OSI参考模型：

应用层：DNS、FTP、HTTP、

表示层

会话层

传输层：UDP、TCP

网络层：IP

数据链路层

物理层

UPD：无连接、不可靠

TCP的三次握手，四次挥手

**建立连接的三次握手：**

第一次握手：建立连接时，客户端发送syn包(syn=j)到服务器，并进入SYN\_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

完成三次握手，客户端与服务器开始传送数据，也就是ESTABLISHED状态。

**结束连接的四次握手**

TCP有一个特别的概念叫做half-close，这个概念是说，TCP的连接是全双工（可以同时发送和接收）连接，因此在关闭连接的时候，必须关闭传和送两个方向上的连接。客户机给服务器一个FIN为1 的TCP报文，然后服务器返回给客户端一个确认ACK报文，并且发送一个FIN报文，当客户机回复ACK报文后（四次握手），连接就结束了。

TIME\_WAIT等待状态，这个状态又叫做2MSL状态，说的是在TIME\_WAIT2发送了最后一个ACK数据报以后， 要进入 TIME\_WAIT状态，这个状态是防止最后一次握手的数据报没有传送到对方那里而准备的（注意这不是四次握手，这是第四次握手的保险状态）。这个状态在 很大程度上保证了双方都可以正常结束，但是，问题也来了。

由于插口的2MSL状态（插口是IP和端口对的意思，socket），使得应用程序在2MSL时间内是无法再次使用同一个插口的，对于客户程序还好 一些，但是对于服务程序，例如httpd，它总是要使用同一个端口来进行服务，而在 2MSL时间内，启动httpd就会出现错误（插口被使用）。为了避免这个错误，服务器给出了一个平静时间的概念，这是说在2MSL时间内，虽然可以重新 启动服务器，但是这个服务器还是要平静的等待2MSL时间的过去才能进行下一次连接。

FIN\_WAIT\_2状态

这就是著名的半关闭的状态了，这是在关闭连接时，客户端和服务器两次握手之后的状态。在这个状态下，应用程序还有接受数据的能力，但是已经无法