

- (2) 哑代理收到了这条 HTTP 请求，但它并不理解 Connection 首部（只是将其作为一个扩展首部对待）。代理不知道 keep-alive 是什么意思，因此只是沿着转发链路将报文一字不漏地发送给服务器（图 4-15b）。但 Connection 首部是个逐跳首部，只适用于单条传输链路，不应该沿着传输链路向下传输。接下来，就要发生一些很糟糕的事情了。
- (3) 在图 4-15b 中，经过中继的 HTTP 请求抵达了 Web 服务器。当 Web 服务器收到经过代理转发的 Connection: Keep-Alive 首部时，会误以为代理（对服务器来说，这个代理看起来就和所有其他客户端一样）希望进行 keep-alive 对话！对 Web 服务器来说这没什么问题——它同意进行 keep-alive 对话，并在图 4-15c 中回送了一个 Connection: Keep-Alive 响应首部。所以，此时 Web 服务器认为它在与代理进行 keep-alive 对话，会遵循 keep-alive 的规则。但代理却对 keep-alive 一无所知。不妙。
- (4) 在图 4-15d 中，哑代理将 Web 服务器的响应报文回送给客户端，并将来自 Web 服务器的 Connection: Keep-Alive 首部一起传送过去。客户端看到这个首部，就会认为代理同意进行 keep-alive 对话。所以，此时客户端和服务端都认为它们在进行 keep-alive 对话，但与它们进行对话的代理却对 keep-alive 一无所知。
- (5) 由于代理对 keep-alive 一无所知，所以会将收到的所有数据都回送给客户端，然后等待源端服务器关闭连接。但源端服务器会认为代理已经显式地请求它将连接保持在打开状态了，所以不会去关闭连接。这样，代理就会挂在那里等待连接的关闭。
- (6) 客户端在图 4-15d 中收到了回送的响应报文时，会立即转向下一条请求，在 keep-alive 连接上向代理发送另一条请求（参见图 4-15e）。而代理并不认为同一条连接上会有其他请求到来，请求被忽略，浏览器就在这里转圈，不会有任何进展了。
- (7) 这种错误的通信方式会使浏览器一直处于挂起状态，直到客户端或服务端将连接超时，并将其关闭为止。¹⁷

95

2. 代理和逐跳首部

为避免此类代理通信问题的发生，现代的代理都绝不能转发 Connection 首部和所有名字出现在 Connection 值中的首部。因此，如果一个代理收到了一个 Connection: Keep-Alive 首部，是不应该转发 Connection 首部，或所有名为 Keep-Alive 的首部的。

注 17：在很多类似的情形下，盲中继和转发的握手信息都会引发问题。