

接。对于每个这样的连接，在客户和服务端中都要分配 TCP 的缓冲区和保持 TCP 变量，这给 Web 服务器带来了严重的负担，因为一台 Web 服务器可能同时服务于数以百计不同的客户的请求。第二，就像我们刚描述的那样，每一个对象经受两倍 RTT 的交付时延，即一个 RTT 用于创建 TCP，另一个 RTT 用于请求和接收一个对象。

在采用持续连接的情况下，服务器在发送响应后保持该 TCP 连接打开。在相同的客户与服务器之间的后续请求和响应报文能够通过相同的连接进行传送。特别是，一个完整的 Web 页面（上例中的 HTML 基本文件加上 10 个图形）可以用单个持续 TCP 连接进行传送。更有甚者，位于同一台服务器的多个 Web 页面在从该服务器发送给同一个客户时，可以在单个持续 TCP 连接上进行。可以一个接一个地发出对对象的这些请求，而不必等待对未决请求（流水线）的回答。一般来说，如果一条连接经过一定时间间隔（一个可配置的超时间隔）仍未被使用，HTTP 服务器就关闭该连接。HTTP 的默认模式是使用带流水线的持续连接。我们把量化比较持续连接和非持续连接性能的任务留作第 2、3 章的课后习题。鼓励读者阅读文献 [Heidemann 1997; Nielsen 1997]。

2.2.3 HTTP 报文格式

HTTP 规范 [RFC 1945; RFC 2616] 包含了对 HTTP 报文格式的定义。HTTP 报文有两种：请求报文和响应报文。下面讨论这两种报文。

1. HTTP 请求报文

下面提供了一个典型的 HTTP 请求报文：

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

通过仔细观察这个简单的请求报文，我们就能知道很多东西。首先，我们看到该报文是用普通的 ASCII 文本书写的，这样有一定计算机知识的人都能够阅读它。其次，我们看到该报文由 5 行组成，每行由一个回车和换行符结束。最后一行后再附加一个回车换行符。虽然这个特定的报文仅有 5 行，但一个请求报文能够具有更多的行或者至少为一行。HTTP 请求报文的第一行叫做**请求行**（request line），其后继的行叫做**首部行**（header line）。请求行有 3 个字段：方法字段、URL 字段和 HTTP 版本字段。方法字段可以取几种不同的值，包括 GET、POST、HEAD、PUT 和 DELETE。绝大部分的 HTTP 请求报文使用 GET 方法。当浏览器请求一个对象时，使用 GET 方法，在 URL 字段带有请求对象的标识。在本例中，该浏览器正在请求对象/somedir/page.html。其版本字段是自解释的；在本例中，浏览器实现的是 HTTP/1.1 版本。

现在我们看看本例的首部行。首部行 Host: www.someschool.edu 指明了对象所在的主机。你也许认为该首部行是不必要的，因为在该主机中已经有一条 TCP 连接存在了。但是，如我们将在 2.2.5 节中所见，该首部行提供的信息是 Web 代理高速缓存所要求的。通过包含 Connection: close 首部行，该浏览器告诉服务器不希望麻烦地使用持续连接，它要求服务器在发送完被请求的对象后就关闭这条连接。User-agent: 首部行用来指明用户代理，即向服务器发送请求的浏览器的类型。这里浏览器类型是 Mozilla/5.0，即 Firefox 浏览器。这个首部行是有用的，因为服务器可以有效地为不同类型的用户代理