

# **TCP and SCTP RTO Restart**

**draft-hurtig-tcpm-rtorestart-01**

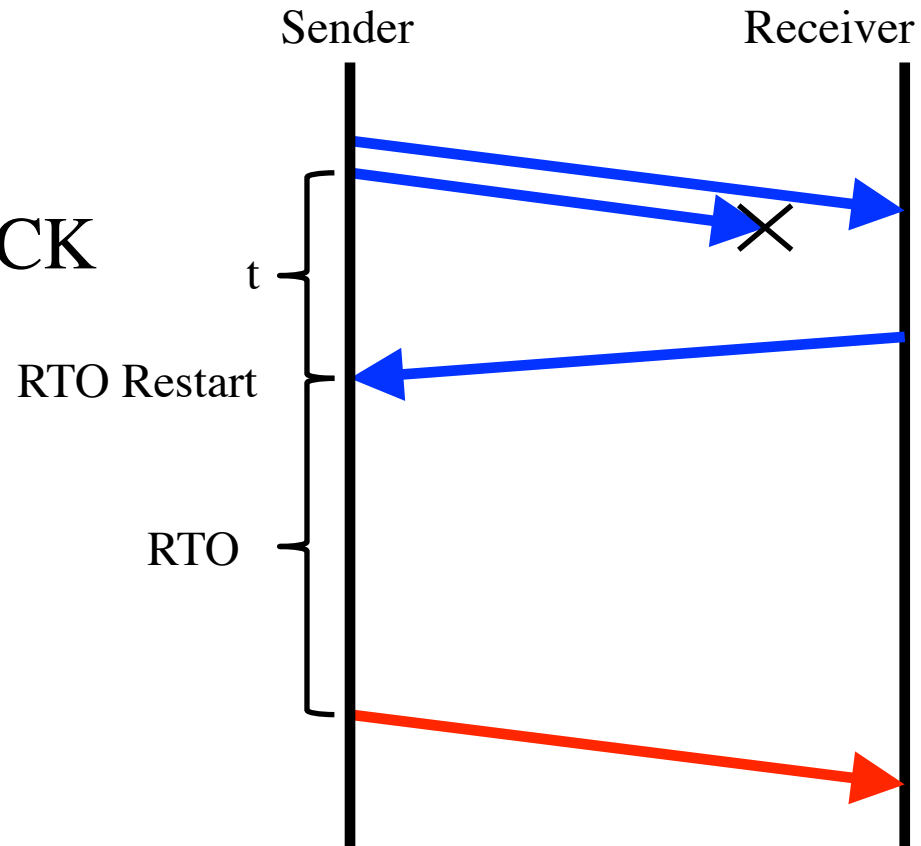
Per Hurtig, Andreas Petlund, Michael Welzl  
perhurt@kau.se, apetlund@ifi.uio.no, michawe@ifi.uio.no

# Motivation

- In some cases TCP/SCTP must use RTO for loss recovery
  - e.g., if a connection has 2 outstanding packets and 1 is lost
- Some solutions exist, but they are not always applicable
  - Limited Transmit (RFC 3042)
    - requires: unsent data, no ack loss
  - Early Retransmit (RFC 5827)
    - requires: 2 outstanding segments, no ack loss, no reordering

# Motivation

- Thus, some flows have to use RTO for loss recovery
- However, the effective RTO often becomes  $RTO = RTO + t$ 
  - Where  $t \approx RTT [+delACK]$
- The reason is that the timer is restarted on each incoming ACK (RFC 6298)



# Impact

- The extra RTT could lead to performance problems for short-lived (e.g. web) and thin streams
  - Thin streams are flows that only use a fraction of the available bandwidth (e.g. online games, chat, VoIP, ...)
  - IETF 78: <http://www.ietf.org/proceedings/78/slides/iccrg-4.pdf>
- 80% of all web flows typically contain 7-8 segments or less [1], which is similar to general TCP flow lengths [2]
  - 2-3 RTTs in slow-start
  - RTO  $\approx$  4 RTTs (Linux and Windows) [3]
- It has previously been shown that web flows use RTOs frequently to recover lost packets [4]

[1] Dukkipati et al., “An argument for increasing TCP's initial congestion window”, ACM CCR, July 2010.

[2] Qian et al., “TCP Revisited: A Fresh Look at TCP in the Wild”, In Proc. of IMC 2009.

[3] Rewaskar et al., “A Performance Study of Loss Detection/Recovery in Real-world TCP Implementations”, In Proc. of ICNP 2007

[4] Balakrishnan et al., “TCP Behavior of a Busy Web Server: Analysis and Improvements”, In Proc. of INFOCOM 1998.

# Impact

- Standard approach no problem when congestion window is large
- Actually, it is beneficial
  - lower risk for spurious RTOs
  - gives FR more time to detect loss
    - smaller congestion window reduction using FR
- This is not the case for short-lived/thin flows
  - congestion window low anyhow

# TCP and SCTP RTO Restart

- To allow retransmissions after exactly RTO seconds, the timer is restarted as:
  - $RTO = RTO - t$
- The modified restart is only used when
  - the number of outstanding segments  $< 4$ ;
  - and there is no unsent data ready for transmission.
- Thus, only flows incapable of FR can use the modified RTO restart

# Costs vs. Benefits

- Benefits
  - reduces RTOs with approximately one RTT for flows incapable of FR
  - isn't more aggressive than allowed by RFC 6298
- Costs
  - more aggressive than the current algorithm
  - requires an extra variable per outstanding segment

# The future

- We have implemented the algorithm in FreeBSD/Linux
- Should this be a WG item?
  - the goal for the draft is experimental