Non-Congestion Robustness (NCR) for TCP

draft-ietf-tcpm-draft-ietf-tcpm-tcp-dcr-03.txt

IETF 62 - March 2005

Sumitha Bhandarkar, A. L. Narasimha Reddy,
Mark Allman, Ethan Blanton

"Hit our satellite with feeling,
Give the people what they paid for"

Status

- Document started as "DCR" (delayed congestion response)
- We have changed to "NCR" (non-congestion robustness)
  - delaying the congestion response is no longer required

- Document has undergone **significant** revision
- Underlying notions **have not** changed

Status (cont.)

- Removed a bunch of text on wireless link-layers
  - Still given as one exploitation of NCR

- Down-scope to only SACK variants of TCP
  - non-SACK NCR is messy and opens new security vulnerabilities (and, it’s 2005!)

Algorithm

- Delay the trigger for fast retransmit by roughly one RTT
  - Gives reordering time to work itself out

- NCR is now based on duplicate ACKs only
  - Removed the time-based trigger

- CURRENT: DupThresh = 3
- NEW: DupThresh = max (3,cwnd)

- Also **must** extend Limited Transmit or fast retransmit will become brittle
  - I.e., segment and ACK loss will force the use of the RTO

Algorithm (cont.)

- Limited Transmit (RFC 3042):
  - Allowed to send previously unsent data on the first and second duplicate ACKs

- Two limited transmit extensions to choose from when using NCR:
  - Careful Limited Transmit or Aggressive Limited Transmit
  - Careful variant is new to this version of the document
  - While retransmit decision is always delayed the variant of LT provides a choice about when to make a congestion control decision

Algorithm (cont.)

- Careful Limited Transmit
  - Send a previously unsent segment on *every second* duplicate ACK
  - *Roughly* the same congestion response we use now
  - Dont change cwnd until retransmission
  - "Standard" slow start-based burst prevention

- Aggressive Limited Transmit
  - Send a previously unsent segment on *every* duplicate ACK
  - Delays the congestion response by *roughly* one RTT

Algorithm (cont.)

- When a host cannot send during the LT phase (due to a lack of data, advertised window constraint, etc.) the algorithm scales the duplicate ACK threshold based on how much data is sent during LT.
  - To preserve robustness
- The algorithm is precisely defined in the draft

Feedback

- Feedback received so far:
  - A number of things to cleanup in the text (better explanations, etc.)
  - Suggest that system could provide a socket option to allow applications to turn NCR on/off
    - Some apps might not want their retransmits delayed
- More feedback needed