

TCP and SCTP RTO Restart

draft-ietf-tcpm-rtorestart-05

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

2015-03-24

R / T E

REDUCING INTERNET TRANSPORT LATENCY

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

- Introduced a new variable (`prevunsnt`) to account for the number of segments that a sender has queued for transmission, but has not yet sent.
- The section on tracking outstanding and previously unsend segments has been extended to give better guidance to implementers:
 - A description of how to track outstanding segments for both packet-based and byte-based stacks.
 - How to track previously unsend segments depending on the segmentation strategy of the TCP stack.
 - The tracking of previously unsend segments is not critical to RTOR; it can be disregarded if it's too complex to implement. The tracking is only done to optimize a corner case.

- The risk of increasing the amount of spurious retransmissions (and their effects) has been further described:
 - The negative effect that needless transmissions has on the network is described.
 - The impact of spurious retransmissions in flows with multiple bursts (e.g. HTTP/1.1 and HTTP/2.0) has been described more closely.
 - Clarified that experimentation in environments with highly variable RTTs (e.g. mobile networks) is useful for further evaluation before being targeted as a proposed standard.
- Clarified that RTOR is suitable for experimentation as a system-wide default.

- Text on relation between RTOR and TLP has been reworked a bit to illustrate the gains and drawbacks of choosing either approach.
- The security considerations section has been rephrased to clarify that the security considerations in RFC6298 apply. No additional security issues, related to RTOR, has been identified.
- Minor improvements to the wording.

Algorithm

When an ACK is received that acknowledges new data:

1. Set $T_{\text{earliest}} = 0$.
2. If the sum of the number of outstanding and previously unsent segments (pre-unsnt) 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 $RTO - T_{\text{earliest}} > 0$.
 - b) RTO, otherwise.

Implementation

- Implementation has been updated and tested with the Linux 4.0-RC4 kernel.
- For detailed information and code, see <http://riteproject.eu>

Questions?