TCP and SCTP RTO Restart

draft-ietf-tcpm-rtorestart-02

Per Hurtig, Anna Brunstrom, Andreas Petlund, Michael Welzl

2014-03-06

R / T E

Reducing Internet Transport Latency

Outline

RTO Restart

Updates to the draft

- Formalized
- Clarifications
- Nits

Implementation

RTO Restart

- As the RTO timer is restarted on an incoming ACK [RFC6298, RFC4960], the effective RTO often becomes $\texttt{RTO} = \texttt{RTO} + t$
  - where $t \approx \texttt{RTT}[+\texttt{delACK}]$
- RTO restart adjusts the RTO so that retransmissions are performed after exactly RTO seconds
- The modified restart is only used when:
  - the number of outstanding segments is < 4
  - there are no unsent data ready for transmission
  - Thus, only flows incapable of FR can use modified RTO restart

Updates to the draft

Formalized

- Algorithm description more formalized [RFC2119]
- It is an OPTIONAL algorithm
- The number of outstandings segments for triggering RTO restart MUST be less than a certain threshold (rrthresh) which SHOULD be set to four

RTO Restart

Clarifications

- Section 3: why RTO restart is active when less than four segments are outstanding
  - Because this is when standard TCP cannot use FR for loss recovery
  - Also, by not always allowing it the risk of triggering RTO instead of FR, or trigger a spurious RTO, is decreased
- Section 4.1: specified that RTO restart can be turned on (default) without causing harm
  - Because it is only active when you can gain something (see above)

RTO Restart

Nits

- Improved the wording throughout the document
- Replaced and updated some references

Implementation

- Updated for the 3.14.0-rc3 Linux kernel
  - Overflow bug removed
- Can be downloaded from http://riteproject.eu
- Currently trying to get it in Linux net-next

Questions?