## TCP和UDP的优缺点及区别

TCP的优点： 可靠，稳定 TCP的可靠体现在TCP在传递数据之前，会有三次握手来建立连接，而且在数据传递时，有确认、窗口、重传、拥塞控制机制，在数据传完后，还会断开连接用来节约系统资源。 TCP的缺点： 慢，效率低，占用系统资源高，易被攻击 TCP在传递数据之前，要先建连接，这会消耗时间，而且在数据传递时，确认机制、重传机制、拥塞控制机制等都会消耗大量的时间，而且要在每台设备上维护所有的传输连接，事实上，每个连接都会占用系统的CPU、内存等硬件资源。 而且，因为TCP有确认机制、三次握手机制，这些也导致TCP容易被人利用，实现DOS、DDOS、CC等攻击。

UDP的优点： 快，比TCP稍安全 UDP没有TCP的握手、确认、窗口、重传、拥塞控制等机制，UDP是一个无状态的传输协议，所以它在传递数据时非常快。没有TCP的这些机制，UDP较TCP被攻击者利用的漏洞就要少一些。但UDP也是无法避免攻击的，比如：UDP Flood攻击…… UDP的缺点： 不可靠，不稳定 因为UDP没有TCP那些可靠的机制，在数据传递时，如果网络质量不好，就会很容易丢包。 基于上面的优缺点，那么： 什么时候应该使用TCP： 当对网络通讯质量有要求的时候，比如：整个数据要准确无误的传递给对方，这往往用于一些要求可靠的应用，比如HTTP、HTTPS、FTP等传输文件的协议，POP、SMTP等邮件传输的协议。 在日常生活中，常见使用TCP协议的应用如下： 浏览器，用的HTTP FlashFXP，用的FTP Outlook，用的POP、SMTP Putty，用的Telnet、SSH QQ文件传输 ………… 什么时候应该使用UDP： 当对网络通讯质量要求不高的时候，要求网络通讯速度能尽量的快，这时就可以使用UDP。 比如，日常生活中，常见使用UDP协议的应用如下： QQ语音 QQ视频 TFTP ……

有些应用场景对可靠性要求不高会用到UPD，比如长视频，要求速率

小结TCP与UDP的区别：

1.基于连接与无连接； 2.对系统资源的要求（TCP较多，UDP少）； 3.UDP程序结构较简单； 4.流模式与数据报模式 ； 5.TCP保证数据正确性，UDP可能丢包，TCP保证数据顺序，UDP不保证。

tcp协议和udp协议的差别 TCP UDP 是否连接 面向连接 面向非连接 传输可靠性 可靠 不可靠 应用场合 传输大量数据 少量数据 速度 慢 快

TCP与UDP区别总结：

1. TCP面向连接（如打电话要先拨号建立连接）;UDP是无连接的，即发送数据之前不需要建立连接
2. TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP尽最大努力交付，即不保证可靠交付
3. TCP面向字节流，实际上是TCP把数据看成一连串无结构的字节流;UDP是面向报文的
4. P没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如IP电话，实时视频会议等）
5. 每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信
6. TCP首部开销20字节;UDP的首部开销小，只有8个字节
7. TCP的逻辑通信信道是全双工的可靠信道，UDP则是不可靠信道

### 三次握手

1. 第一次握手：客户端发送一个TCP标志位SYN=1，ACK=0的数据包给服务端，并随机会产生一个Seq=J.当服务端接收到这个数据后，服务端由SYN=1可知客户端是想要建立连接；
2. 第二次握手：服务端要对客户端的联机请求进行确认，向客户端发送应答号ACK=1、SYN=1        确认号Ack=J+1，此值是客户端的序列号加1，还会产生一个随机的序列号Seq=K，这样就告诉客户端可以进行连接；
3. 第三次握手：客户端收到数据后检查Ack是否为J+1，以及标志位ACK的值是否为1，若为1，则会发送ACK=1、确认号码Ack=K+1，告诉服务端,你的请求连接被确认，连接可以建立，Client和Server进入ESTABLISHED状态，完成三次握手，随后Client与Server之间可以开始传输数据了。

### 四次挥手

由于TCP连接时全双工的，因此，每个方向都必须要单独进行关闭，这一原则是当一方完成数据发送任务后，发送一个FIN来终止这一方向的连接，收到一个FIN只是意味着这一方向上没有数据流动了，即不会再收到数据了，但是在这个TCP连接上仍然能够发送数据，直到这一方向也发送了FIN。首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭，上图描述的即是如此。 1. 第一次挥手：Client发送一个FIN，用来关闭Client到Server的数据传送，Client进入FIN\_WAIT\_1状态。 2. 第二次挥手：Server收到FIN后，发送一个ACK给Client，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号），Server进入CLOSE\_WAIT状态。 3. 第三次挥手：Server发送一个FIN，用来关闭Server到Client的数据传送，Server进入LAST\_ACK状态。 4. 第四次挥手：Client收到FIN后，Client进入TIME\_WAIT状态，接着发送一个ACK给Server，确认序号为收到序号+1，Server进入CLOSED状态，完成四次挥手。