Analysis for the Differences Between Standard Congestion Control Schemes

draft-nishida-ccwg-standard-cc-analysis-01

Yoshifumi Nishida

nsd+ietf@gmail.com

Background

- Congestion Control standards should provide consistent guidelines, shouldn’t contradict
  - Also, they should be transport protocol agnostic in general

- Analyzing the impacts of Congestion Controls on the Internet won’t be easy
  - It would require long term analysis
  - Having a reference for several checking points could be useful

What’s in this draft?

- A list for differences on certain topics in CC standards
  - TCP Reno(RFC5681), QUIC Reno(RFC9002), CUBIC(RFC9438)
  - Difference between TCP Reno and QUIC Reno
    - Ideally, TCP Reno and QUIC Reno should not be different with regard to aggressiveness
  - Difference between Reno and CUBIC in terms of fairness
    - Ideally, Reno and CUBIC should coexist ‘mostly’ fairly
      - It’s fine CUBIC archives better performance, but shouldn’t push away Reno

Differences between TCP and QUIC Reno (1)

- Initial Window
  - RFC5681 .. Up to 4 segments or 4380 bytes
    - RFC6928 allows TCP connections to use up to 10 segments or 14600 bytes, but it’s an experimental.
  - RFC9002 .. Up to 10 segments or 14720 bytes

- Minimum RTO
  - RFC6928 .. 1 sec
  - RFC9002 .. No minimum RTO

- Loss Window
  - RFC5681 .. 1 Segment
  - RFC9002 .. 2 Segments

Differences between TCP and QUIC Reno (2)

- Window Growth in Slow Start
  - RFC9002 .. cwnd += *number_of_acked_bytes*
  - RFC5681 .. cwnd += min(*number_of_acked_bytes*, *1* SMSS)
    - Increases at most 1 SMSS by a single ACK
    - It mentions RFC3465, but it's not recommended to use it.
      - Also, RFC3465 is experimental, although 9002 is more aggressive than RFC3465 as there is no L factor

- Slow Start Threshold After Packet loss
  - RFC9002 .. half value of congestion window when packet loss is detected
  - RFC5681 .. half value of flight size instead of congestion window
    - Also, RFC5681 basically prohibits to use congestion window here

Differences between Reno and CUBIC

- Multiplicative Window Decrease Factor
  - RFC5681 .. Use 0.5
  - RFC9438 .. Use 0.7
    - 0.7 might not be too aggressive, but might not be fair with RFC5681

  - Reno-Friendly model in CUBIC aims to mitigate the fairness issue between Reno and CUBIC
    - However, this model might need further analysis
      - Detailed explanation is described in the draft

Goals of this document

- Published as an Informational RFC for a reference
- Clarify differences between congestion control standards
  - They should provide the consistent guidelines to avoid conflicts
  - Initiate discussions for the next steps
- Could be used as a reference for future analysis on the impacts of congestion controls on the Internet

Appendix

Is Reno-Friendly Model in CUBIC a valid model?

- If this model was designed to make CUBIC to be fair with Reno
  - Reno uses AIMD(1, 0.5) while CUBIC uses AIMD(0.529, 0.7) to be compatible with Reno
    - Green and Orange parts should have the areas of the same size
      - But, this presumes that both have the same congestion epoch, which might not be always true
  - Further analysis might be required