QUIC Recovery Update

Ian Swett, IETF 104

Review of Key Points

Packet numbers monotonically increase in send order

ACK frames acknowledge one or more ranges of packets

An explicit ack_delay is present in each ACK frame

Peer sends max_ack_delay during handshake

IETF 103 overview of QUIC recovery from tcpm.

IETF 104 overview of QUIC recovery from tcpm.

Update since Bangkok (since -16)

Almost as many changes as all previous revisions to recovery

Only 6 design issues open, which we'll discuss next

Time and packet threshold loss detection (#1974)

draft 16

Time threshold or Packet threshold, if you only do packet threshold, also do early retransmit with a timer

draft 19

Always both time and packet threshold

Packet is lost whenever either threshold is hit

Merge TLP + RTO -> PTO (#2114)

draft 16

2 TLPs, then RTO

TLP sends one packet, RTO sends two packets

TLP = max(1.5*SRTT+MaxAckDelay, kMinTLPTimeout)

RTO = max(SRTT + 4*RTTVAR + MaxAckDelay, kMinRTO)

Merge TLP + RTO -> PTO (#2114)

draft 19

Timeout: PTO = smoothed_rtt + max(4*rttvar, kGranularity) +

max_ack_delay

Send: One or two packets with retransmittable frames

Note: min_rto replaced by kGranularity, max_ack_delay

Define Persistent Congestion (#2365, #2244)

draft 16

CWND is reduced to MinCWND when all packets prior to the acknowledged RTO packet are lost

Issue: Sending new packets extends the TLP and RTO timeouts, avoiding collapsing CWND even if no packets are ACKed

Define Persistent Congestion (#2365, #2244)

draft 19

Persistent Congestion: ACK frame establishes loss of all in-flight packets sent over a long enough period of time

Period = 3 * PTO timeout (similar to 2 TLPs + RTO)

Limit Ack Delay to Max Ack Delay (#2099)

draft 16

```
// Adjust for ack delay if it's plausible.
if (latest_rtt - min_rtt > ack_delay):
    latest_rtt -= ack_delay
```

Issue: A peer could use this to artificially decrease PTO

Note: min_rtt never includes ack_delay

Limit Ack Delay to Max Ack Delay (#2099)

draft 19

```
// Limit ack_delay by max_ack_delay
ack_delay = min(ack_delay, max_ack_delay)
// Adjust for ack delay if it's plausible.
if (latest_rtt - min_rtt > ack_delay):
    latest_rtt -= ack_delay
```

Discarding Recovery State (#2327)

New since draft 16

Caused by 0-RTT rejection or dropping keys

- Those packets can't be acknowledged
  - Discard all recovery state for those packets
  - Remove them from bytes in flight

Pace packets or Reset CC after Idle (#2023)

New since draft 16

When exiting idle, must limit the burst to IW

If not pacing, the CWND is set to IW, enter Slow Start

Move Psuedocode to the Appendix (#2408)

The text is intended to be normative and complete

Ideally the pseudocode would compile in Python

- Currently does not compile

Better text/pseudocode for multiple PN spaces (#2417, #2451, #2485)

New since draft 16

Specifies psuedocode for multiple packet number space loss detection, with necessary data structures duplicated

Since -16

- Unify TLP and RTO into a single PTO; eliminate min RTO, min TLP and min crypto timeouts; eliminate timeout validation (#2114, #2166, #2168, #1017)
- Redefine how congestion avoidance in terms of when the period starts (#1928, #1930)
- Document what needs to be tracked for packets that are in flight (#765, #1724, #1939)
- Integrate both time and packet thresholds into loss detection (#1969, #1212, #934, #1974)
- Reduce congestion window after idle, unless pacing is used (#2007, #2023)
- Disable RTT calculation for packets that don't elicit acknowledgment (#2060, #2078)
- Limit ack_delay by max_ack_delay (#2060, #2099)
- Initial keys are discarded once Handshake are available (#1951, #2045)
- Reorder ECN and loss detection in pseudocode (#2142)
- Only cancel loss detection timer if ack-eliciting packets are in flight (#2093, #2117)

Since -17

- After Probe Timeout discard in-flight packets or send another (#2212, #1965)
- Endpoints discard initial keys as soon as handshake keys are available (#1951, #2045)
- 0-RTT state is discarded when 0-RTT is rejected (#2300)
- Loss detection timer is cancelled when ack-eliciting frames are in flight (#2117, #2093)
- Packets are declared lost if they are in flight (#2104)
- After becoming idle, either pace packets or reset the congestion controller (#2138, 2187)
- Process ECN counts before marking packets lost (#2142)
- Mark packets lost before resetting crypto_count and pto_count (#2208, #2209)
- Congestion and loss recovery state are discarded when keys are discarded (#2327)

Since -18

- Change IW byte limit to 14720 from 14600 (#2494)
- Update PTO calculation to match RFC6298 (#2480, #2489, #2490)
- Improve loss detection's description of multiple packet number spaces and pseudocode (#2485, #2451, #2417)
- Declare persistent congestion even if non-probe packets are sent and don't make persistent congestion more aggressive than RTO verified was (#2365, #2244)
- Move pseudocode to the appendices (#2408)
- What to send on multiple PTOs (#2380)