TCP 拥塞避免算法浏览

LeanCloud 郭瑞

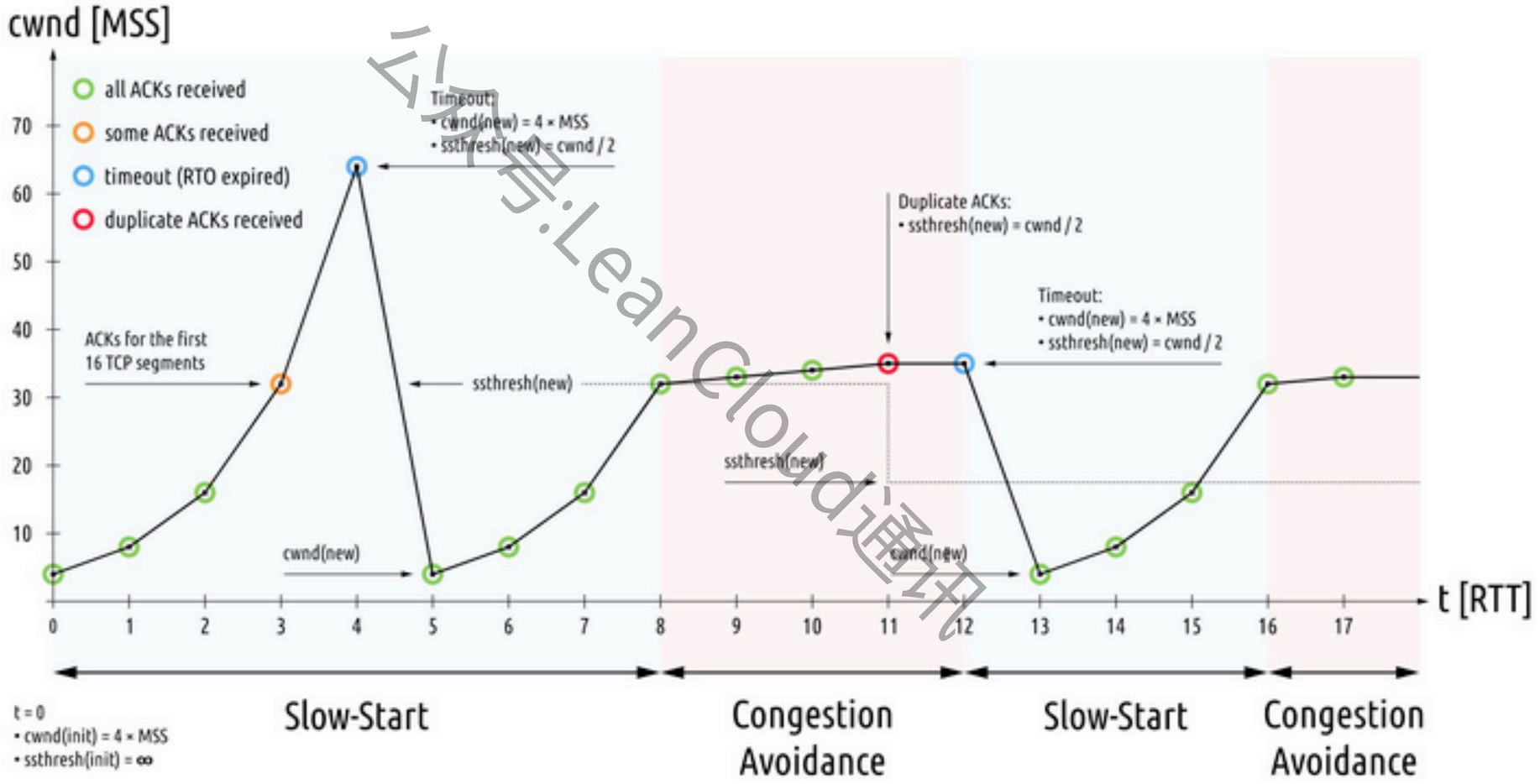
主要内容

1. TCP 部分内容回顾
2. Reno
3. BIC
4. Cubic
5. Vegas
6. BBR

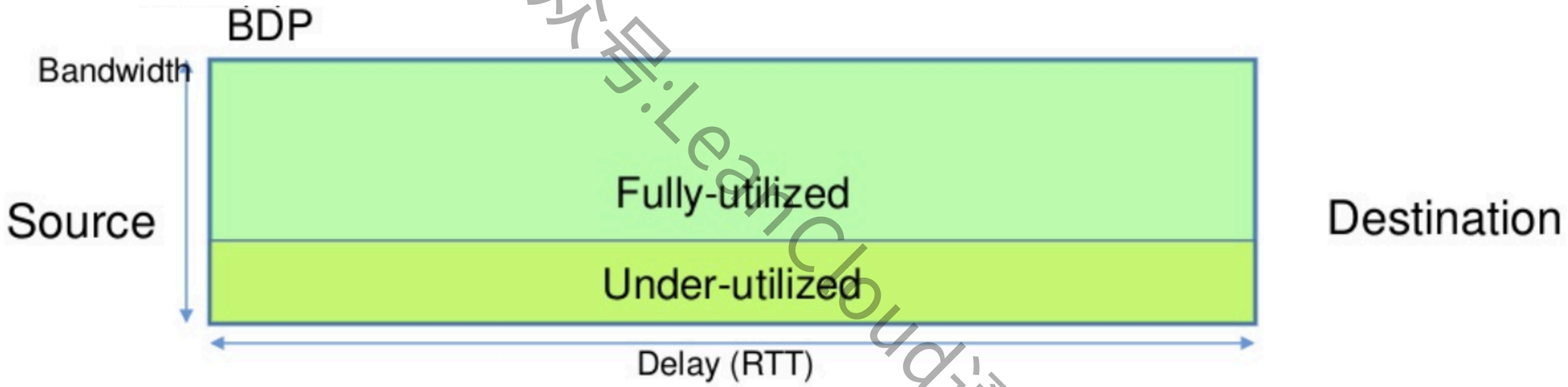
TCP 内的两个窗口

- RWND, Receiver Window
- CWND, Congestion Window

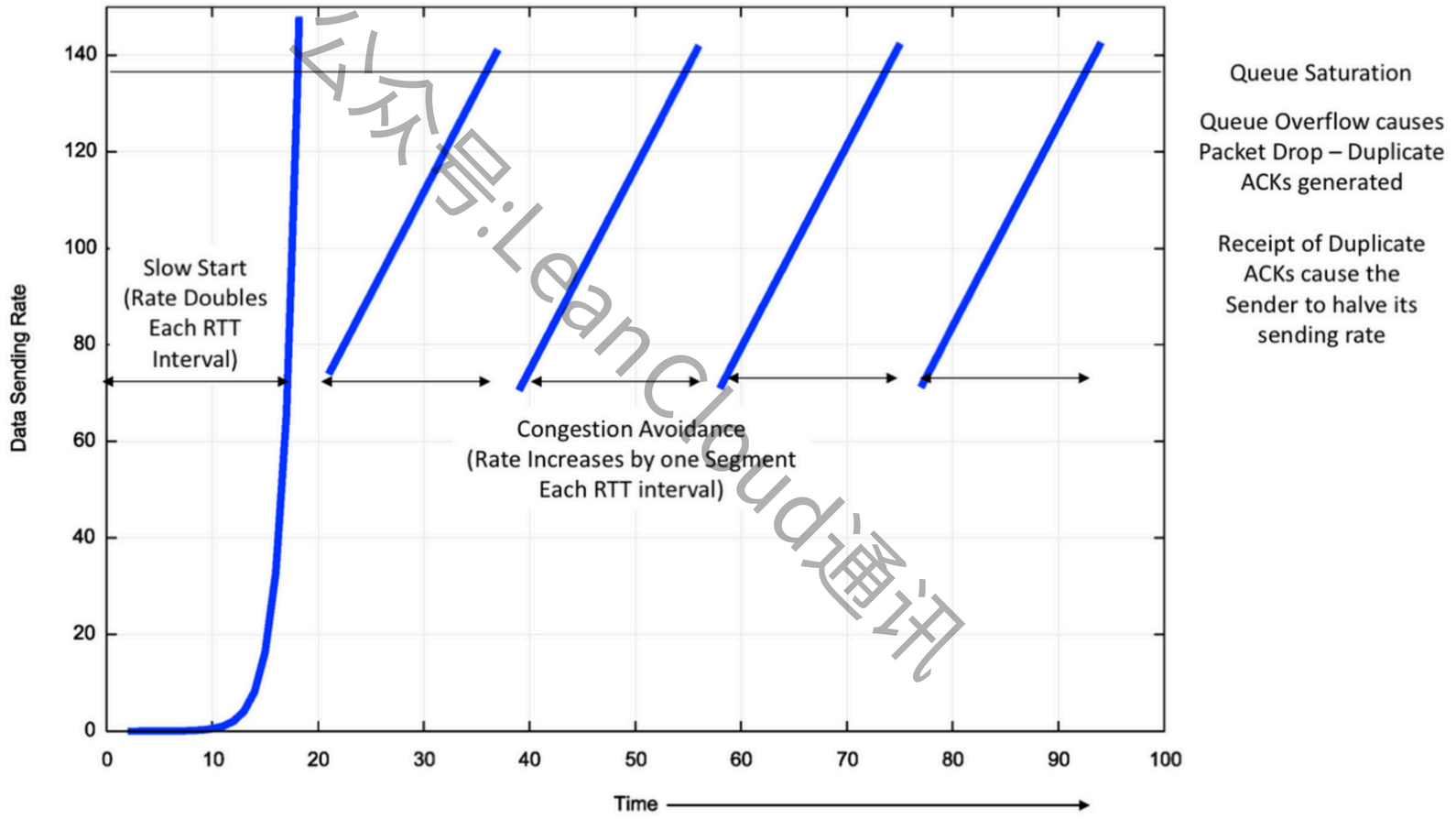
CWND 变化过程



BDP



Reno



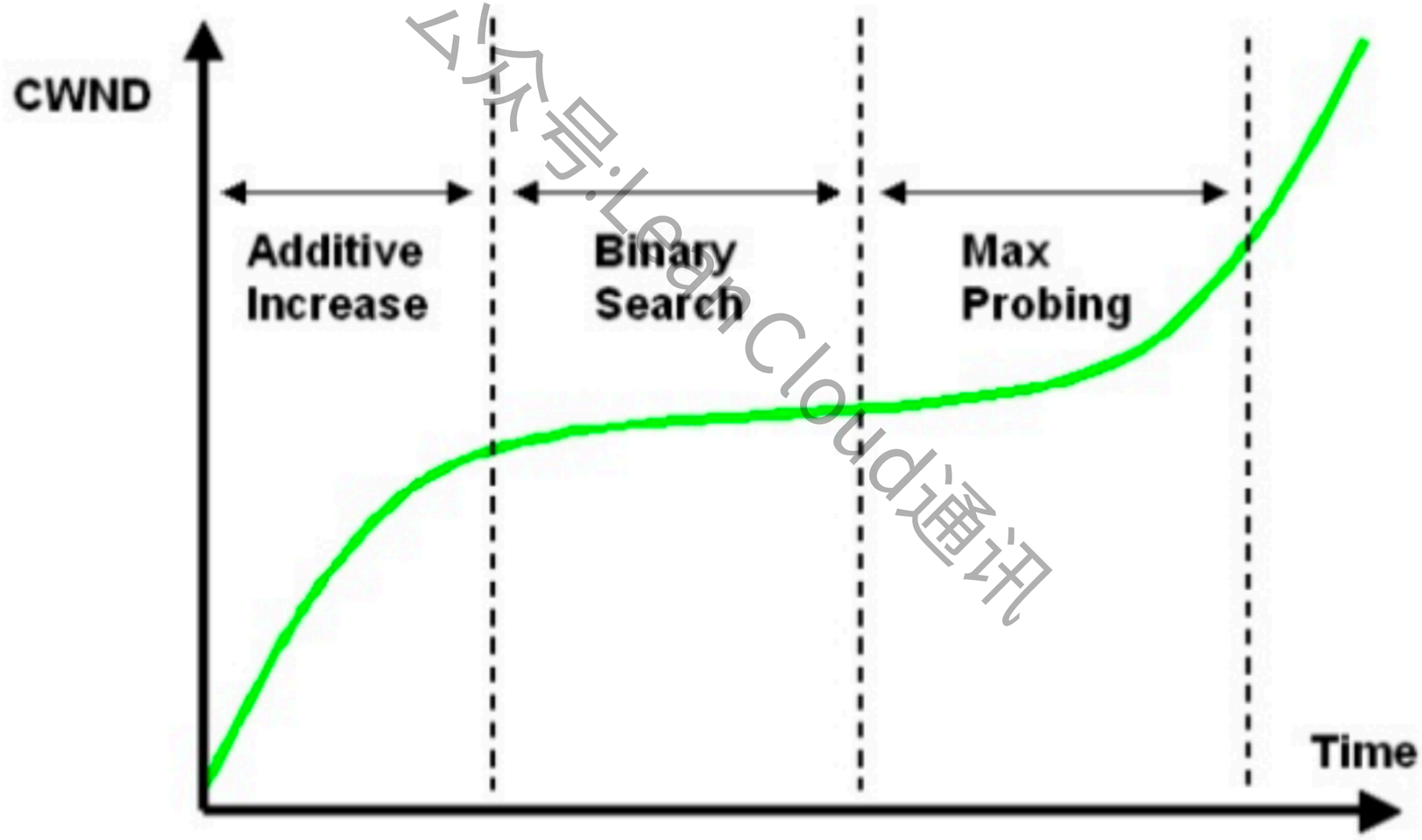
Reno 的假设

- 丢包一定因为网络出现拥塞
- 网络的 RTT 和带宽稳定不容易变化
- 将速率减半以后，一定能清空 Buffer

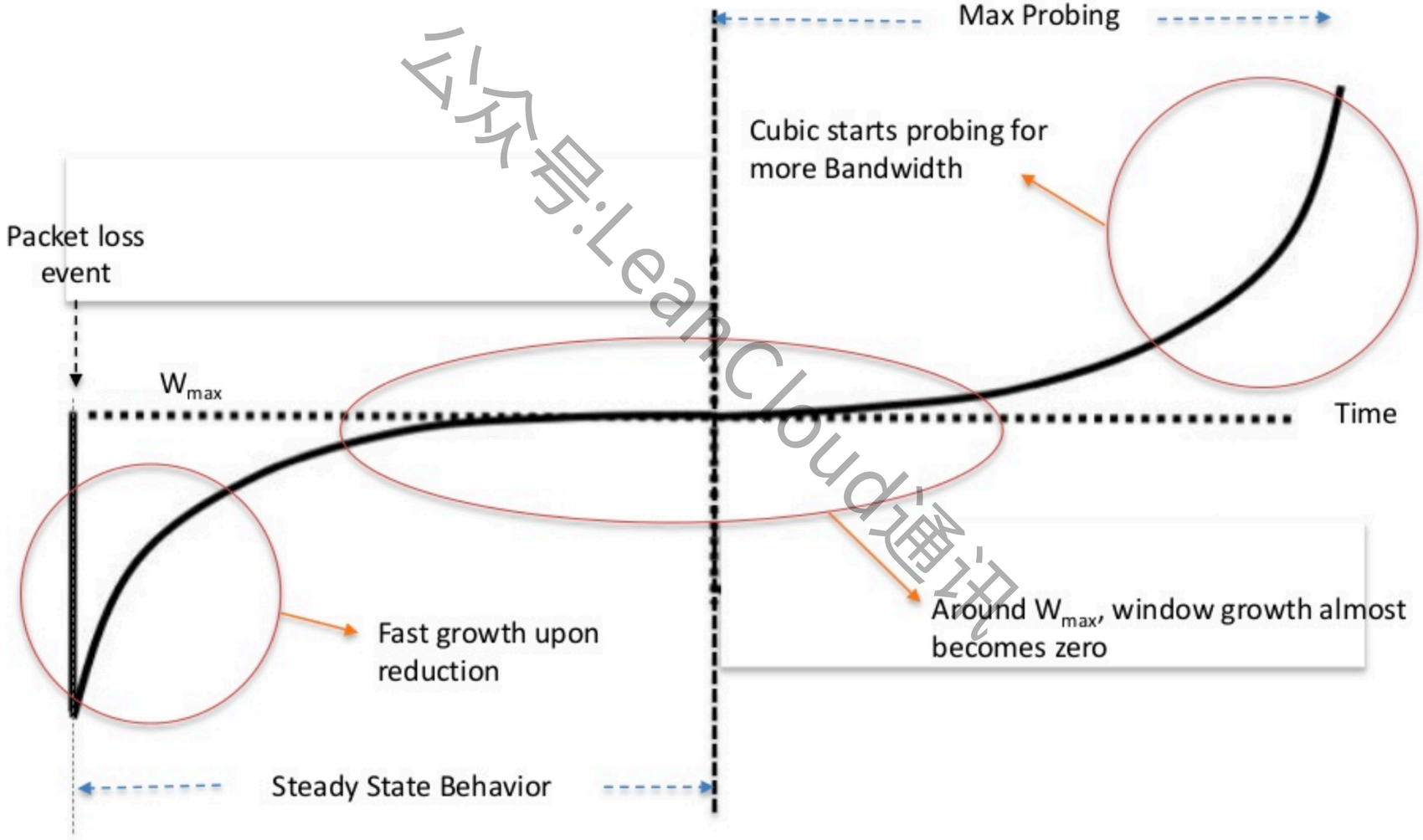
Reno 问题

- 受链路 Buffer 影响很大
- 对高带宽网络利用率低
- 对共享链路的其它 RTT 较大的连接不友好

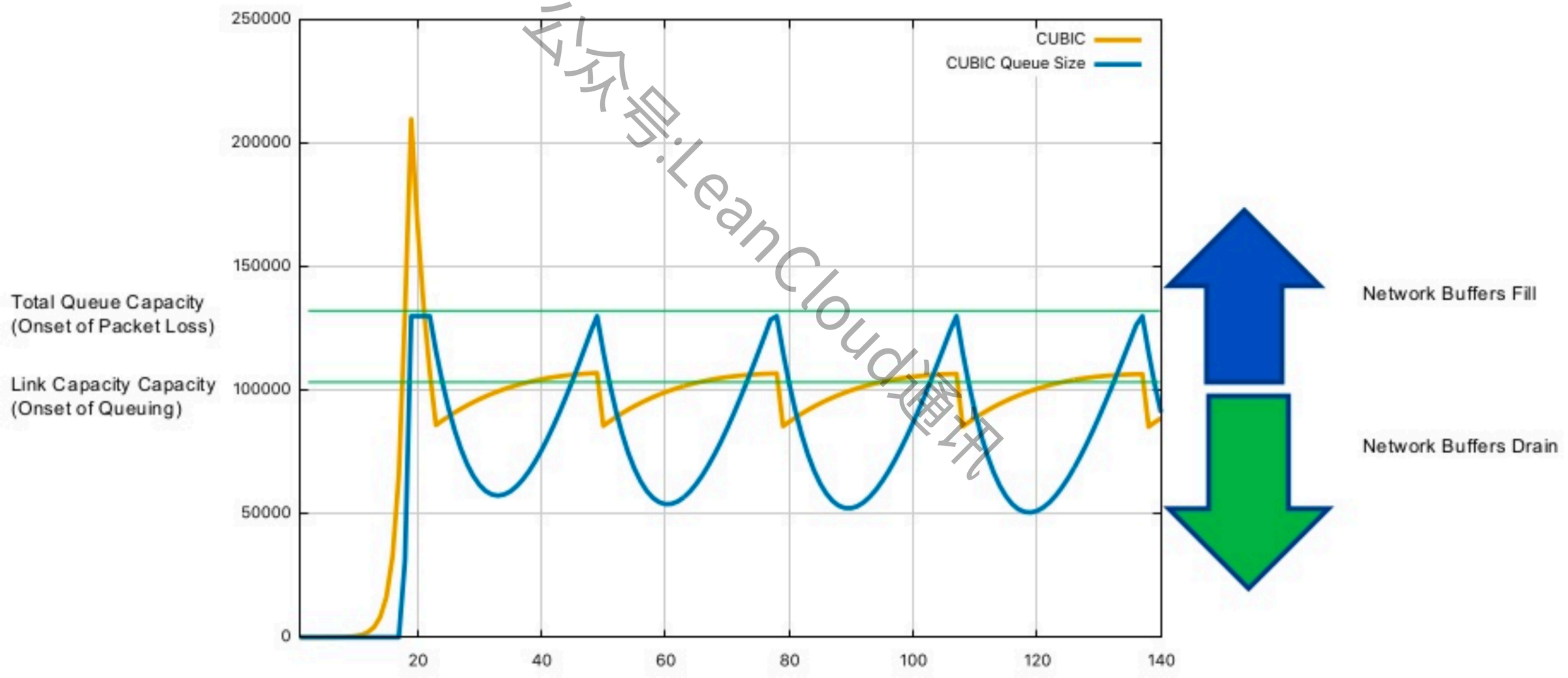
BIC



Cubic



Cubic



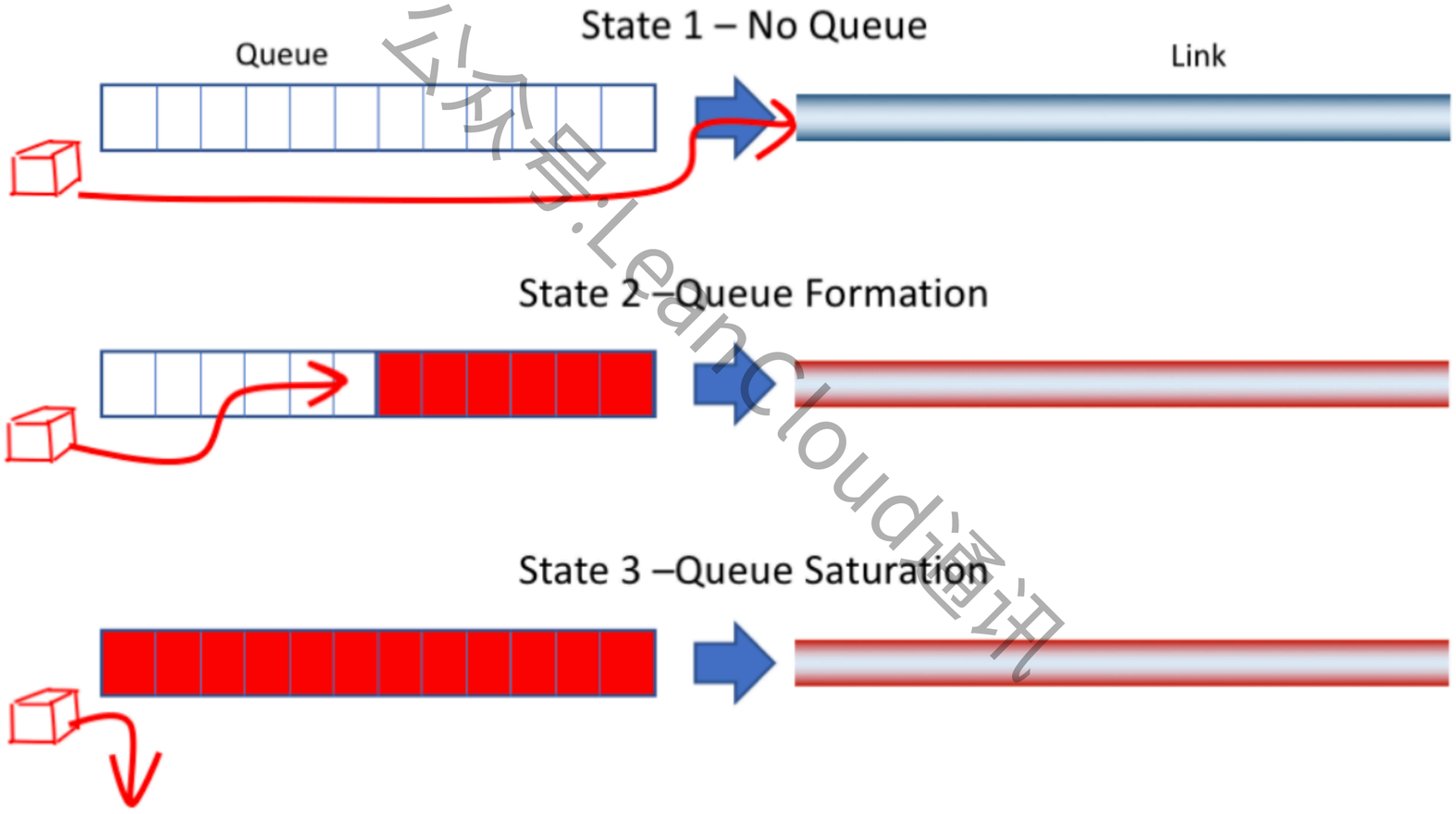
Cubic 优点

- 因为跟 RTT 无关所以更公平
- 更适合 BDP 大的网络

Cubic 缺点

- 当 Bandwidth 变化时候，CWND 跟随慢
- 更易导致 Bufferbloat

队列模型



Vegas

- 会监控 RTT
- 会尝试增加发送速率来探测链路带宽
- 如果丢包或者 RTT 增大就降低发送速率

Vegas

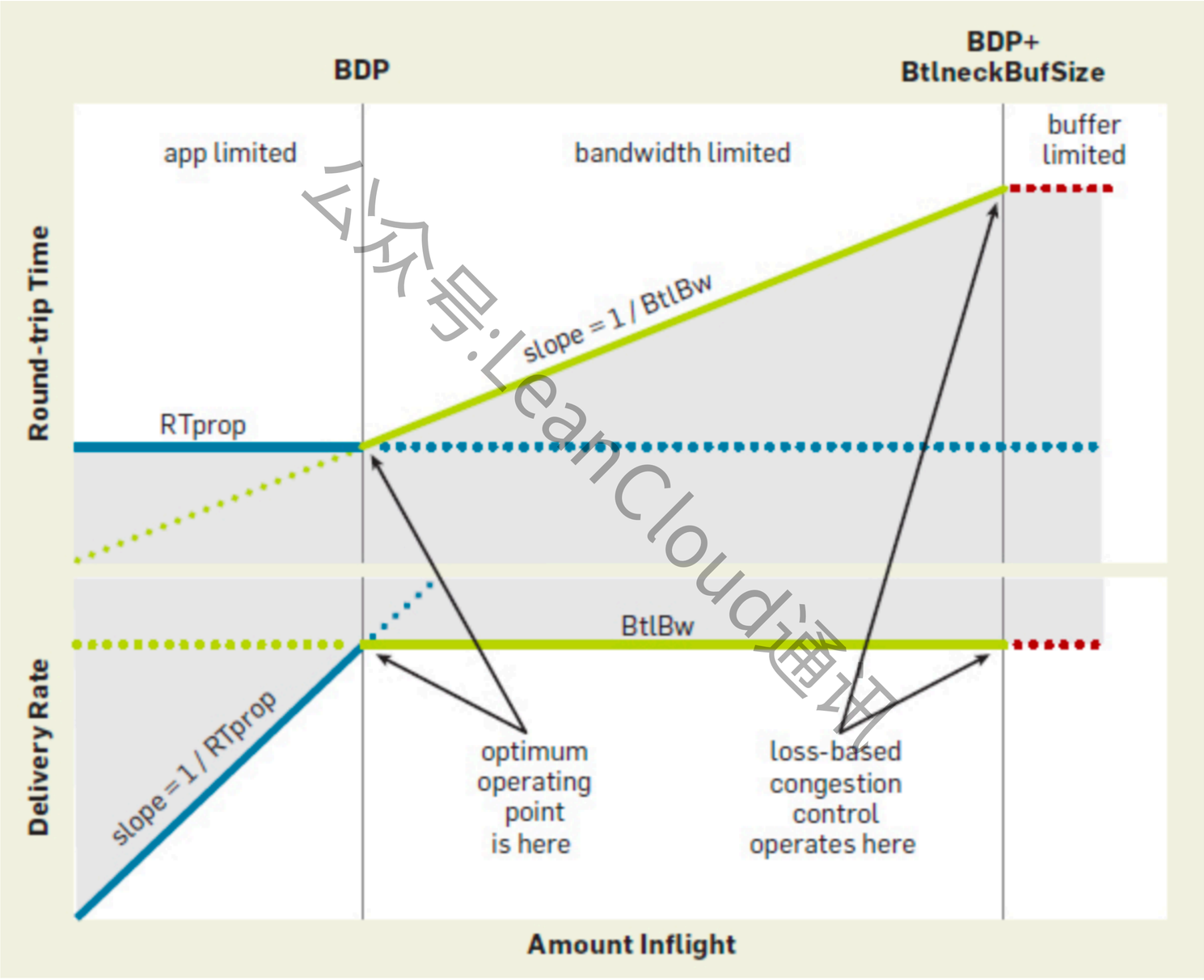
- CWND 增长是线性的，不能很好利用网络传输速率
- 不能跟基于丢包的算法共存

估计 BtlBw 和 RTprop

$$RTT_t = RTprop_t + \eta_t$$

$$\widehat{RTprop} = RTprop + \min(\eta_t) = \min(RTT_t) \quad \forall t \in [T - W_R, T]$$

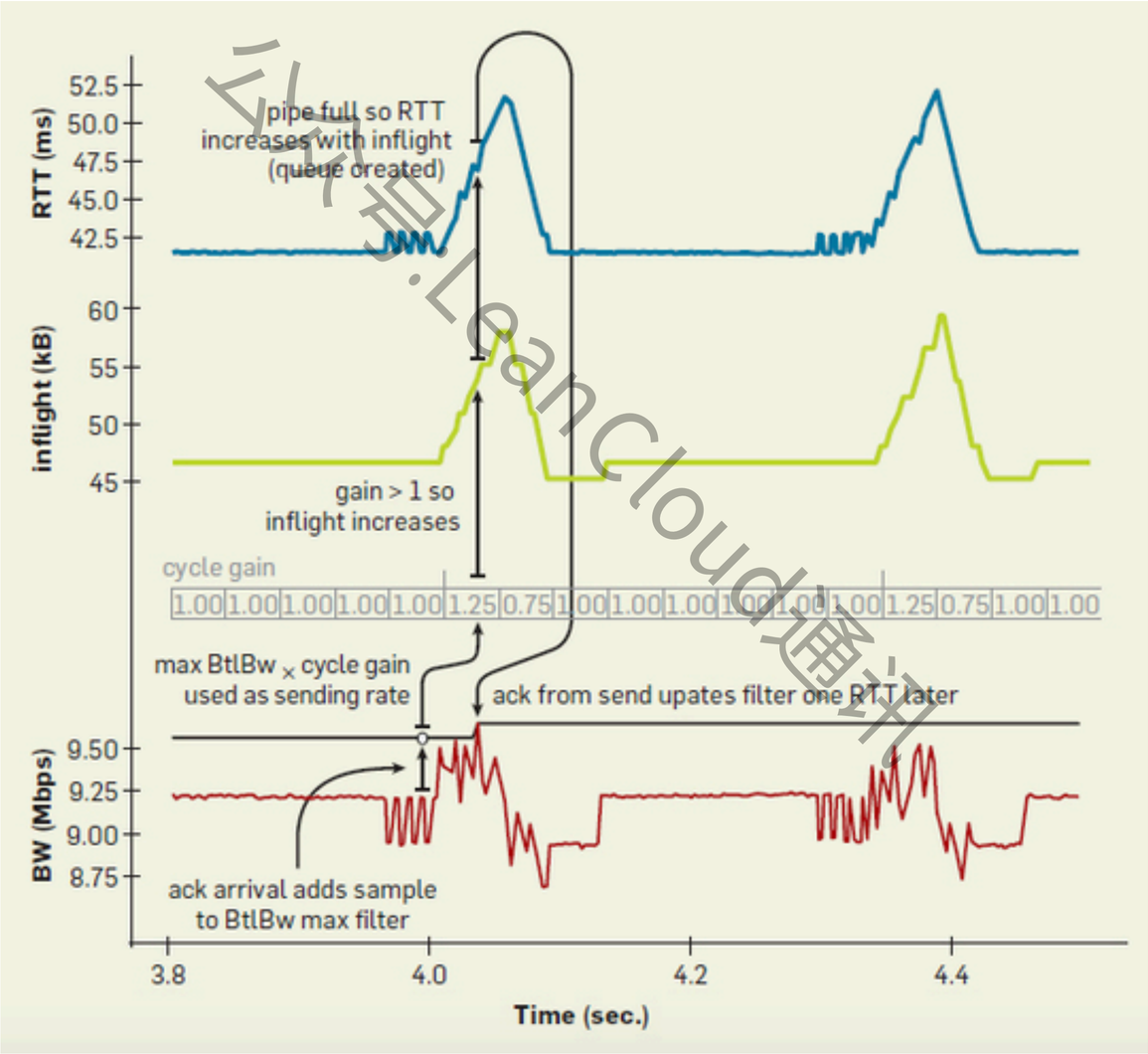
$$\widehat{BtlBw} = \max(deliveryRate_t) \quad \forall t \in [T - W_B, T]$$



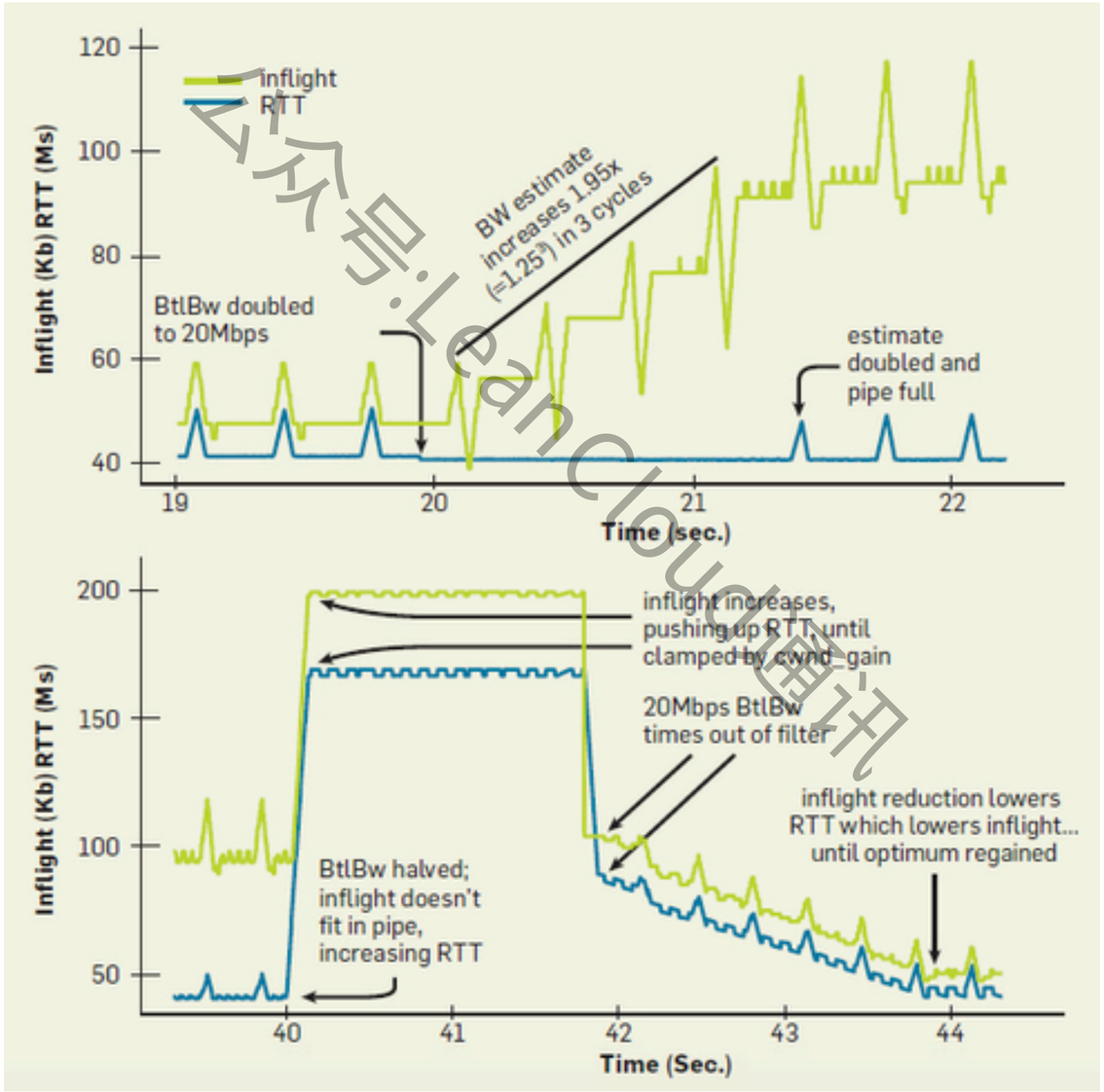
BBR 状态

- Startup
- Drain
- ProbeBW
- ProbeRTT

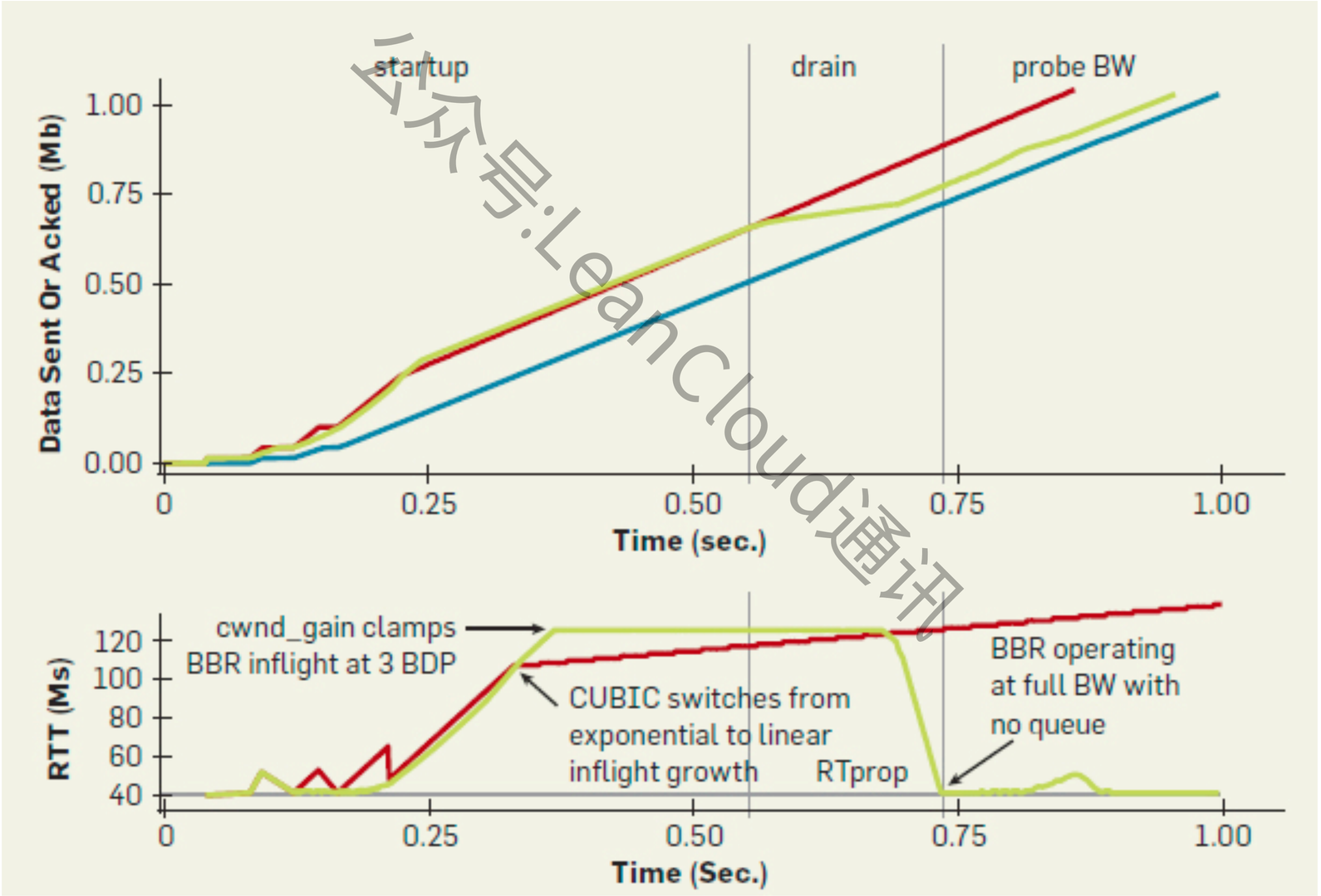
BBR 工作过程



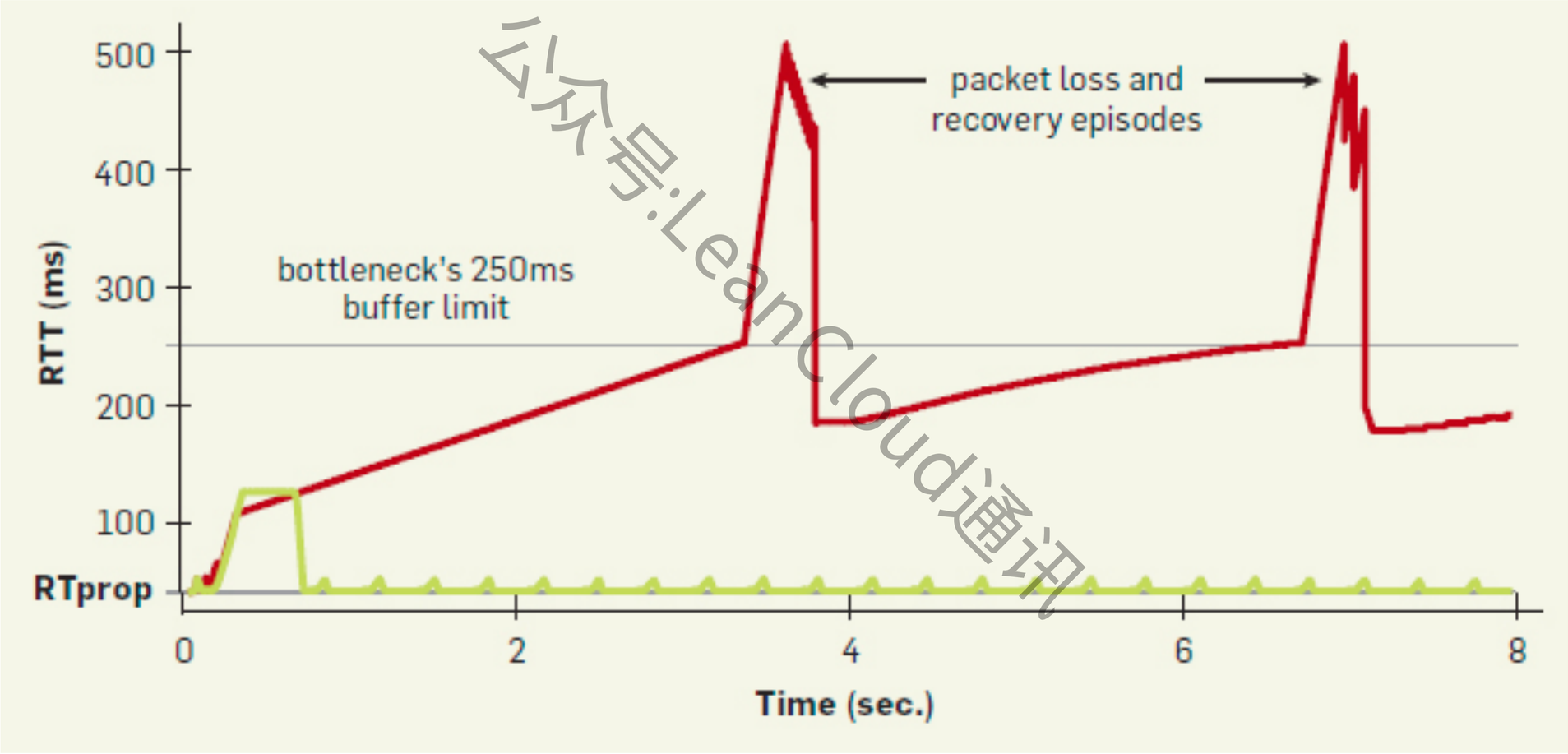
带宽变化时 BBR 工作过程



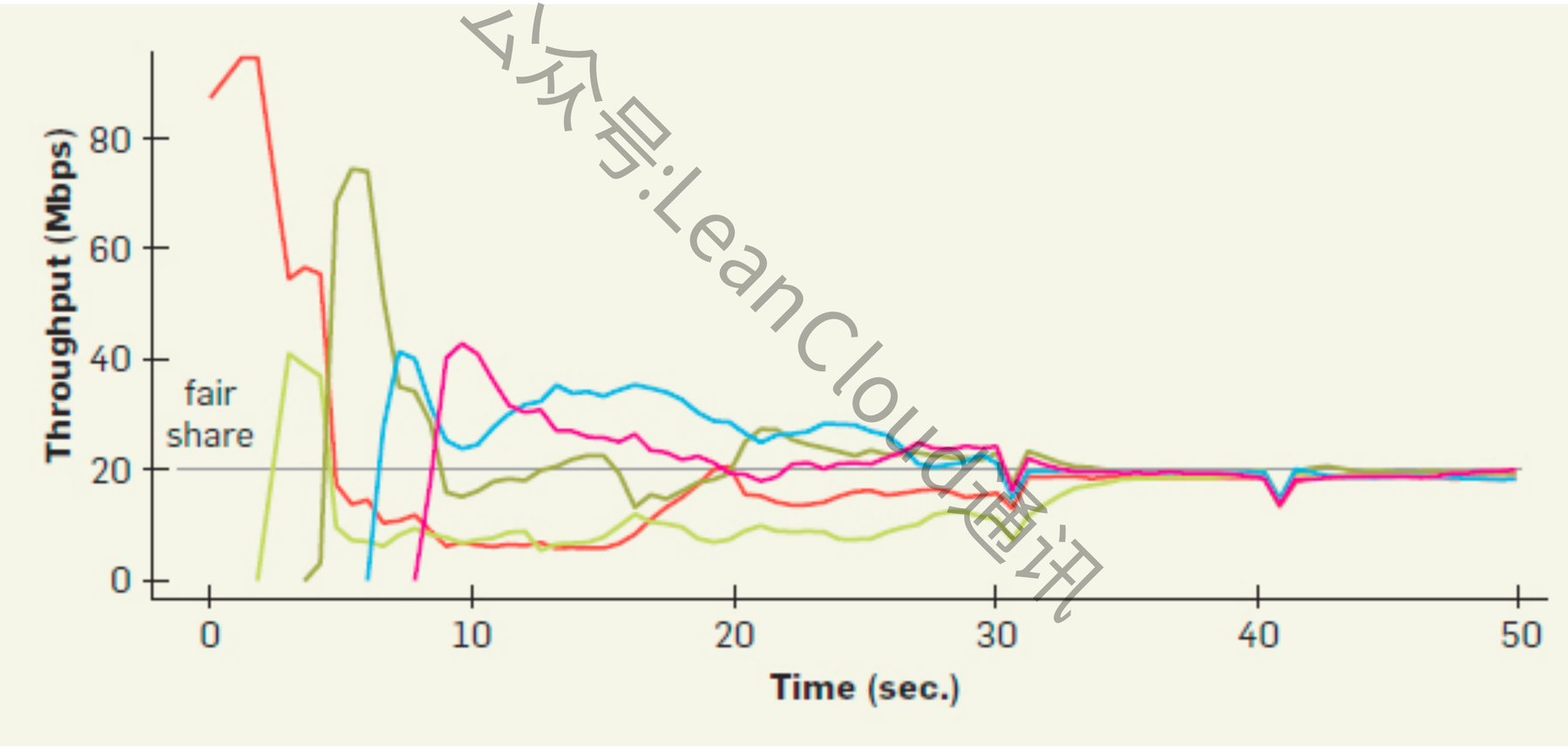
BBR Startup 阶段和 Cubic 对比



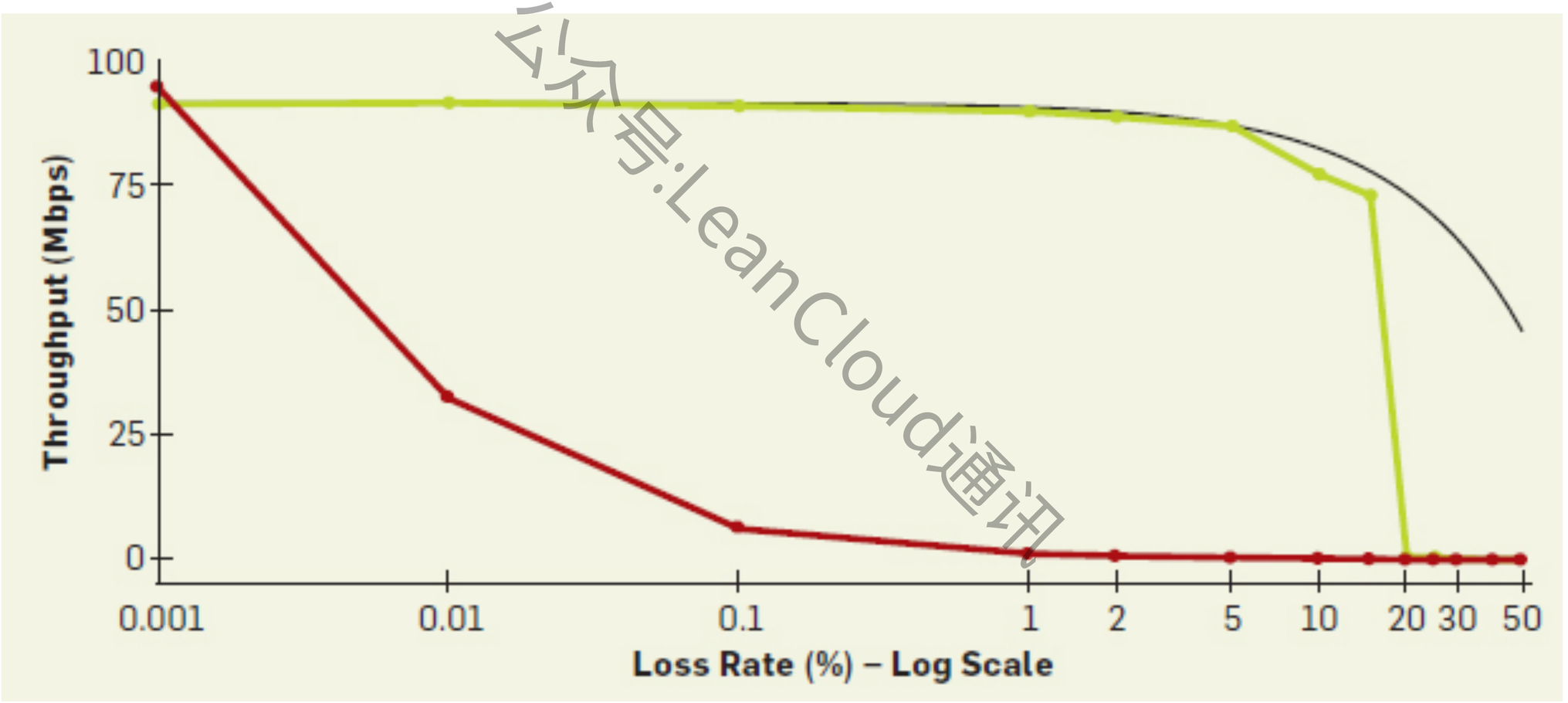
BBR 和 Cubic 对比



多个 BBR 连接共存



BBR 容忍丢包



谢谢

参考列表

- 1. https://blog.apnic.net/2017/05/09/bbr-new-kid-tcp-block/
- 2. https://tools.ietf.org/id/draft-cardwell-iccrg-bbr-congestion-control-00.html#modulating-cwnd-in-loss-recovery
- 3. https://www.slideshare.net/deawooKim/cubic-kdw
- 4. https://cacm.acm.org/magazines/2017/2/212428-bbr-congestion-based-congestion-control/fulltext
- 5. https://queue.acm.org/detail.cfm?id=3022184
- 6. https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/IFIP-Networking-2018-TCP-BBR.pdf
- 7. https://www.slideshare.net/apnic/ausnog-2019-tcp-and-bbr?qid=1a825414-5bbd-4ff0-8057-b81198d79313&v=&b=&from_search=1
- 8. http://blog.cerowrt.org/post/a_bit_about_bbr/
- 9. https://code.woboq.org/linux/linux/net/ipv4/tcp_bbr.c.html