

TCP重传:

报文重传是TCP最基本的错误恢复功能，它的目的是防止报文丢失。

报文丢失的可能原因是很多的，包括应用故障、路由设备过载、或者临时的服务宕机等。

重传时间过短:在网络中由于拥塞引起丢包时，频繁的重传会进一步加剧网络拥塞。引起丢包，恶化网络传输性能。

重传时间过长:接收方长时间无法完毕数据接收。引起长时间占用连接线路造成资源耗损、传输效率较低等问题。

针对上述问题，TCP中设计了超时重传机制。该机制规定当发送方A向B发送数据包P1时，开启时长为RT0的重传定时器，假设A在RT0内未收到B对P1的确认报文，则觉得P1在网络中丢失。此时又一次发送P1。由此，引出RT0大小的设定问题。对若干次时间取平均值，该值用于确定终于RT0值。在终于RT0值确定之前，确定每一次报文传输是否有丢包发生使用重传计时器。

当报文发送之后，但接收方尚未发送TCP ACK报文，发送方假设源报文丢失并将其重传。

重传之后，RT0值加倍；假设在2倍RT0值到达之前还是没有收到ACK报文，就再次重传，假设仍然没有收到ACK，那么RT0值再次加倍。如此持续下去，每次重传RT0都翻倍，知道收到ACK报文或发送方达到配置的最大重传次数。

1) 超时重传

超时重传机制来保证TCP传输的可靠性

每次发送数据包时，发送的数据包都有序列号（seq），接收端收到数据后，会回复ack进行确认，表示某一seq号数据已经收到。发送方在发送了某个seq包后。等待一段时间，假设没有收到相应的ack回复，就会觉得报文丢失，会重传这个数据包。

2)

TCP拥塞控制能够提升网络利用率，降低丢包率，并保证网络资源对每条数据流的公平性。

拥塞控制主要有几种：慢启动、阻塞避免、快速重传以及快速恢复。

拥塞控制的最终受控变量是发送端向网络一次连续写入的数据量（即收到其中第一个数据的确认应答之前的所有数据），称为发送窗口。如果发送窗口值太小，则会引起网络延迟，如果太大，则容易导致网络拥塞。

慢启动

为了防止网络拥塞，TCP采用一种慢启动算法，对发送数据量，引入了拥塞窗口，在慢启动时，将这个拥塞窗口设为1个报文段发送数据。之后每收到一次确认应答，拥塞窗口就加1个报文段。

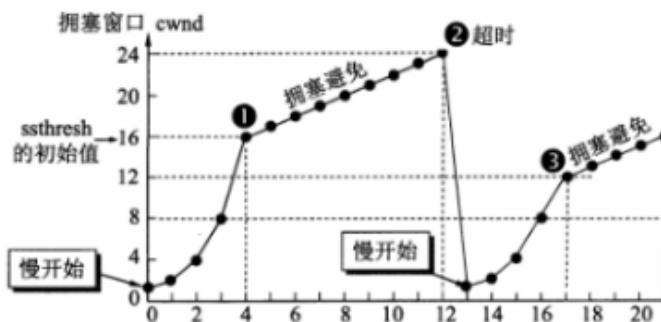
慢启动算法的步骤如下：

1. 初始化cwnd（拥塞窗口）= 1，表示可以传一个MSS大小的数据。
2. 经过一个RTT（收到一轮的确认）， $cwnd = cwnd * 2$ 。
4. ssthresh(slow start threshold, 慢开始门限)，是一个上限，当 $cwnd \geq ssthresh$ 时，就会进入拥塞避免算法。

拥塞避免算法

拥塞避免算法是让拥塞窗口缓慢地增大，收到每一轮的确认后，将拥塞窗口的值加1，而不是加倍，这样拥塞窗口的值按照线性规律缓慢的增长。在慢启动中，当 $cwnd \geq ssthresh$ 时，就会进入“拥塞避免算法”。一般来说ssthresh的值是65535，单位是字节，当cwnd达到这个值后，算法如下。

无论是在慢开始阶段还是拥塞避免阶段，只要发送方判断出现拥塞（其根据就是没有收到确认，虽然没有收到确认可能是其他原因的分组丢失，但是因为无法判定，所以都当作拥塞来处理），就把慢开始门限设置为出现拥塞时的发送窗口大小的一半。然后把拥塞窗口设置为1，执行慢开始算法。



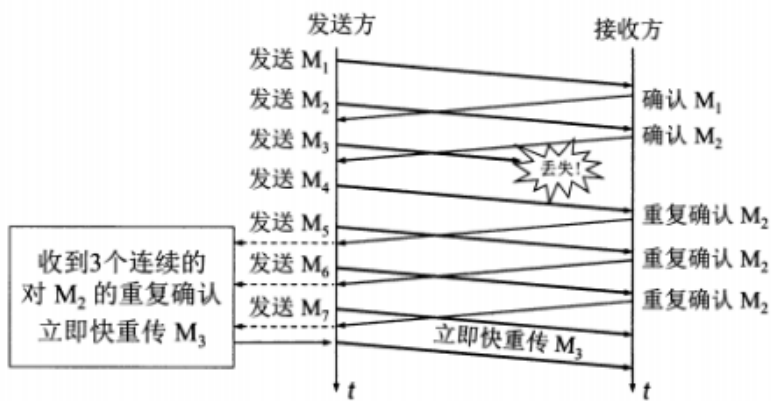
加法增大与乘法减小

乘法减小：无论是慢开始阶段还是拥塞避免，只要出现了网络拥塞（超时），就把慢开始门限值ssthresh减半

加法增大：执行拥塞避免算法后，拥塞窗口线性缓慢增大，防止网络过早出现拥塞

快重传

1. 快重传要求接收方在收到一个失序的报文段后就立即发出重复确认（为的是使发送方早知道有报文段没有到达对方）而不要等到自己发送数据时捎带确认。快重传算法规定，发送方只要一连收到三个重复确认就应当立即重传对方尚未收到的报文段，而不必继续等待设置的重传计时器时间到期。



2. 由于不需要等待设置的重传计时器到期，能尽早重传未被确认的报文段，能提高整个网络的吞吐量。

快恢复（与快重传配合使用）

1. 采用快恢复算法时，慢开始只在TCP连接建立时和网络出现超时时才使用。
2. 当发送方连续收到三个重复确认时，就执行“乘法减小”算法，把ssthresh门限减半。但是接下去并不执行慢开始算法。
3. 考虑到如果网络出现拥塞的话就不会收到好几个重复的确认，所以发送方现在认为网络可能没有出现拥塞。所以此时不执行慢开始算法，而是将cwnd设置为ssthresh的大小，然后执行拥塞避免算法。

注意

发送方窗口的上限值 = $\text{Min}(\text{接受窗口 } rwnd, \text{拥塞窗口 } cwnd)$

$rwnd > cwnd$ 接收方的接收能力限制发送方窗口的最大值

$rwnd < cwnd$ 网络的拥塞限制发送方窗口的最大值