

## 17.2 客户端驱动的协商

对于服务器来说，收到客户端请求时只是发回响应，在其中列出可用的页面，让客户端决定要看哪个，这是最容易的事情。很显然，这是服务器最容易实现的方式，而且客户端很可能选择到最佳的版本（只要列表中有让客户端选择的足够信息）。不利之处是每个页面都需要两次请求：第一次获取列表，第二次获取选择的副本。这种技术速度很慢且过程枯燥乏味，让用户厌烦。

从实现原理上来说，服务器实际上有两种方法为客户端提供选项：一是发送回一个 HTML 文档，里面有到该页面的各种版本的链接和每个版本的描述信息；另一种方法是发送回 HTTP/1.1 响应时，使用 300 Multiple Choices 响应代码。客户端浏览器收到这种响应时，在前一种情况下，会显示一个带有链接的页面；在后一种情况下，可能会弹出对话框，让用户做选择。不管怎么样，决定是由客户端的浏览器用户作出的。

除了增加时延并且对每个页面都要进行繁琐的多次请求之外，这种方法还有一个缺点：它需要多个 URL：公共页面要一个，其他每种特殊页面也都要一个。因此，比如说原始的请求地址是 `www.joes-hardware.com`，Joe 的服务器可能会回复某个页面，该页面里面有到 `www.joes-hardware.com/english` 和 `www.joes-hardware.com/french` 的链接。如果客户端想加书签的话，是要加在原始的公共页面上呢，还是加在选中的页面上呢？如果用户想把这个网站推荐给他的朋友，是告知 `www.joes-hardware.com` 这个地址好呢，还是只告诉他们讲英语的朋友 `www.joes-hardware.com/english` 这个地址？

396

## 17.3 服务器驱动的协商

在前一节中，我们了解了客户端驱动的协商存在的若干缺点。大部分缺点都涉及客户端和服务端之间通信量的增长，这些通信量用来决定什么页面才是对请求的最佳响应。减少额外通信量的一种方法是让服务器来决定发送哪个页面回去，但为了做到这一点，客户端必须发送有关客户偏好的足够信息，以便服务器能够作出准确的决策。服务器通过客户端请求的首部集来获得这方面的信息。

有以下两种机制可供 HTTP 服务器评估发送什么响应给客户端比较合适。

- 检查内容协商首部集。服务器察看客户端发送的 Accept 首部集，设法用相应的响应首部与之匹配。
- 根据其他（非内容协商）首部进行变通。例如，服务器可以根据客户端发送的 User-Agent 首部来发送响应。