Analysis for the Differences Between Standard Congestion Control Schemes

draft-nishida-tcpm-standard-cc-analysis-00

Yoshifumi Nishida

nsd+ietf@gmail.com

What’s in this draft?

- A list for difference between 3 congestion control standards
  - Reno(RFC5681), QUIC Reno(RFC9002), CUBIC (8312bis)
- Motivations
  - As a record for CUBIC’s Fairness discussions
  - As a reference for future discussions of congestion control principles
    - It would be better these 3 standards have the same principles
  - It might encourage further analysis
- Out of focus
  - Evaluations for the difference between the docs.
    - The purpose of the doc is ONLY to provide a list
    - It may take time to evaluate these differences

Differences between RFC5681 and RFC9002 (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
    - It also mentions pacing
- Loss Window
  - RFC5681 .. 1 Segment
  - RFC9002 .. 2 Segments
- Loss Detection Schemes
  - RFC9002 .. Specifies TCP RACK TLP type loss detection scheme
  - RFC5681 .. No description for TCP RACK-TLP

Differences between RFC5681 and RFC9002 (2)

- Slow Start Threshold After Packet loss
  - RFC9002 .. half value of congestion window when packet loss is detected.
    - It also mentions using RFC7661
  - RFC5681 .. half value of flight size instead of congestion window
  - Also, RFC5681 basically prohibits to use cwnd here

> *" Implementation Note:*
> *An easy mistake to make is to simply use cwnd, rather than FlightSize, which in some implementations may incidentally increase well beyond rwnd. "*

Differences between RFC5681 and RFC9002 (3)

- 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.

Differences between RFC5681 and RFC9002 (4)

- Loss Recovery Algorithm
  - Definition of end of recovery period
    - RFC9002 .. One of any packets sent during the Recovery period is acknowledged
    - RFC5681 .. All lost segments found before recovery period are acknowledged
  - This will mean RFC9002 can exit from recovery more easily than RFC5681
    - Increase congestion window right after loss via CA while RFC5681 stays in recovery
    - If we keep losing a few packets every RTT, RFC9002 will repeat CA and Recovery
      - This means RFC9002 decreases cwnd two times slower than RFC5681

Differences between RFC5681 and 8312bis (1)

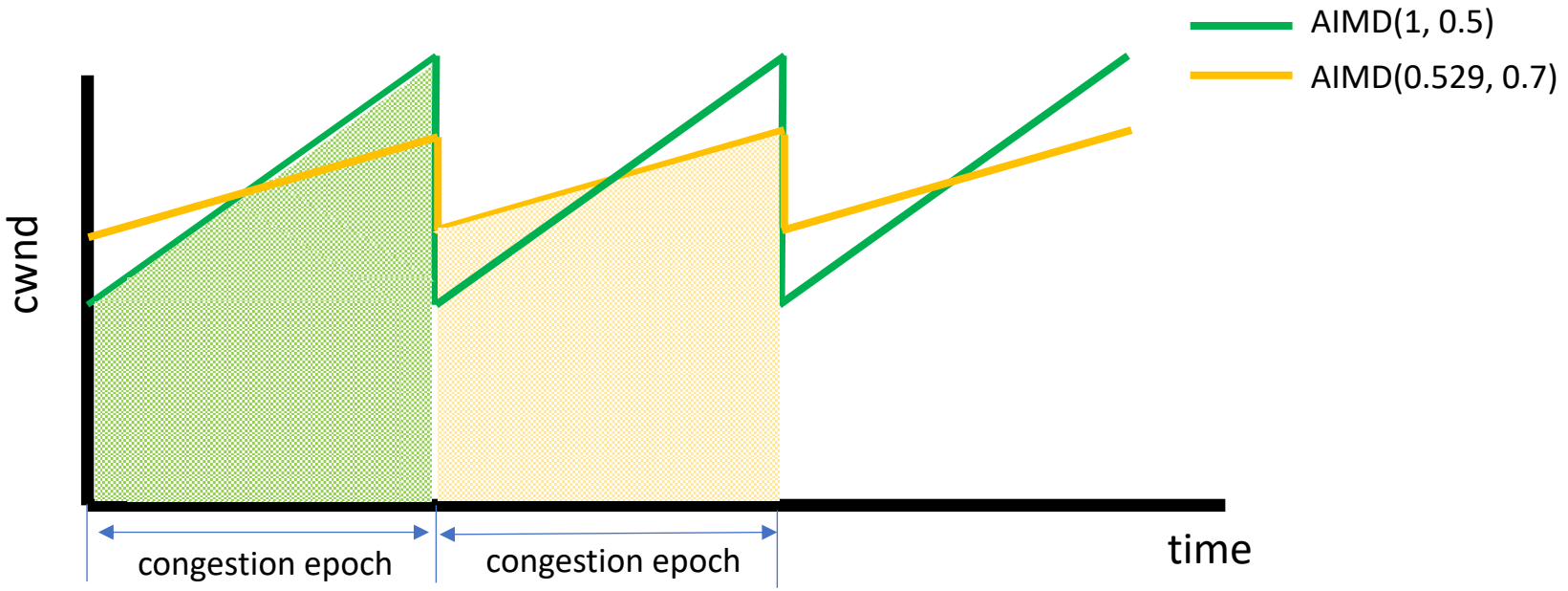
- Multiplicative Window Decrease Factor
  - RFC5681 .. Use 0.5
  - RFC8312bis .. Use 0.7
    - 0.7 might not be too aggressive, but might not be fair with RFC5681

    - Jacobson, V., "Congestion Avoidance and Control", CCR 1988

      ”.. you should reduce your window by half because the bandwidth available to you has been reduced by half. And, if there are more than two conversations sharing the bandwidth, halving your window is conservative "

Differences between RFC5681 and 8312bis (2)

- 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



Discussion Points

- Are these differences acceptable or need to be sorted out?
  - If it needs to be sorted out, how do we proceed?
    - Do we want to update RFC5681?
- Is it worth for publishing this draft as a reference?