山东大学 计算机 学院

计算机网络 课程实验报告

学号：	姓名：	班级：
实验题目： 实验二		
实验学时：2h	实验日期： 2023.03.06	

实验目的：

了解基本的 GET/响应交互以及 HTTP 消息格式、掌握检索大型 HTML 文件以及检索带有嵌入对象的 HTML 文件的方法和 HTTP 身份验证及安全性

硬件环境：

Windows10 家庭版

软件环境：

Wireshark

实验步骤与内容：

实验内容：

实验一：

1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
2. What languages (if any) does your browser indicate that it can accept to the server?
3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?
4. What is the status code returned from the server to your browser?
5. When was the HTML file that you are retrieving last modified at the server?
6. How many bytes of content are being returned to your browser?
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.

实验二：

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

实验三：

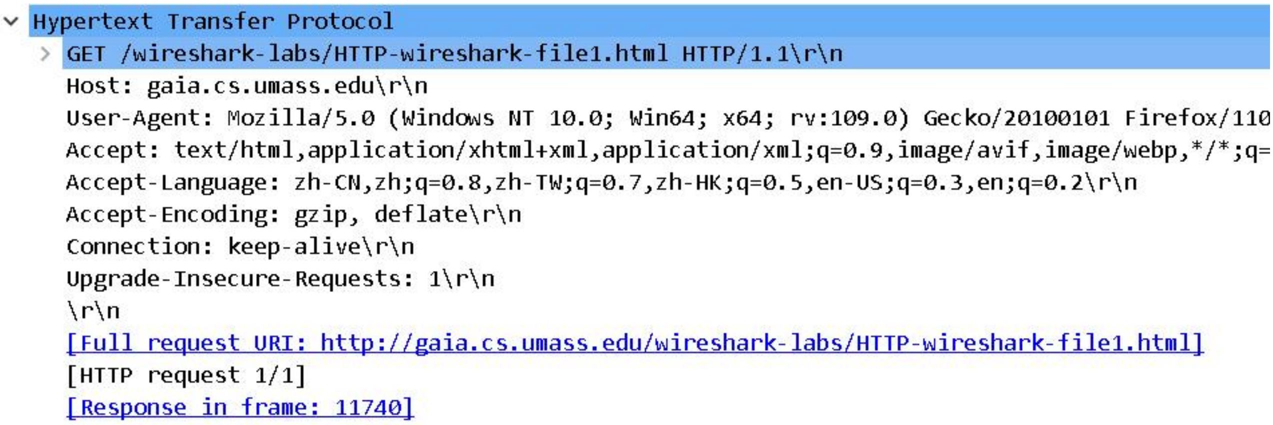
12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?
14. What is the status code and phrase in the response?
15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

实验四：
16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?
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.

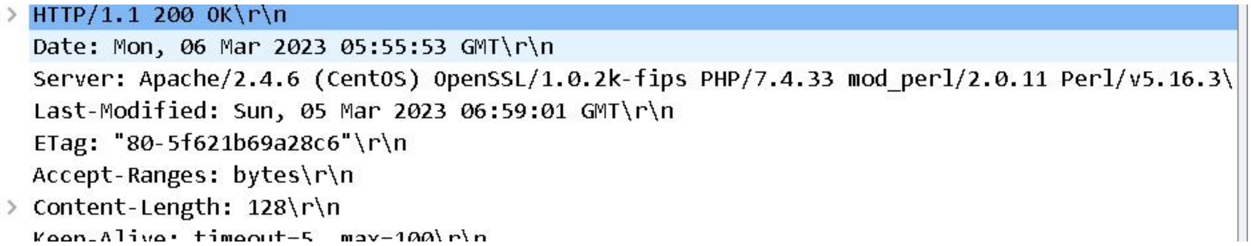
实验五：
18. What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?
19. When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

实验步骤：
先打开 Wireshark，然后根据实验指导书，并打开相应的网站，进行相关抓包，同时查看详细请求。
实验一：
1. 浏览器和服务器运行的 HTTP 都是 1.1 版
2. 可接受的语言是



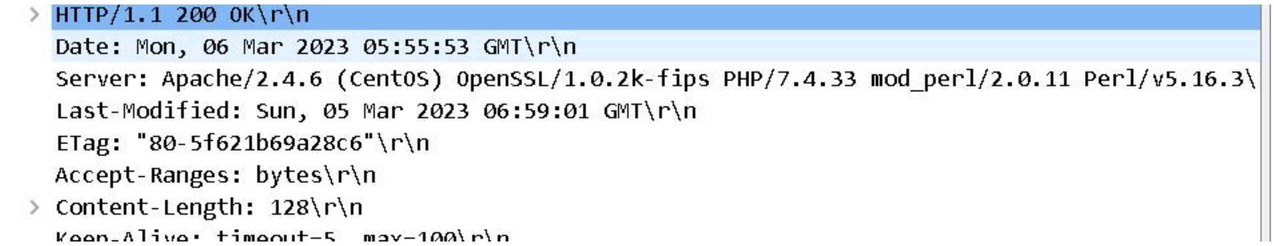
```
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/110
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
    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
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
    [HTTP request 1/1]
    [Response in frame: 11740]
```

3. 本机的 IP 地址为 172.25.160.154，服务器的 IP 地址为：128.119.245.12
4. 200 OK
5. 上次修改时间为



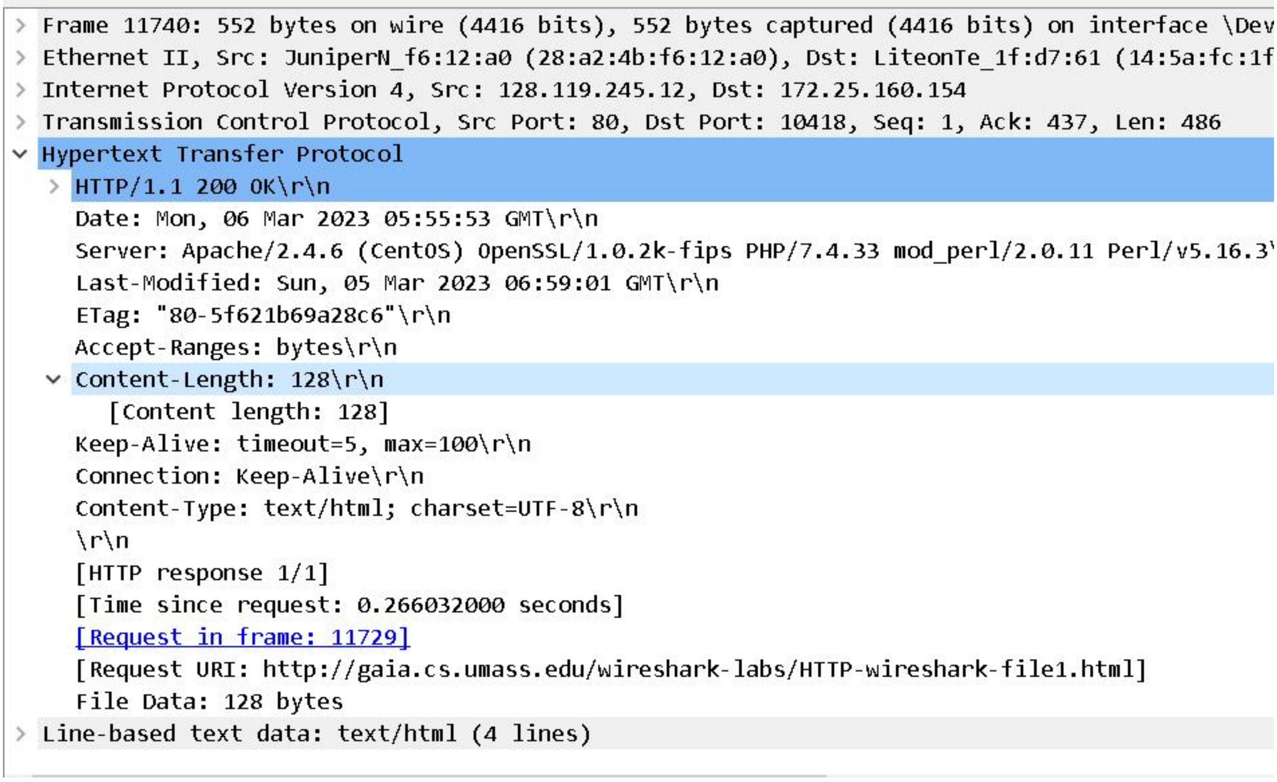
```
> HTTP/1.1 200 OK\r\n
  Date: Mon, 06 Mar 2023 05:55:53 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3\
  Last-Modified: Sun, 05 Mar 2023 06:59:01 GMT\r\n
  ETag: "80-5f621b69a28c6"\r\n
  Accept-Ranges: bytes\r\n
> Content-Length: 128\r\n
  Keep-Alive: timeout=5, max=100\r\n
```

6.



```
  > HTTP/1.1 200 OK\r\n
    Date: Mon, 06 Mar 2023 05:55:53 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\
    Last-Modified: Sun, 05 Mar 2023 06:59:01 GMT\r\n
    ETag: "80-5f621b69a28c6"\r\n
    Accept-Ranges: bytes\r\n
  > Content-Length: 128\r\n
    Keep-Alive: timeout=5, max=100\r\n
```

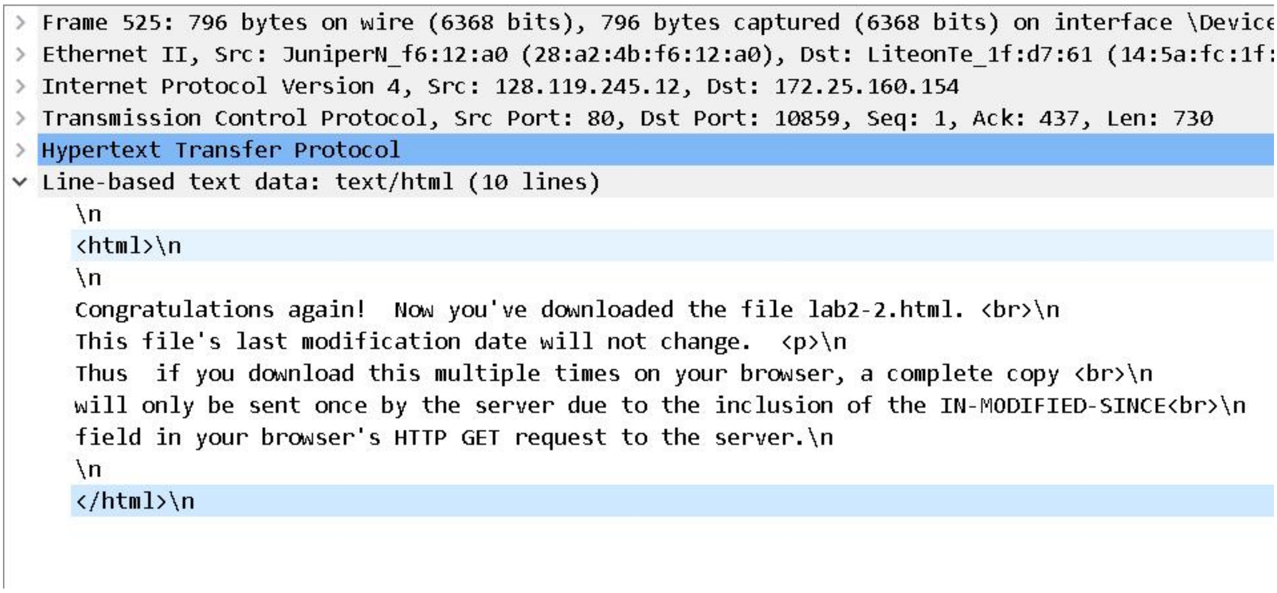
7.



```
> Frame 11740: 552 bytes on wire (4416 bits), 552 bytes captured (4416 bits) on interface \Dev
> Ethernet II, Src: JuniperN_f6:12:a0 (28:a2:4b:f6:12:a0), Dst: LiteonTe_1f:d7:61 (14:5a:fc:1f
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.25.160.154
> Transmission Control Protocol, Src Port: 80, Dst Port: 10418, Seq: 1, Ack: 437, Len: 486
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Mon, 06 Mar 2023 05:55:53 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\
    Last-Modified: Sun, 05 Mar 2023 06:59:01 GMT\r\n
    ETag: "80-5f621b69a28c6"\r\n
    Accept-Ranges: bytes\r\n
  v Content-Length: 128\r\n
      [Content length: 128]
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.266032000 seconds]
    [Request in frame: 11729]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
    File Data: 128 bytes
> Line-based text data: text/html (4 lines)
```

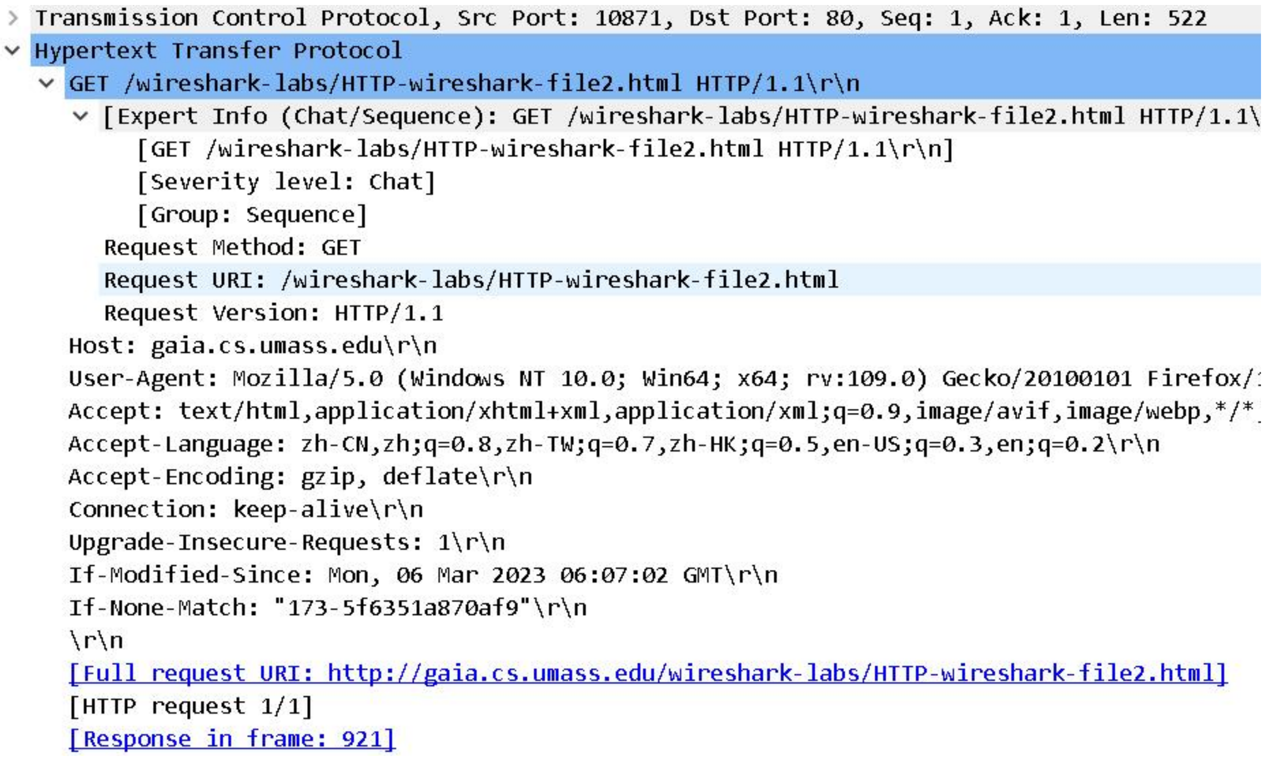
实验二：

1. 不能看见
2. 是



```
> Frame 525: 796 bytes on wire (6368 bits), 796 bytes captured (6368 bits) on interface \Device
> Ethernet II, Src: JuniperN_f6:12:a0 (28:a2:4b:f6:12:a0), Dst: LiteonTe_1f:d7:61 (14:5a:fc:1f:
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.25.160.154
> Transmission Control Protocol, Src Port: 80, Dst Port: 10859, Seq: 1, Ack: 437, Len: 730
> Hypertext Transfer Protocol
v Line-based text data: text/html (10 lines)
    \n
    <html>\n
    \n
    Congratulations again!  Now you've downloaded the file lab2-2.html. <br>\n
    This file's last modification date will not change.  <p>\n
    Thus  if you download this multiple times on your browser, a complete copy <br>\n
    will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
    field in your browser's HTTP GET request to the server.\n
    \n
    </html>\n
```

3. 是

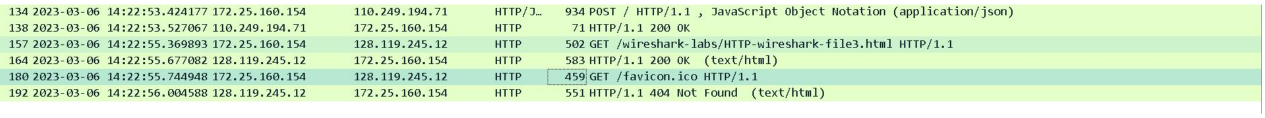


```
> Transmission Control Protocol, Src Port: 10871, Dst Port: 80, Seq: 1, Ack: 1, Len: 522
v Hypertext Transfer Protocol
  v GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    v [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\
        [GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /wireshark-labs/HTTP-wireshark-file2.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*
    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
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    If-Modified-Since: Mon, 06 Mar 2023 06:07:02 GMT\r\n
    If-None-Match: "173-5f6351a870af9"\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 1/1]
    [Response in frame: 921]
```

4. 状态代码为 304，短语是 Not Modified。服务器并没有显示返回文件内容，这是因为我们并没有对内容做修改，而之前的内容已经被缓存，所以不会再返回一次。

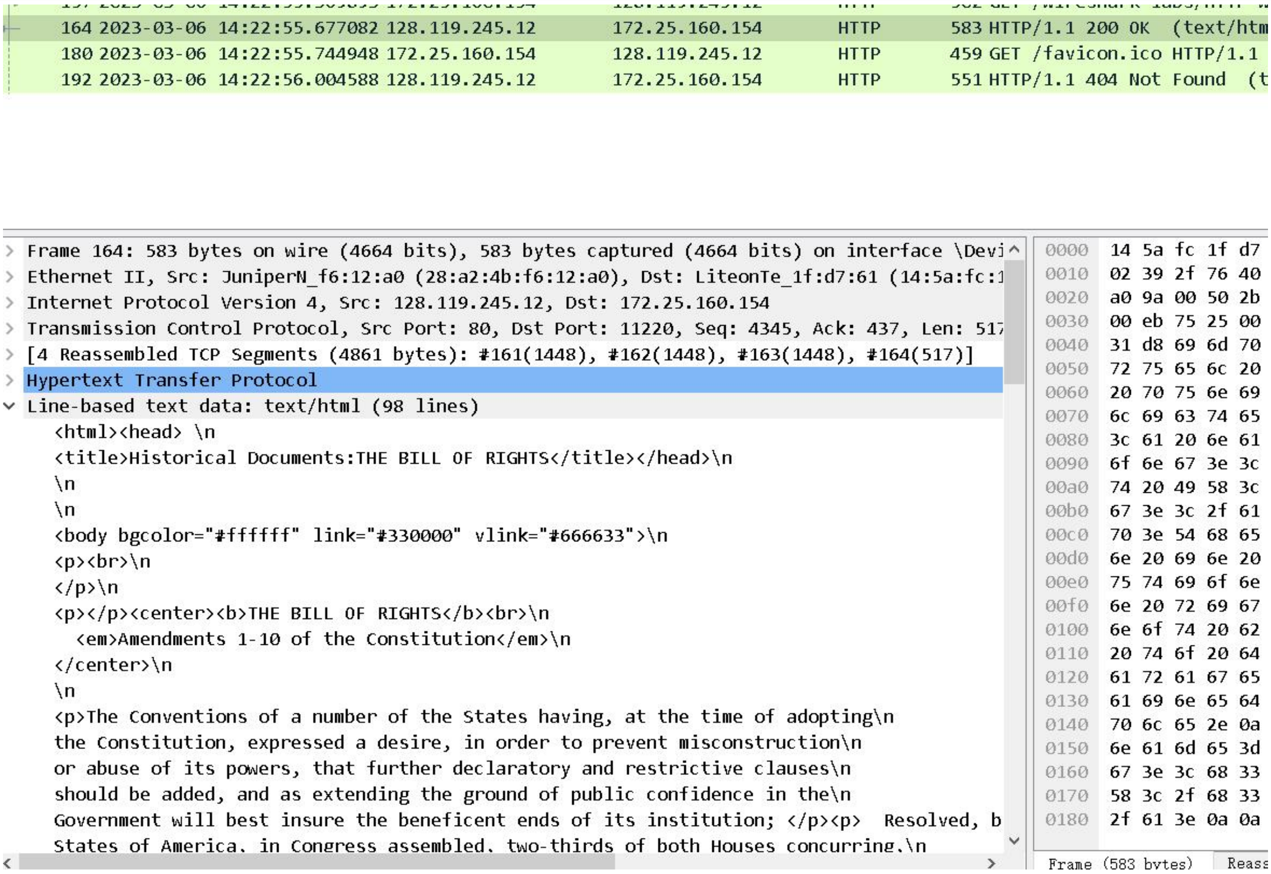
实验三：

1. 两条 GET 请求消息



```
134 2023-03-06 14:22:53.424177 172.25.160.154   110.249.194.71   HTTP/J…  934 POST / HTTP/1.1 , JavaScript Object Notation (application/json)
138 2023-03-06 14:22:53.527067 110.249.194.71   172.25.160.154   HTTP      71 HTTP/1.1 200 OK
157 2023-03-06 14:22:55.369893 172.25.160.154   128.119.245.12   HTTP     502 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
164 2023-03-06 14:22:55.677082 128.119.245.12   172.25.160.154   HTTP     583 HTTP/1.1 200 OK  (text/html)
180 2023-03-06 14:22:55.744948 172.25.160.154   128.119.245.12   HTTP     459 GET /favicon.ico HTTP/1.1
192 2023-03-06 14:22:56.004588 128.119.245.12   172.25.160.154   HTTP     551 HTTP/1.1 404 Not Found  (text/html)
```

2.



```
164 2023-03-06 14:22:55.677082 128.119.245.12   172.25.160.154   HTTP   583 HTTP/1.1 200 OK  (text/htm
180 2023-03-06 14:22:55.744948 172.25.160.154   128.119.245.12   HTTP   459 GET /favicon.ico HTTP/1.1
192 2023-03-06 14:22:56.004588 128.119.245.12   172.25.160.154   HTTP   551 HTTP/1.1 404 Not Found  (t
```

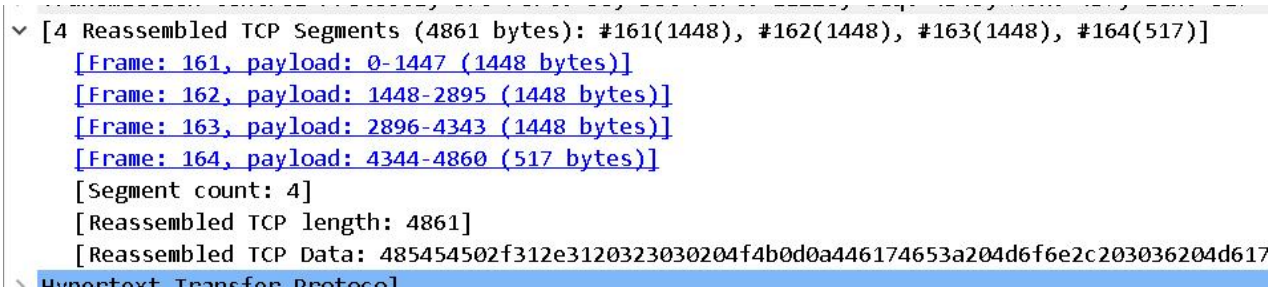
```
> Frame 164: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface \Devi
> Ethernet II, Src: JuniperN_f6:12:a0 (28:a2:4b:f6:12:a0), Dst: LiteonTe_1f:d7:61 (14:5a:fc:1
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.25.160.154
> Transmission Control Protocol, Src Port: 80, Dst Port: 11220, Seq: 4345, Ack: 437, Len: 517
> [4 Reassembled TCP Segments (4861 bytes): #161(1448), #162(1448), #163(1448), #164(517)]
> Hypertext Transfer Protocol
v Line-based text data: text/html (98 lines)
    <html><head> \n
    <title>Historical Documents:THE BILL OF RIGHTS</title></head>\n
    \n
    \n
    <body bgcolor="#ffffff" link="#330000" vlink="#666633">\n
    <p><br>\n
    </p>\n
    <p></p><center><b>THE BILL OF RIGHTS</b><br>\n
      <em>Amendments 1-10 of the Constitution</em>\n
    </center>\n
    \n
    <p>The Conventions of a number of the States having, at the time of adopting\n
    the Constitution, expressed a desire, in order to prevent misconstruction\n
    or abuse of its powers, that further declaratory and restrictive clauses\n
    should be added, and as extending the ground of public confidence in the\n
    Government will best insure the beneficent ends of its institution; </p><p>  Resolved, b
    States of America, in Congress assembled, two-thirds of both Houses concurring,\n
```

```
0000  14 5a fc 1f d7
0010  02 39 2f 76 40
0020  a0 9a 00 50 2b
0030  00 eb 75 25 00
0040  31 d8 69 6d 70
0050  72 75 65 6c 20
0060  20 70 75 6e 69
0070  6c 69 63 74 65
0080  3c 61 20 6e 61
0090  6f 6e 67 3e 3c
00a0  74 20 49 58 3c
00b0  67 3e 3c 2f 61
00c0  70 3e 54 68 65
00d0  6e 20 69 6e 20
00e0  75 74 69 6f 6e
00f0  6e 20 72 69 67
0100  6e 6f 74 20 62
0110  20 74 6f 20 64
0120  61 72 61 67 65
0130  61 69 6e 65 64
0140  70 6c 65 2e 0a
0150  6e 61 6d 65 3d
0160  67 3e 3c 68 33
0170  58 3c 2f 68 33
0180  2f 61 3e 0a 0a
```

Frame (583 bytes) Reass

3. 200 OK

4. 需要 4 个 TCP 段



```
v [4 Reassembled TCP Segments (4861 bytes): #161(1448), #162(1448), #163(1448), #164(517)]
    [Frame: 161, payload: 0-1447 (1448 bytes)]
    [Frame: 162, payload: 1448-2895 (1448 bytes)]
    [Frame: 163, payload: 2896-4343 (1448 bytes)]
    [Frame: 164, payload: 4344-4860 (517 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 4861]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a204d6f6e2c203036204d617
```

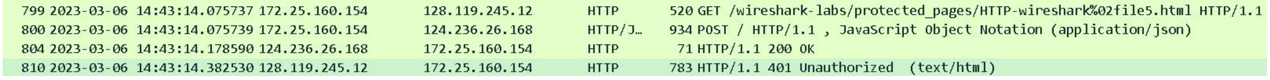
实验四：

1. 发送了三条 HTTP GET 请求消息，其中这些 GET 请求信息发送到了两个 IP 地址，一个为 128.119.245.12，一个为 178.79.137.164

2. 从两个网站并行下载，原因如下：我们可以看到对每张图片都发送了 GET 命令，这样可以提高响应速度，同时依次对两张图片发送 GET 命令，应该是并行下载。

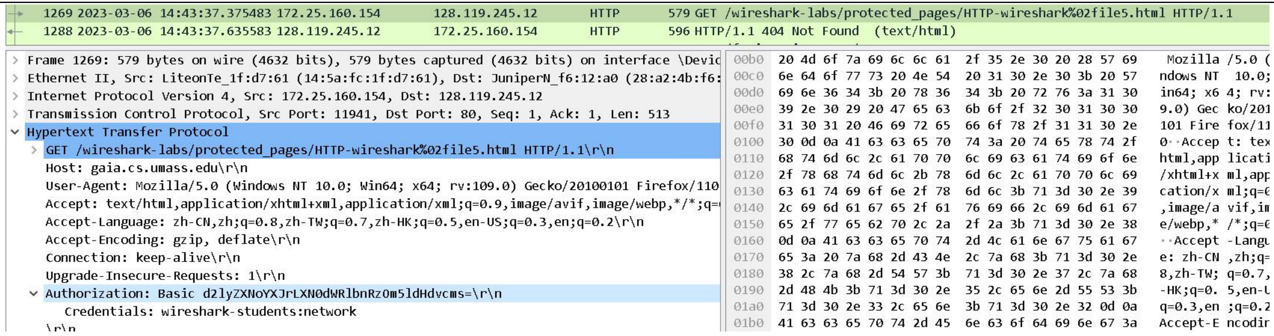
实验五：

1. 初次响应是 401 Unauthorized



```
799 2023-03-06 14:43:14.075737 172.25.160.154   128.119.245.12   HTTP     520 GET /wireshark-labs/protected_pages/HTTP-wireshark%02file5.html HTTP/1.1
800 2023-03-06 14:43:14.075739 172.25.160.154   124.236.26.168   HTTP/J…  934 POST / HTTP/1.1 , JavaScript Object Notation (application/json)
804 2023-03-06 14:43:14.178590 124.236.26.168   172.25.160.154   HTTP      71 HTTP/1.1 200 OK
810 2023-03-06 14:43:14.382530 128.119.245.12   172.25.160.154   HTTP     783 HTTP/1.1 401 Unauthorized  (text/html)
```

2. 第二次多了认证部分



```
1269 2023-03-06 14:43:37.375483 172.25.160.154    128.119.245.12    HTTP    579 GET /wireshark-labs/protected_pages/HTTP-wireshark%02file5.html HTTP/1.1
1288 2023-03-06 14:43:37.635583 128.119.245.12    172.25.160.154    HTTP    596 HTTP/1.1 404 Not Found  (text/html)
```

```
> Frame 1269: 579 bytes on wire (4632 bits), 579 bytes captured (4632 bits) on interface \Devic
> Ethernet II, Src: LiteonTe_1f:d7:61 (14:5a:fc:1f:d7:61), Dst: JuniperN_f6:12:a0 (28:a2:4b:f6:
> Internet Protocol Version 4, Src: 172.25.160.154, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 11941, Dst Port: 80, Seq: 1, Ack: 1, Len: 513
v Hypertext Transfer Protocol
  > GET /wireshark-labs/protected_pages/HTTP-wireshark%02file5.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/110
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
    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
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  v Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
      Credentials: wireshark-students:network
    \r\n
```

```
00b0  20 4d 6f 7a 69 6c 6c 61  2f 35 2e 30 20 28 57 69    Mozilla /5.0 (
00c0  6e 64 6f 77 73 20 4e 54  20 31 30 2e 30 3b 20 57   ndows NT  10.0;
00d0  69 6e 36 34 3b 20 78 36  34 3b 20 72 76 3a 31 30   in64; x6 4; rv:
00e0  39 2e 30 29 20 47 65 63  6b 6f 2f 32 30 31 30 30   9.0) Gec ko/201
00f0  31 30 31 20 46 69 72 65  66 6f 78 2f 31 31 30 2e   101 Fire fox/11
0100  30 0d 0a 41 63 63 65 70  74 3a 20 74 65 78 74 2f   0··Accep t: tex
0110  68 74 6d 6c 2c 61 70 70  6c 69 63 61 74 69 6f 6e   html,app licati
0120  2f 78 68 74 6d 6c 2b 78  6d 6c 2c 61 70 70 6c 69   /xhtml+x ml,app
0130  63 61 74 69 6f 6e 2f 78  6d 6c 3b 71 3d 30 2e 39   cation/x ml;q=0
0140  2c 69 6d 61 67 65 2f 61  76 69 66 2c 69 6d 61 67   ,image/a vif,im
0150  65 2f 77 65 62 70 2c 2a  2f 2a 3b 71 3d 30 2e 38   e/webp,* /*;q=0
0160  0d 0a 41 63 63 65 70 74  2d 4c 61 6e 67 75 61 67   ··Accept -Langu
0170  65 3a 20 7a 68 2d 43 4e  2c 7a 68 3b 71 3d 30 2e   e: zh-CN ,zh;q=
0180  38 2c 7a 68 2d 54 57 3b  71 3d 30 2e 37 2c 7a 68   8,zh-TW; q=0.7,
0190  2d 48 4b 3b 71 3d 30 2e  35 2c 65 6e 2d 55 53 3b   -HK;q=0. 5,en-U
01a0  71 3d 30 2e 33 2c 65 6e  3b 71 3d 30 2e 32 0d 0a   q=0.3,en ;q=0.2
01b0  41 63 63 65 70 74 2d 45  6e 63 6f 64 69 6e 67 3a   Accept-E ncodin
```

结论分析与体会：

通过具体查看每一个 HTTP 请求，了解其中的具体内容，对 HTTP 中的相关内容有了更加深刻的理解。了解了 HTTP 基本的 GET 请求，响应式交互，HTTP 消息格式，检索大型 HTML 文件，检索带有嵌入对象的 HTML 文件，以及 HTTP 身份验证和安全性，对 HTTP 有了进一步的认知