draft-ietf-tcpm-rack-09

{ycheng, ncardwell}@google.com
tcpm IETF108 7/27/20

Main changes

A substantial rewrite

1. Focusing on integrating RACK-TLP as whole
2. Incorporating WGLC reviews (let me know if I missed any)

New motivation, high-level design, reordering rationale sections

Two new examples of RACK-TLP scenario timelines

Motivation: what RACK-TLP can do that the 3-DUPACK heuristic couldn’t

1. Quickly detect packet drops in short flows or at the end of an application data flight.

2. Detect lost retransmissions.

3. Tolerate low reordering degree in *time* distance
    a. E.g. deliver P100, P1, P2, … P99. Reordering degree: Sequence: 99*SMSS. Time: <<RTT

High-level design (sec 3)

Overarching goal: Ack-triggered Fast recovery as much as possible. RTO recovery as last resort

1. RACK: detect losses via ACK events as much as possible, to repair losses at round-trip time-scales:

   Segment S is lost if S.sent_time + RTT + reo_wnd < Now

2. TLP: gently probe to solicit additional ACK to trigger (1) to avoid RTO and subsequent congestion window reset

Reordering window adaptation (sec 3.3.2)

Reordering window is dynamically adapted as follows:

1. If no reordering seen: **zero** if 3-DUPACKs or already in recovery
2. If reordering seen: start from min_RTT/4
3. For every round that observes DSACK, linearly increase window until it reaches SRTT. After 16 recoveries w/o any DSACK seen, go to (2)

Rationale:
Short flows recover quickly with controlled risk of spurious retransmission
Long flows adapt to (low time-degree) reordering
Low initial window with bounded max to disincentivize excessive network reordering

How TLP recovers faster via RACK (sec 3.4)

```
Event  TCP DATA SENDER                            TCP DATA RECEIVER
     1.   Send P0, P1, P2, P3          -->
          [P1, P2, P3 dropped by network]
     2.                                <--          Receive P0, ACK P0

     3a.  2RTTs after (2), TLP timer fires
     3b.  TLP: retransmits P3          -->

     4.                                <--         Receive P3, SACK P3

     5a.  Receive SACK for P3
     5b.  RACK: marks P1, P2 lost
     5c.  Retransmit P1, P2            -->
          [P1 retransmission dropped by network]

     6.                                <--    Receive P2, SACK P2 & P3

     7a.  RACK: marks P1 retransmission lost
     7b.  Retransmit P1                -->
     8.                                <--          Receive P1, ACK P3
```

MUST, SHOULD, MAY changes

+ Reordering window SHOULD adapt based on DSACK if eligible
+ Reordering timer SHOULD be used to quickly recover
+ TLP requires RACK, RACK requires SACK
+ TLP sender SHOULD cancel any other pending RTO, ZWP, RACK timer when (re)arming PTO
+ (Implicit MUST) at most one TLP probe at a time
- TLP.max_ack_delay of 200ms => implementation-specific

Relationship to other RFCs

- Replace/subsume as an alternative:
  - Conservative Loss Recovery based on SACK [RFC6675]
  - Early Retransmit [RFC5827]

- Complementary & compatible:
  - Limited Transmit [RFC3042]
  - RTO Restart [RFC7765]
  - F-RTO [RFC5682]
  - RTO [RFC6298]
  - Eifel [RFC3522]