



图 3-54 加性增、乘性减的拥塞控制

TCP AIMD 算法基于大量的工程见解和在运行网络中的拥塞控制经验而开发。在 TCP 研发后的十年, 理论分析显示 TCP 的拥塞控制算法用做一种分布式异步优化算法, 使得用户和网络性能的几个重要方面被同时优化 [Kelly 1998]。拥塞控制的丰富理论已经得到发展 [Srikant 2004]。

### 5. 对 TCP 吞吐量的宏观描述

给出 TCP 的锯齿状行为后, 自然要考虑一个长存活期的 TCP 连接的平均吞吐量 (即平均速率) 可能是多少。在这个分析中, 我们将忽略在超时事件后出现的慢启动阶段。(这些阶段通常非常短, 因为发送方很快就以指数增长离开该阶段。) 在一个特定的往返间隔内, TCP 发送数据的速率是拥塞窗口与当前 RTT 的函数。当窗口长度是  $w$  字节, 且当前往返时间是 RTT 秒时, 则 TCP 的发送速率大约是  $w/\text{RTT}$ 。于是, TCP 通过每经过 1 个 RTT 将  $w$  增加 1 个 MSS 探测出额外的带宽, 直到一个丢包事件发生为止。当一个丢包事件发生时, 用  $W$  表示  $w$  的值。假设在连接持续期间 RTT 和  $W$  几乎不变, 那么 TCP 的传输速率在  $W/(2 \times \text{RTT})$  到  $W/\text{RTT}$  之间变化。

这些假设导出了 TCP 稳态行为的一个高度简化的宏观模型。当速率增长至  $W/\text{RTT}$  时, 网络丢弃来自连接的分组; 然后发送速率就会减半, 进而每过一个 RTT 就发送速率增加  $\text{MSS}/\text{RTT}$ , 直到再次达到  $W/\text{RTT}$  为止。这一过程不断地自我重复。因为 TCP 吞吐量 (即速率) 在两个极值之间线性增长, 所以我们有

$$\text{一条连接的平均吞吐量} = \frac{0.75 \times W}{\text{RTT}}$$

通过这个高度理想化的 TCP 稳态动态性模型, 我们可以推出一个将连接的丢包率与可用带宽联系起来的有趣表达式 [Mahdavi 1997]。这个推导将在课后习题中概要给出。一个根据经验建立的并与测量数据一致的更复杂模型参见 [Padhye 2000]。

### 6. 经高带宽路径的 TCP

认识到下列事实是重要的: TCP 拥塞控制已经演化了多年并仍在继续演化。对当前 TCP 变量的总结和 TCP 演化的讨论, 参见 [Floyd 2001, RFC 5681, Afanasyev 2010]。以往对因特网有益的东西 (那时大量的 TCP 连接承载的是 SMTP、FTP 和 Telnet 流量), 不一定对当今 HTTP 主宰的因特网或具有难以想象的服务的未来因特网还是有益的。