参考：<http://www.cnblogs.com/leenuxqt/p/3202527.html>

[区别下RPC与HTTP](http://www.cnblogs.com/leenuxqt/p/3202527.html)

**OSI网络结构的七层模型**

各层的具体描述如下：

　　第七层：应用层     定义了用于在网络中进行通信和数据传输的接口 - 用户程式；提供标准服务，比如虚拟终端、文件以及任务的传输和处理；   
　　第六层：表示层     掩盖不同系统间的数据格式的不同性； 指定独立结构的数据传输格式； 数据的编码和解码；加密和解密；压缩和解压缩   
　　第五层：会话层     管理用户会话和对话； 控制用户间逻辑连接的建立和挂断；报告上一层发生的错误   
　　第四层：传输层     管理网络中端到端的信息传送； 通过错误纠正和流控制机制提供可靠且有序的数据包传送； 提供面向无连接的数据包的传送；   
　　第三层：网络层     定义网络设备间如何传输数据； 根据唯一的网络设备地址路由数据包；提供流和拥塞控制以防止网络资源的损耗   
　　第二层：数据链路层 定义操作通信连接的程序； 封装数据包为数据帧； 监测和纠正数据包传输错误   
　　第一层：物理层      定义通过网络设备发送数据的物理方式； 作为网络媒介和设备间的接口；定义光学、电气以及机械特性。

 在上述7层中，http协议是应用层协议。HTTP协议是超文本传送协议(HyperText Transfer Protocol)的缩写，它是万维网(World Wide Web,www,也简称为Web)的基础。HTTP协议设计之初就是为了实现Web的想法。HTTP协议位于TCP/IP协议栈的应用层。基于HTTP协议的浏览器/服务器模式（B/S模式）的信息交换过程，分四个过程：建立连接、发送请求信息、发送响应信息、关闭连接。

**关于RPC的基本概念介绍如下**

英文原义：Remote Procedure Call Protocol

中文释义：（RFC-1831）远过程调用协议  
　　 注解：一种通过网络从远程计算机程序上请求服务，而不需要了解底层网络技术的协议。RPC协议假定某些传输协议的存在，如TCP或UDP，为通信程序之间携带信息数据。在OSI网络通信模型中，RPC跨越了传输层和应用层。RPC使得开发包括网络分布式多程序在内的应用程序更加轻易。  
　　 RPC采用客户机/服务器（C/S）模式。请求程序就是一个客户机，而服务提供程序就是一个服务器。首先，调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息的到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用过程接收答复信息，获得进程结果，然后调用执行继续进行。  
RPC信息协议由两个不同结构组成：调用信息和答复信息。

同时也注意到了这样的信息

[远程通信的几种选择（RPC，Webservice，RMI，JMS的区别）](http://blog.csdn.net/shan9liang/article/details/8995023)

[几种基于HTTP协议的RPC性能比较](http://bbs.chinaunix.net/thread-1762282-1-1.html)

RPC和Socket的区别

<http://blog.163.com/fanning_7213/blog/static/249650520113124540501/>

RPC(Remote Procedure Call，远程过程调用)是建立在Socket之上的,出于一种类比的愿望,在一台机器上运行的主程序,可以调用另一台机器上准备好的子程序,就像LPC(本地过程调用).

    越底层，代码越复杂、灵活性越高、效率越高；越上层，抽象封装的越好、代码越简单、效率越差。Socket和RPC的区别再次说明了这点。

不论是程序员在编写基于C/S(客户端服务器)的程序时,还是网络工程师在处理RPC问题时,他们问的最多的就是RPC和Socket有什么区别和联系?   
　 　RPC(Remote Procedure Call，远程过程调用)是建立在Socket之上的,出于一种类比的愿望,在一台机器上运行的主程序,可以调用另一台机器上准备好的子程序,就像 LPC(本地过程调用).RPC带来了开发C/S程序的简单可靠的手段,它通过一种叫XDR的数据表达方法描述数据,程序员书写伪代码,然后由 rpcgen程序翻译为真正的可编译的C语言源代码,再编译成真正的Client端和Server端程序。   
　　RPC作为普遍的C/S开发方法,开发效率高效,可靠.但RPC方法的基本原则是以模块调用的简单性忽略通讯的具体细节,以便程序员不用关心C/S之间的通讯协议,集中精力对付实现过程.这就决定了 RPC生成的通讯包不可能对每种应用都有最恰当的处理办法,与Socket方法相比,传输相同的有效数据,RPC占用更多的网络带宽.   
　　RPC是在Socket的基础上实现的,它比socket需要更多的网络和系统资源.另外,在对程序优化时,程序员虽然可以直接修改由rpcgen产生的令人费解的源程序,但对于追求程序设计高效率的RPC而言,获得的简单性则被大大削弱.   
RPC与是Socket的类比