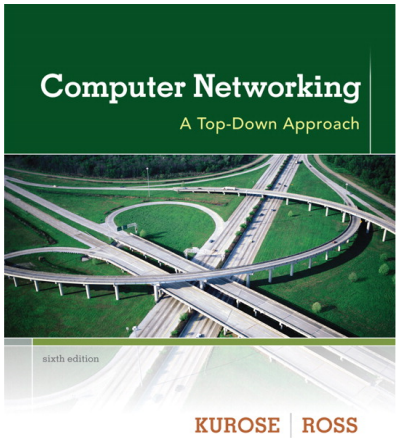
Wireshark 实验: HTTP v6.1



Computer Networking: A Top-Down Approach, 6th ed.,
J.F. Kurose and K.W. Ross

"不闻不若闻之，闻之不若见之；见之不若知之，知之不若行之；学至于行而止矣。"

——《荀子•儒效篇》

翻译：张瑶、谢小芳、朱媚、刘丹丹、胡雨豪
审校：秦大力

我们已经在 Wireshark Introductory 实验（Wireshark Lab: Getting Started v6.0，文件名：Wireshark_Intro_v6.0.doc）中学习了 Wireshark 分组嗅探器，现在我们准备使用 Wireshark 来研究一些协议。在本实验中，我们将探讨 HTTP 协议的几个方面：基本的 GET/response 交互、HTTP 报文格式、获取一个 HTML 大文件、获取带嵌入对象的 HTML 文件以及 HTTP 身份验证与安全。在开始这些实验之前,可以先回顾一下教材的第 2.2 节。[1]

一、 基本的 HTTP GET/response

让我们开始探索 HTTP，先下载一个非常简单的 HTML 文件，这个文件很短且不包含任何嵌入的对象。请执行以下操作：

- 启动 web 浏览器。
- 按照 Introductory 实验描述的方法启动 Wireshark（注意：此时还不要开始捕获）。在 display-filter-specification 文本框输入“http”（仅字母，不包括引号），这样只有捕获到的 HTTP 报文将会显示在 packet-listing 窗口（在这里我们只对 HTTP 协议感兴趣，而不想看到所有捕获到的数据包）。
- 等待一分多钟（我们很快就会明白为什么），然后开始捕获。
- 输入以下网址到你的浏览器
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
  你的浏览器应该会显示一个非常简单的、仅一行文本的 HTML 文件。
- 停止 Wireshark 捕获。

[1] 请参考教材中的章节和图表：*Computer Networks, A Top-down Approach, 6th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2012.*

你的 Wireshark 窗口应类似于图 1 所示。如果你没有合适的互联网连接来运行 Wireshark 并完成上述抓包过程，也可以下载一个根据上面的抓包过程步骤创建的跟踪文件。[2]

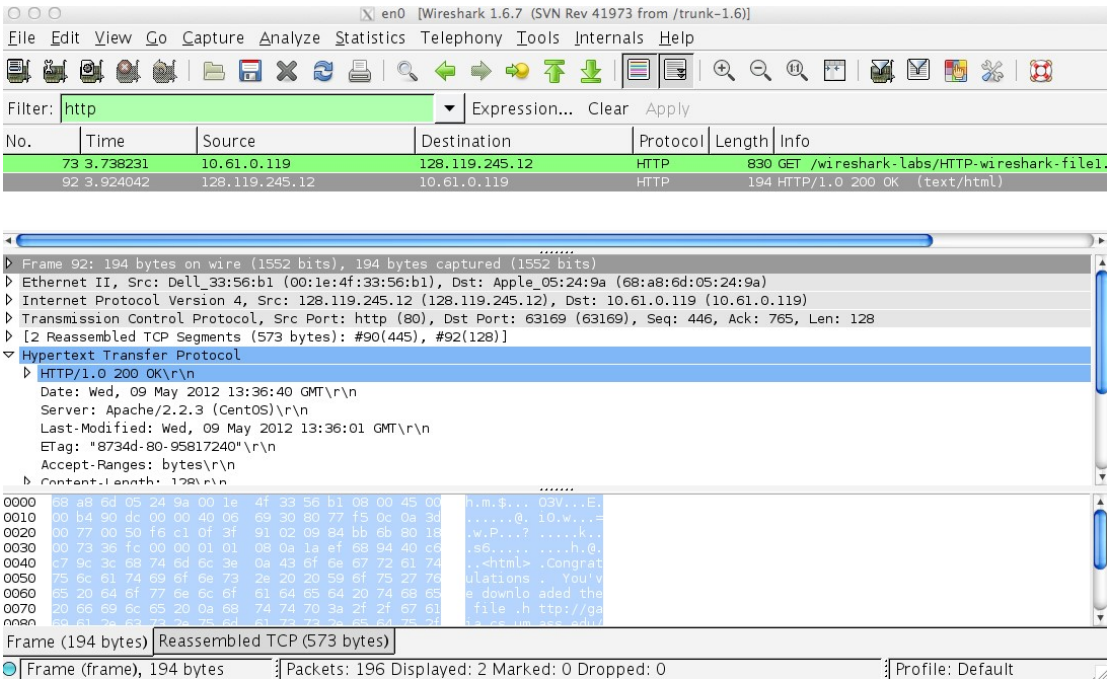


图 1：在浏览器获取了 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html 之后的 WireShark 界面

图 1 中的示例在 packet-listing 窗口中显示了两个捕获的 HTTP 报文：GET 报文（从浏览器到 gaia.cs.umass.edu web 服务器）和从服务器返回浏览器的响应报文。数据包内容（packet-contents）窗口显示当前所选择报文（即图 1 中的 packet-listing 窗口高亮显示的 HTTP OK 报文）的细节内容。回想一下,HTTP 报文包含在一个 TCP segment 内，而 TCP segment 包含在一个 IP 数据报中，IP 数据报则包含在一个以太网帧内，因此除了 HTTP 报文之外，Wireshark 还可以逐层显示帧、以太网帧、IP 数据报、TCP 报文段的信息。我们希望尽量不显示非 http 数据（这里我们仅对 HTTP 感兴趣，并将在以后的实验中研究其他协议），所以要确保 packet-contents 窗口中 Frame、Ethernet、IP 以及 TCP 行最左边的方块上显示一个加号或

[2] 下载 zip 文件（http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip）并提取 trace 文件 http-ethereal-trace-1，这个 trace 文件是在作者的电脑上运行 Wireshark 并执行抓包过程得到的结果。一旦你下载了 trace 文件，就可以将其加载到 Wireshark 并查看抓包的结果（使用 *File* 下拉菜单,选择 *Open*，然后选择 trace 文件 http-ethereal-trace-1）。这种方式的效果也类似于图 1。(在不同的操作系统或不同版本的 Wireshark 上，Wireshark 用户界面会有少许差异)

者指向右边的小三角形，而 HTTP 行最左边的方块上显示一个减号或者向下的小三角形（这意味着有关 HTTP 报文的所有信息将被显示）。

（注意：你应该忽略任何对图标文件对象 favicon.ico 的 HTTP GET 请求和响应。针对这个文件对象的请求是由浏览器自动发出的，在本实验中我们将忽略该文件。实际上，favicon.ico 图标是网站的缩略标志，可以显示在浏览器标签、地址栏左边和收藏夹中。通常情况下，网页开发者都应该提供该文件。）

通过查看 HTTP GET 和响应报文中的信息，回答下列问题。回答以下问题时，你应该打印出 GET 请求和响应报文（参见 Wireshark Introductory 实验中的关于如何做到这一点的解释），并指出回答问题的过程中你是在哪里找到相关信息的。提交实验报告时，请给出相关信息在 HTTP 报文当中的位置（例如：某问题的相关信息位于某个首部行，则注明该首部行的 Key 即可）。

1. 你的浏览器运行 HTTP 版本 1.0 还是 1.1？服务器运行的 HTTP 版本呢？
2. 浏览器显示可以接受服务器提供的哪种自然语言（如果有的话）？
3. 你的计算机和 gaia.cs.umass.edu 服务器 IP 地址分别是什么?
4. 服务器返回给浏览器的状态码是什么？
5. 请写出你得到的 HTML 文件在服务器端的最后修改时间。
6. 在 HTTP 响应报文中，找到返回给浏览器的内容的字节数。
7. 检查数据包内容窗口中的原始数据,能够从这些原始数据中找到任何没有在 packet-listing 窗口中显示的首部行吗？如果有的话,请写出其中的一个。

在回答上面 5 个问题的过程中，你可能会惊奇地发现，你获得的文档是在下载文档的前一分钟内被最后修改的。这是因为（对于这个特定的文件）gaia.cs.umass.edu 服务器每分钟都修改文件的最后修改时间，也就是将文件的最后修改时间设置为当前时间。因此，如果你在两次连续访问同一文件的过程中等待一分多钟，那么文件的最后修改时间将会发生变化，因此你的浏览器会下载一个“新”文件的副本。

二、 HTTP 条件 GET/响应

回忆一下教材第 2.2.6 节的内容，大多数 web 浏览器在检索 HTTP 对象时，会进行对象缓存，也就是条件 GET。执行以下步骤之前,请确保你的浏览器缓存是空的（例如在 Firefox 中，选择 *Tools → Clear Recent History*，并勾选 *Cache* 复选框；在 Internet Explorer 中,可以选择*工具 → Internet 选项 → 删除...*。这些操作将从清空你的浏览器缓存) 。接下来完成以下步骤：

- 启动 web 浏览器,并使用上面描述的方法清空你的浏览器缓存。
- 启动 Wireshark

- 在浏览器中输入以下网址
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
  浏览器应该显示一个非常简单的、有五行文本的 HTML 文件。
- 在浏览器中快速再次输入同样的网址(或简单地点击浏览器的刷新按钮)。
- 停止捕获,并在 display-filter-specification 窗口输入“http”，以便只有捕捉到的 HTTP 报文会显示在 packet-listing 窗口中。
- （注意：如果你不能上网，也可以在 Wireshark 中使用我们提供的 trace 文件 http-ethereal-trace-2 来回答下面的问题；请参见脚注 2，该 trace 文件是在作者的电脑上执行上述步骤的结果。)

请回答下列问题：

8. 检查第一个浏览器发送给服务器的 HTTP GET 请求的内容，在这个 HTTP GET 中，你看到了“IF-MODIFIED-SINCE”行吗?
9. 检查服务器响应的内容。服务器明确地返回了文件内容吗？你是如何知道的？
10. 现在检查第二个 HTTP GET 请求的内容。你能找到 HTTP GET 中的“IF-MODIFIED-SINCE”行吗？如果可以，那么“IF-MODIFIED-SINCE”首部行的值是什么？
11. 在针对第二个 HTTP GET 的响应报文中，服务器返回的 HTTP 状态码和短语是什么？这一次，服务器返回了请求的文件内容吗？请解释一下。

三、 获取大文件

到目前为止，我们的示例中所获取的文件都是很简短的 HTML 文件。接下来我们来看看，当我们下载一个大的 HTML 文件时会发生什么。请完成以下操作：

- 启动 Web 浏览器，并确保浏览器的缓存已清除，如上所述。
- 启动 Wireshark
- 在浏览器中输入以下 URL
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
  浏览器将会显示美国权利法案，这个文件是相当大的一个文本。
- 停止捕获，并在 display-filter-specification 窗口中输入“http”，以便只显示捕获的 HTTP 报文。
- （注意：如果无法在实时网络连接上运行 Wireshark，则可以使用 trace 文件 http-ethereal-trace-3 来回答以下问题，参见脚注 2。）

在 packet-listing 窗口中，应该可以找到 HTTP GET 报文，然后是针对该请求的多个 TCP 响应数据包。这里需要解释一下：回忆一下教材的第 2.2 节（见教材中的

图 2.9），HTTP 响应报文由状态行、首部行、一个空行以及实体主体组成。在本实验小节中，响应报文中的实体正是浏览器所请求的整个 HTML 大文件。但这个 HTML 文件相当长，有 4500 字节且无法放在一个 TCP 数据包当中。因此，这个 HTTP 响应报文被 TCP 分成了几个部分，每个部分再分别封装在单独的 TCP 段中（参见教材中的图 1.24）。在 Wireshark 的最新版本中，Wireshark 将每个 TCP 段划定为单独的数据包，而实际上，单个 HTTP 响应报文被分割在多个 TCP 报文段的这种情形，可以在 TCP 段 Info 列中的“TCP segment of a reassembled PDU”和 Wireshark 注释中看出来（见图 2）。Wireshark 的早期版本使用“Continuation”短语表示单个的 HTTP 报文被分割存放在多个 TCP 段中的情形。这里强调一下，HTTP 协议中是没有所谓的“Continuation”报文的！

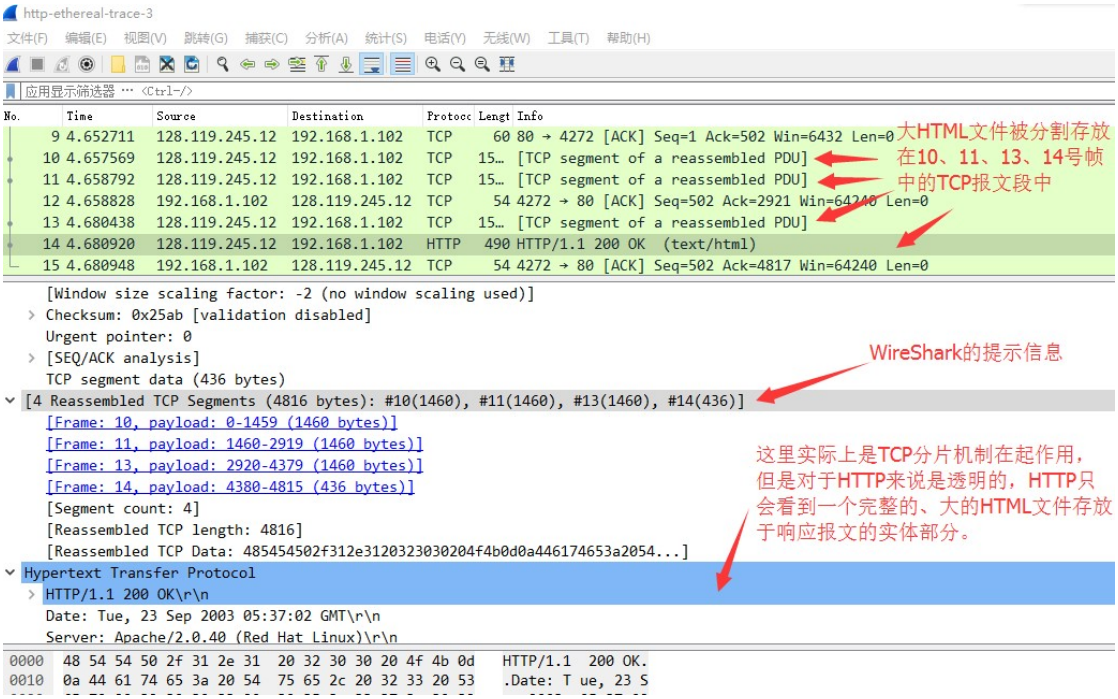


图 2：HTML 大文件在 TCP 中的分片

请回答下列问题：

12. 您的浏览器发送多少个 HTTP GET 请求报文？哪一个数据帧包含了请求获取权利法案（Bill of Rights）的 GET 报文，请写出该数据帧的序号（见 packet-listing 窗口的 No.列）？
13. 哪一个数据帧包含了针对 GET 请求的响应报文？请写出该数据帧的序号。
14. 响应报文中的状态码和短语是什么？
15. 需要多少个 TCP 段来传输包含权利法案文本的单个 HTTP 响应报文？

四、 包含嵌入对象的 HTML 文档

我们已经看到，对于 HTML 大文件，Wireshark 是如何显示捕获到的数据包。接下来我们看看，当浏览器下载一个包含嵌入对象的 HTML 文件时会发生什么情况。“包含嵌入对象的 HTML 文件”是指该文件包含了存储在其他服务器上的一些对象（在下面的示例中,“对象”是一些图像文件）。

先执行以下操作:

- 启动 web 浏览器,并确保你的浏览器的缓存清除,正如上面所讨论的。
- 启动 Wireshark
- 输入以下网址到浏览器
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
  浏览器应该显示一个简短的 HTML 文件和两个图片，这两个图片是在 HTML 文件中被引用的。也就是说，图片本身并不包含在 HTML 文件中，而是在 HTML 文件中包含了图像的 URL。正如课本中所讨论的，浏览器必须从 URL 所指定的网站获取这些图片。其中一张图片是本书出版商的标志，从 www.aw-bc.com 获取；另一张图片是本书第五版的封面（这是我最喜欢的封面)，存储在 manic.cs.umass.edu 服务器中。
- 停止捕获并在 display-filter-specification 窗口输入“http”，以便只显示捕获的 HTTP 报文。
- （注意：如果不能上网，可以使用 trace 文件 http-ethereal-trace-4 来回答下面的问题，参见脚注 2）

请回答下列问题：

16. 你的浏览器发送了多少个 HTTP GET 请求报文？这些 GET 请求被发送到哪一个 IP 地址？
17. 你的浏览器分别从两个网站下载图片时，是并行地下载还是一个接一个地下载的？请说明你的理由。

五、 HTTP 身份认证

最后，让我们尝试访问一个受密码保护的网站，并研究在访问过程中的所交换的 HTTP 报文。URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html 是一个用密码保护的页面，用户名为 wireshark-students，密码是 network。请访问该页面并执行以下操作：

- 如上所述，确保浏览器的缓存被清除，然后重启浏览器。
- 启动 Wireshark

- 在浏览器中输入以下 URL
  http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
  在弹出框中输入所需的用户名和密码。
- 停止捕获并在 display-filter-specification 窗口输入“http”，以便只显示捕获的 HTTP 报文。
- （注意：如果不能上网，可以使用 trace 文件 http-ethereal-trace-5 来回答下面的问题，参见脚注 2）

现在让我们来检查 Wireshark 的输出信息。建议先阅读一下补充材料“HTTP Authentication Schemes”或者《HTTP 权威指南》的第 12 章“基本认证机制”。

请回答下列问题：

18. 第一个 HTTP GET 请求报文的服务器响应（状态码和状态短语）是什么？
19. 当浏览器第二次发出 HTTP GET 报文时，该报文中新添加的字段是什么？

在客户端的 HTTP GET 报文中的“Authorization”首部行,你输入的用户名和密码（wireshark-students 和 network）被编码为“Basic”之后的字符串（d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=）。虽然看起来用户名和密码被加密了，但该字符串只是以一种称为 Base64 的格式进行了编码,而并**没有**被加密！你可以访问 http://www1.tc711.com/tool/BASE64.htm，并输入 base64 编码的字符串 d2lyZXNoYXJrLXN0dWRlbnRz 进行解码。哇！你会看到该字符串已从 Base64 编码转换为 ASCII 编码，因此可以看到用户名！而要查看密码，请输入字符串 Om5ldHdvcms=并解码。因为任何人都可以下载像 Wireshark 这样的工具，并通过网络适配器抓取数据包（不仅仅是他们自己的数据包），任何人都可以从 Base64 转换为 ASCII（就像你刚刚做的一样！），所以你应该知道 Web 网站上的简单密码并不安全，除非采取了一些其他的安全措施。

不要害怕！正如我们将在第 8 章中看到的，存在许多使 WWW 访问更安全的方法。很明显,我们将需要超出 HTTP 基本认证框架的更多安全手段或方法！