draft-dukkipati-tcppm-tcp-loss-probe-01

N. Dukkipati, N. Cardwell, Y. Cheng, M. Mathis

TCPM WG @IETF 86, 12 March 2013.
Tail loss probe (TLP) recap

Problem
  Timeout recovery is 10-100x longer than fast recovery.
  Tail drops in short transactions are very common.
  70% of losses on Google services recovered via timeouts.

Goal
  Reduce tail latency of short Web transactions.

Approach
  Convert RTOs to fast recovery.
  Retransmit the last packet in 2 RTTs to trigger FR.

Impact
  Reduced RTO events by 15%.
  Reduced HTTP response time 6% on average, 10% at tail (99%).
  0.48% overhead in TLP probes.
TLP example
Changes between -00 and -01

New section on FACK threshold based recovery.

Experiment results with TLP loss detection algorithm.

Scheduling PTO at min(PTO, RTO).

Several minor edits:
   Why is PTO 2.RTT (and not RTT, 3.RTT...)?
   TCP Loss Probe -> Tail Loss Probe.
   Use of one probe (versus multiple probes) per tail loss episode.
   Decision against 1-byte retransmission.
   Referenced Rescue retransmission.
   Relation to RTO-restart.
FACK threshold based recovery

SND.FACK is the highest sequence number known to have been received plus one.*

Threshold based recovery algorithm:

\[
\text{If } (\text{SND.FACK} - \text{SND.UNA}) > \text{dupack threshold}: \\
\quad \rightarrow \text{Invoke Fast Retransmit and Fast Recovery.}
\]

A very effective algorithm for invoking loss recovery. In Linux as the default since 1998.

Detecting TLP repaired losses

Problem:
Must invoke congestion control if TLP repairs loss and the only loss is last segment.

Approach: Count duplicate ACKs for TLP retransmissions.
  TLP episode: N consecutive TLP segments for same tail loss.
  End of TLP episode: ACK above SND.NXT.
  Expect to receive N TLP dupacks before episode ends.
  No loss: sender receives N TLP dupacks.
  Loss: sender recvs <N TLP dupacks.
  On detecting loss, reduce cwnd and ssthresh as in fast recovery.
Loss detection example

Client

Server

cwnd == 10

2.RTT

loss probe: 10

TLPHighRxt = 10

ACK: 10

TLPHighRxt remains 10

ACK: 11

Recd. no dupack

cwnd = 7

TLPHighRxt = 0
Loss detection experiment results

Loss detection algorithm found ~33% of TLP retransmissions repaired a loss.

Latency with loss detection is slightly better than without.

Single byte probe is not very useful in practice.
When is a TLP sent?

\[ PTO = 2 \times SRTT \]

\[ \text{if } (\text{FlightSize} == 1) \]
\[ PTO = \max(PTO, 1.5 \times SRTT + WC\text{DelAckT}) \]

\[ PTO = \max(PTO, 10\text{ms}) \]

\[ PTO = \min(RTO, PTO) \]

RTO is rearmed to "now + RTO" at the time of sending a TLP.
WG adoption

Work in progress.
  Sent upstream to Linux.
  Submitted a research paper.

Key: TCP should have a mechanism to deal directly with tail losses.

TLP is a simple, practical, easily-deployable scheme:
  Trades a small amount of bandwidth for latency.
  Keeps RTO conservative to reduce spurious timeouts and cwnd reductions (e.g., mobile).
  Works with other features like RTO-restart, F-RTO, cwnd undo, limited transmit, early retransmit, better RTO estimation.

Ready to be adopted as WG item.