TCP and SCTP RTO Restart

draft-ietf-tcpm-rtorestart-03

Per Hurtig, Anna Brunstrom, Andreas Petlund, Michael Welzl

2014-07-21

R/TE

Reducing Internet Transport Latency

Outline

RTO Restart

Updates to the draft

Outstanding Algorithm Issue

Experiments

- Fully Controlled
- Realistic Loss
- Web Page

Implementation

RTO Restart

- As the RTO timer is restarted on an incoming ACK [RFC6298, RFC4960], the effective RTO often becomes $\texttt{RTO} = \texttt{RTO} + \texttt{RTT}[+\texttt{delACK}]$
- 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

- Added a section to generalize the tracking of outstanding segments
  - to cover both byte-based and packet-based implementations
- Updated the document to use "RTOR" instead of "RTO Restart" when referring to the modified algorithm
- Moved document terminology to a section of its own, and added a description of the `rrthresh` variable there as well
- Clarified the relationship between fast retransmit and RTOR
- Improved the wording throughout the document

Outstanding Algorithm Issue

When an ACK is received that acknowledges new data:

1. Set T_earliest = 0
2. If the following two conditions hold:
   a) The number of outstanding segments is less than a RTOR threshold (rrthresh). The rrthresh SHOULD be set to four
   b) There is no unsent data ready for transmission

   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 "RTO - T_earliest" seconds (for the current value of RTO)

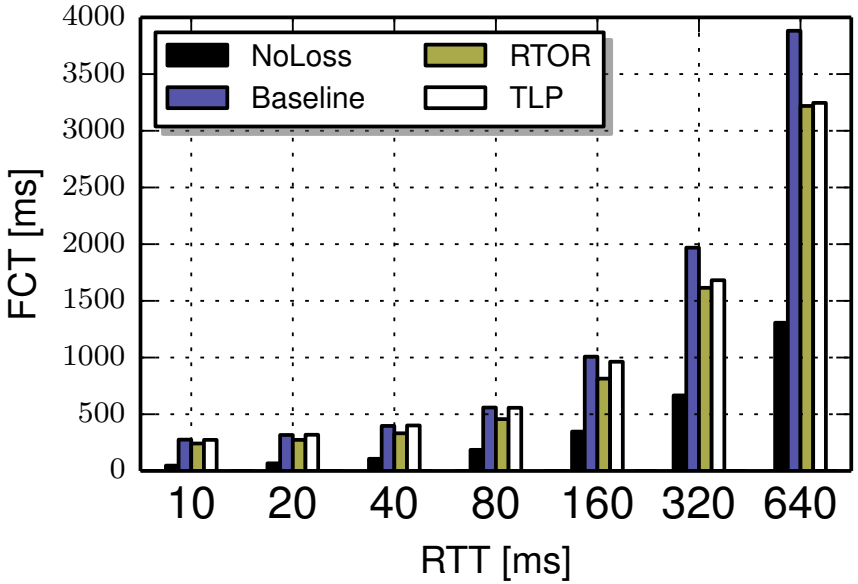
Outstanding Algorithm Issue

- Attempted fix in current draft rearms the retransmission timer after a new data transmission, if needed
  - this behavior causes other problems
- A possible alternative solution changes the conditions (in 2) to:
  - the number of outstanding and unsent segments is less than a RTOR threshold (rrthresh)

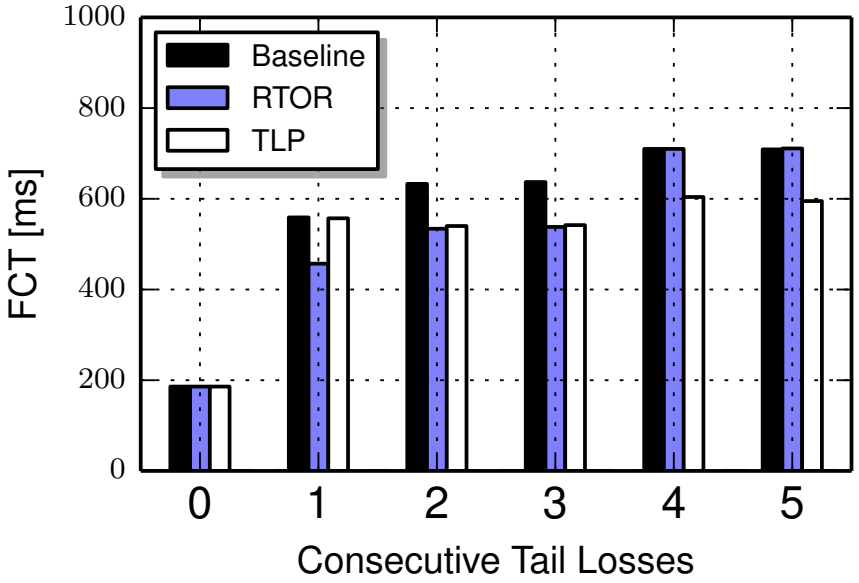
Experiments

- Fully controlled – fixed-size flows with tail loss
- Realistic loss – trace-driven background traffic
- Web pages – web page downloads with correlated loss patterns

Fully Controlled

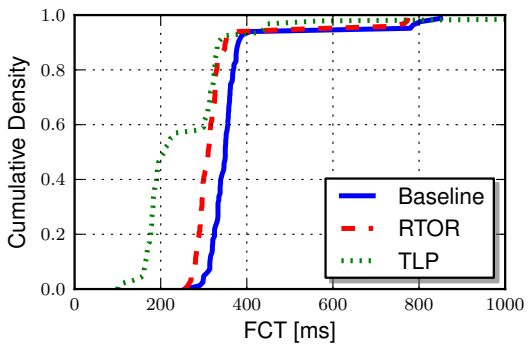


(a) Loss of the last segment

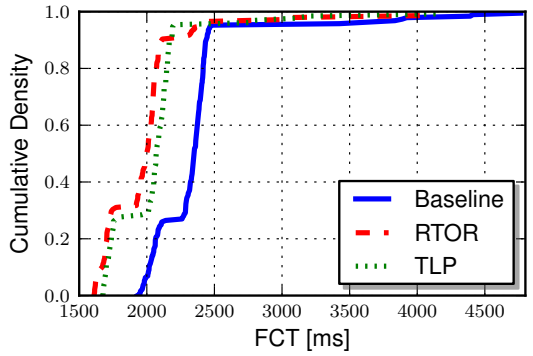


(b) Loss of the n last segments, 80ms RTT

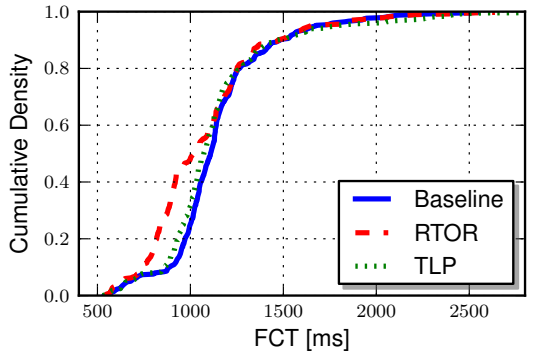
Realistic Loss



(a) 4 segments, 10ms baseline RTT

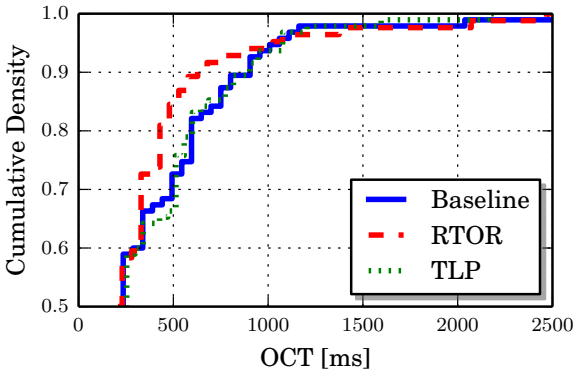


(b) 4 segments, 320ms baseline RTT

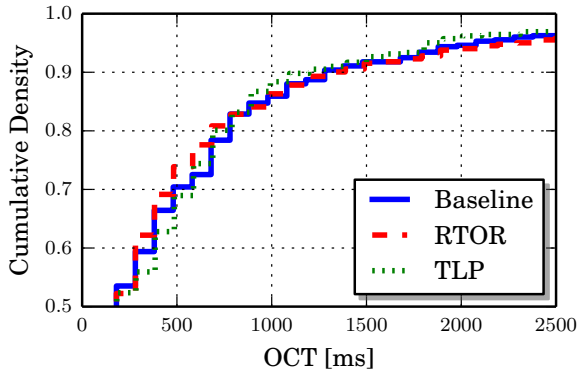


(c) 100 segments, 80ms baseline RTT

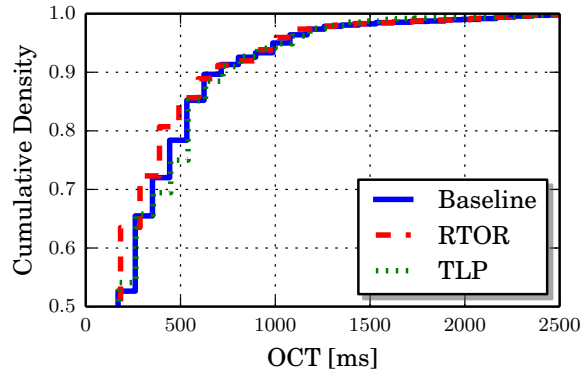
Web Page



(a) Small-sized website



(b) Medium-sized website



(c) Large website

Implementation

- Updated for the 3.15 Linux kernel
- Can be downloaded from http://riteproject.eu

Questions?