

Socket

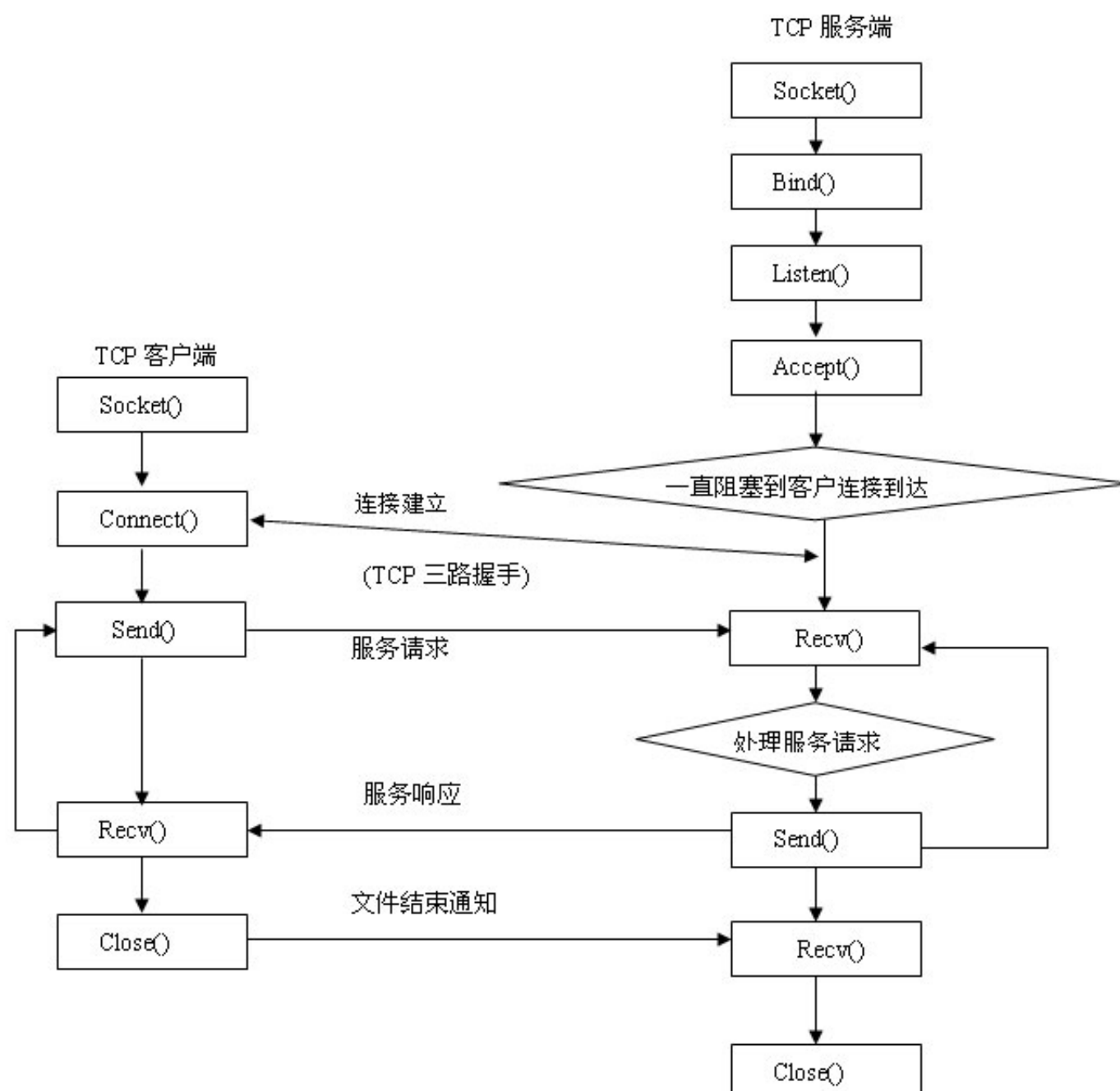
网络上的两个程序通过一个双向的通信连接实现数据的交换，这个连接的一端称为一个socket。建立网络通信连接至少需要一对端口号(socket)。socket本质是编程接口(API)，对TCP/IP的封装，TCP/IP也要提供可供程序员做网络开发所用的接口，这就是Socket编程接口

iOS Socket

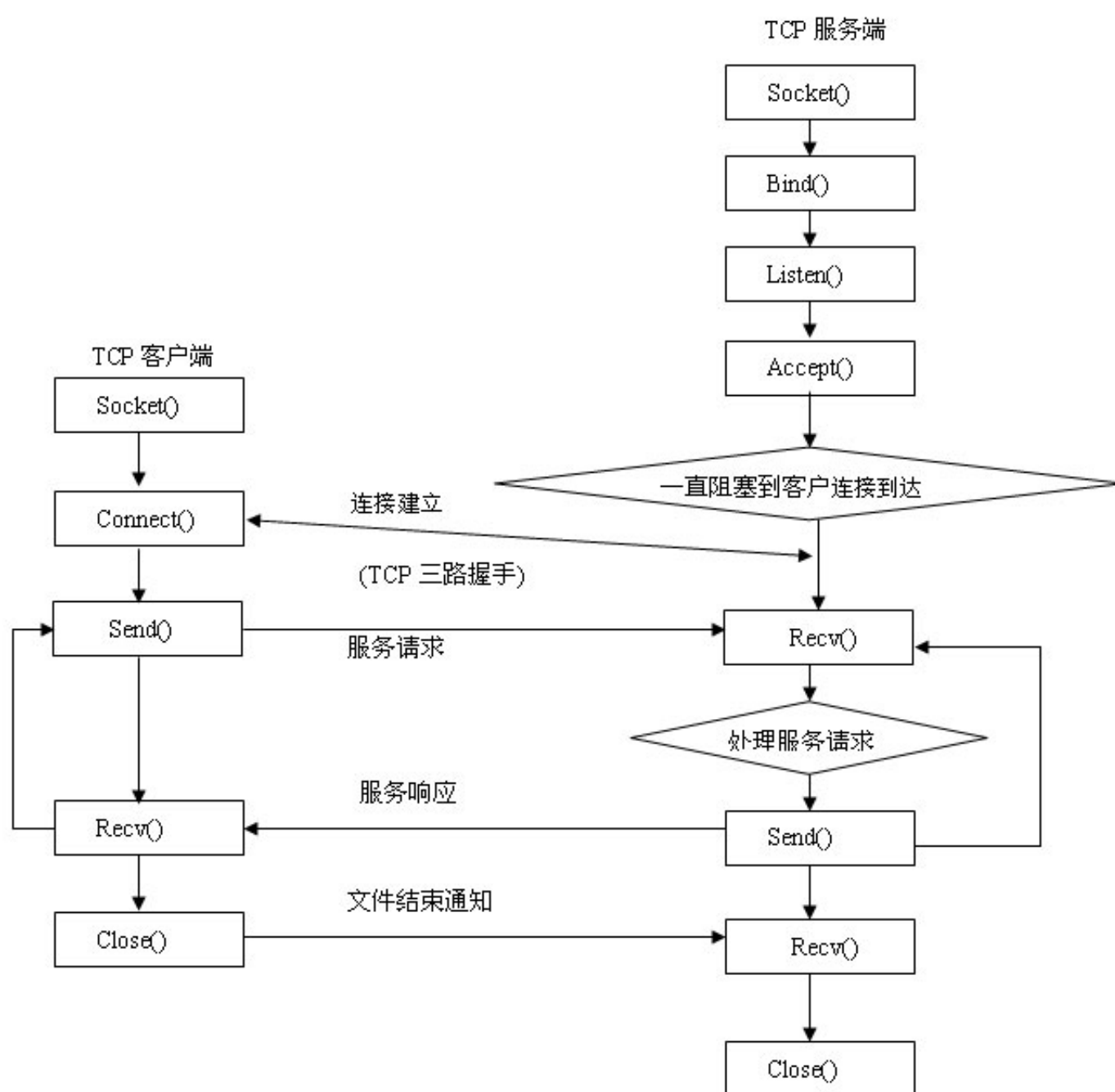
- BSD Socket
- CFSocket
- AsyncSocket
- [CocoaAsyncSocket](#)

基本框架

- 基本TCP客户-服务器程序设计基本框架



- 基本UDP客户-服务器程序设计基本框架流程图



常用的Socket类型有两种：流式Socket（`SOCK_STREAM`）和数据报式Socket（`SOCK_DGRAM`）。

流式是一种面向连接的Socket，针对于面向连接的TCP服务应用；

数据报式Socket是一种无连接的Socket，对应于无连接的UDP服务应用。

拓展：

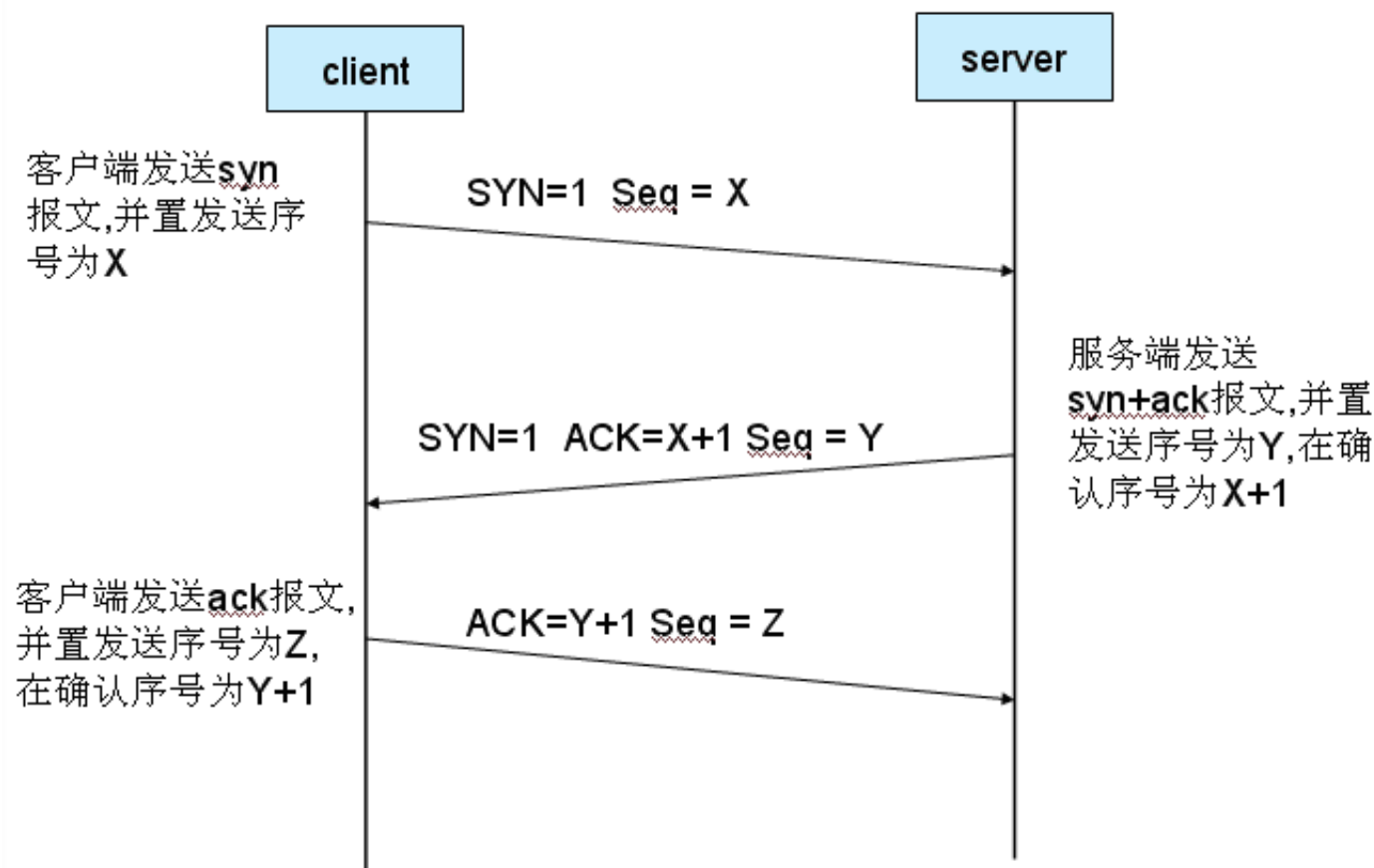
- 三次握手

第一次握手：客户端发送一个TCP的SYN标志位置1的包指明客户打算连接的服务器的端口，以及初始序号X,保存在包头的序列号(Sequence Number)字段里。

第二次握手：服务器发回确认包(ACK)应答。即SYN标志位和ACK标志位均为1同时，将确认序号(Acknowledgement Number)设置为客户的序列号加1以，即X+1。

第三次握手：客户端再次发送确认包(ACK) SYN标志位为0，ACK标志位为1。并且把服务器发来ACK的序号字段+1，放在确定字段中发送给对方.并且在数据段放写序列号的+1。

TCP 三次握手



- 四次挥手

编程中，任何一方执行close()操作即可产生挥手操作。

因为当Server端收到Client端的SYN连接请求报文后，可以直接发送SYN+ACK报文。其中ACK报文是用来应答的，SYN报文是用来同步的。但是关闭连接时，当Server端收到FIN报文时，很可能并不会立即关闭SOCKET，所以只能先回复一个ACK报文，告诉Client端，“你发的FIN报文我收到了”。只有等到我Server端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送。故需要四步握手。

TCP 四次挥手

