@[toc]

# TCP 可靠传输

* **传输层** ： 使用TCP 实现可靠传输
* **网络层**：提供尽最大努力交付，不可靠传输

**可靠**：保证接收方进程从缓存区读出的字节流与发送方发出的字节流是完全一样的

## TCP 实现可靠传输的机制

1、校验 2、序号 3、确认 4、重传

一个字节占一个序号。  
序号字段指的是一个报文段第一个字节的序号。

* 确认重传不分家，TCP的发送方在**规定的时间**（重传时间）内没有收到确认就要重传已发送的报文段。*超时重传*
* TCP采用自适应算法，动态改变重传时间RTSs（加权平均往返时间）。

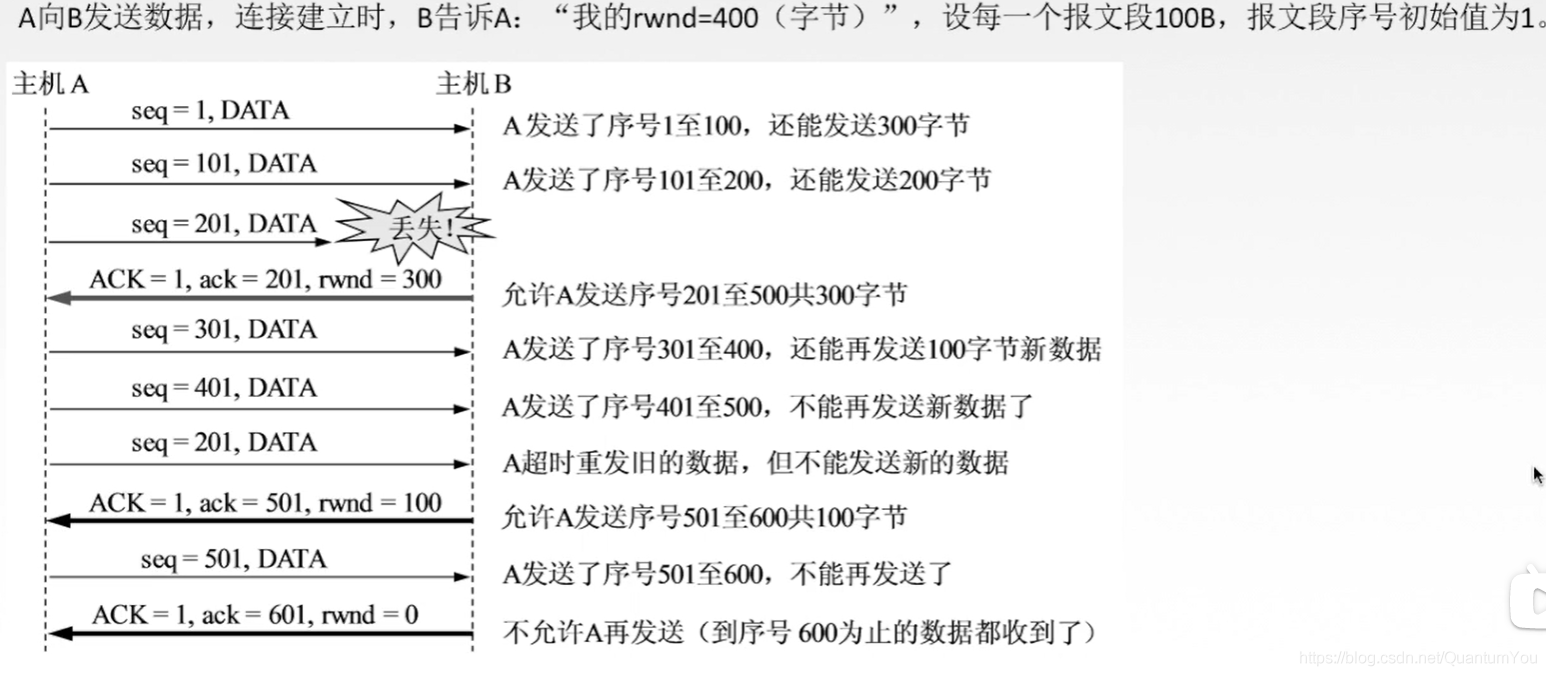
**冗余ACK**（冗余确认）

* 每当比期望序号大的失序报文段到达时，发送一个冗余ACK,指明下一个期待字节的序号。  
  发送方已发送1,2,3,4,5报文段
* 接收方收到1,返回给1的确认(确认号为2的第一个字节)
* 接收方收到3,仍返回给1的确认(确认号为2的第一个字节)
* 接收方收到4,仍返回给1的确认(确认号为2的第一个字节)
* 接收方收到5,仍返回给1的确认(确认号为2的第一个字节)
* 发送方收到3个对于报文段1的冗余ACK==> 认为2报文段丢失，重传2号报文段 *快速重传*

# TCP流量控制

* **流量控制**：让发送方慢点，要让接收方来得及接收
* TCP利用滑动窗口机制实现流量控制。
* 在通信过程中，接收方根据自己接收缓存的大小，动态地调整发送方的发送窗口大小，即接收窗口rwnd(接收方设置确认报文段的窗口字段来将rwnd通知给发送方），发送方的发送窗口取接收窗口rwnd和拥塞窗口cwnd的最小值。

TCP 协议不是利用停等协议



* TCP为每一个连接设有一个持续计时器，只要TCP连接的一方收  
  到对方的零窗口通知，就启动持续计时器。
* 若持续计时器设置的时间到期，就发送一个**零窗口探测报文段**接收方收到探测报文段时给出现在的窗口值。
* 若窗口仍然是0,那么发送方就重新设置持续计时器。

# TCP 拥塞控制

* **出现拥塞的条件**： 对资源需求的总和 > 可用资源  
  网络中有许多资源同时呈现供应不足 => 网络性能变坏 =>网络吞吐量将随输入负荷增大而下降
* **拥塞控制** : 防止过多的数据注入到网络中。 *全局性*

拥塞控制与流量控制的区别：

* 拥塞控制一对多 ，流量控制一对一

## 拥塞控制的四种算法

* 慢开始、拥塞避免、快重传、快恢复

假定  
1、数据单方向传送，而另一个方向只传送确认  
2、接收方总是有足够大的缓存空间，因而发送窗口大小取决于拥塞程度

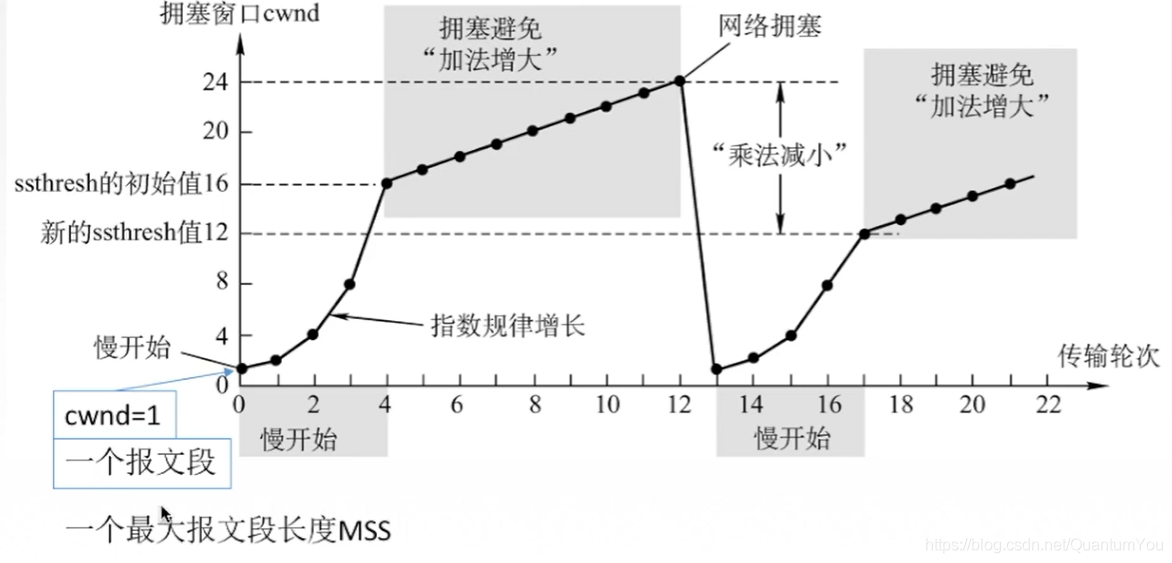
发送窗口=Min { 接收窗口rwnd,拥塞窗口cwnd }

* **接收窗口** ： 接收方根据接受缓存设置的值，并告知给发送方，反映接收方容量
* **拥塞窗口** ： 发送方根据自己估算的网络拥塞程度而设置的窗口值，反映网络当前容量。

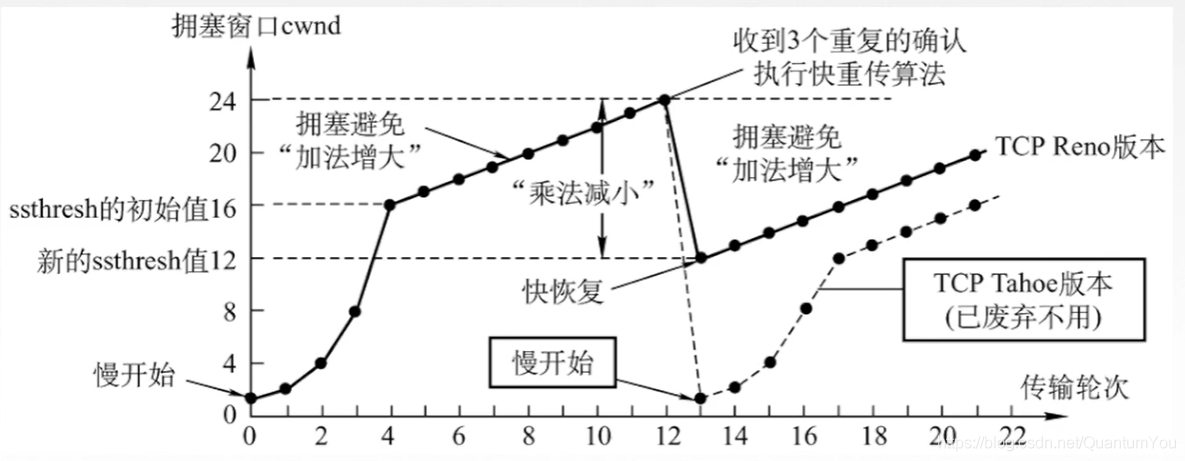
### 慢开始、拥塞避免

**一个传输轮次** ：发送了一批报文段，并收到它们的确认的时间。

* 一个往返时延RTT. 开始发送一批拥塞窗口内的报文段到开始发送下一批拥塞窗口内的报文段的时间。



### 快重传和快恢复



# 本章总结

