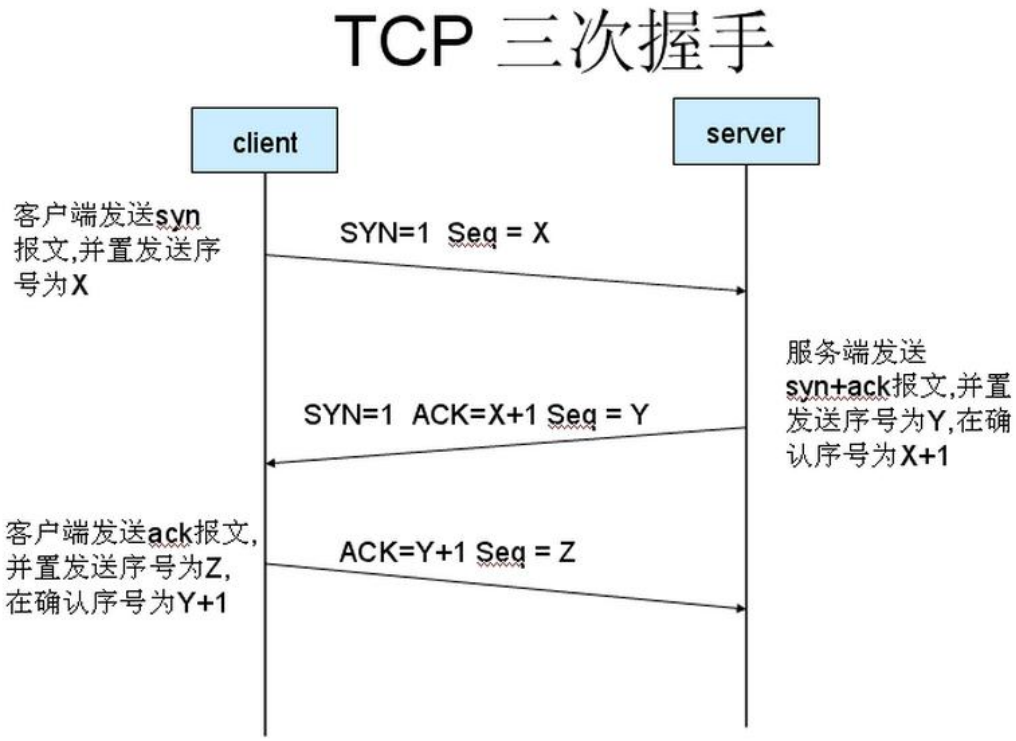
TCP协议

OSI七层模型	TCP/IP概念层模型	功能	TCP/IP协议族
应用层	应用层	文件传输，电子邮件，文件服务，虚拟终端	TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet
表示层		数据格式化，代码转换，数据加密	没有协议
会话层		解除或建立与别的接点的联系	没有协议
传输层	传输层	提供端对端的接口	TCP，UDP
网络层	网络层	为数据包选择路由	IP，ICMP，RIP，OSPF，BGP，IGMP
数据链路层	链路层	传输有地址的帧以及错误检测功能	SLIP，CSLIP，PPP，ARP，RARP，MTU
物理层		以二进制数据形式在物理媒体上传输数据	ISO2110，IEEE802，IEEE802.2

TCP协议是一个可靠的、面向连接、面向字节流的传输层协议。

TCP三次握手



步骤：

A: 客户端发送**SYN**报文给服务端并置**同步序列号seq=x**

B: 服务端收到后发送**ACK**、**SYN** 报文并置**seq=y**、**ACK=x+1**

C: 客户端发送**ACK**报文，**ACK=Y+1**、**seq=z**

Q1: TCP为什么需要三次握手，最后一次不要可不可以？

1. 双方互相确认发送接收能力 **同步初始化序列号** （TCP是可靠的）

   第一次握手：确认了客户端的发送能力和服务端的接收能力

   第二次握手：确认了客户端的接收能力和服务端的发送能力

   前两次握手虽然确认了客户端和服务端的接收、发送能力，但服务端没有确认客户端具备接收能力

2. 防止重复连接

   防止旧的重复连接引起的连接混乱 （网络拥堵造成的重复连接）

   假如只有两次握手的话，服务端就只能接收连接或拒绝连接，但并不清楚是正常的连接请求，还是过期的连接请求。这样在第三次握手时，客户端可以根据服务端返回的序列号判断是不是过期连接（序列号是增加的），如果是过期连接就可以向服务端发起断开请求，反之发送确认连接来建立真正的连接。

Q2: 什么是半连接队列？

服务端第一次收到客户端的SYN时，进入SYN_RCVD状态，此时双方并没有正式建立连接，服务端会将此种状态下的连接请求放入队列中，这个队列就叫半连接队列。

当然，还有全连接队列，就是已经完成三次握手，建立起连接的就会放入全连接队列。

注 服务端发送SYN+ACK包后，如果未收到客户端的ACK，服务端就会进行首次重传，等待一段时间后还是未收到，进行第二次重传。重传的次数取决于系统规定的重传次数，一旦超过，就会将连接移除出半连接队列。

Q3: ISN(initial Sequence Number) 是固定的？

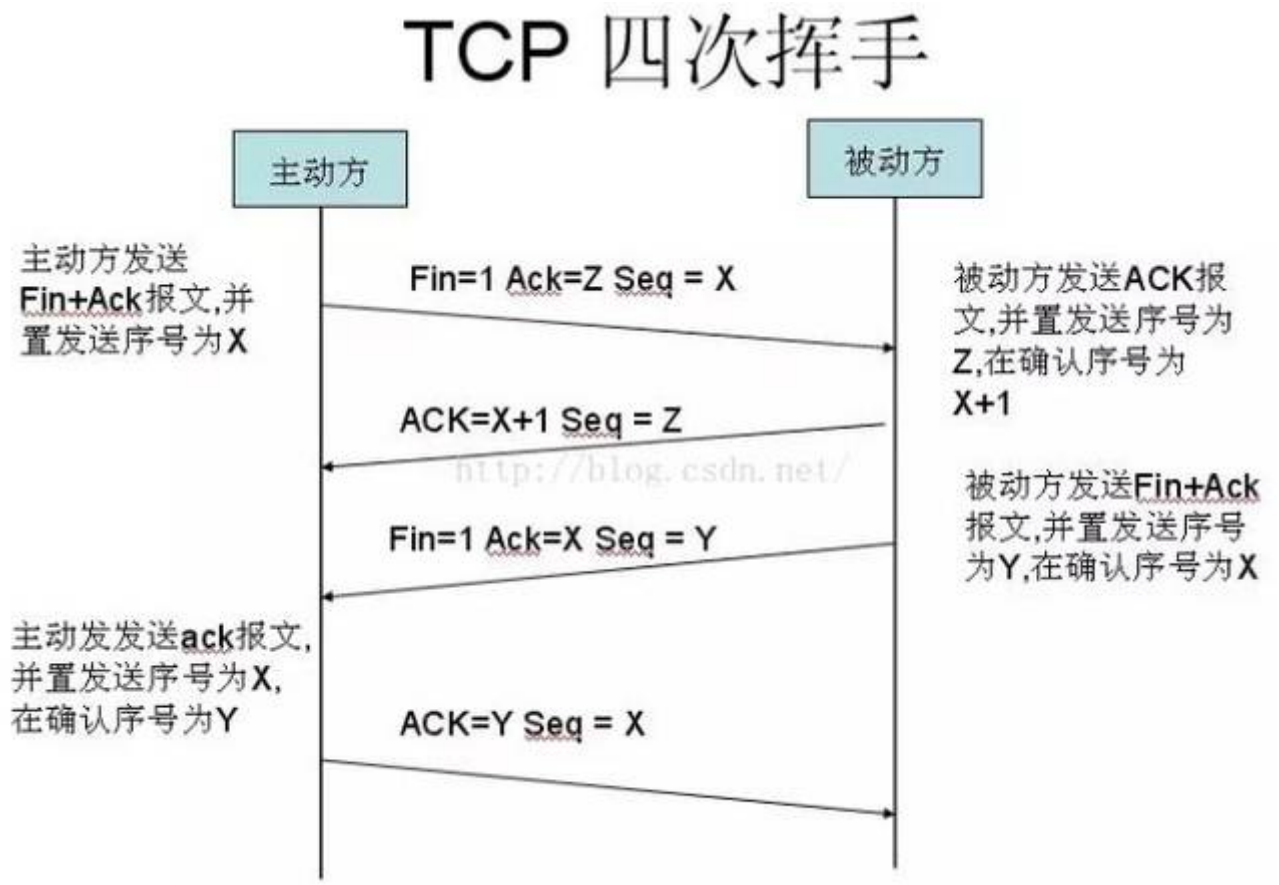
当一端为建立连接发送自己的SYN时，会选择一个初始序号seq。随时间变化，所有每个连接都有自己特有的序列号。

三次握手重要的一个功能就是双方交换自己的ISN, 以便让对方知道接下来接收数据的时候是按怎样的序列号就行组装数据(数据不一定是一次性发送的，根据数据的大小进行合理的分组发送，这样有可能因为网络问题导致延迟)

Q4: 三次握手可以携带数据？

第三次握手是可以携带数据的，此时客户端已经知晓服务端具备接收能力。 前两次是不可以的，以免恶意攻击（不理会服务端的接收发送能力恶意发送大量SYN，会让服务端耗费大量的时间、内存来处理这些数据）

TCP四次挥手



Q1: 为什么需要四次？

在连接过程中，当服务端收到客户端的SYN时，可以直接发送SYN+ACK报文。SYN是用来同步的，ACK是用来确认的。当客户端向服务端发送关闭请求FIN后，很可能并不会立即关闭SOCKET，而是先给客户端回复ACK，告知客户端“你发的关闭请求我收到了”。只有等服务端把所有的报文都发送了，才可以关闭，才会向客户端发送关闭请求。

TCP的“半关闭”状态，也就是说TCP允许连接的一端在结束它自己的发送后还可以接收来自另一端的数据。

Q2: 四次挥手释放连接时，为什么需要等待2MSL(MSL:报文段最大生存时间)?

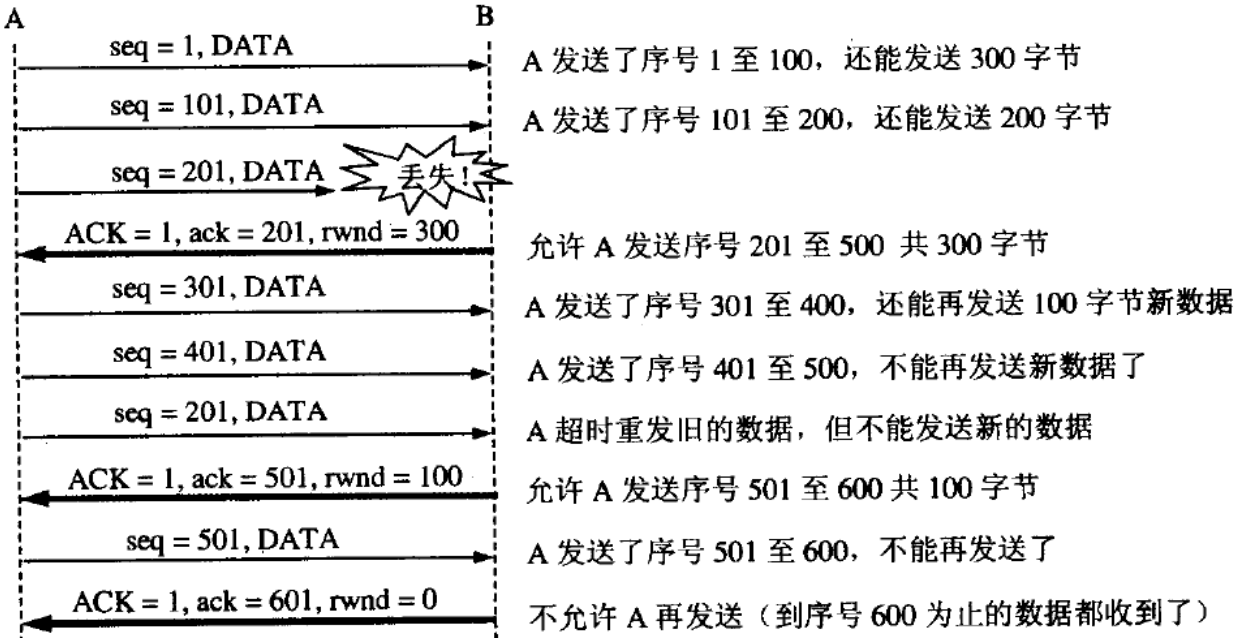
为了保证客户端发送的最后一个ACK报文段可以顺利传输到服务端。因为这个ACK可能会丢失，导致服务端收不到对FIN-ACK的确认报文，服务端会重发FIN-ACK，接着客户端会重发确认ACK，重新启动时间等待计时器。最后确保客户端和服务端都可以成功的关闭。

1. 保证客户端的最后一个ACK可以到达服务端

2. 防止“已失效的连接请求报文段”出现在网络。客户端发送完最后一个ACK，经过2MSL，就可以使本连接内所产生的所有报文端都可以消失在网络中。（确保客户端、服务端正常关闭，不再接收请求）

TCP流量控制

所谓的流量控制就是让发送方的发送速率不要太快，利用**滑动窗口**机制可以很方便的在tcp连接上实现对发送方的流量控制。TCP的窗口单位是字节（面向字节流），不是报文段，发送方的发送窗口不能超过接收方给出的接收窗口的数值



TCP拥塞控制

拥塞控制和流量控制的区别：

拥塞控制是防止过多的数据注入到网络中，避免网络中路由器和链路过载。拥塞控制的前提是网络能承受现有的负荷。这是个全局性的问题，涉及主机、路由器、以及与降低网络传输性能有关的一切因素。

流量控制是减缓发送方的数据发送速率，以便接收方来得及接收处理。是点对点通信量的控制，是个端对端的问题。