实验1 网络协议分析工具Wireshark的使用

实验所属系列：《计算机网络系统》课内实验 实验对象：本科

相关课程及专业：软件工程 实验时数（学分）：**2**学时

实验类别：课内上机

实验开发教师：计算机网络系统课程组

**【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。

**【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。

前置知识：了解Wireshark基本设置方法，熟悉利用Wireshark抓取本机网络通信数据包的基本步骤，熟悉过滤捕获结果的操作技巧。

**【实验环境】**

安装Windows操作系统的PC机1台，装有Wireshark软件及Winpcap驱动。

**【实验设备】**

PC机1台。

**【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。

**【预备知识】**

Wireshark用户界面和基本操作。

**【实验步骤】**

**1.1 HTTP GET/Response基本消息交互**

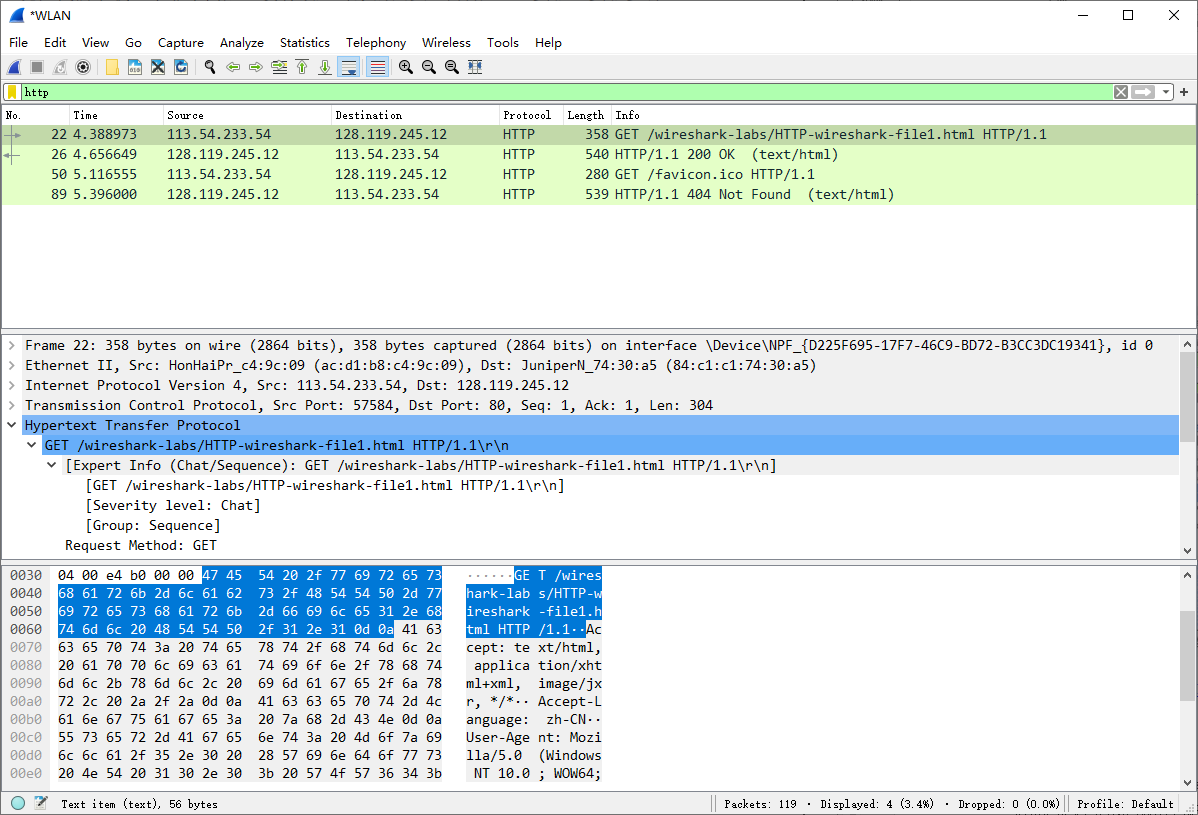
首先下载一个非常简短且没有嵌入对象的网页，开始HTTP实验。实验步骤如下：

* 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。



**Figure1:** 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，HTTP OK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？



1. 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？



1. 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？



1. 从服务器返回到你的浏览器的状态码是什么？



1. 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？



1. 返回到你浏览器上的文件内容有多少字节？



1. 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。

**是的，没有看到任何报头。**

在回答的上述第5个问题时，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。

**1.2 有条件的HTTP GET/Response消息交互**

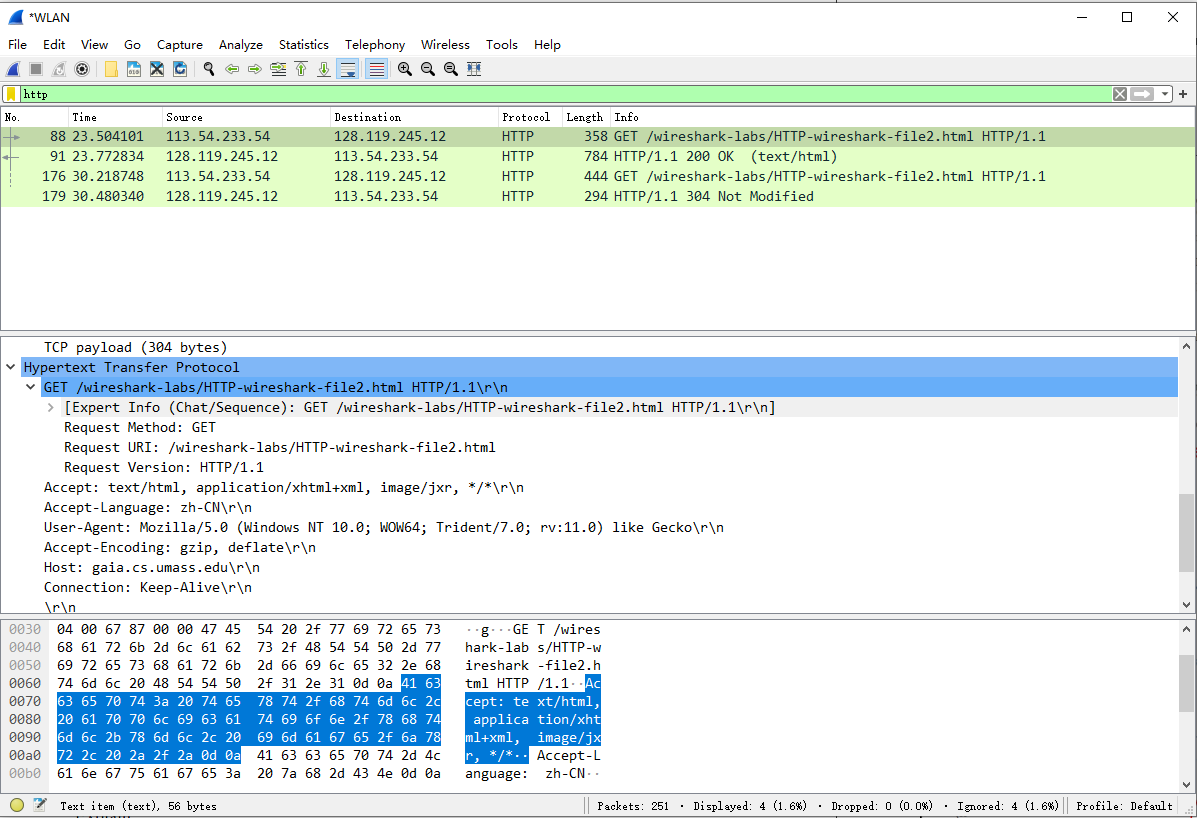
回顾在课本内容，大多数网页浏览器都缓存下载的网页对象，当下载一个网页对象时，执行有条件的GET操作。在执行下列操作之前，请确保已经清空你的浏览器缓存。（在火狐浏览器中，可选择工具->清空最近历史记录并清空缓存，或者在IE浏览器中，选择工具->Internet选项->删除文件；这些操作会清空你的浏览器缓存）。现在完成下面的实验步骤：

* 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；

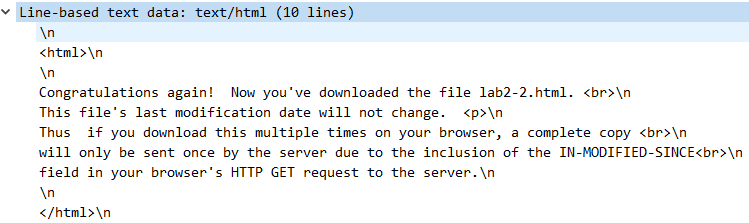


回答下列问题：

1. 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？

**第一个HTTP GET消息中无“IF-MODIFIED-SINCE”，第二个有。**

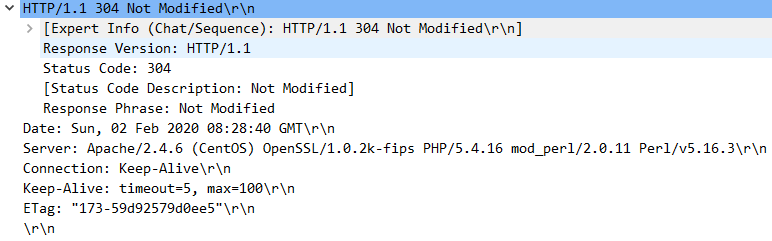
1. 查看服务器响应消息的内容，服务器是否明确发送了文件内容？你是怎么辨别？



1. 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？



1. 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。



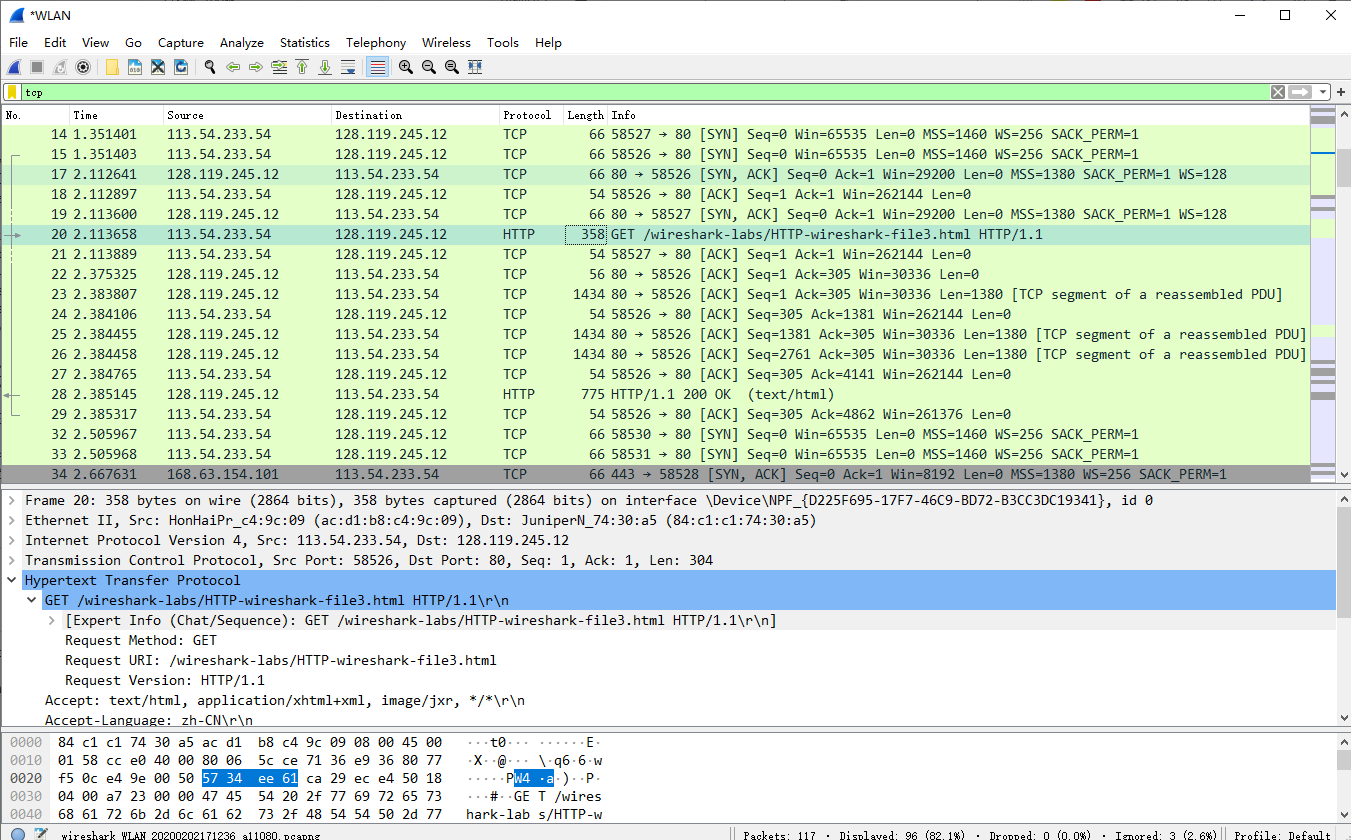
**1.3 获取长文档**

在目前的实验例子中，我们下载的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HTTPGET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。**

回答下面的问题：

1. 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？

**浏览器发送了1个HTTP GET请求信息。在抓取的流量中，序号为20的数据包发出了下载人权法案的GET请求消息。**

1. 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？

**数据包23。**

1. 在响应消息中的状态码和状态信息是什么？

200 OK。

1. 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？

**三个数据包，23、25、26。**

**1.4 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：

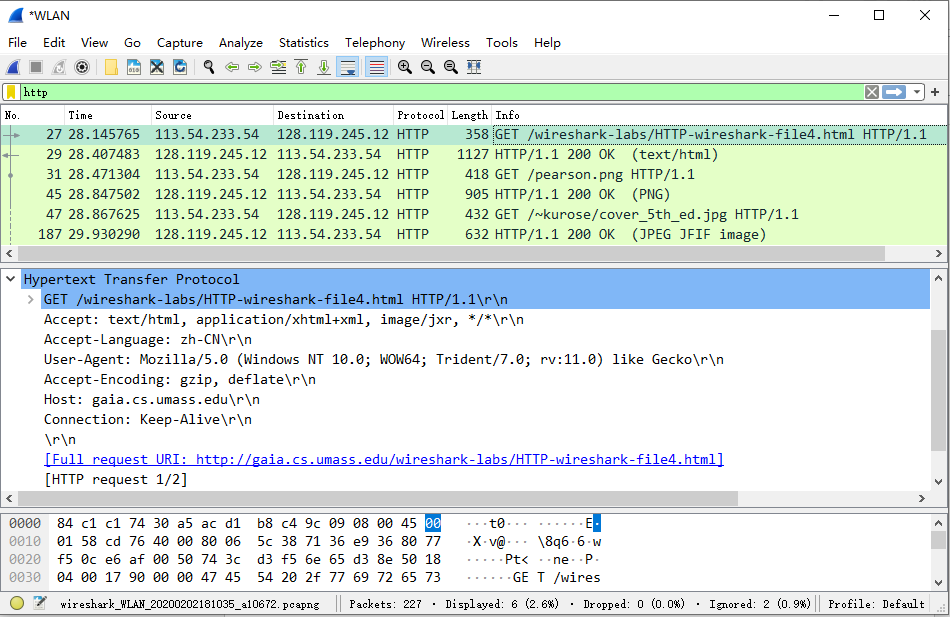
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>

或

[http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file5.html](http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.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。

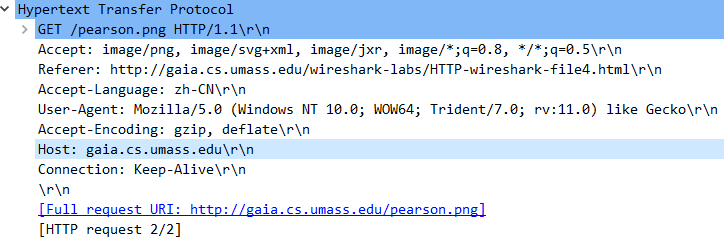
* 停止Wireshark抓取流量，然后在显示过滤器参数窗口中输入”http”， 在报文列表窗口中只显示HTTP消息。

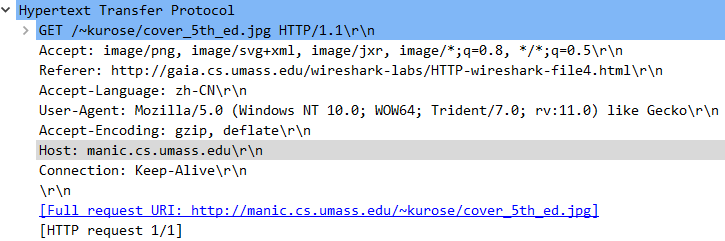


回答下面的问题：

1. 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？

**3个HTTP GET消息。这些消息表面上都发送到目的地址128.119.245.12，但后两个HTTP GET的实际请求RUI是：**





1. 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。

**并行下载。获取两个图片的HTTP GET分别是31、47，但响应31的HTTP OK是45，在数据包47之前就已经发送。**

**1.5 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，

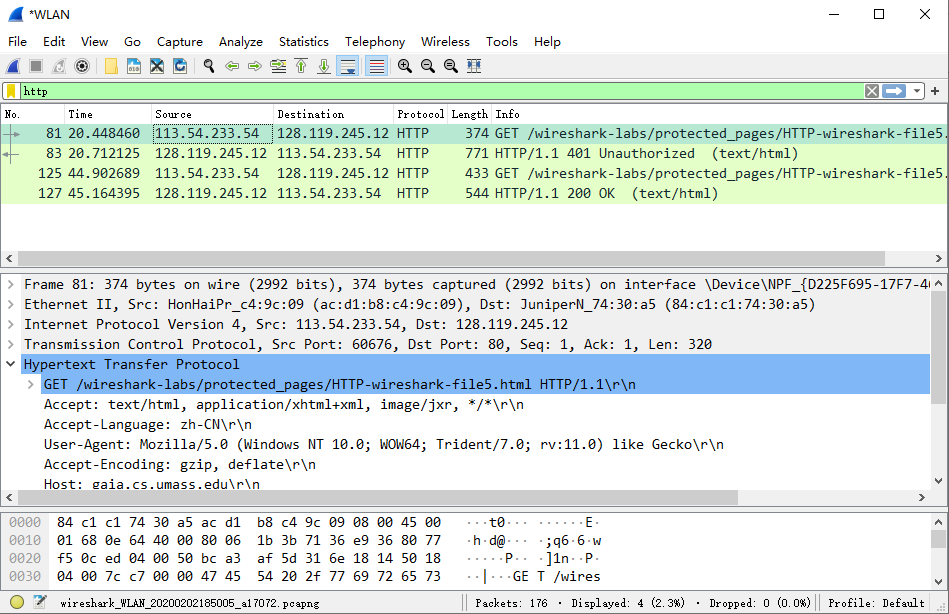
<http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html>

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。



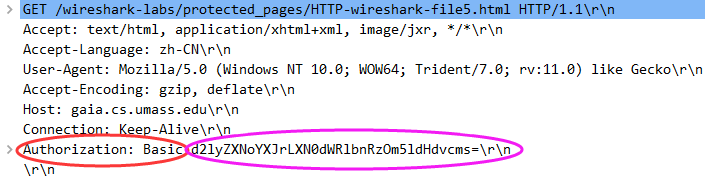
回答下列问题：

1. 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）？

**数据包81是最初的HTTP GET，数据包83是服务器的REPLY。服务器发回的响应消息是401 Unauthorized。**

1. 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？

**包含了Authorization: Basic。**



你输入的用户名（**wireshark-students**）和密码（**network**），是以下面这样一串字符形式（**d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=**），附在客户端发出的HTTP GET消息中“**Authorization:Basic**”报文头之后。虽然你输入的用户名和密码看上去经过加密，但是它们仅仅是进行了基于Base64格式的简单编码，而用户名和密码实际上并没有加密！如果想验证这一点，请登录http://www.motobit.com/util/base64-decoder-encoder.asp，然后输入这个Base64编码的字符串**d2lyZXNoYXJrLXN0dWRlbnRz** 并解码。这个字符串由Base64编码翻译为了ASCII编码，因而可以看到你的用户名！再输入那个字符串的剩余部分**Om5ldHdvcms=**，然后点击解码，即可查看密码。由于任何人都可以下载一个类似Wireshark这样的抓包工具来抓取经过他们网络适配器的数据包(并不只包括他们自己的流量)，并且任何人都可将Base64编码转化成ASCII编码 (就像你刚才做的那样)。因此，你需要明确WWW站点上的那些简单密码并不安全，除非进一步采取了其他安全措施。

**【注意事项】**

1、如果机器有多个网卡，则使用Wireshark时应选择适当的网卡或者在捕获时用ether host MAC地址指定，MAC地址中各个数字间可以用冒号或者短横线分隔。

2、在Wireshark中也可以预先不设定捕获的过滤字符串，即默认抓取所有包。待抓取停止后，再利用Filter表达式来提取显示需要的内容，但其语法和捕获前设定过滤字符串的语法有区别。

3、为简化和清晰起见，在分析时应忽略那些错误的或重传的数据包。

**【实验报告】**

1．说明捕获抓包过程。

2．完成实验步骤中的问题并进行结果分析。