

TCP and SCTP RTO Restart

draft-ietf-tcpm-rtorestart-04

Per Hurtig, Anna Brunstrom, Andreas Petlund, [Michael Welzl](#)

2014-11-10

R / T E

REDUCING INTERNET TRANSPORT LATENCY

Outline

RTO Restart

Updates to the draft

Algorithmic Changes

Experimental Results and Implementation

RTO Restart

- As the RTO timer is restarted on an incoming ACK [[RFC6298](#), [RFC4960](#)], the effective RTO often becomes $RTO = RTO + RTT[+delACK]$
- RTO restart adjusts the RTO so that retransmissions are performed after exactly RTO seconds
- The modified restart is only applied when FR can not be used

Updates to the draft

- Changed the algorithm to allow RTOR when there is unsend data available, but the cwnd does not allow transmission.
 - change discussed at IETF 90
- Changed the algorithm to not trigger if " $RTO - T_{\text{earliest}} \leq 0$ ", to avoid that ACKs to previous retransmissions trigger premature timeouts.
 - problem discussed on tcpm mailing list
- Made minor adjustments throughout the document to adjust for the algorithmic change.
- Improved the wording throughout the document.

Algorithmic Changes

When an ACK is received that acknowledges new data:

1. Set $T_{\text{earliest}} = 0$.
2. If the total number of **outstanding and previously unsend** segments is less than an RTOR threshold ($rrthresh$), set T_{earliest} to the time elapsed since the earliest outstanding segment was sent.
3. Restart the retransmission timer so that it will expire after (for the current value of RTO):
 - a) $RTO - T_{\text{earliest}}$, if $RTO - T_{\text{earliest}}$ is > 0 .
 - b) RTO , otherwise.

Experimental Results and Implementation

- Experimental results on the performance of RTOR presented at last meeting, complemented with info on spurious retransmissions here
 - Fully controlled – fixed-size flows with tail loss: no spurious retransmissions
 - Realistic loss – trace-driven background traffic: give the numbers for RTOR and baseline
 - * Baseline: 2.2×10^{-4}
 - * RTOR: 2.9×10^{-4}

– Web pages – web page downloads with correlated loss patterns (fraction spurious):

* Baseline: 4.8×10^{-5}

* RTOR: 5.9×10^{-5}

- Implementation has been updated with the latest algorithm changes
- For detailed information and code, see <http://riteproject.eu>

Questions?