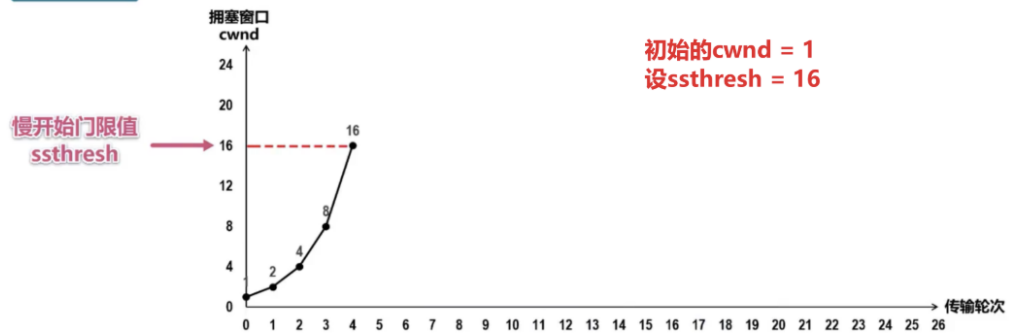


- TCP的拥塞控制

- 基于滑动窗口，cwnd：发送窗口

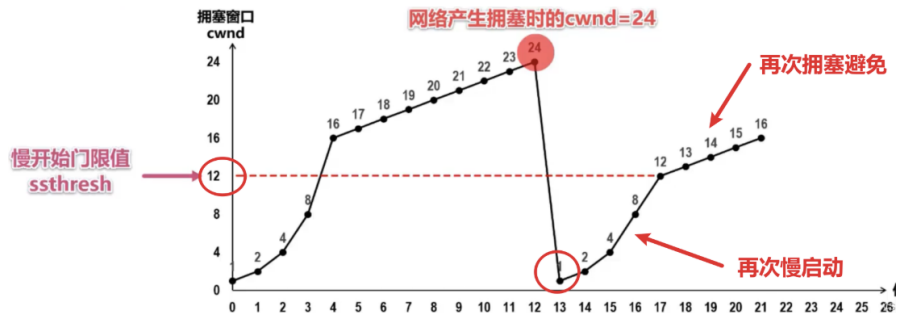
- 慢开始

- 通常在一条TCP连接开始时，cwnd被设置为1个MSS（最大报文段），也即cwnd=1
- 每当TCP发送方将发送窗口的数据发送完，并顺利接收到所有的确认后，就会将拥塞窗口大小翻倍，也即慢启动阶段，cwnd以指数形式增长。
- 拥塞窗口会一直增长直到到达慢开始门限sssthresh，开始执行拥塞避免算法



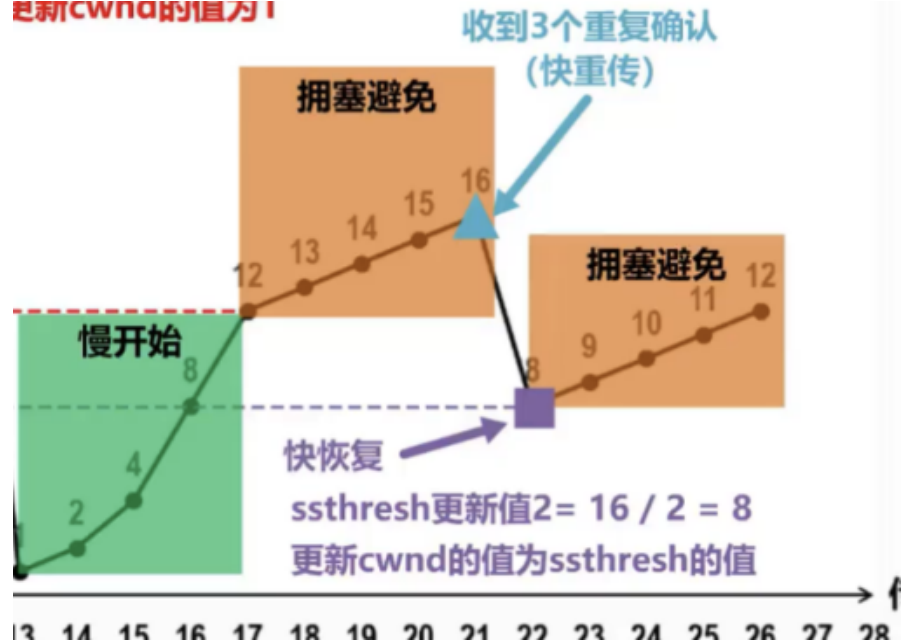
- 拥塞避免

- 该阶段的拥塞窗口变为线性增长，每次cwnd+1，也即每次增加一个MSS
- 随着拥塞窗口的增加，发送速率不断提高，当TCP遇到：
 - 分组超时重传时，即认为发生了网络拥塞
 - 更新cwnd的值为1，此时将更新sssthresh的值为当前拥塞窗口的一半，更新为24的一半即12
 - 然后继续执行慢启动—拥塞避免



- 收到连续的3个重复确认，则认为是正常的网络包丢失，而不是网络拥塞造成的（这正是快重传算法的功劳）
 - 重传丢失的分组

- 执行快恢复算法
更新cwnd的值为1



- 快恢复
 - 如果发送方收到了3个重复确认，就执行快恢复算法
 - 将慢开始门限ssthresh和拥塞窗口cwnd都设置为当前拥塞窗口的一半，然后执行拥塞避免算法。
- 快重传
 - 让发送方尽快重传，而不是等待超时重传计时器超时再重传。
- 一个综合了4种拥塞控制算法的例子

