**TCP三次握手的简单理解**

**为什么要进行三次握手？**

众所周知，TCP协议与UDP协议的最大不同在于TCP协议使用的是面向连接的，可靠的数据传输协议，既然是可靠的数据传输协议，那么在传输数据之前，就必须先建立连接，那么怎样才能建立连接呢？那就是通过TCP三次握手实现的。

事实上，TCP三次握手中蕴含了许多有趣的故事，举个我曾看过的一个例子，大概内容如下：红军和蓝军都想消灭一波敌人，但是单凭他们一个军队的力量都不足以消灭这波敌人，因此他们想到了一起合作，于是红军向蓝军发了一封电报，内容是约定好早上8点一起向敌军进攻，由于他们不确定蓝军是否一定能收到电报，所以只有收到蓝军的回复之后才会进行进攻，而蓝军也是同样的想法，因为他们不确定红军一定能收到自己的回复而在约定好的时间发动进攻，所以他们只有收到红军的回复后才发动进攻......

问怎样才能保证这次战役一定胜利呢？答案是不可能的，因为双方都对于自己发出的消息对方是否一定接收得到存在质疑，所以，这样的通信将一直进行下去，结果将是使胜利的几率一直接近100%，但是却永远达不到100%，那么有什么好方法解决这种现状呢？一种方法就是提前约定好，在几次通信之后，就发动进攻，而不是一直继续下去。而TCP协议就是使用这种方法，在三次通信之后就建立连接，在这个例子之中就是：红军率先发送消息，蓝军收到消息之后立即回复，红军收到蓝军的回复后立即向蓝军发送消息确认自己已收到蓝军的回复，之后红军不管蓝军是否收到自己的回复，都按约定好的时间发动进攻，而蓝军只有收到红军第二次发的消息之后才按计划发动进攻。

**三次握手的过程**

首先要简单了解ACK和SYN是什么： ``` ACK： 响应数据包 SYN： 主动建立连接数据包 ```

第一步：客户端随机选取一个大于1024的端口作为程序的沟通接口，在TCP报头中，包含带有SYN的主动连接，记录下发送给服务器的SYN序号。

第二步：服务器收到客户端的SYN后，回复给客户端一个同时带有SYN和ACK的TCP数据包，同时，将收到的客户端发送的SYN序号加1，便于客户端确认，同时重新发送一个SYN序号，然后等待客户端响应。

第三步：客户端收到服务器的响应之后，验证服务器ACK序号，并将服务器SYN序号加1，并返回包含ACK的TCP数据包。

第四步：服务器收到来自客户端的响应之后，验证ACK序号，建立连接。

ACK=1

Seq:20002

ACK=1 Seq:10001

ACK=1 Seq:10002

SYN=1 Seq:20001

SYN

ACK/SYN

ACK