



## TCP的重发机制

- 毕竟出错会发生 → 考虑重发 → 定时器超时时延的设计。
- Tr层超时时间的设计:
  $t_{out} \approx \lceil 2t_p \rceil$  设  $t_p \sim G(\mu, \sigma)$  where  $\sigma_{Tr} \gg \sigma_{DL}$

- 采用线性动态系统的思想:
 系统的当前状态由系统的惯性与测量（新息）的凸组合来估计。
 
$$T_n = \alpha T_{n-1} + (1-\alpha) \hat{T}_n \quad t_{out} = \beta T_n$$
 测量的内容：从报文发出到收回确认之间的时间间隔  
 可以保证稳态情况下的收敛（图）
- However, 测量环节存在着不确定性 (uncertainty)
- Karn算法思想:  $T_n = \gamma T_{n-1}$   
 重发发生时，不再信赖测量，而是信赖惯性 (inertia)，并逐步放大 (why? )

## TCP的编号机制

- 异于HDLC
  - 按字节编号：
- 同于HDLC
  - 发送序号定义：待传报文段首字节序号；
  - 确认序号定义：已收到数据的末字节序号 + 1 (期望下次收到的首字节序号)；
  - piggybacking

## TCP的流控机制

- 流控的定义
  - 如何传递控制参数——TCP header中的WIN字段
- 基于通知窗口的流控：
  - 发送窗口 = 接收窗口 (通知窗口) : 选择重传ARQ
  - 例：(滑动窗口的工作)

## TCP的流控机制

- 基于拥塞窗口的流控：
  - 发送端根据网络拥塞情况自行决定窗口尺寸；
  - 慢启动和拥塞避免技术；
- 实际中的TCP流控——木桶法则：

发送窗口 =  $\text{Min}[\text{通知窗口}, \text{拥塞窗口}]$

拥塞窗口原理

## TCP面向连接的管理

- Establish a TCP connection — 三次握手原理

三次握手的过程

二次握手的fail case

