山东大学 计算机科学与技术 学院

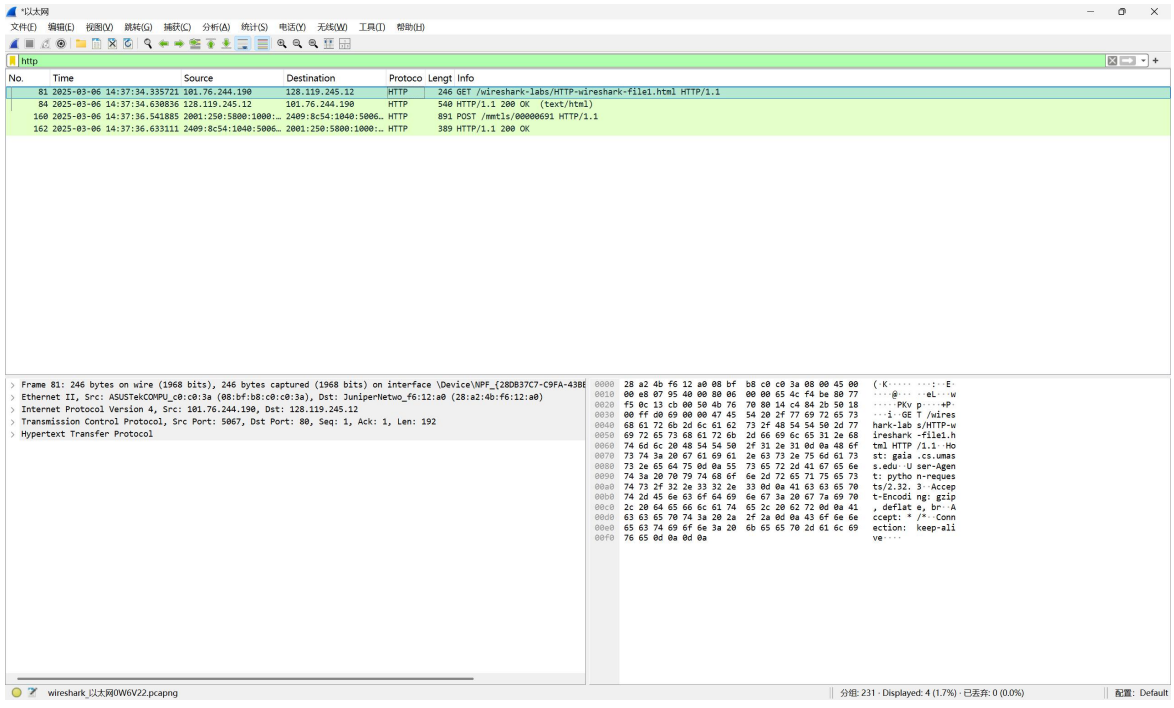
新兴网络技术与实践 课程实验报告

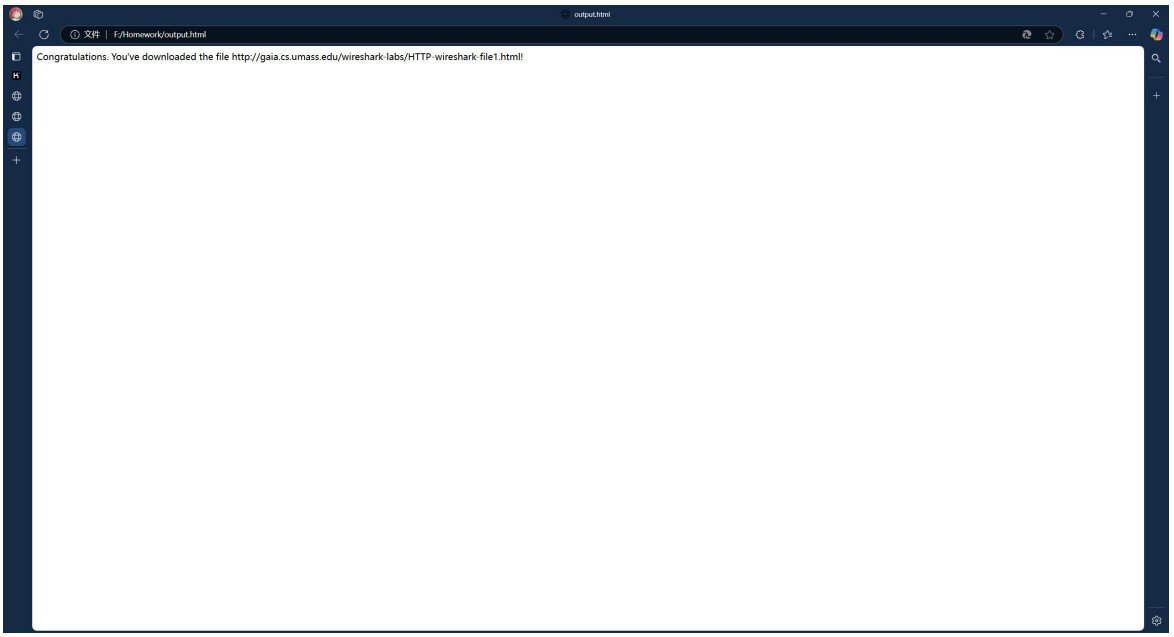
学号：202300130183	姓名：宋浩宇	班级：23 级智能班
实验题目：Wireshark HTTP		
实验学时：2	实验日期：2025/3/12	
实验目的：了解 HTTP 协议		

实验结果：

以下为对

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html 的 HTTP 请求和返回结果

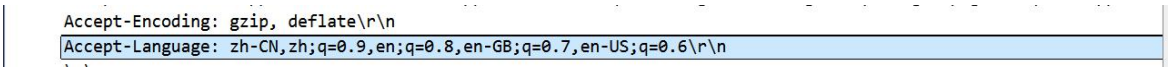




1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

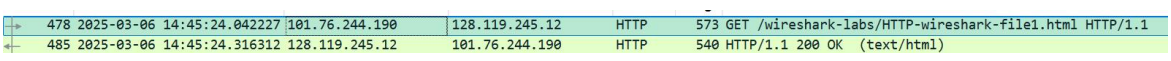
从我们截获到的 HTTP 的请求中的结果可以看到，我的电脑上使用的是 HTTP1.1，服务器使用的也是 HTTP1.1

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



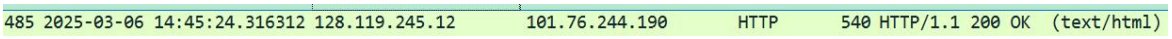
如图所示，我的浏览器表示可以接受的语言有简体中文、中文、英文、英文（英国）、英文（美国）

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



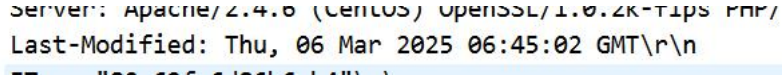
如图所示，我的电脑的 IP 地址是 101.76.244.190，gaia.cs.umass.edu 服务器的 IP 地址是 128.119.245.12

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



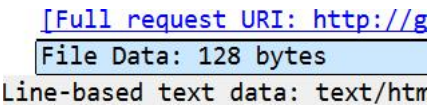
如图所示，浏览器从服务器收到的返回状态码是 200（HTTP OK）

5. When was the HTML file that you are retrieving last modified at the server?



如图所示，我们获取的这个 HTML 文件的最后修改时间是 2025 年 3 月 6 日，星期四，上午 6 点 45 分 23 秒，格林尼治标准时间。换算成北京时间就是 14:45:23

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

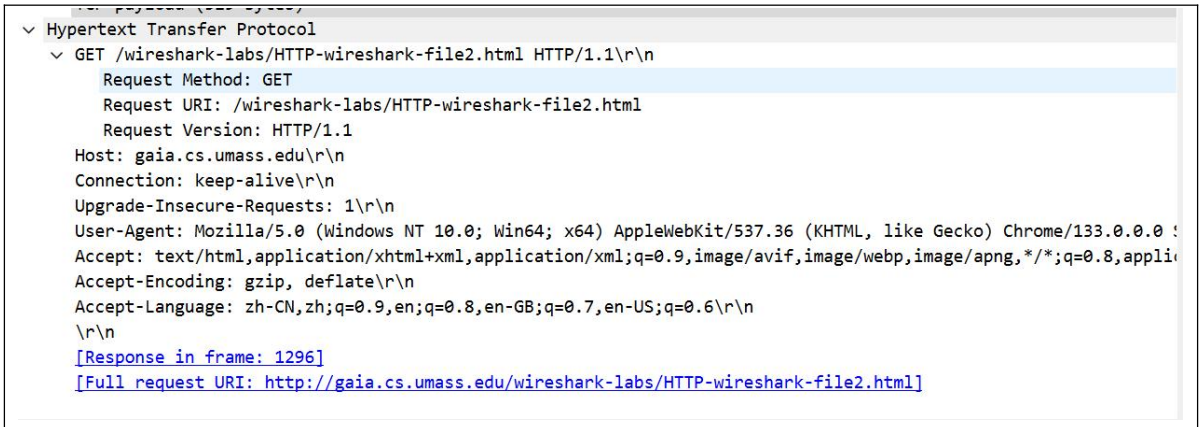


如图所示，大小为 128 字节

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.

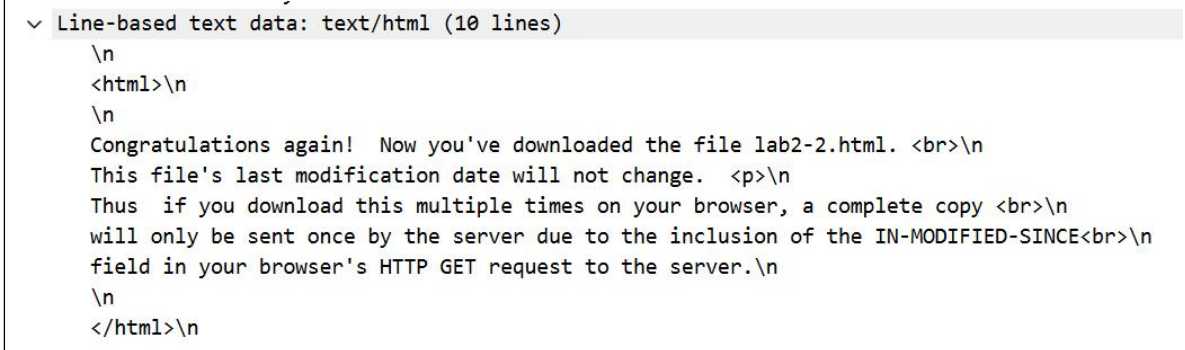
比如这个，Line-based test data: text/html

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?



如图，在第一次访问的时候并没有 IF-MODIFIED-SINCE 这个字段

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



如图，服务器明确返回了文件的内容，以上就是这个 html 文件的完整内容，由此即可判断。

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?

```
Hypertext Transfer Protocol
  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
      Request Method: GET
      Request URI: /wireshark-labs/HTTP-wireshark-file2.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    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/133.0.0.0 S
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
    If-None-Match: "173-62fa70a76157e"\r\n
    If-Modified-Since: Thu, 06 Mar 2025 06:59:01 GMT\r\n
    \r\n
    [Response in frame: 1350]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

如图，进行刷新操作时的 HTTP GET 请求中是包括了 IF-MODIFIED-SINCE 这个内容的。后边跟着的信息是我们第一次发出请求时收到返回结果中的上一次修改的时间，如下图所示：

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Thu, 06 Mar 2025 07:01:23 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Thu, 06 Mar 2025 06:59:01 GMT\r\n
    ETag: "173-62fa70a76157e"\r\n
    Accept-Ranges: bytes\r\n
  > Content-Length: 371\r\n
```

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.

```
1344 2025-03-06 15:01:31.129342 101.76.244.190  128.119.245.12   HTTP  685 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
1350 2025-03-06 15:01:31.413813 128.119.245.12  101.76.244.190   HTTP  294 HTTP/1.1 304 Not Modified
```

第二个返回的状态码是 304，Not Modified，这一次服务器并没有给出明确的 html 文件的内容，

```
> Frame 1350: 294 bytes on wire (2352 bits), 294 bytes captured (2352 bits) on interface \Device\NPF_{28DB37C7-C9FA-43
> Ethernet II, Src: JuniperNetwo_f6:12:a0 (28:a2:4b:f6:12:a0), Dst: ASUSTekCOMPU_c0:c0:3a (08:bf:b8:c0:c0:3a)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 101.76.244.190
> Transmission Control Protocol, Src Port: 80, Dst Port: 6900, Seq: 1, Ack: 632, Len: 240
> Hypertext Transfer Protocol
```

可以看到返回信息里没有Line-based test data: text/html 这个字段

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?

No.	Time	Source	Destination	Protoco	Lengt	Info
219	2025-03-06 15:12:37.264346	101.76.244.190	128.119.245.12	TCP	66	7852 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
242	2025-03-06 15:12:37.541689	128.119.245.12	101.76.244.190	TCP	66	80 → 7852 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
243	2025-03-06 15:12:37.541805	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
244	2025-03-06 15:12:37.542187	101.76.244.190	128.119.245.12	HTTP	573	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
280	2025-03-06 15:12:37.819576	128.119.245.12	101.76.244.190	TCP	60	80 → 7852 [ACK] Seq=1 Ack=520 Win=30336 Len=0
281	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=1 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
282	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=1461 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
283	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=2921 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
284	2025-03-06 15:12:37.821009	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4381 Win=65280 Len=0
285	2025-03-06 15:12:37.821031	128.119.245.12	101.76.244.190	HTTP	535	HTTP/1.1 200 OK (text/html)
286	2025-03-06 15:12:37.821042	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4862 Win=65024 Len=0
436	2025-03-06 15:12:42.822028	128.119.245.12	101.76.244.190	TCP	60	80 → 7852 [FIN, ACK] Seq=4862 Ack=520 Win=30336 Len=0
437	2025-03-06 15:12:42.822116	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4863 Win=65024 Len=0

我的浏览器只发出了一次 GET 请求消息，其中就是图中这个背景为深色被选中的数据包包含了获取目标数据的 GET 请求消息。

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

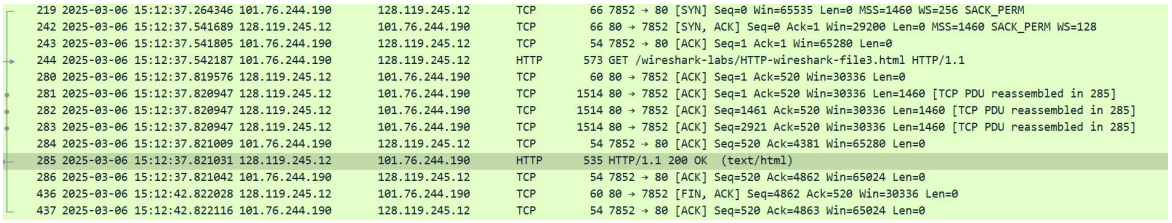
No.	Time	Source	Destination	Protoco	Lengt	Info
219	2025-03-06 15:12:37.264346	101.76.244.190	128.119.245.12	TCP	66	7852 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
242	2025-03-06 15:12:37.541689	128.119.245.12	101.76.244.190	TCP	66	80 → 7852 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
243	2025-03-06 15:12:37.541805	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
244	2025-03-06 15:12:37.542187	101.76.244.190	128.119.245.12	HTTP	573	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
280	2025-03-06 15:12:37.819576	128.119.245.12	101.76.244.190	TCP	60	80 → 7852 [ACK] Seq=1 Ack=520 Win=30336 Len=0
281	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=1 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
282	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=1461 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
283	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=2921 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
284	2025-03-06 15:12:37.821009	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4381 Win=65280 Len=0
285	2025-03-06 15:12:37.821031	128.119.245.12	101.76.244.190	HTTP	535	HTTP/1.1 200 OK (text/html)
286	2025-03-06 15:12:37.821042	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4862 Win=65024 Len=0
436	2025-03-06 15:12:42.822028	128.119.245.12	101.76.244.190	TCP	60	80 → 7852 [FIN, ACK] Seq=4862 Ack=520 Win=30336 Len=0
437	2025-03-06 15:12:42.822116	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4863 Win=65024 Len=0

如图，被选中的深色的数据包包含了 HTTP GET 请求响应相关的状态码和短语。

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

依旧参照上图，状态码为 200，短语为 OK

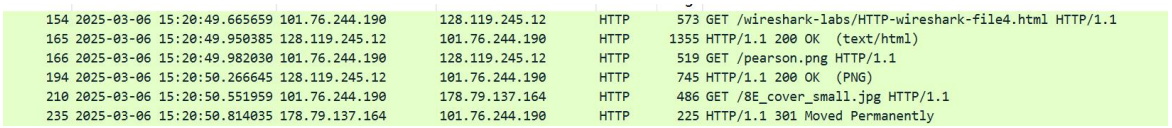
15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?



219	2025-03-06 15:12:37.264346	101.76.244.190	128.119.245.12	TCP	66	7852 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
242	2025-03-06 15:12:37.541689	128.119.245.12	101.76.244.190	TCP	66	80 → 7852 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
243	2025-03-06 15:12:37.541805	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
244	2025-03-06 15:12:37.542187	101.76.244.190	128.119.245.12	HTTP	573	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
280	2025-03-06 15:12:37.819576	128.119.245.12	101.76.244.190	TCP	60	80 → 7852 [ACK] Seq=1 Ack=520 Win=30336 Len=0
281	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=1 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
282	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=1461 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
283	2025-03-06 15:12:37.820947	128.119.245.12	101.76.244.190	TCP	1514	80 → 7852 [ACK] Seq=2921 Ack=520 Win=30336 Len=1460 [TCP PDU reassembled in 285]
284	2025-03-06 15:12:37.821009	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4381 Win=65280 Len=0
285	2025-03-06 15:12:37.821031	128.119.245.12	101.76.244.190	HTTP	535	HTTP/1.1 200 OK (text/html)
286	2025-03-06 15:12:37.821042	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4862 Win=65024 Len=0
436	2025-03-06 15:12:42.822028	128.119.245.12	101.76.244.190	TCP	60	80 → 7852 [FIN, ACK] Seq=4862 Ack=520 Win=30336 Len=0
437	2025-03-06 15:12:42.822116	101.76.244.190	128.119.245.12	TCP	54	7852 → 80 [ACK] Seq=520 Ack=4863 Win=65024 Len=0

通过 wireshark 追踪流的功能我们可以找到所有带着数据的 TCP 数据包，一共是 11 个。而其中用于向我的电脑传递网页上的这些数据的是 6 个（从 HTTP GET 请求发出后开始计算）。

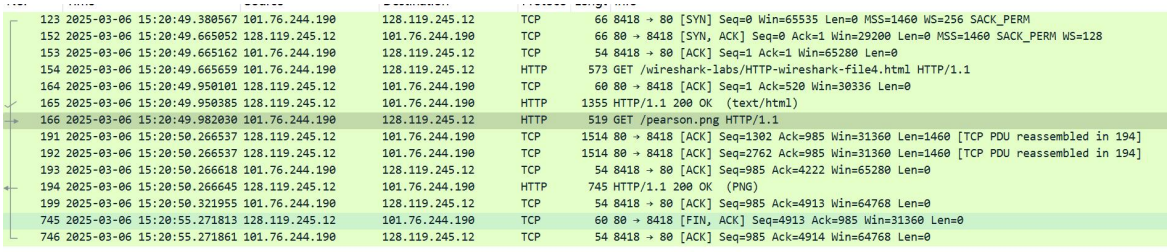
16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?



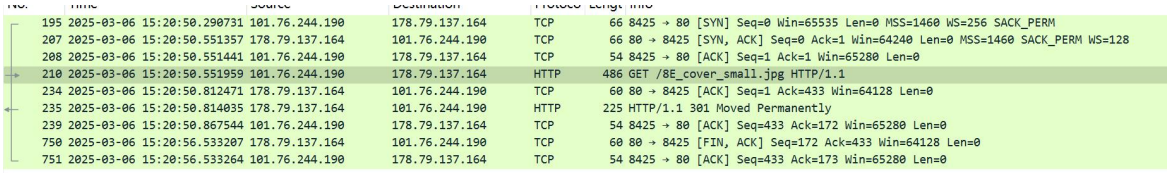
154	2025-03-06 15:20:49.665659	101.76.244.190	128.119.245.12	HTTP	573	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
165	2025-03-06 15:20:49.950385	128.119.245.12	101.76.244.190	HTTP	1355	HTTP/1.1 200 OK (text/html)
166	2025-03-06 15:20:49.982030	101.76.244.190	128.119.245.12	HTTP	519	GET /pearson.png HTTP/1.1
194	2025-03-06 15:20:50.266645	128.119.245.12	101.76.244.190	HTTP	745	HTTP/1.1 200 OK (PNG)
210	2025-03-06 15:20:50.551959	101.76.244.190	178.79.137.164	HTTP	486	GET /8E_cover_small.jpg HTTP/1.1
235	2025-03-06 15:20:50.814035	178.79.137.164	101.76.244.190	HTTP	225	HTTP/1.1 301 Moved Permanently

如图，我的浏览器一共发出了三次 HTTP GET 请求，分别发到 128.119.245.12、128.119.245.12、178.79.137.164 这几个 IP 地址。

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.



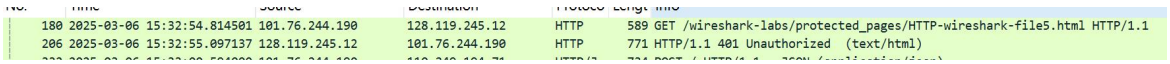
123	2025-03-06 15:20:49.380567	101.76.244.190	128.119.245.12	TCP	66	8418 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
152	2025-03-06 15:20:49.665052	128.119.245.12	101.76.244.190	TCP	66	80 → 8418 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
153	2025-03-06 15:20:49.665162	101.76.244.190	128.119.245.12	TCP	54	8418 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
154	2025-03-06 15:20:49.665659	101.76.244.190	128.119.245.12	HTTP	573	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
164	2025-03-06 15:20:49.950101	128.119.245.12	101.76.244.190	TCP	60	80 → 8418 [ACK] Seq=1 Ack=520 Win=30336 Len=0
165	2025-03-06 15:20:49.950385	128.119.245.12	101.76.244.190	HTTP	1355	HTTP/1.1 200 OK (text/html)
166	2025-03-06 15:20:49.982030	101.76.244.190	128.119.245.12	HTTP	519	GET /pearson.png HTTP/1.1
191	2025-03-06 15:20:50.266537	128.119.245.12	101.76.244.190	TCP	1514	80 → 8418 [ACK] Seq=1302 Ack=985 Win=31360 Len=1460 [TCP PDU reassembled in 194]
192	2025-03-06 15:20:50.266537	128.119.245.12	101.76.244.190	TCP	1514	80 → 8418 [ACK] Seq=2762 Ack=985 Win=31360 Len=1460 [TCP PDU reassembled in 194]
193	2025-03-06 15:20:50.266618	101.76.244.190	128.119.245.12	TCP	54	8418 → 80 [ACK] Seq=985 Ack=4222 Win=65280 Len=0
194	2025-03-06 15:20:50.266645	128.119.245.12	101.76.244.190	HTTP	745	HTTP/1.1 200 OK (PNG)
199	2025-03-06 15:20:50.321955	101.76.244.190	128.119.245.12	TCP	54	8418 → 80 [ACK] Seq=985 Ack=4913 Win=64768 Len=0
745	2025-03-06 15:20:55.271813	128.119.245.12	101.76.244.190	TCP	60	80 → 8418 [FIN, ACK] Seq=4913 Ack=985 Win=31360 Len=0
746	2025-03-06 15:20:55.271861	101.76.244.190	128.119.245.12	TCP	54	8418 → 80 [ACK] Seq=985 Ack=4914 Win=64768 Len=0



No.	Time	Source	Destination	Protocol	Length	Info
195	2025-03-06 15:20:50.290731	101.76.244.190	178.79.137.164	TCP	66	8425 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
207	2025-03-06 15:20:50.551357	178.79.137.164	101.76.244.190	TCP	66	80 → 8425 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
208	2025-03-06 15:20:50.551441	101.76.244.190	178.79.137.164	TCP	54	8425 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0
210	2025-03-06 15:20:50.551959	101.76.244.190	178.79.137.164	HTTP	486	GET /8E_cover_small.jpg HTTP/1.1
234	2025-03-06 15:20:50.812471	178.79.137.164	101.76.244.190	TCP	60	80 → 8425 [ACK] Seq=1 Ack=433 Win=64128 Len=0
235	2025-03-06 15:20:50.814035	178.79.137.164	101.76.244.190	HTTP	225	HTTP/1.1 301 Moved Permanently
239	2025-03-06 15:20:50.867544	101.76.244.190	178.79.137.164	TCP	54	8425 → 80 [ACK] Seq=433 Ack=172 Win=65280 Len=0
750	2025-03-06 15:20:56.533207	178.79.137.164	101.76.244.190	TCP	60	80 → 8425 [FIN, ACK] Seq=172 Ack=433 Win=64128 Len=0
751	2025-03-06 15:20:56.533264	101.76.244.190	178.79.137.164	TCP	54	8425 → 80 [ACK] Seq=433 Ack=173 Win=65280 Len=0

我判断我的浏览器应该是依次下载的两张图片而不是并行下载，因为通过比较获取图片用的这两个 HTTP GET 请求和返回的时间可以看出来，以及 TCP 数据包接收到的时间可以判断，第二个 GET 请求的发起时间是在第一个请求的图片下载完之后。

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



No.	Time	Source	Destination	Protocol	Length	Info
180	2025-03-06 15:32:54.814501	101.76.244.190	128.119.245.12	HTTP	589	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
206	2025-03-06 15:32:55.097137	128.119.245.12	101.76.244.190	HTTP	771	HTTP/1.1 401 Unauthorized (text/html)

如图所示，在我进行第一次访问的时候，服务器返回的状态码是 401. 短语是 Unauthorized 表示缺少身份信息。

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

```
Hypertext Transfer Protocol
  GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
      Request Method: GET
      Request URI: /wireshark-labs/protected_pages/HTTP-wireshark-file5.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
  > Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\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/133.0.0.0 Saf
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
    \r\n
    [Response in frame: 529]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html]
```

如图，在浏览器第二次发出 HTTP GET 请求时，多了一个 Authorization 字段。

问题及收获：

问题：因为浏览器有在第一次访问 HTTP 链接时会自动重定向的问题，导致我无法在第一次 GET 时获取到状态码 200 的情况，因此我编写了一个 python 脚本来进行 HTTP GET 来获取状态码 200 用于实验报告的截图中。

收获：了解到了 HTTP GET 请求中一些字段的意义，包括发送时的 Authurization、IF-MODIFIED-SINCE，接受时的 Line-based test data: text/html，还有 TCP 数据包也会用于在 HTTP 请求中作为伴随的传输数据的介质。还有就是我所使用的浏览器（Edge）在加载一个页面中绑定到其他 URL 的资源的时候（可能）是依次加载而不是并行加载的。