Ack Delay

QUIC Interim, New York, September 2018

When and how much to delay acks

Current: max delay of 1 RTT in transport draft and recovery recommends a 25ms delayed ack timer.

Key Points:

Congestion controllers may need frequent feedback **#1428**

The sender uses max ack delay when calculating TLP and RTO timeouts **#1438**

When and how much to delay acks

Senders

Pick congestion controller, so want control over ACK rate

Need to know max ack delay for TLP and RTO

Receivers

Send ACKs, implement the policy

Are incented to send as few ACKs as possible

3 parameters

Fraction of RTT - Provides delivery rate samples

Time - Max Ack Delay for TLP and RTO

‘Retransmittable’ bytes received - limits unused CWND

Constraints

Alarms wake up late, never early

Implementations don’t want to schedule alarms ‘too soon’. ie: <1ms in the future

Option 1: 1-size fits all

Common suggestion

Don’t delay more than ¼ RTT; use 25 ms delayed ack timer

Limit packets received before sending ack to 10 (vs 2 in TCP)

Con: Reno throughput decreases with ¼ RTT ACKing

Option 2: Sender dictates receiver behavior

Sender impls cong controller, so it should determine policy

Could be fraction of RTT, time, bytes, or some combination

Con: Receiver may not be able to implement ACK policy

Option 3: Receiver dictates own behavior

Similar to the TCP Max Ack Delay (MAD) proposal, which keeps current TCP model of receiver determining it, but adds explicit communication.

Con: Sender needs to adapt cong controller behavior

Option 4: Sender AND Receiver have control

Sender requests no more than fraction of RTT ack delay

Receiver communicates max ack delay

Related Issues

Max Data Received before sending an ACK #1428

Reno is the documented congestion controller

Reno is primarily ACK-clocked

Sending ACKs less frequently increases the amount of time Reno is CWND limited, particularly during slow start

Proposal: Sender sends a transport param indicating retransmittable bytes received before sending an ACK

Explicit Max Ack Delay #981

- TCP has a proposal for explicitly communicating MAD
- Current text causes a spurious TLP the first time an ACK is delayed(by 25ms) if the RTT is ~10ms

Options:

1. Assume 25ms ack delay until we have ‘enough’ data
2. Add a transport param to communicate MAD

Max Ack Delay Permanent Increase #1438

Max Ack Delay is currently a max over the entire connection

Max Ack Delay informs the TLP and RTO timeouts

If an ACK is lost, and the largest_acked does not increase, the next ACK may be sent with a very long ack delay

Proposal: If the largest_acked does not change, set the ack_delay to 0 to indicate largest_acked is old

Removing MinRTO #1017

MinRTO is currently 200ms

RTO already includes MaxAckDelay

TCP MAD proposal removes MinRTO and instead bases it on explicit max ack delay

Proposal: Remove MinRTO