

- 404 Not Found: 被请求的文档不在服务器上。
- 505 HTTP Version Not Supported: 服务器不支持请求报文使用的 HTTP 协议版本。

你想看一下真实的 HTTP 响应报文吗？这正是我们高度推荐的事，而且也很容易做到。首先用 Telnet 登录到你喜欢的 Web 服务器上，接下来输入一个只有一行的请求报文去请求放在该服务器上的某些对象。例如，假设你看到命令提示，键入：

```
telnet cis.poly.edu 80
```

```
GET /~ross/ HTTP/1.1
Host: cis.poly.edu
```

(在输入最后一行后连续按两次回车。)这就打开一个到主机 cis.poly.edu 的 80 端口的 TCP 连接，并发送一个 HTTP 请求报文。你将会看到一个携带包括 Ross 教授主页的 HTML 基本文件的响应报文。如果你只是想看一下 HTTP 协议的报文行，而不是获取对象本身的话，那么可以用 HEAD 代替 GET。最后，用 /~banana/ 代替 /~ross/，看看你得到什么类型的响应报文。

在本节中，我们讨论了 HTTP 请求报文和响应报文中的一些首部行。HTTP 规范中定义了很多可以被浏览器、Web 服务器和 Web 缓存服务器插入的首部行。我们只提到了全部首部行中的少数几个，在 2.2.5 节中我们讨论网络 Web 缓存时还会涉及其他几个。一本可读性很强的文献是 [Krishnamurty 2001]，它对 HTTP 协议（包括它的首部行和状态码）进行了广泛讨论。

浏览器是如何决定在一个请求报文中包含哪些首部行的呢？Web 服务器又是如何决定在一个响应报文中包含哪些首部行呢？浏览器产生的首部行与很多因素有关，包括浏览器的类型和协议版本（例如，HTTP/1.0 浏览器将不会产生任何 1.1 版本的首部行）、浏览器的用户配置（如喜好的语言）、浏览器当前是否有一个缓存的但是可能超期的对象版本。Web 服务器的表现也类似：在产品、版本和配置上都有差异，所有这些都会影响响应报文中包含的首部行。

2.2.4 用户与服务器的交互：cookie

我们前面提到了 HTTP 服务器是无状态的。这简化了服务器的设计，并且允许工程师们去开发可以同时处理数以千计的 TCP 连接的高性能 Web 服务器。然而一个 Web 站点通常希望能够识别用户，可能是因为服务器希望限制用户的访问，或者因为它希望把内容与用户身份联系起来。为此，HTTP 使用了 cookie。cookie 在 [RFC 6265] 中定义，它允许站点对用户进行跟踪。目前大多数商务 Web 站点都使用了 cookie。

如图 2-10 所示，cookie 技术有 4 个组件：①在 HTTP 响应报文中的一个 cookie 首部行；②在 HTTP 请求报文中的一个 cookie 首部行；③在用户端系统中保留有一个 cookie 文件，并由用户的浏览器进行管理；④位于 Web 站点的一个后端数据库。使用图 2-10，我们通过一个典型的例子看看 cookie 的工作过程。假设 Susan 总是从家中 PC 使用 Internet Explorer 上网，她首次与 Amazon.com 联系。我们假定过去她已经访问过 eBay 站点。当请求报文到达该 Amazon Web 服务器时，该 Web 站点将产生一个唯一识别码，并以此作为索引在它的后端数据库中产生一个表项。接下来 Amazon Web 服务器用一个包含 Set-cookie: 首部的 HTTP 响应报文对 Susan 的浏览器进行响应，其中 Set-cookie: 首部含有该识别码。例如，该首部行可能是

```
Set-cookie: 1678
```