Advancing Proportional Rate Reduction to Proposed Standard(?)

draft-mathis-tcpm-rfc6937bis-00

Matt Mathis, Nandita Dukkipati, Yuchung Cheng

Adopt this as a tcpm work item?

PRR recap (RFC6937 experimental)

PRR is a special congestion control effective only during fast recovery

- When inflight >= ssthresh, send at loss_beta*rate_before_loss
  (e.g. loss_beta = 0.5 for Reno (aka rate-halving), 0.7 for Cubic)
- When inflight < ssthresh, send at the same or twice the delivery_rate
  (more later)

- Used by all congestion control modules in Linux during fast recovery
  - Can be more dominant than the actual C.C. for lossy flows that’re in fast recovery constantly (e.g. video streaming through policers)

Current Status

- PRR is widely deployed
  - At least three major OSs: Linux, Windows, (NetFlix) BSD
  - Vast majority of Web traffic for years
- No changes to algorithms published in RFC 6937
  - PRR-CRB - Conservative Reduction Bound - strict packet conversion during loss recovery
  - PRR-SSRB - Slowstart Reduction Bound - one extra segment per ACK during loss recovery
- 2015 Heuristic to dynamically select which reduction bound
  - Only use PRR-SSRB when making good forward progress
    - ACKs that advanced snd.una and report no new losses
  - Resolves some pathological cases with token bucket policers
    - CC estimates ssthresh before it can possibly measure the token rate
    - The heuristic makes the best of a bad situation

Tentative path forward

- Adopt as a tcpm work item
- Update the text
  - Normative RFC 2119 language
  - Add MAY use the heuristic...
  - Trim redundant and obsolete language
    - RFC 6937 repeats itself and is much longer than necessary
    - Focus on what an implementer needs to know
    - Use non-normative references to RFC 6937 for prior measurement work, etc