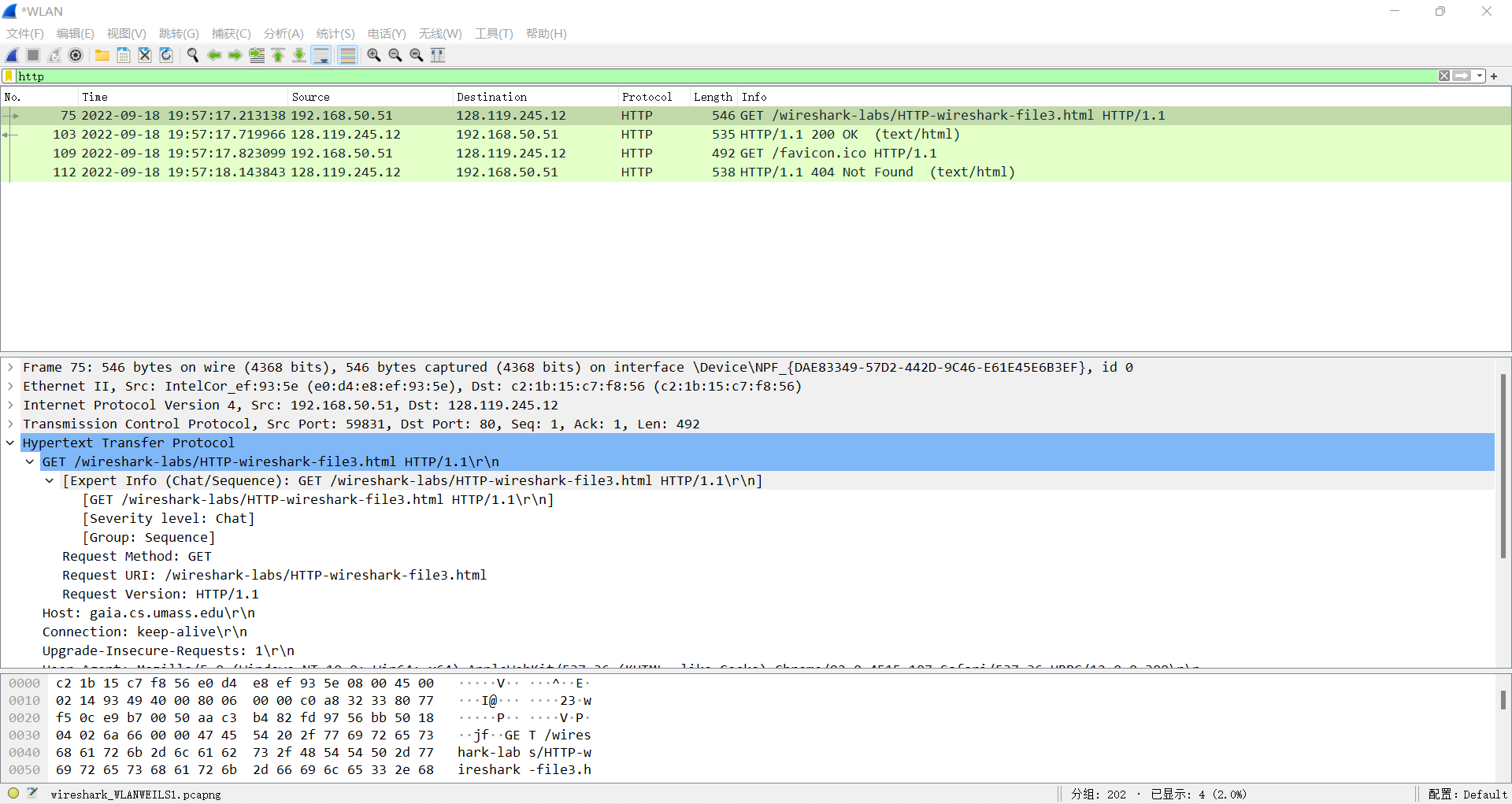
**Step 3：Retrieving Long Documents**

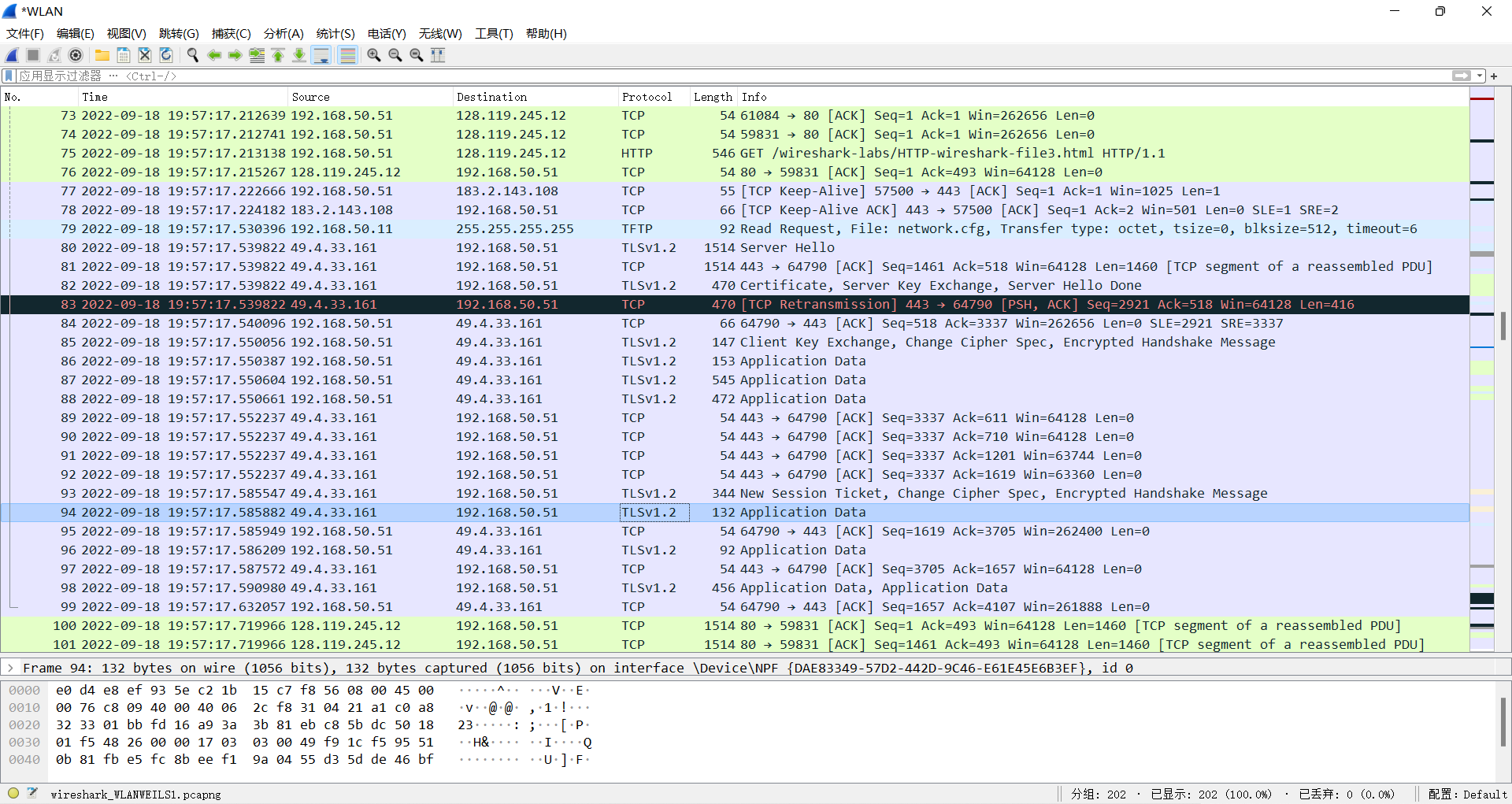
打开浏览器并清除缓存内容，打开Wireshark抓包并登入以下URL：<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html> 。观察到如下界面：



**（图1-13 浏览器登入界面显示）**

实验过程中选择http过滤，Wireshark抓包结果如图1-14（注意HTTP协议中不包含“Continuation”报文）。



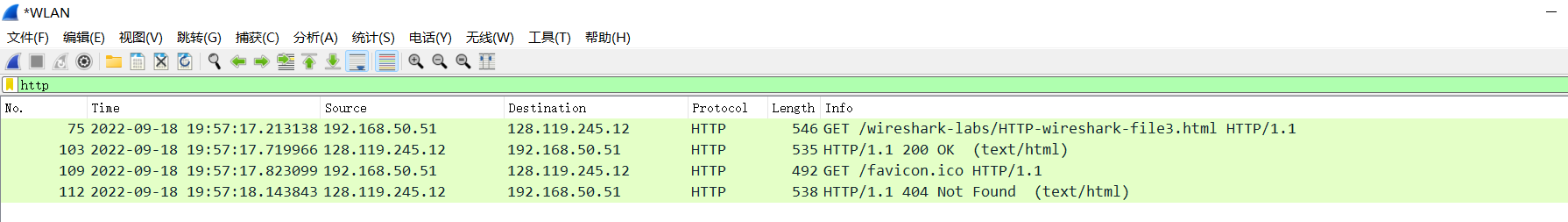


**（图1-14 Wireshark抓包结果）**

**Q&A：**

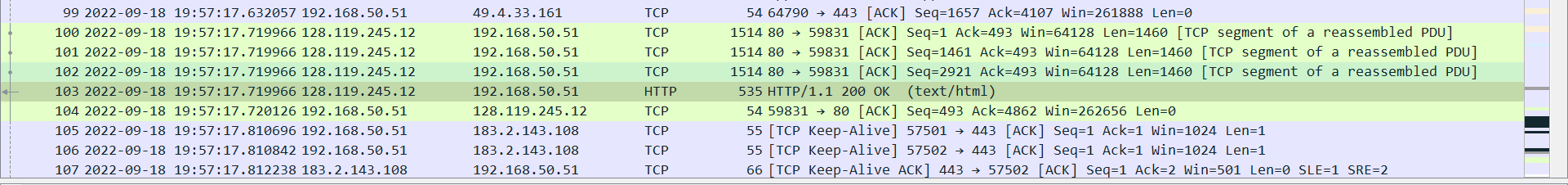
**12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?**

**Ans：**一共发送了2个GET request messages(在不忽略favicon请求的情况下，若忽略，则为1个message)；The first one（number 75）.



**13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?**

**Ans：**103;(200 OK);

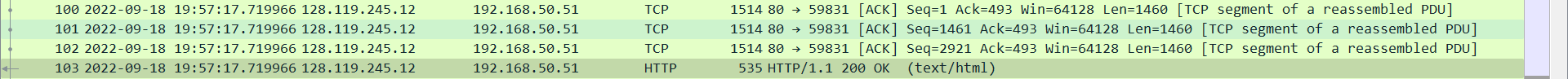


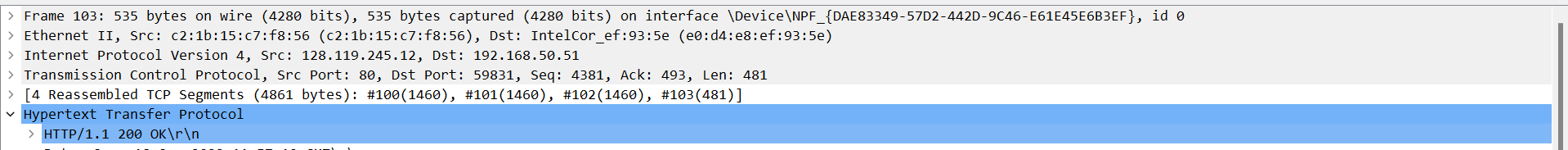
**14. What is the status code and phrase in the response?**

**Ans：**200 OK

**15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?**

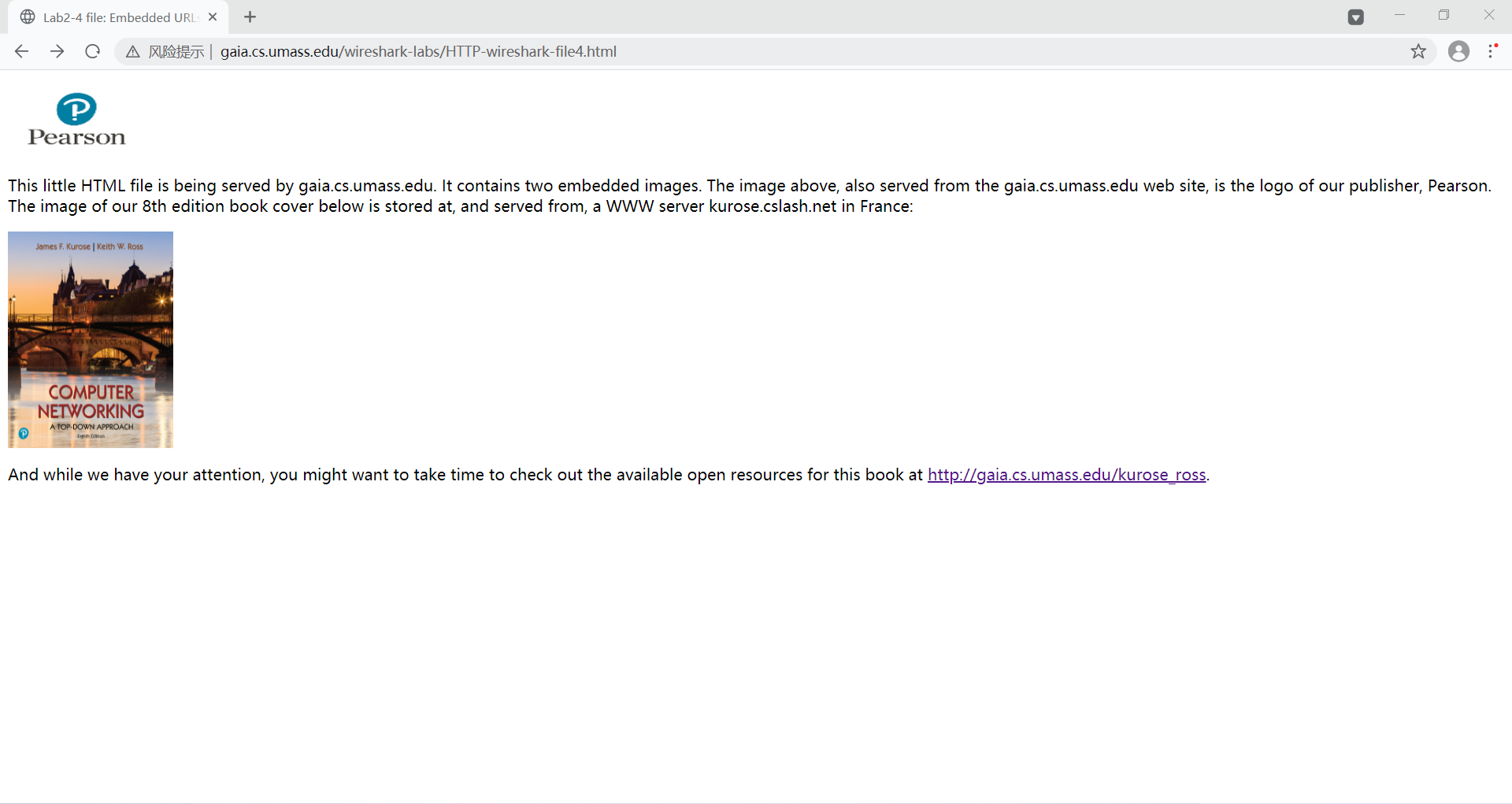
**Ans：**4（分别为#100、#101、#102、#103）.





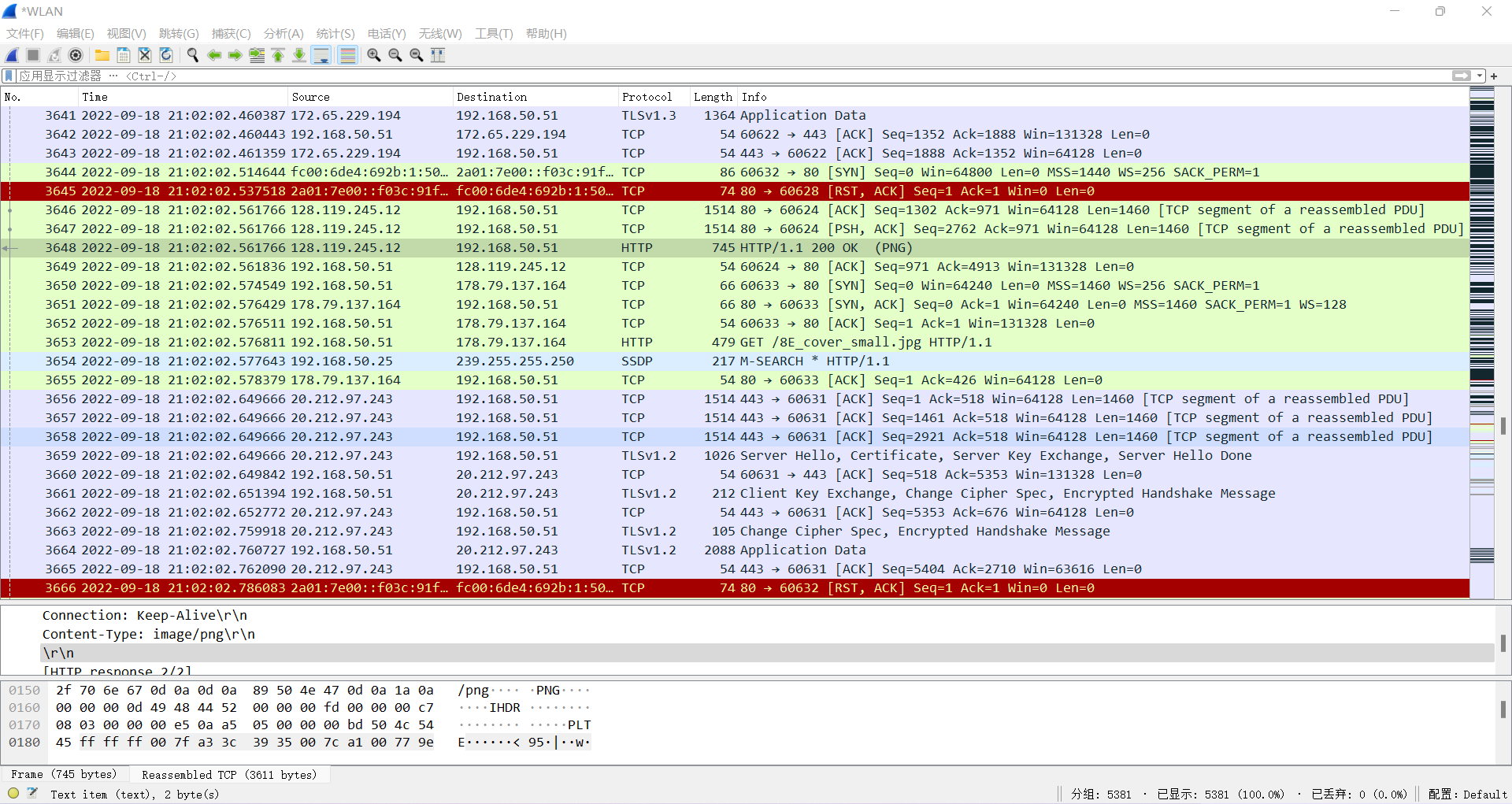
**Step 4：HTML Documents with Embedded Objects**

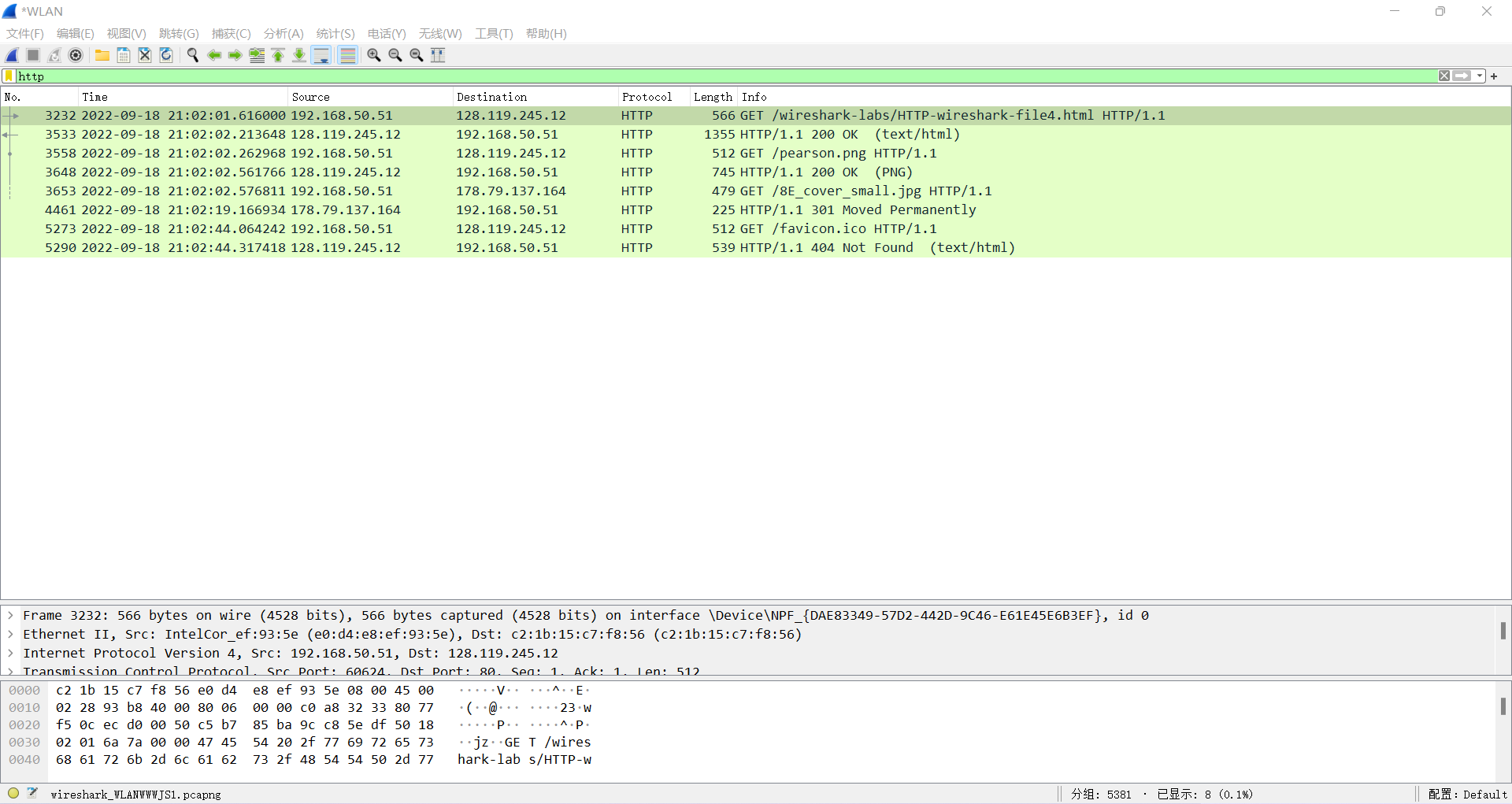
打开浏览器并清除缓存内容，打开Wireshark抓包并登入以下URL：<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html> 。观察到如下界面：



**（图1-15 浏览器登入界面显示）**

Wireshark抓包结果如图1-16。





**（图1-16 Wireshark抓包结果）**

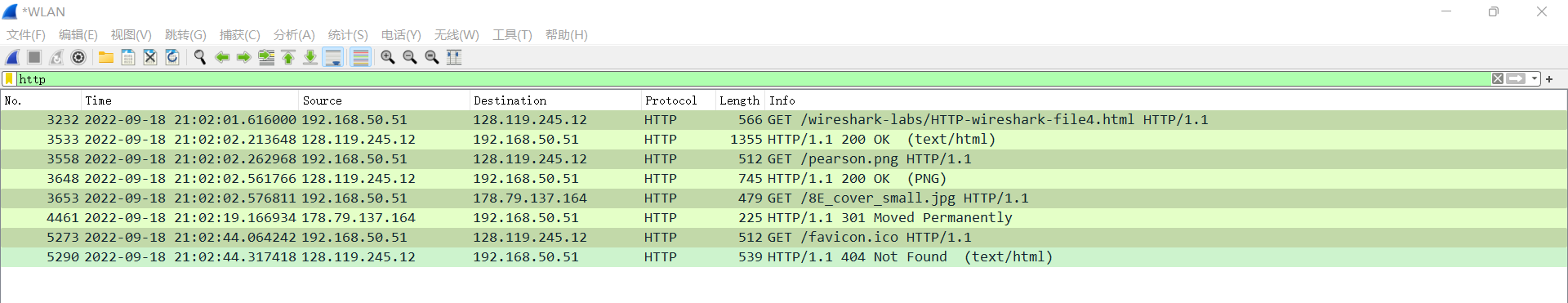
**Q&A：**

**16. How many HTTP GET request messages did your browser send? To which**

**Internet addresses were these GET requests sent?**

**Ans：**My browser sent 4 HTTP GET request messages in total(在不忽略favicon请求的情况下，若忽略，则为3个messages).

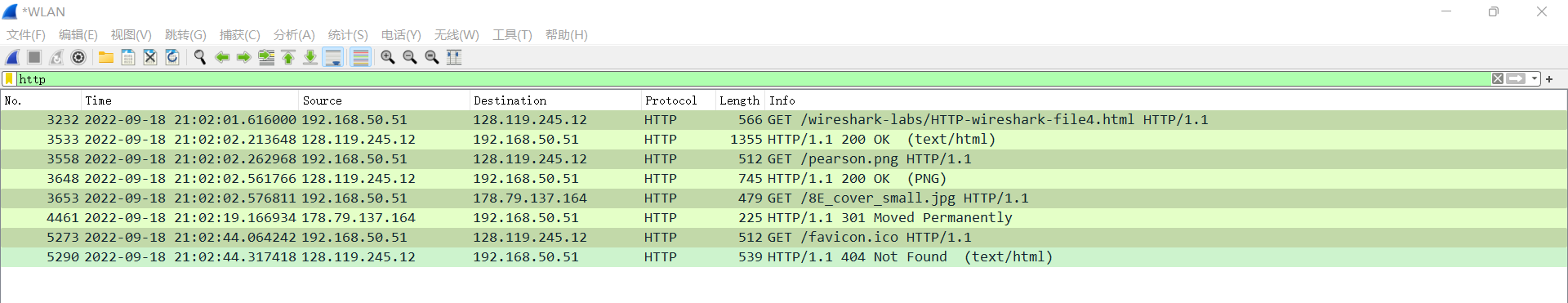
Adresses:前两个（以及最后一个）的地址均为128.119.245.12。第三个的地址为178.79.137.164。

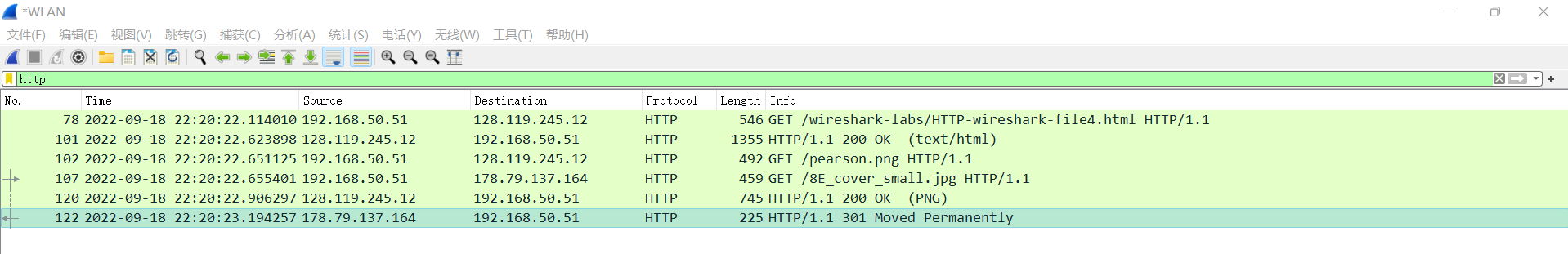


**17. Can you tell whether your browser downloaded the two images serially, or**

**whether they were downloaded from the two web sites in parallel? Explain.**

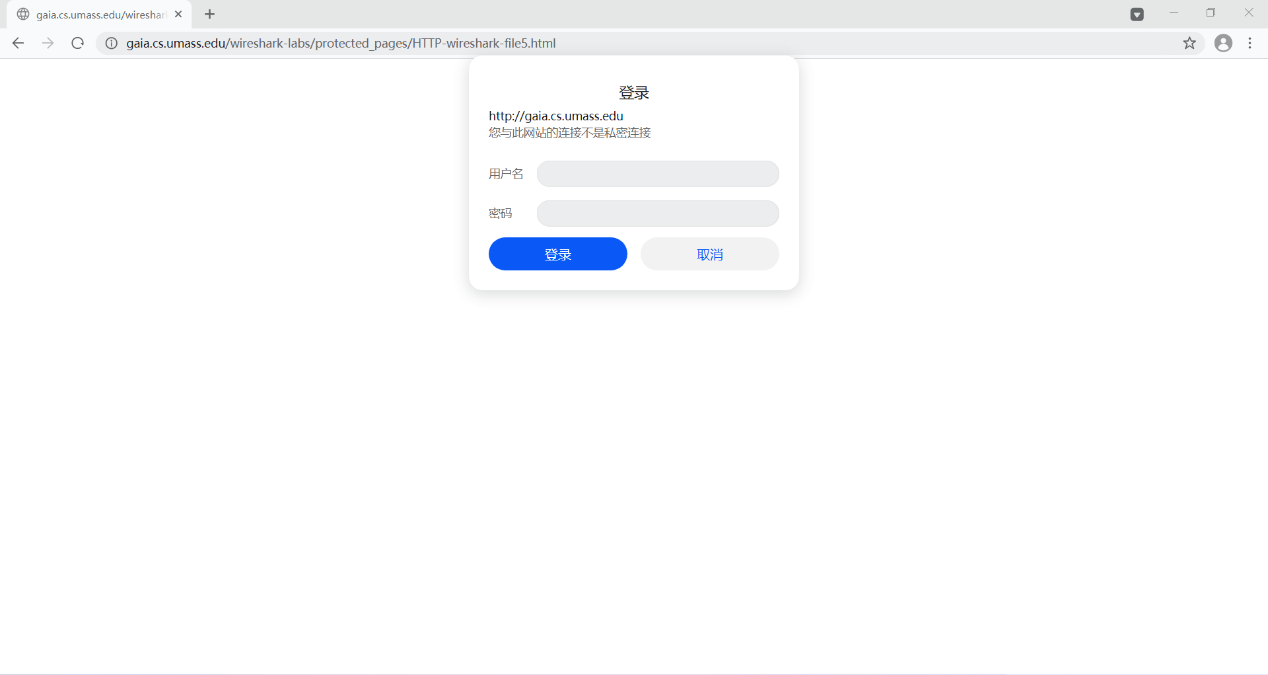
**Ans：**并行的。在第一次实验（Microsoft Edge）结果中实验结果显示两个包并不是同时发送的（实验结果如下）。第一个GET包在21：02：02.262968时发送，在21：02：02.561766时收到回复。而第二个包在21：02：02.576811时才发送。为保证准确性，以下换用另一个浏览器（华为浏览器）再次试验。实验结果如下：两个包发送时间间隔极小，可以证明二者是并行发送的。分析得出，两次实验结果存在差异的可能原因是不同浏览器的处理方式不同以及两张图片请求的服务器不同，请求时间等具有一定差异。





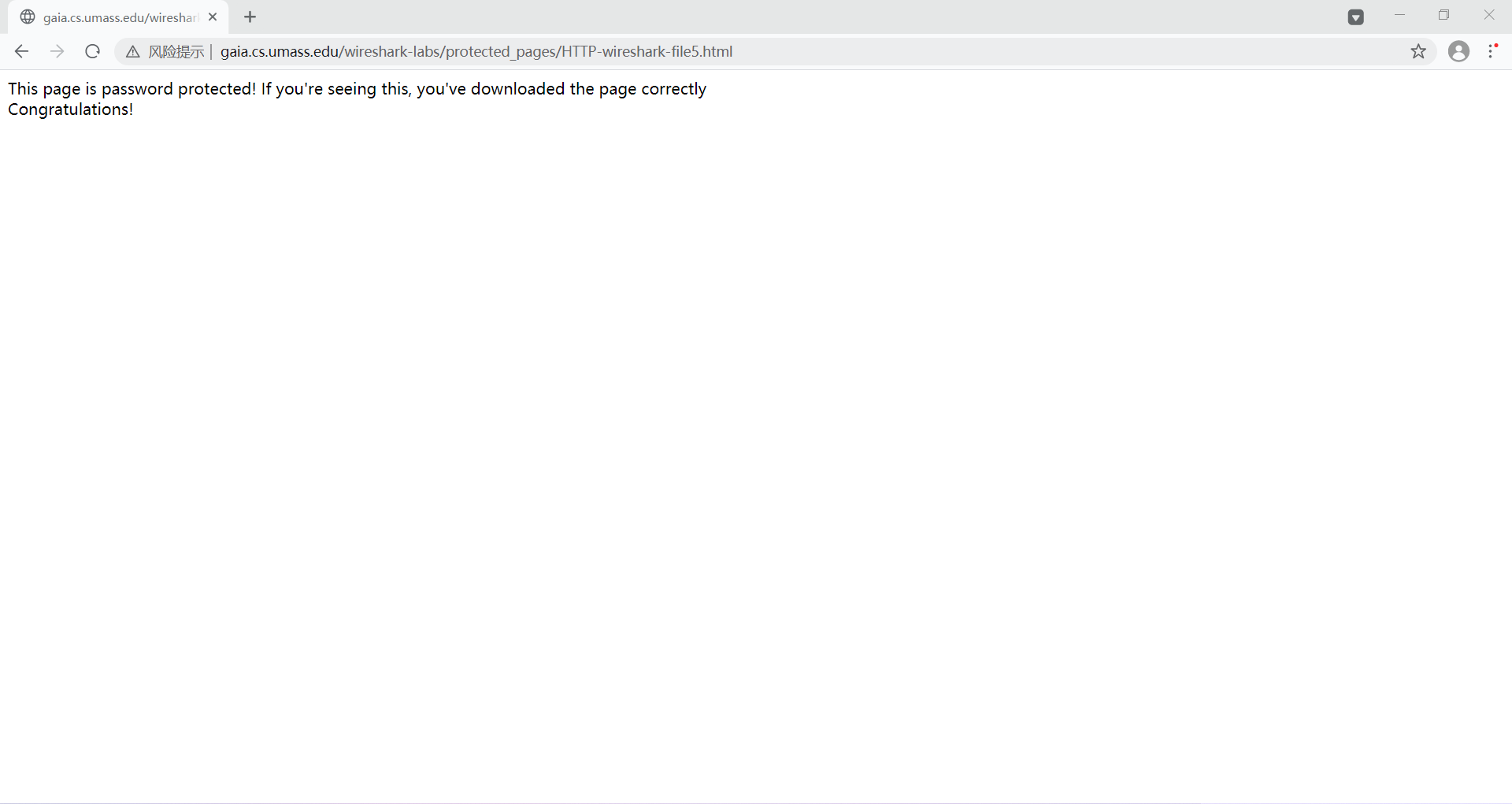
**Step 5：HTTP Authentication**

打开浏览器并清除缓存内容，重启浏览器。打开Wireshark抓包并登入以下URL：<http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html> 。观察到如下界面：



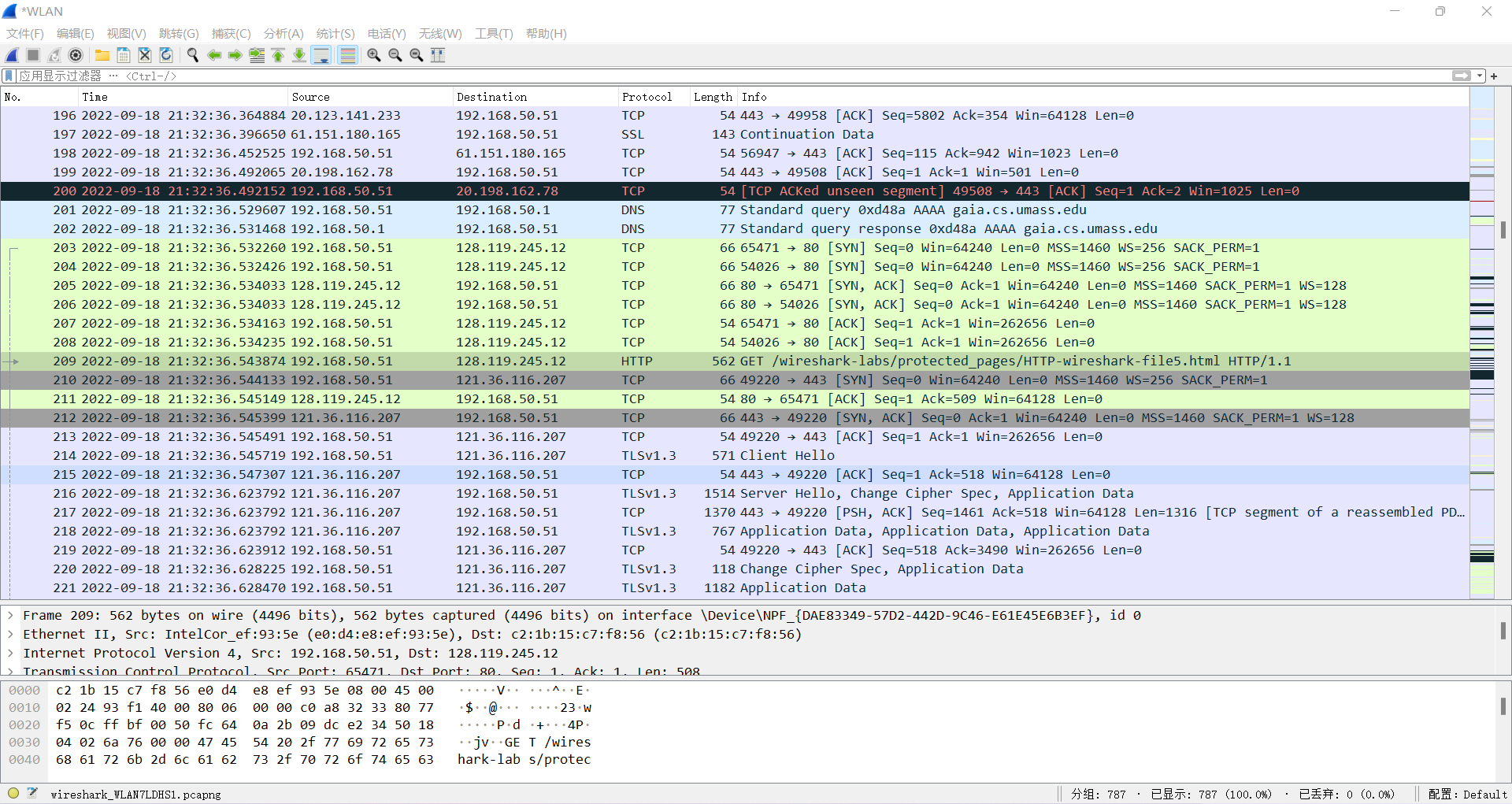
**（图1-17 浏览器登入界面显示）**

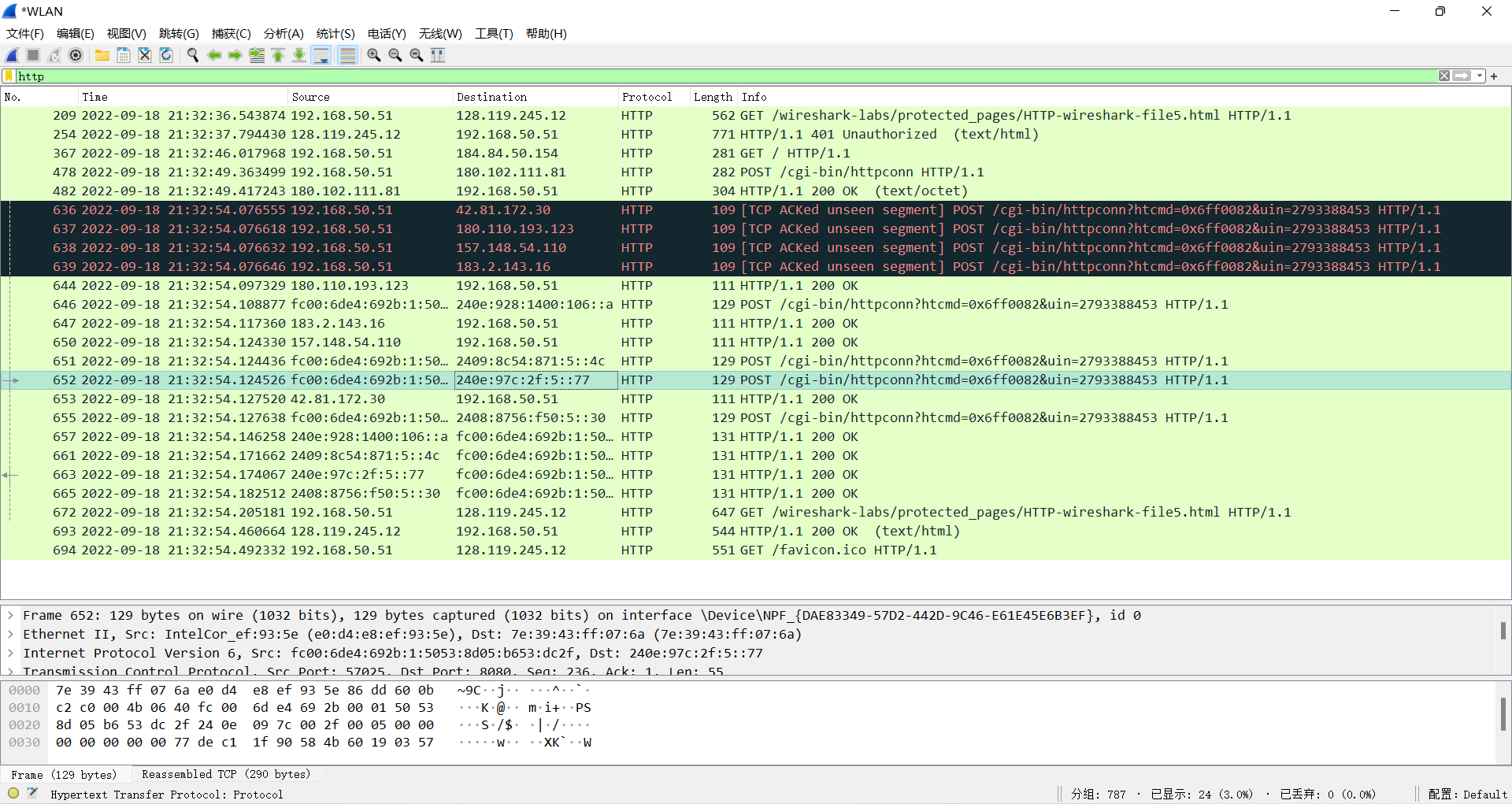
输入用户名和密码，观察到如下界面：



**（图1-18 输入密码后成功登入显示）**

Wireshark抓包显示如下：



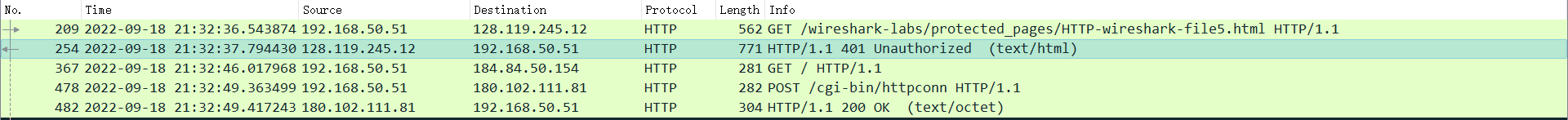


**（图1-19 Wireshark抓包显示）**

**Q&A:**

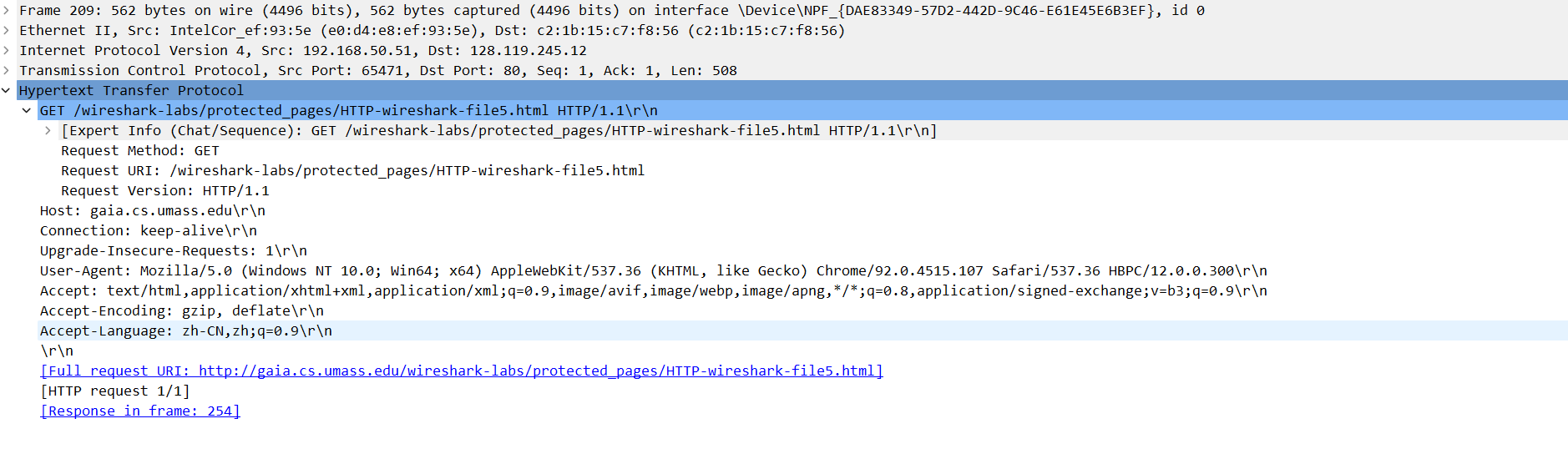
**18. What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?**

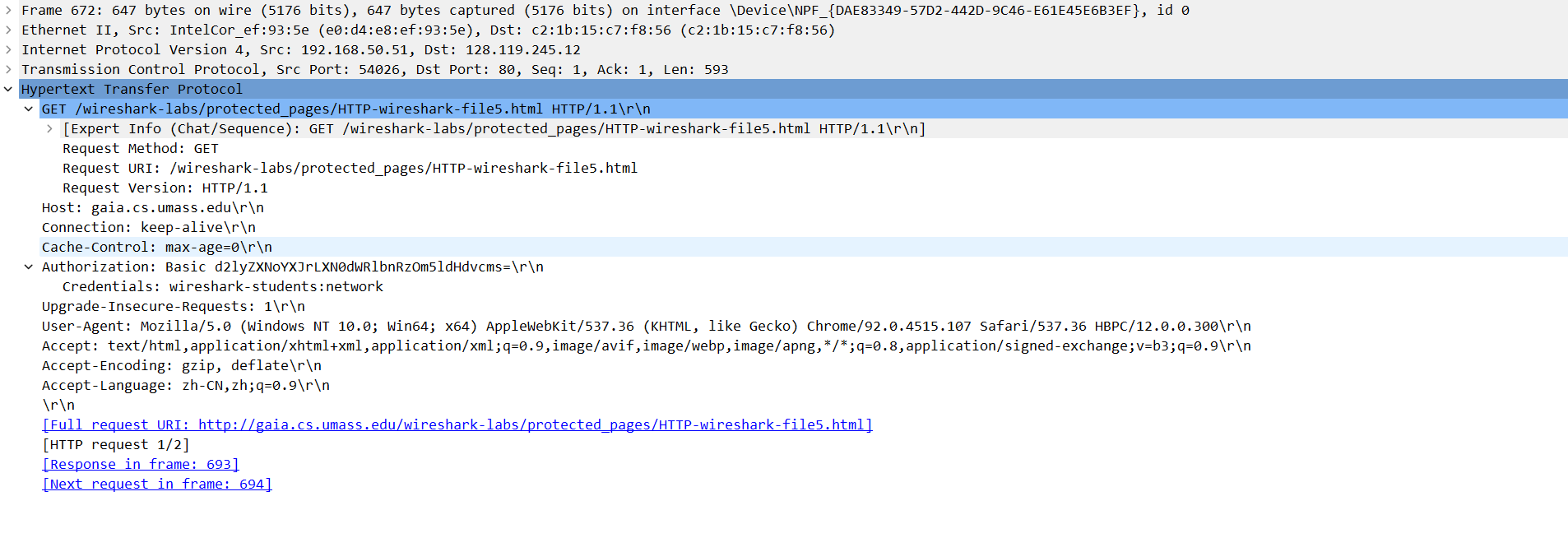
**Ans：**401 Unauthorized。



**19. When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?**

**Ans：**增添了Cache-Control和Authorization两个部分（对比如下图）。





1. **补充内容：**

**favicon**

在第一次抓包过程中，观察到Wireshark除了抓到了相应的GET/OK包，在过程也向服务器发送GET /favicon.ico HTTP/1.1报文。查阅资料了解到：favicon，即Favorites Icon的缩写，是指显示在浏览器收藏夹、地址栏和标签标题前面的个性化图标。 以图标的方式区别不同的网站。由于实验网址中并未进行相关的设置，浏览器返回404 Not Found报文，并记录进错误日志。

**缓存控制**

在http中，控制缓存开关的字段有两个：Pragma 和 Cache-Control。其中Pragma含有Pragma和Expires两个字段。当Pragma字段的值为no-cache时，表示禁用缓存，Expires的值为一个GMT时间，表示该缓存的有效时间。如果一个报文中同时出现Pragma和Cache-Control时，以Pragma为准。同时出现Cache-Control和Expires时，以Cache-Control为准。即优先级从高到低为 Pragma -> Cache-Control -> Expires

值得注意的是，Cache-Control除了在响应中使用，在请求中也可以使用。

**Base64编码**

Base64即包括小写字母a-z、大写字母A-Z、数字0-9、符号"+"、"/"一共64个字符的字符集，（若算上“=”，实际是65个字符）。将任何符号转换为该字符集中的字符的过程叫base64编码。在编码时，需要首先将字符串（图片等）转换成二进制序列，然后按每6个二进制位为一组，分成若干组，如果不足6位，则低位补0。每6位组成一个新的字节，高位补00，构成一个新的二进制序列，最后根据base64索引表中的值找到对应的字符从而构成相应的字符列。