TCP 拥塞控制

TCP 拥塞控制四大阶段

- 慢启动
- 拥塞避免
- 快速重传
- 快速恢复

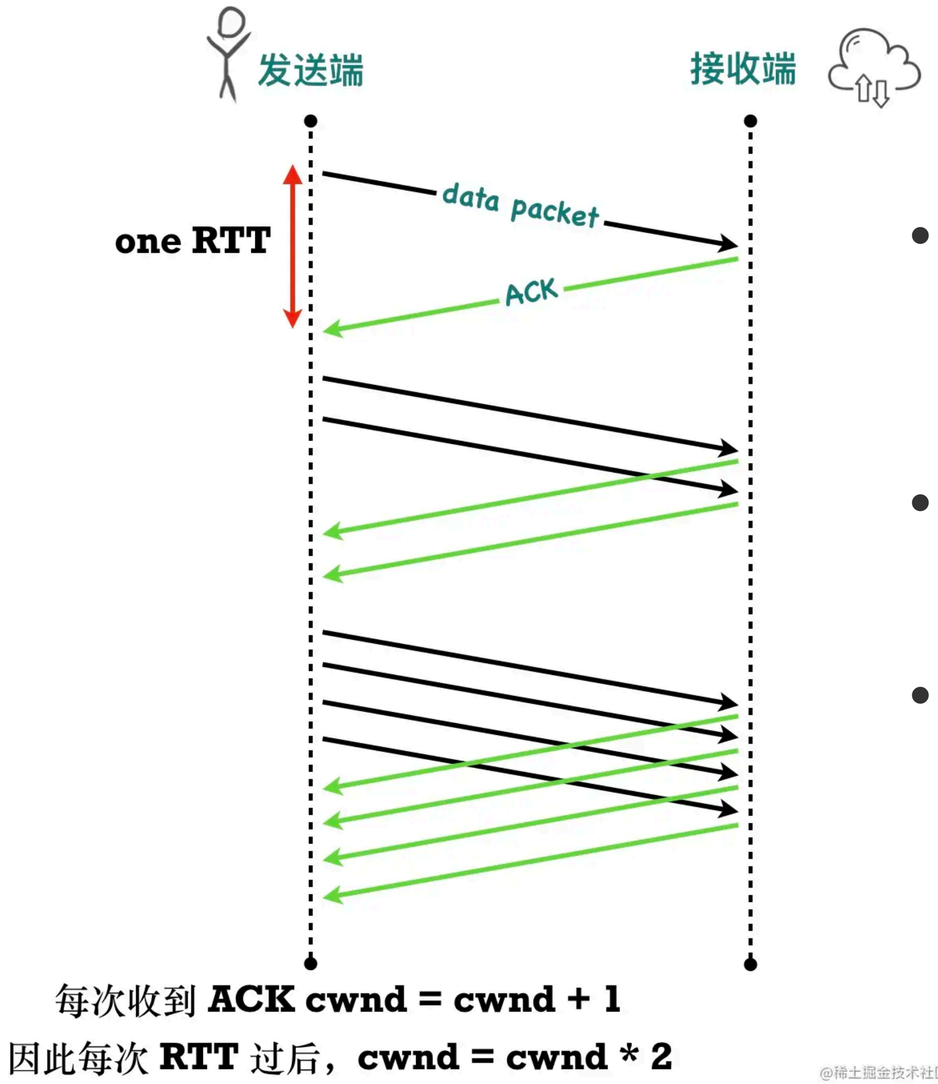
拥塞控制 | 慢启动

在连接建立之初，应该发多少数据给接收端才是合适的呢？

你不知道对端有多快，如果有足够的带宽，你可以选择用最快的速度传输数据，但是如果是一个缓慢的移动网络呢？如果发送的数据过多，只是造成更大的网络延迟。这是基于整个考虑，每个 TCP 连接都有一个拥塞窗口的限制，最初这个值很小，随着时间的推移，每次发送的数据量如果在不丢包的情况下，“慢慢”的递增，这种机制被称为「慢启动」

拥塞控制

慢启动

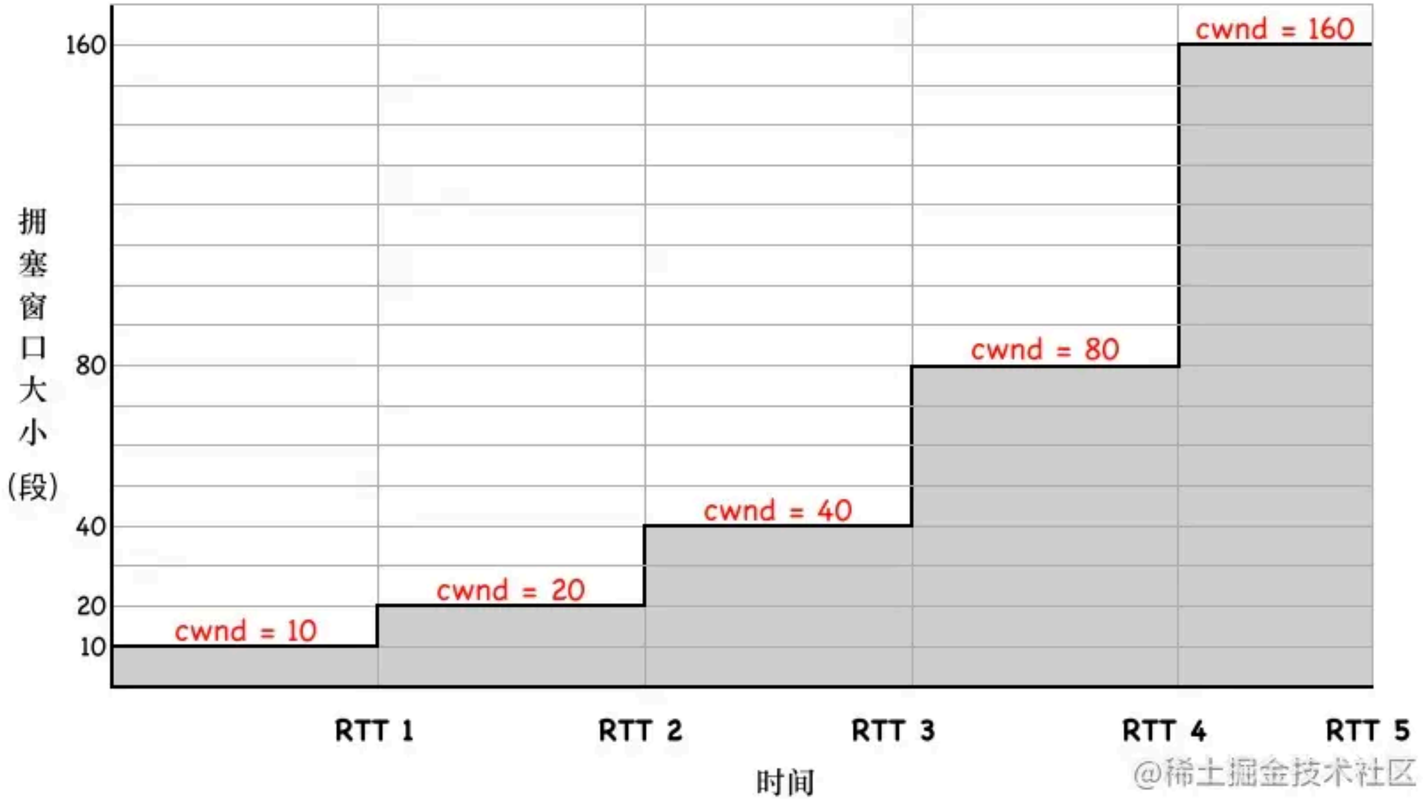


- 第一步，三次握手以后，双方通过 ACK 告诉了对方自己的接收窗口（rwnd）的大小，之后就可以互相发数据了
- 第二步，通信双方各自初始化自己的「拥塞窗口」（Congestion Window，cwnd）大小。
- 第三步，cwnd 初始值较小时，每收到一个 ACK，cwnd + 1，每经过一个 RTT，cwnd 变为之前的两倍。

@稀土掘金技术社区

拥塞窗口随时间的变化关系

慢启动



慢启动阈值

（Slow Start Threshold，ssthresh）

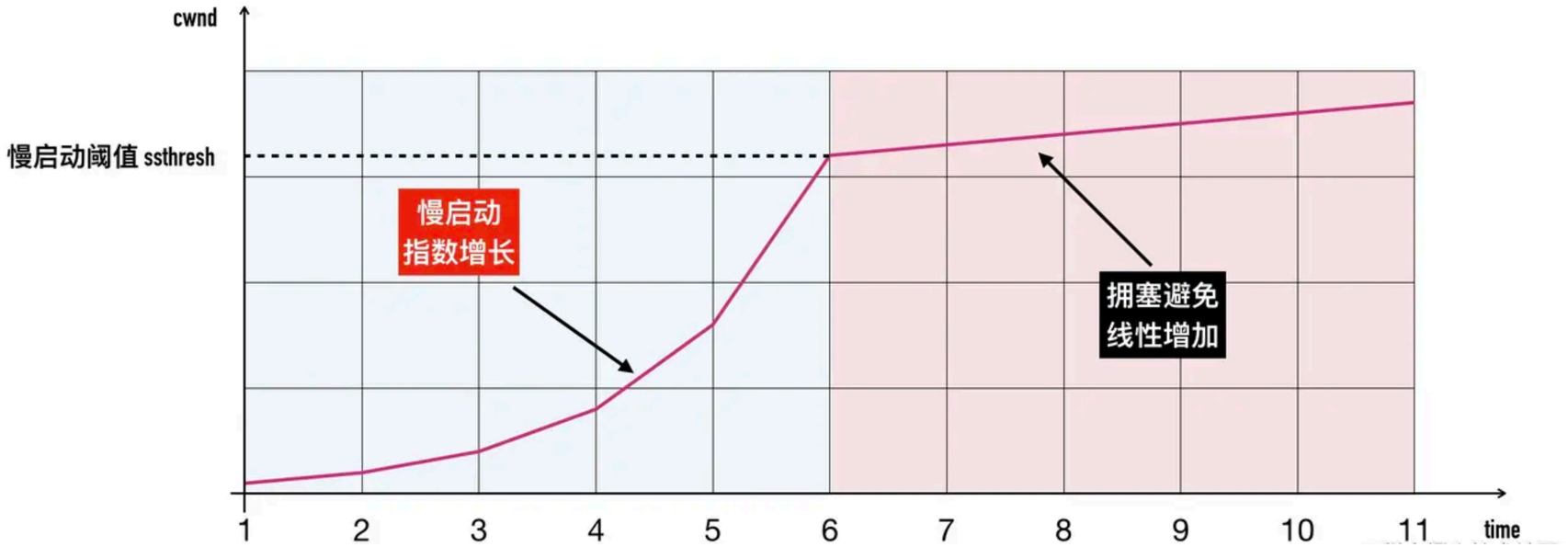
慢启动拥塞窗口（cwnd）肯定不能无止境的指数级增长下去，否则拥塞控制就变成了「拥塞失控」了，它的阈值称为「慢启动阈值」（Slow Start Threshold，ssthresh）。ssthresh 就是一道刹车，让拥塞窗口别涨那么快。

- 当 cwnd < ssthresh 时，拥塞窗口按指数级增长（慢启动）
- 当 cwnd > ssthresh 时，拥塞窗口按线性增长（拥塞避免）

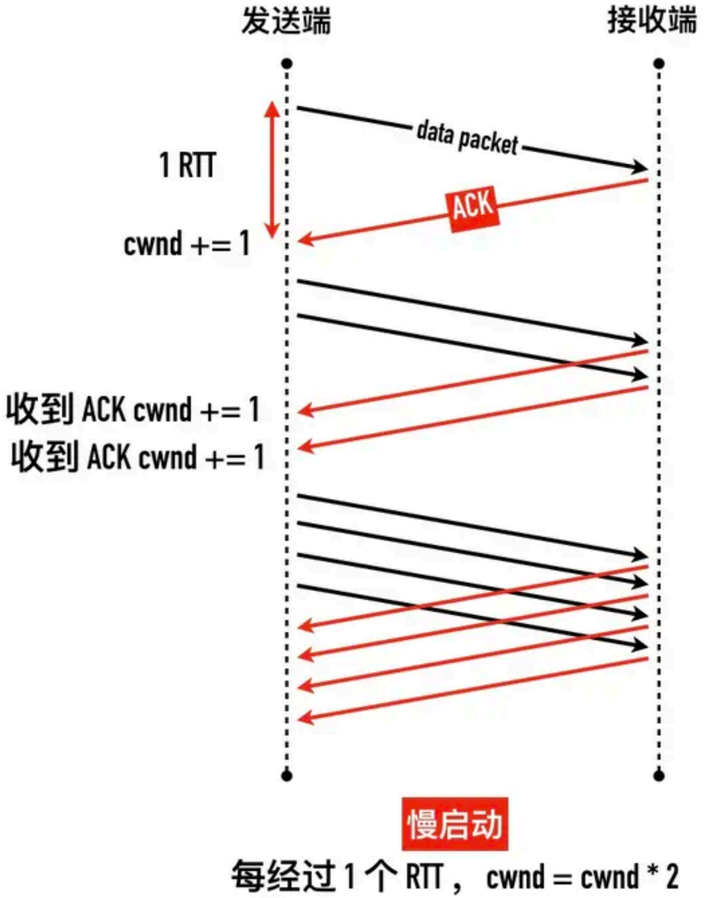
拥塞避免

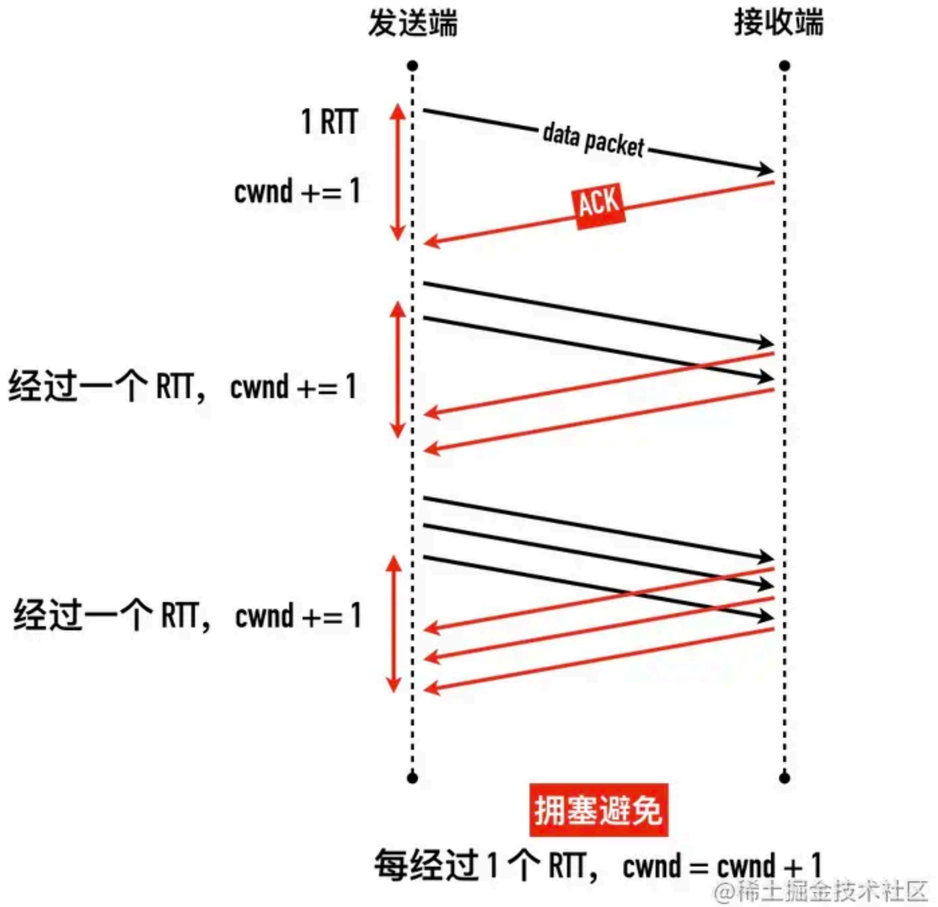
（Congestion Avoidance）

当 cwnd > ssthresh 时，拥塞窗口进入「拥塞避免」阶段，在这个阶段，每一个往返 RTT，拥塞窗口大约增加 1 个 MSS 大小，直到检测到拥塞为止。



慢启动VS拥塞避免

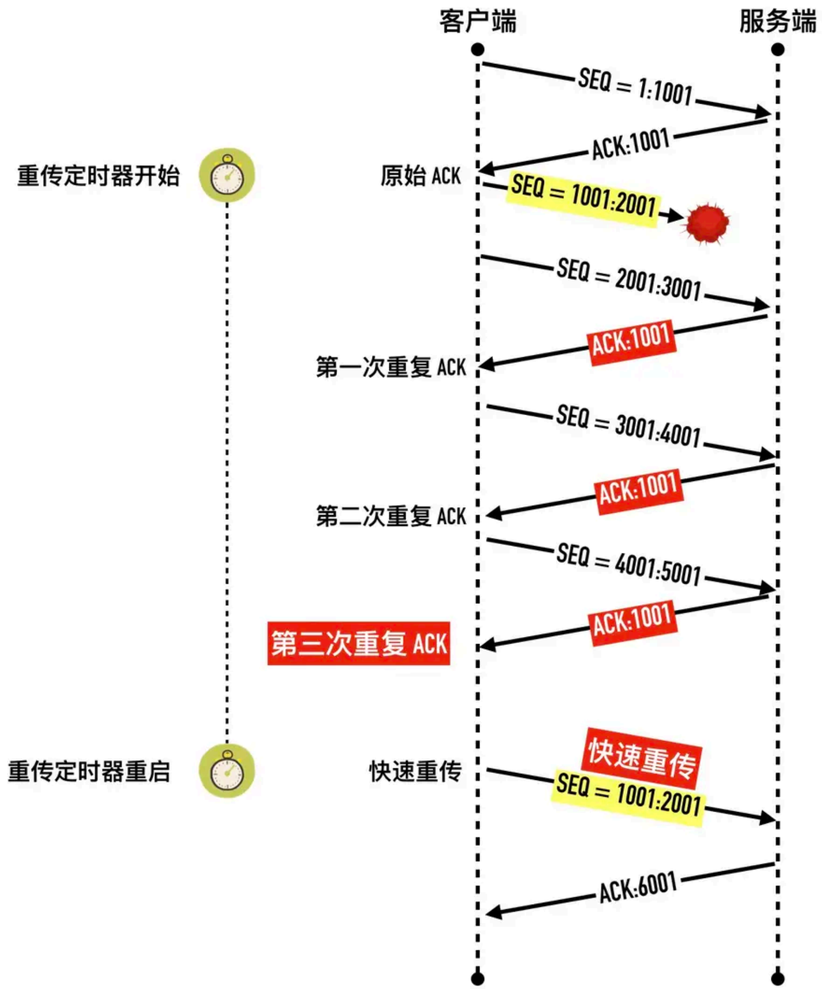




@稀土掘金技术社区

快速重传

(Fast Retransmit)



快速恢复

(Fast Retransmit)

当收到三次重复 ACK 时，进入快速恢复阶段。解释为网络轻度拥塞。

- 拥塞阈值 ssthresh 降低为 cwnd 的一半：ssthresh = cwnd / 2
- 拥塞窗口 cwnd 设置为 ssthresh
- 拥塞窗口线性增加

一个实际包分析