HyStart++: Modified Slow Start for TCP

draft-ietf-tcpm-hystartplusplus-04

TCPM, IETF 113

March 23, 2022

Praveen Balasubramanian, Yi Huang, Matt Olson

Microsoft

HyStart++ Recap

- Slow Start can overshoot ideal send rate & cause massive packet loss
  - Increased retransmissions
  - Time spent in recovery
  - Sometimes results in RTO (retransmission timeout)

- HyStart++
  - Simple modification to Slow Start
  - Only use Delay Increase algorithm from original HyStart
  - Compensate for premature slow start exit
  - Use max of Limited Slow Start (RFC3742) and Congestion Avoidance
  - Define tuning constants based on measurements and deployment experience

Jitter Resiliency and Simplification

- Standard slow start (RFC 5681)
- Only use Delay Increase algorithm from original HyStart
- Upon exit from slow start, enter Conservative Slow Start (CSS)
- Under CSS increase cwnd as a fraction of standard slow start
- If measured RTT shrinks during CSS, exit was spurious, resume HyStart++
- Else enter congestion avoidance

- Rationale: Instead of trying to compensate for early exit, add detection for spurious exits to be able to resume slow start

Algorithm Details

- On each ACK in slow start
  - Update the cwnd per standard slow start
  - If taking an RTT sample, measure current round's MinRTT
- For each round in slow start (round approximates an RTT)
  - Remember last round's minRTT
  - *If at least N_RTT_SAMPLE RTT samples taken and currentRoundMinRTT and lastRoundMinRTT are valid*
  - Check if currentRoundMinRTT is greater than lastRoundMinRTT + Threshold
  - If yes, set ssthresh = cwnd, cssBaselineRtt = currentRoundMinRTT, exit slow start and enter Conservative Slow Start (LSS)
- CSS lasts CSS_ROUNDS rounds.
- For each ACK in CSS
  - Update the cwnd as "standard slow start cwnd" / CSS_GROWTH_DIVISOR
- For each round in CSS
  - Remember last round's minRTT
  - If at least N_RTT_SAMPLE RTT samples taken
  - Check if currentRoundMinRTT is less than cssBaseLineRtt
  - If yes, declare exit as spurious and resume HyStart++
  - Else enter congestion avoidance
- Exit HyStart++ on first congestion signal
- SHOULD use on first slow start and MAY use after idle

Changes in draft 04

- Randall proposed "set cssBaseLineMinRTT to (lastRoundMinRtt + RttThresh)" but experiments show that it causes worse perf in networks with jitter
- Neal suggested "if (SND.UNA > windowEnd)" for determining the end of a round but its also inaccurate
- Only logic change: simplify and remove dependency on LOW_SSTHRESH
- Some editorial improvements

Status & Next Steps

- Addressed reviews
  - Bob Briscoe
  - Jeremy Harris
  - Neal Cardwell
- Answered questions
  - Randall Stewart
  - Neal Cardwell
- Implementations
  - Microsoft Windows TCP CUBIC
  - Cloudflare's QUIC library (quiche) and its production QUIC traffic
  - FreeBSD TCP CUBIC

- Ready for Working Group Last Call