BBR Startup cwnd Gain: a Derivation

July 2018

By Ian Swett and the Google BBR team

[Analytic Derivation]

[Validation via Emulation]

This document presents an analytic derivation of the BBR STARTUP cwnd gain constant of 2.0.

Analytic Derivation

To find the smoothest, gentlest approach that roughly matches the growth dynamics of unpaced Reno/CUBIC, we want the BBR sending rate to double each RTT (each unit time interval).

The goal is not to ensure BBR is never cwnd-limited during STARTUP, but that the cwnd limit never slows down BBR's bandwidth growth.

To keep things simple, this model ignores ACK aggregation effects.

From Step (6) of the pacing gain derivation,

$$DataSentInRoundTripUpTo(t) \;=\; \int_{t-1}^{t} PacingRate(t)\,dt$$

If we call the cwnd gain *K*, and consider a simple scenario where the connection has not yet saturated the bottleneck and thus *RTT* = *min_rtt*, then BBR calculates the cwnd at a particular time as:

$$cwnd(t) \;=\; K \,*\, EstimatedBandwidth(t) \,*\, RTT$$

We model cwnd as being exactly the data sent in the last round trip:

$$cwnd(t) \; = \; DataSentInRoundTripUpTo(t)$$

Then we can combine the previous two equations to obtain the following:

$$K \; * \; EstimatedBandwidth(t) \; * \; RTT \; = \; DataSentInRoundTripUpTo(t)$$

And if the RTT is one time unit, then:

$$K \; * \; EstimatedBandwidth(t) \; = \; DataSentInRoundTripUpTo(t)$$

$$K \; * \; EstimatedBandwidth(t) \; = \int_{t-1}^{t} PacingRate(t)\, dt$$

Solving for K:

$$K \; * \; EstimatedBandwidth(t) \; = \int_{t-1}^{t} PacingRate(t)\, dt$$

$$K \; = \int_{t-1}^{t} PacingRate(t)\, dt \,/\, EstimatedBandwidth(t)$$

From Step (4) of the pacing gain derivation, the estimated bandwidth at time *t* is a function of data sent in the prior round trip, the round trip ending at *t - 1*, thus:

$$K \; = \int_{t-1}^{t} PacingRate(t)\, dt \,/\, DataSentInRoundTripUpTo(t\, -\, 1)$$

From Step (6) of the pacing gain derivation, this is equivalent to:

$$K \; = \int_{t-1}^{t} PacingRate(t)\, dt \,/ \int_{t-2}^{t-1} PacingRate(t)\, dt$$

$$K \; = \; 2$$

Thus the cwnd gain that fits these constraints is 2.

Validation via Simulation

Simulation using the QUIC BbrSenderTest shows that cwnd gain values of 2.3 and above all result in the same amount of time to exit STARTUP in a high bandwidth deeply buffered network simulation where STARTUP exited at MAX cwnd. Lower-bandwidth cases had a discontinuity when one fewer round was required at certain cwnd gains, which decreased the total time in STARTUP.

As expected, SRTT when exiting STARTUP was correlated with cwnd gain, and all cwnd gains from 2 to 3 achieve the same final Max BW.

The simulation sent an ACK for every 2 packets, which may be the cause of the discrepancy relative to the analytic derivation.