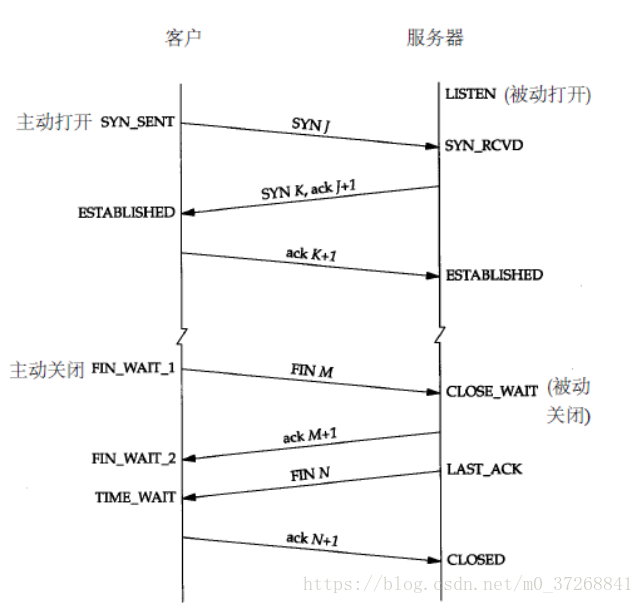
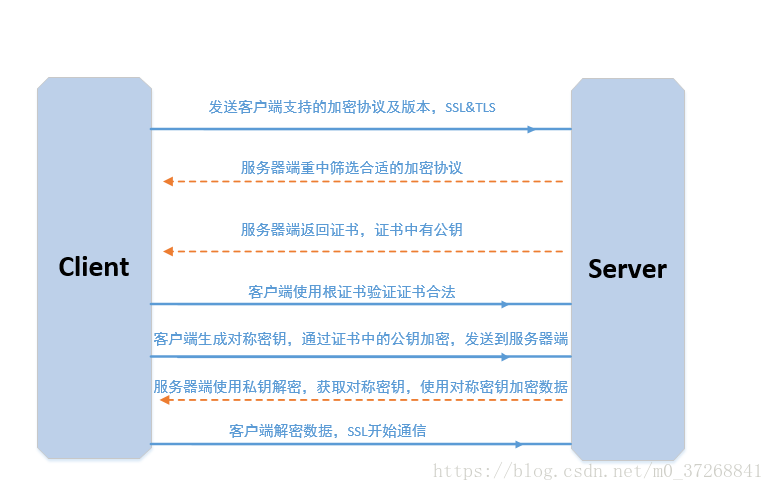
**Tcp连接过程**



**https连接过程**



**持久连接**

在 HTTP 1.0 版本中，并没有官方的标准来规定 Keep-Alive 如何工作，因此实际上它是被附加到 HTTP 1.0协议上，如果客户端浏览器支持 Keep-Alive ，那么就在HTTP请求头中添加一个字段 Connection: Keep-Alive，当服务器收到附带有 Connection: Keep-Alive 的请求时，它也会在响应头中添加一个同样的字段来使用 Keep-Alive 。这样一来，客户端和服务器之间的HTTP连接就会被保持，不会断开（超过 Keep-Alive 规定的时间，意外断电等情况除外），当客户端发送另外一个请求时，就使用这条已经建立的连接。

在 HTTP 1.1 版本中，默认情况下所有连接都被保持，如果加入 "Connection: close" 才关闭。目前大部分浏览器都使用 HTTP 1.1 协议，也就是说默认都会发起 Keep-Alive 的连接请求了，所以是否能完成一个完整的 Keep-Alive 连接就看服务器设置情况。

* HTTP Keep-Alive 简单说就是保持当前的TCP连接，避免了重新建立连接。
* HTTP 长连接不可能一直保持，例如 Keep-Alive: timeout=5, max=100，表示这个TCP通道可以保持5秒，max=100，表示这个长连接最多接收100次请求就断开。
* HTTP 是一个无状态协议，这意味着每个请求都是独立的，Keep-Alive 没能改变这个结果。另外，Keep-Alive也不能保证客户端和服务器之间的连接一定是活跃的，在 HTTP1.1 版本中也如此。唯一能保证的就是当连接被关闭时你能得到一个通知，所以不应该让程序依赖于 Keep-Alive 的保持连接特性，否则会有意想不到的后果。
* 使用长连接之后，客户端、服务端怎么知道本次传输结束呢？两部分：1. 判断传输数据是否达到了Content-Length 指示的大小；2. 动态生成的文件没有 Content-Length ，它是分块传输（chunked），这时候就要根据 chunked 编码来判断，chunked 编码的数据在最后有一个空 chunked 块，表明本次传输数据结束，详见[这里](http://www.cnblogs.com/skynet/archive/2010/12/11/1903347.html)。什么是 chunked 分块传输呢？下面我们就来介绍一下。

### Transfer-Encoding分块传输

一个简易的 Server 的流程如下：

* 1.建立连接，接受一个客户端连接。
* 2.接受请求，从网络中读取一条 HTTP 请求报文。
* 3.处理请求，访问资源。
* 4.构建响应，创建带有 header 的 HTTP 响应报文。
* 5.发送响应，传给客户端。

省略流程 3，大体的程序与调用的函数逻辑如下：

* socket() 创建套接字
* bind() 分配套接字地址
* listen() 等待连接请求
* accept() 允许连接请求
* read()/write() 数据交换
* close() 关闭连接