

# TCP，UDP协议的区别

类型	特点			性能		应用场景	首部字节
	是否面向连接	传输可靠性	传输形式	传输效率	所需资源		
TCP	面向连接	可靠	字节流	慢	多	要求通信数据可靠 (如文件传输、邮件传输)	20-60
UDP	无连接	不可靠	数据报文段	快	少	要求通信速度高 (如域名转换)	8个字节 (由4个字段组成)

UDP在传送数据之前不需要先建立连接，远地主机在收到UDP报文之后，不需要给出任何确认；虽然UDP不提供可靠交付，在某些情况下UDP确实是一种有效的工作方式（一般用于即时通信），比如QQ语音，QQ视频，直播等；

TCP提供面向连接的服务，在传送数据之前必须建立连接，数据传输完毕之后要释放连接。TCP不提供广播或者多播服务，由于TCP要提供可靠的，面向连接的传输服务，（具体可靠体现在传输数据之前三次握手，数据传递时，有确认，窗口，重传，拥塞控制等机制，传输数据之后要四次挥手结束）这些机制难以避免的增加了很多开销，所以使得协议数据单元首部增大很多，还要占用很多处理资源，所以TCP一般用于传输，发送和接收邮件，远程登录等场景；

## TCP协议如何保证可靠传输

- 应用数据被分割成TCP认为最合适发送的数据块；
- TCP给发送的每一个包进行编号，接收方向上层返回数据时要先排序，返回有序的数据；
- 校验和**，TCP将保持他首部和数据的校验和，这是一个端到端的校验和，目的是检测数据在传输过程中任何变化，如果发现有错误，就丢弃这个报文段和不确认收到这个报文段；
- TCP的接收端丢弃重复的数据；
- 流量控制**，TCP连接的两端都有一块固定大小的缓存区，TCP接收端只允许发送端发送接收端缓冲区能接纳的数据，当接收方来不及处理发送发数据，能提示对方降低发送速率，放在包丢失；
- 拥塞控制**，当网络拥塞的时候，减少发送数据；
- ARQ协议**，基本原理是没发送完一个分组就停止发送，等待对方确认，在收到确认之后再发下一个分组；
- 超时重传**，当TCP发出一个段后，他启动一个定时器，等待目的端确认收到这个报文段，如果不能及时的收到这个确认，将会重发这个报文段；

## ARQ协议

连续 ARQ 协议可提高信道利用率。发送方维持一个发送窗口，凡位于发送窗口内的分组可以连续发送出去，而不需要等待对方确认。接收方一般采用累计确认，对按序到达的最后一个分组发送确认，表明到这个分组为止的所有分组都已经正确收到了。

优点：信道利用率高，容易实现，及时确认丢失，也不必重传 缺点：不能像发送方反应已经正确接收到的所有分组的信息。比如：发送方发送了 5条 消息，中间第三条丢失（3号），这时接收方只能对前两个发送确认。发送方无法知道后三个分组的下落，而只好把后三个全部重传一次。这也叫 Go-Back-N（回退 N），表示需要退回来重传已经发送过的 N 个消息。

## 滑动窗口和流量控制

TCP利用滑动窗口实现流量控制，流量控制是为了控制发送方发送速率，保证接收方来得及接收。

接收方发送的确认报文中的窗口字段可以控制发送窗口的大小，从而影响发送方的发送速率，将窗口设置为0，发送方不能在发送；

## 拥塞控制

在某段时间，若对网络中某一资源的需求超过了该资源所能提供的可用部分，网络的性能就要变坏。这种情况就叫拥塞。拥塞控制所要做的都有一个前提，就是网络能够承受现有的网络负荷。拥塞控制是一个全局性的过程，涉及到所有的主机，所有的路由器，以及与降低网络传输性能有关的所有因素。相反，流量控制往往是点对点通信量的控制，是个端到端的问题。流量控制所要做到的就是抑制发送端发送数据的速率，以便使接收端来得及接收。

为了进行拥塞控制，TCP发送方要维持一个**拥塞窗口**cwnd的变量，拥塞窗口的大小取决于网络的拥塞程度，并且动态变化，发送方的发送窗口，取拥塞窗口和接收方是接收窗口中较小的一个；

TCP的拥塞控制采用四种算法，即**慢开始**，**拥塞避免**，**快重传**和**快恢复**。

**慢开始**：慢开始算法的思路是当主机开始发送数据时，如果立即把大量数据字节注入到网络，那么可能会引起网络阻塞，因为现在还不知道网络的符合情况。经验表明，较好的方法是先探测一下，即由小到大逐渐增大发送窗口，也就是由小到大逐渐增大拥塞窗口数值。cwnd初始值为1，每经过一个传播轮次，cwnd加倍。**拥塞避免**：拥塞避免算法的思路是让拥塞窗口cwnd缓慢增大，即每经过一个往返时间RTT就把发送方的cwnd加1。**快重传和快恢复**：在TCP/IP中，快速重传和恢复（fast retransmit and recovery, FRR）是一种拥塞控制算法，它能快速恢复丢失的数据包。没有FRR，如果数据包丢失了，TCP将会使用定时器来要求传输暂停。在暂停的这段时间内，没有新的或复制的数据包被发送。有了FRR，如果接收机接收到一个不按顺序的数据段，它会立即给发送机发送一个重复确认。如果发送机接收到三个重复确认，它会假定确认件指出的数据段丢失了，并立即重传这些丢失的数据段。有了FRR，就不会因为重传时要求的暂停被耽误。当有单独的数据包丢失时，快速重传和恢复（FRR）能最有效地工作。当有多个数据信息包在某一段很短的时间内丢失时，它则不能很有效地工作。