

电子科技大学信息与软件工程学院

实 验 报 告

学 号 2018091618008

姓 名 袁昊男

(实验) 课程名称 计算机网络系统

理论教师 王伟东

实验教师 文淑华

电 子 科 技 大 学

实 验 报 告

学生姓名：袁昊男 学号：2018091618008 指导教师：文淑华

实验地点：在线实验 实验时间：2020.04.16

一、实验名称：网络协议分析工具 Wireshark 的使用

二、实验学时：2 学时

三、实验目的：

学习使用网络协议分析工具 Wireshark 的方法，并用它来分析 HTTP 网络协议。

四、实验原理：

抓取网络数据包查看其中内容并进行分析是网络协议分析的基本方法之一。Wireshark（前称 Ethereal）即是一个网络数据包分析软件，其功能是抓取网络数据包，并尽可能显示出最为详细的网络数据包资料。

网络数据包分析软件的功能可想象成“电工技师使用电表来量测电流、电压、电阻”的工作——只是将场景移植到网络上，并将电线替换成网络线。

Wireshark 可用来检测网络问题，检查信息安全相关问题，可用于学习和分析网络协议，也可以用于测试新的通讯协议。

Wireshark 不是入侵检测软件（Intrusion Detection Software, IDS）。对于网络上的异常流量行为，Wireshark 不会产生警示或是任何提示。但仔细分析 Wireshark 抓取的数据包能够帮助使用者对于网络行为有更清楚的了解。Wireshark 不会修改网络数据包的内容，它只会反映出目前通信中的数据包信息。Wireshark 本身也不会向网络发送数据包。

Wireshark 具有如下功能：

- 支持 UNIX 和 Windows 平台
- 在接口实时捕捉包
- 能详细显示包的详细协议信息
- 可以打开/保存捕捉的包
- 可以导入导出其他捕捉程序支持的包数据格式

- 可以通过多种方式过滤包
- 多种方式查找包
- 通过过滤以多种色彩显示包
- 创建多种统计分析

对于网络协议的分析，基本的方式是逐层查看 Wireshark 抓取到的数据包中的协议字段部分（比如 IP 头部，TCP 头部等等），或者直接查看某层头部，从而加深对相关协议的理解。

五、实验内容：

- 1、利用 Wireshark 分析 HTTP GET/Response 基本消息交互；
- 2、利用 Wireshark 观察和分析有条件的 HTTP GET/Response 消息交互；
- 3、利用 Wireshark 观察和分析 HTTP 获取长文档机制；
- 4、利用 Wireshark 分析网页中嵌入有对象（图片）的获取流程；
- 5、了解 HTTP 的认证机制。

六、实验器材（设备、元器件）：

安装有网络协议分析工具 Wireshark 软件的 PC 机一台。

七、实验步骤：

1、HTTP GET/Response 基本消息交互

首先下载一个非常简短且没有嵌入对象的网页，开始 HTTP 实验。实验步骤如下：

- 打开浏览器；
- 打开 Wireshark 软件，但是先不要开始抓取流量。在显示过滤参数窗口中输入“http”（只是字母，不包括引号），这样只有 HTTP 消息才会随后显示在报文显示窗口中。（我们目前只关注 HTTP 协议，而不希望显示所有被捕获的数据包）。
- 等待一分多钟（我们会很快知道为什么），然后开始用 Wireshark 抓包。
- 在浏览器中输入如下地址：
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>
你的浏览器应该会显示这个非常简单且只有一行文字的网页。
- 停止 Wireshark 抓取数据包。

- 你的 Wireshark 窗口应该如图 1 所示类似。如果你无法运行 Wireshark 抓取实时网络流量，那么你下载一个流量文件，该文件是按照上述步骤进行实验时所抓取的流量。

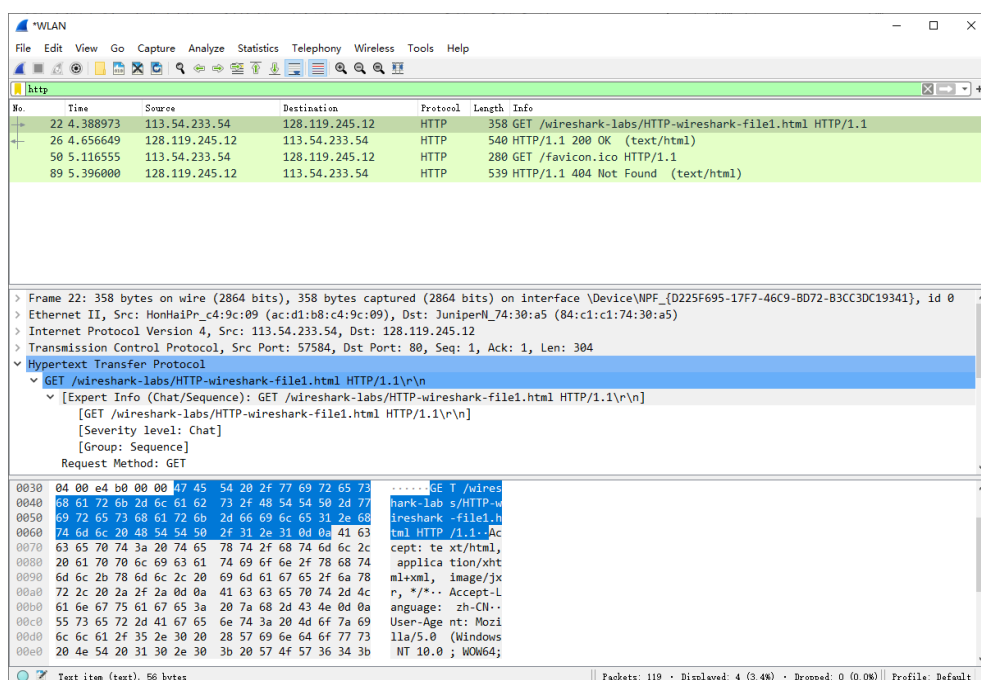


图1: 浏览器访问<http://gaia.cs.umass.edu/wireshark-labs/HTTPwireshark-file1.html> 页面，而由Wireshark所抓取的数据包显示

图 1 的报文显示窗口中，显示了 Wireshark 抓取的四个 HTTP 消息，后两个消息是关于 favico.ico 图表的 HTTP GET 和 Response 消息。如果显示有这个文件的下载数据包，是由于你的浏览器自动地询问服务器是否有图标文件可被显示在浏览器地址栏的 URL 旁边。本实验忽略这个文件下载所生成的容易误导的数据包。

前两个消息：一个是 GET 信息（从你的浏览器到 gaia.cs.umass.edu 服务器），另一个是从服务器到浏览器的 Response 响应信息。报文内容窗口显示了被选消息的一些细节信息。在这个例子中，HTTPOK 消息在报文显示窗口中高亮显示。试回忆 HTTP 消息是封装在 TCP 报文段中，TCP 报文段被装入 IP 数据报承载传输，而 IP 数据报又被装入一个以太网数据帧；Wireshark 可以显示出数据帧层，以太网层，IP 层，和 TCP 层等数据包信息。因为本实验只关注 HTTP，我们想尽量减少显示非 HTTP 数据，而其他协议会在后面实验中讨论，因此需要在上述各层信息域的最左侧设置带加号的小方框或者是向右指向的三角形，这样的配置表明信息隐藏或尚未显示，并且在 HTTP 那一行设置一个减号或者向下指向的三角形，即可显示出 HTTP 的所有信息。

查看 HTTP GET 和 Response 消息，请回答下列问题。回答问题时，要

求提供 GET 和 Response 消息的截图，而且标出在消息的什么位置找到了这些用于回答问题的信息。

- (1) 你的浏览器所使用的 HTTP 协议是 1.0 版还是 1.1 版？服务器运行的 HTTP 协议是什么版本？
- (2) 如果可能，你的浏览器标明其支持的什么语言，可以访问服务器？
- (3) 你的计算机 IP 地址是多少？gaia.cs.umass.edu 服务器的 IP 地址是什么？
- (4) 从服务器返回到你的浏览器的状态码是什么？
- (5) 你的浏览器下载的网页文件，在服务器上最后一次修改的时间是什么时候？
- (6) 返回到你浏览器上的文件内容有多少字节？
- (7) 在报文内容窗口查看原始数据的内容，你是否看到一些数据的报头并没有显示在报文列表窗口中？如果看到，请写出一个。

在回答的上述第 5 个问题，你可能会意外的发现那个下载的文档是在你下载该文档最近一分钟前刚刚被修改的。这是因为，对这个特定文件而言，gaia.cs.umass.edu 服务器每隔一分钟重复将文件的最近修改时间设定为当前时间。因此，如果你在访问服务器期间多等待一分钟再下载这个文件，那么该文件就会显示最近被修改过，因而你的浏览器就会下载一个该文档的“新”版本。

2、有条件的 HTTP GET/Response 消息交互

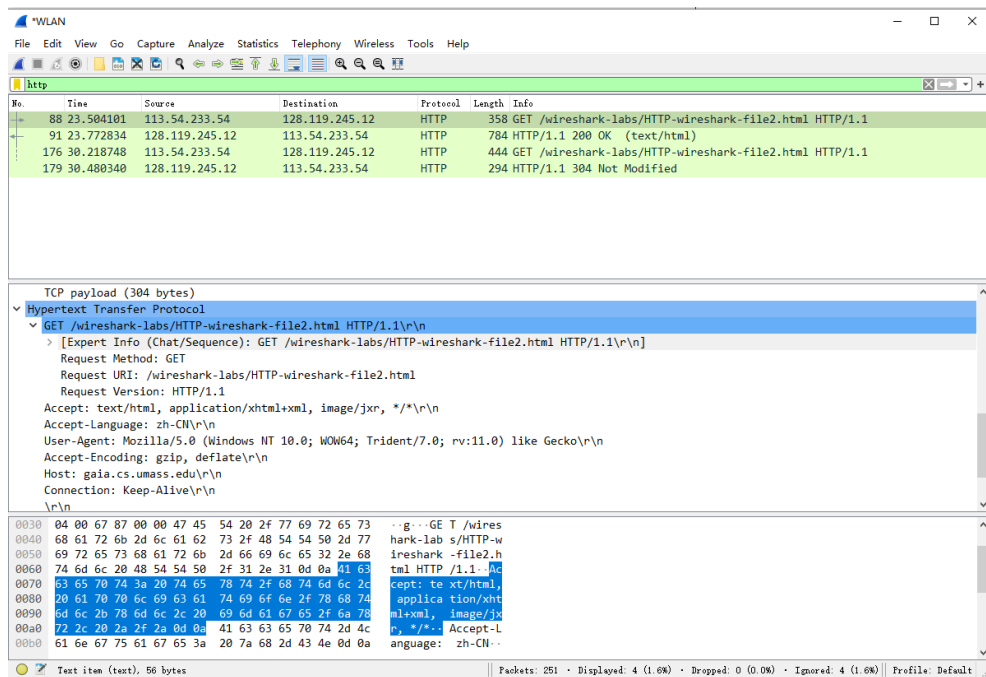
回顾在课本内容，大多数网页浏览器都缓存下载的网页对象，当下载一个网页对象时，执行有条件的 GET 操作。在执行下列操作之前，请确保已经清空你的浏览器缓存。（在火狐浏览器中，可选择工具->清空最近历史记录并清空缓存，或者在 IE 浏览器中，选择工具->Internet 选项->删除文件；这些操作会清空你的浏览器缓存）。现在完成下面的实验步骤：

- 打开网页浏览器，并确保你的浏览器已经按上述方法被清空缓存；
- 打开 Wireshark 软件；
- 在浏览器中键入如下 URL 地址，

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>

你的浏览器应该会显示一个非常简单且有 5 行文字的网页。

- 迅速在浏览器中再次输入同样的 URL（或者仅点击浏览器的刷新按钮）。
- 停止 Wireshark 捕获流量，然后在显示过滤器参数窗口中输入“http”，在报文列表窗口仅仅显示 HTTP 消息；



回答下列问题：

- (8) 查看从你的浏览器发给服务器的第一个 HTTP GET 请求消息的内容。你是否在 HTTP GET 消息中，看到一行“IF-MODIFIED-SINCE”文字？
- (9) 查看服务器响应消息的内容，服务器是否明确发送了文件内容？你是怎么辨别？
- (10) 现在查看从你浏览器发给服务器的第二个 HTTP GET 请求消息的内容。在 HTTP GET 中，你是否看到一行“IF-MODIFIED-SINCE”文字？如果看到，那么在“IF-MODIFIED-SINCE”报头字段之后紧跟的是什么信息？
- (11) 服务器响应第二个 HTTP GET 请求后返回的 HTTP 状态码是什么？服务器是否明确返回了文件内容吗？请解释。

3、获取长文档

在目前的实验例子中，我们下载文档都是既简单又短小的网页文件。下面，我们来看看下载一个长网页文件，将会发生什么。执行如下实验操作：

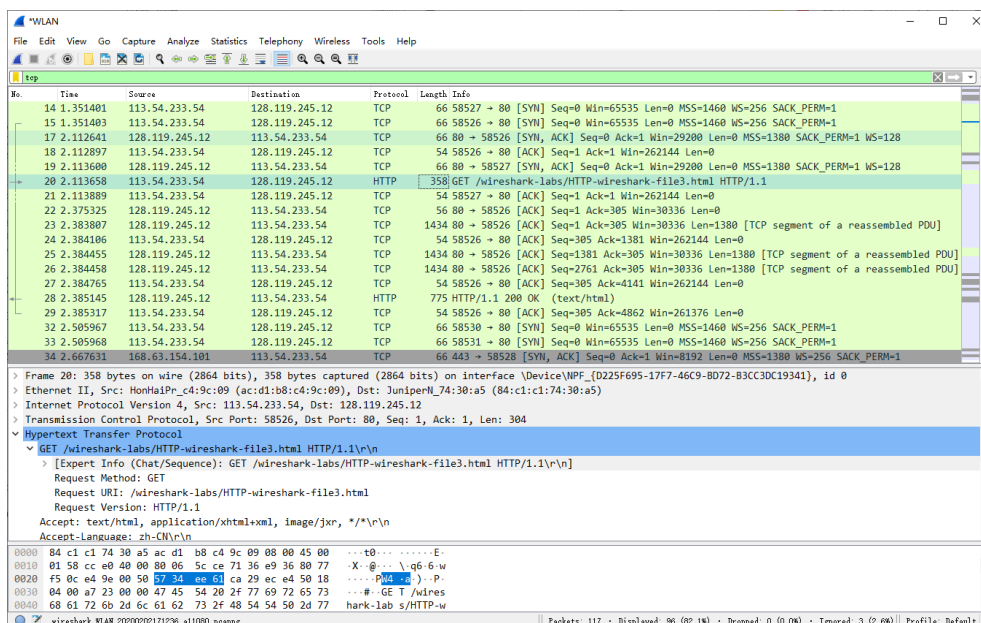
- 打开你的浏览器，并确保浏览器的缓存已按照之前介绍的方法被清空。
- 打开 Wireshark 软件。
- 在你的浏览器中键入下面的地址：

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

你的浏览器中会显示这篇相当长的美国人权法案文件。

- 停止 Wireshark 抓取流量，然后在显示过滤器参数窗口中输入“tcp”，在

报文列表窗口中只显示 HTTP、TCP 消息。



在报文列表窗口中，可看到 HTTP GET 请求消息，紧随其后的是一个响应此 HTTP GET 请求的多数据包 TCP Response 消息。这个多数据包响应消息值得更多笔墨解释。试回顾课本内容，HTTP 响应消息包括一个状态行，多个报头字段行，一个空行，最后跟着一个网页实体所组成。在这个 HTTP GET 请求的例子中，相应的实体部分是整个请求的网页文件。在这个例子中，这个网页文件相当长，而 4500 字节的长度对于一个 TCP 报文段来说太大了。单个的 HTTP 响应信息被 TCP 分割成几块，每一块数据由一个单独的 TCP 报文段承载（可参见课本图 1.24）。在最近几个版本中，Wireshark 将每个 TCP 报文段当作一个单独的数据包，将从一个单独的 HTTP 响应消息被分割成的多个 TCP 报文段，在 Wireshark 显示信息栏中标注为“单一 PDU 的重组 TCP 报文段”（TCP segment of a reassembled PDU）。Wireshark 早期版本使用“连续”字样来表示一个 HTTP 消息的整个内容被切割成多个 TCP 报文段。在此我们强调，HTTP 协议中没有“连续”消息。

以上截图中，HTTP GET 是数据包 No. 20，HTTP OK 响应是 No. 28。携带长文本的 HTTP 响应数据包是 23、25、26。注意数据包 24 是客户机到服务器的 TCP ACK 报文。

回答下列问题：

- (12) 你的浏览器发送了多少个 HTTP GET 请求信息？在抓取的流量中，哪个序号的数据包发出了下载人权法案的 GET 请求消息？
- (13) 在流量文件中，哪个序号的数据包承载了响应 HTTP GET 请求的返回消息的状态码以及状态信息？

(14)在响应消息中的状态码和状态信息是什么？

(15)HTTP 响应消息需要有多少个数据包,用以承载包含数据的 TCP 报文段,传输这篇人权法案文本？

4、 嵌有对象的网页文档

当浏览器下载有嵌入对象的文件,例如网页中嵌入有存贮在另外服务器的其他对象(如下面的例子所示,图片文件),我们将学习 Wireshark 如何分析相关流量。执行如下实验操作:

- 打开你的浏览器,并确保浏览器的缓存已按照之前介绍的方法被清空。
- 打开 Wireshark 软件。
- 在你的浏览器中键入下面的地址:

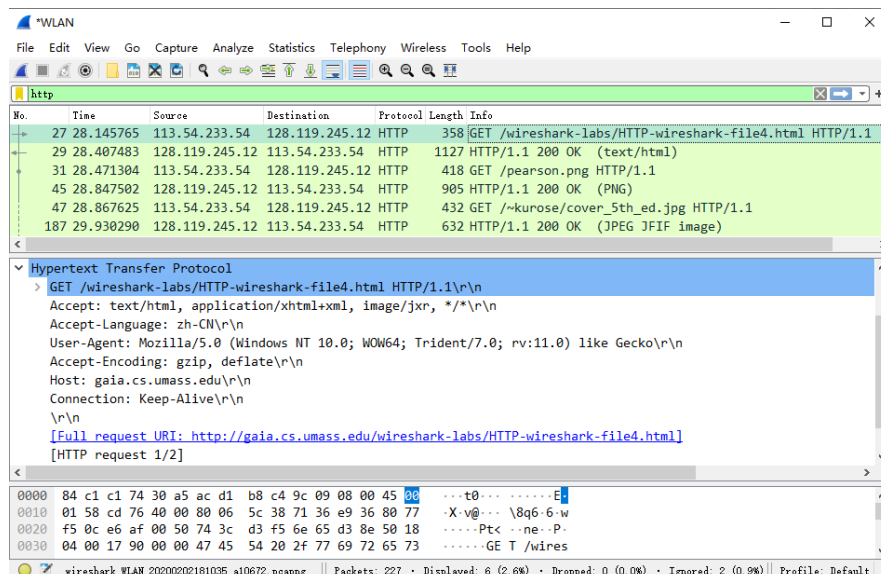
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>

或

<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file5.html>

你的浏览器中会显示嵌入有两张图片的一个短小的网页文件,该文件中引用了这两张图片,即图片本身并不属于这个网页文件,而是图片的地址嵌入到了网页文件中。正如课本中所讨论,你的浏览器将必须从图片地址指出的网站下载这些图标文件。

- 停止 Wireshark 抓取流量,然后在显示过滤器参数窗口中输入“http”,在报文列表窗口中只显示 HTTP 消息。



回答下列问题:

(16)你的浏览器发送了多少个 HTTP GET 请求信息? 这些 GET 请求消息发送到了那些互联网地址?

(17) 你是否能够判断你的浏览器是顺序依次下载的这两个图片文件，亦或是并行从两个网站下载？请解释。

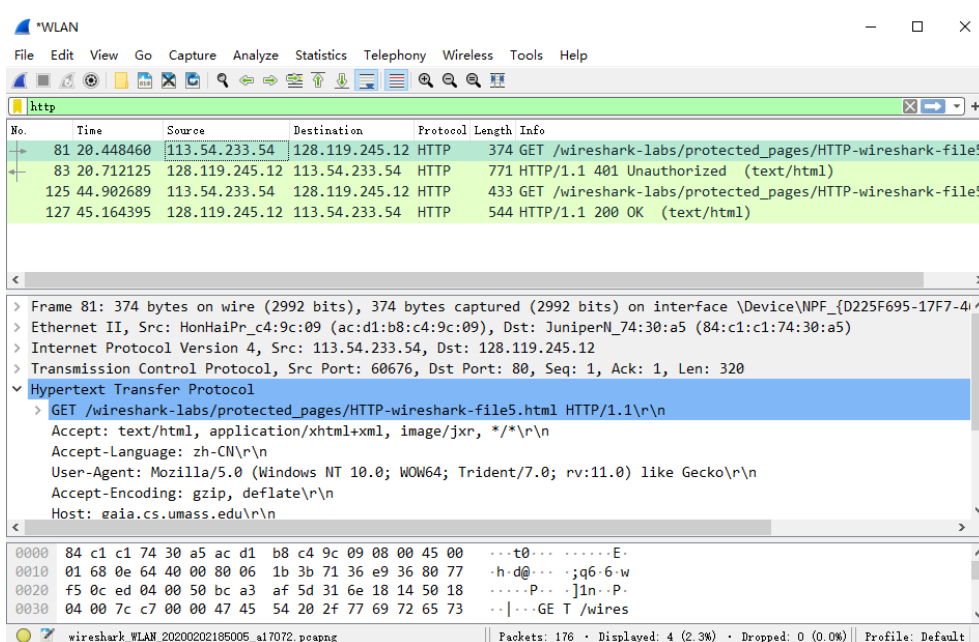
5、HTTP 认证机制

最后，我们尝试访问一个受密码保护的网站，并研究浏览器访问这样一个网站所生成的 HTTP 消息序列。这个 URL 地址，

http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html

是由密码保护的，用户名为“wireshark-students”（去掉引号），密码为“network”（同样去掉引号）。那么我们来访问这个受密码保护的所谓“安全”网站。执行下面的实验步骤：

- 按前述方法，确保清空浏览器的缓存，再关闭浏览器。然后，重新打开浏览器。
- 打开 Wireshark 软件。
- 在浏览器中键入如下 URL：
http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html
- 在弹出的窗口中，键入用户名和密码。
- 停止 Wireshark 抓取流量，然后在显示过滤器参数窗口中输入“http”，在报文列表窗口中只显示 HTTP 消息。



回答下列问题：

- (18) 对应于你的浏览器发出的最初 HTTP GET 请求消息，服务器发回的响应消息是什么（状态码或状态信息）？
- (19) 当你的浏览器第二次发送 HTTP GET 请求消息时，这个 GET 消息中包含了哪些新的字段？

八、实验结果与分析（含重要数据结果分析或核心代码流程分析）

- 1、 你的浏览器所使用的 HTTP 协议是 1.0 版还是 1.1 版？服务器运行的 HTTP 协议是什么版本？

```
Hypertext Transfer Protocol
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n]
Request Method: GET
Request URI: /wireshark-labs/HTTP-wireshark-file1.html
Request Version: HTTP/1.1
```

```
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
Response Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
```

答：浏览器与服务器使用的均为 HTTP 1.1 版本。

- 2、 如果可能，你的浏览器标明其支持的什么语言，可以访问服务器？

```
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
\r\n
```

答：支持 zh-CN 和 en。

- 3、 你的计算机 IP 地址是多少？ gaia.cs.umass.edu 服务器的 IP 地址是什么？

No.	Time	Source	Destination	Protocol	Length	Info
94	13.919989	192.168.3.7	128.119.245.12	HTTP	546	GET /wireshark-l
109	14.278407	128.119.245.12	192.168.3.7	HTTP	552	HTTP/1.1 200 OK
111	14.344743	192.168.3.7	128.119.245.12	HTTP	478	GET /favicon.ico
114	14.693753	128.119.245.12	192.168.3.7	HTTP	550	HTTP/1.1 404 Not

答：本机 IP 地址为 192.168.3.7，服务器 IP 地址为 128.119.245.12。

- 4、 从服务器返回到你的浏览器的状态码是什么？

```
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
Response Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
```

答：状态码为“200 OK”。

- 5、你的浏览器下载的网页文件，在服务器上最后一次修改的时间是什么时候？

```
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod_perl/2.0.11 Perl/v5.16.3\r\n
Last-Modified: Sun, 19 Apr 2020 05:59:01 GMT\r\n
ETag: "80-5a39e7aa79c21"\r\n
```

答：在服务器端的最后修改时间为：Sun, 19 Apr 2020 05:59:01 GMT。

- 6、返回到你浏览器上的文件内容有多少字节？

```
Accept-Ranges: bytes\r\n
Content-Length: 128\r\n
Keep-Alive: timeout=5, max=100\r\n
```

答：返回文件内容有 128 字节。

- 7、在报文内容窗口查看原始数据的内容，你是否看到一些数据的报头并没有显示在报文列表窗口中？如果看到，请写出一个。

答：没有看到任何报头。

- 8、查看从你的浏览器发给服务器的第一个 HTTP GET 请求消息的内容。你是否在 HTTP GET 消息中，看到一行“IF-MODIFIED-SINCE”文字？

```
Hypertext Transfer Protocol
GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
[Expert Info (Chat/Sequence): 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
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
\r\n
```

答：第一个 GET 消息中无“IF-MODIFIED-SINCE”文字。

- 9、查看服务器响应消息的内容，服务器是否明确发送了文件内容？你是怎么辨别？

```
Line-based text data: text/html (10 lines)
\r\n
<html>\r\n
\r\n
Congratulations again! Now you've downloaded the file lab2-2.html. <br>\r\n
This file's last modification date will not change. <p>\r\n
Thus if you download this multiple times on your browser, a complete copy <br>\r\n
will only be sent once by the server due to the inclusion of the IF-MODIFIED-SINCE<br>\r\n
field in your browser's HTTP GET request to the server.\r\n
\r\n
</html>\r\n
```

答：明确发送了文件内容，因为有“Line-based text data”行。

- 10、现在查看从你浏览器发给服务器的第二个 HTTP GET 请求消息的内容。在 HTTP GET 中，你是否看到一行“IF-MODIFIED-SINCE”文字？如果看到，那么在“IF-MODIFIED-SINCE”报头字段之后紧跟的是什么信息？

```
If-Modified-Since: Sun, 19 Apr 2020 05:59:01 GMT\r\n\r\n[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

答：在第二个 GET 消息中可以看到一行“IF-MODIFIED-SINCE”文字。客户端第二次请求此 URL 时，根据 HTTP 协议的规定，浏览器会向服务器传送 If-Modified-Since 报头，询问该时间之后文件是否有被修改过，所以其后紧跟的是本地浏览器存储的文件修改时间。

- 11、服务器响应第二个 HTTP GET 请求后返回的 HTTP 状态码是什么？服务器是否明确返回了文件内容吗？请解释。

```
Hypertext Transfer Protocol
  HTTP/1.1 304 Not Modified\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 304 Not Modified\r\n]
    Response Version: HTTP/1.1
    Status Code: 304
    [Status Code Description: Not Modified]
```

答：状态码为“304 Not Modified”；没有明确返回文件内容，因为报文段中无“Line-based text data”行。

- 12、你的浏览器发送了多少个 HTTP GET 请求信息？在抓取的流量中，哪个序号的数据包发出了下载人权法案的 GET 请求消息？

No.	Time	Source	Destination	Protocol	Length	Info
15	2.015596	192.168.3.7	128.119.245.12	HTTP	546	GET /wireshark-l
26	2.325816	128.119.245.12	192.168.3.7	HTTP	727	HTTP/1.1 200 OK
29	2.399402	192.168.3.7	128.119.245.12	HTTP	478	GET /favicon.ico

答：一共 2 个 GET 请求信息。序号为 15 的数据包发出了下载人权法案的 GET 请求消息。

- 13、在流量文件中，哪个序号的数据包承载了响应 HTTP GET 请求的返回消息的状态码以及状态信息？

20	2.319807	128.119.245.12	192.168.3.7	TCP	66	80 → 52902 [ACK]
21	2.324976	128.119.245.12	192.168.3.7	TCP	1466	80 → 52902 [ACK]

答：数据包 21。

- 14、在响应消息中的状态码和状态信息是什么？

```
0040 da d1 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f ..HTTP/1 .1 200 0
0050 4b 0d 0a 44 61 74 65 3a 20 53 75 6e 2c 20 31 39 K..Date: Sun, 19
0060 20 41 70 72 20 32 30 32 30 20 31 32 3a 35 38 3a Apr 2020 12:58:
0070 34 31 20 47 4d 54 0d 0a 53 65 72 76 65 72 3a 20 41 GMT..Server:
```

答：状态码和状态信息为“200 OK”。

- 15、 HTTP 响应消息需要有多少个数据包,用以承载包含数据的 TCP 报文段,传输这篇人权法案文本?

No.	Time	Source	Destination	Protocol	Length	Info
21	2.324976	128.119.245.12	192.168.3.7	TCP	1466	80 → 52902 [ACK]
22	2.325177	128.119.245.12	192.168.3.7	TCP	1466	[TCP Previous seq
23	2.325264	192.168.3.7	128.119.245.12	TCP	78	52902 → 80 [ACK]
24	2.325472	128.119.245.12	192.168.3.7	TCP	1466	[TCP Out-of-Order
25	2.325538	192.168.3.7	128.119.245.12	TCP	66	52902 → 80 [ACK]

答: 需要 21、22、24 号,共 3 个数据包来传输这篇文本。

- 16、 你的浏览器发送了多少个 HTTP GET 请求信息? 这些 GET 请求消息发送到了那些互联网地址?

No.	Time	Source	Destination	Protocol	Length	Info
101	9.204040	192.168.3.7	128.119.245.12	HTTP	546	GET /wireshark-l
108	9.522673	128.119.245.12	192.168.3.7	HTTP	1139	HTTP/1.1 200 OK
111	9.551464	192.168.3.7	128.119.245.12	HTTP	478	GET /pearson.png
155	9.860841	128.119.245.12	192.168.3.7	HTTP	877	HTTP/1.1 200 OK
159	9.862613	192.168.3.7	128.119.245.12	HTTP	492	GET /~kurose/cov
349	11.147323	128.119.245.12	192.168.3.7	HTTP	584	HTTP/1.1 200 OK
352	11.153888	192.168.3.7	128.119.245.12	HTTP	478	GET /favicon.ico
354	11.507099	128.119.245.12	192.168.3.7	HTTP	550	HTTP/1.1 404 Not

答: 发送了 4 个 HTTP GET 请求。这四个请求首先发送到了服务器,IP 地址为 128.119.245.12; 后三个请求的实际请求 URL 分别为 <http://gaia.cs.umass.edu/pearson.png>、
http://manic.cs.umass.edu/~kurose/cover_5th_ed.jpg、
<http://gaia.cs.umass.edu/favicon.ico>。

- 17、 你是否能够判断你的浏览器是顺序依次下载的这两个图片文件,亦或是并行从两个网站下载? 请解释。

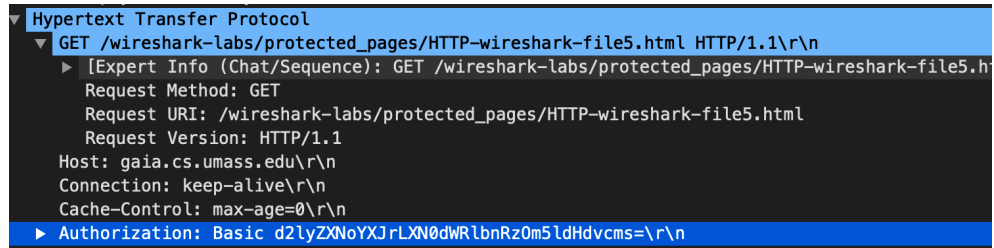
答: 并行下载。获取两个图片文件的 GET 请求的数据包序号分别为 111、159, 而 GET 请求是并行发出的, 且两个图片文件的实际资源地址不相同, 因此并行下载, 且相应各自 GET 的“200 OK”数据包跟在对应的 GET 请求之后。

- 18、 对应于你的浏览器发出的最初 HTTP GET 请求消息, 服务器发回的响应消息是什么(状态码或状态信息)?

Hypertext Transfer Protocol	
HTTP/1.1 401 Unauthorized\r\n	
▶ [Expert Info (Chat/Sequence): HTTP/1.1 401 Unauthorized\r\n]	
Response Version: HTTP/1.1	
Status Code: 401	
[Status Code Description: Unauthorized]	

答: 响应消息为“401 Unauthorized”。

- 19、 当你的浏览器第二次发送 HTTP GET 请求消息时，这个 GET 消息中包含了哪些新的字段？



```
Hypertext Transfer Protocol
  GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.h
      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 d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcm5l\r\n
```

答：包含了“Authorization: Basic”字段。该字段的值“d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcm5l”实际上是输入的用户名（wireshark-students）和密码（network）基于 Base64 格式的简单编码，而用户名和密码实际上并没有加密。

九、总结及心得体会：

本次实验使用 Wireshark 分析了 HTTP GET/Response 基本消息交互、HTTP 获取长文档机制、网页中嵌入有对象（图片）的获取流程，了解了 HTTP 的认证机制。掌握了网络协议分析工具 Wireshark 的基础使用方法及应用，掌握了分析 HTTP 网络协议的方法与技巧，并对教材上讲授的有关 HTTP、TCP 协议的理论知识有了更深刻的认识。

十、对本实验过程及方法、手段的改进建议：

本次实验还可以使用 Wireshark 对其他网站进行抓包分析，这样更能激发学生的动手实践兴趣，体现《计算机网络系统》课程实践性强的特点，让学生充分讲理论与实践相结合，加深对相关网络协议、数据通信过程的理解与掌握。

报告评分：

指导教师签字：