Acks and Recovery

Ian Swett, IETF Seattle Interim

QUIC Ack Principles

1. Once a packet is acknowledged, it can’t be un-acked (aka: it’s irrevocable)
2. ACK frames are never retransmitted, but the information within them is
3. QUIC favors forward progress over perfect feedback
    a. ie: It’s better to ack a new packet than include an old packet that’s already been acked
    b. Even given 50% loss, the odds of losing 255 acks in a row is LOW
    c. If an acknowledgement is missed, the worst thing that happens is a spurious retransmission.
4. NEVER acknowledge packets that have not been received
5. NEVER acknowledge packets with more packet protection

ACK_FRAME: When to send them

1. You MUST NOT send an ACK in response to an ACK only packet
2. You SHOULD send an ACK less frequently than every incoming packet
3. You SHOULD send an ACK sooner(possibly immediately) when it exposes a new gap
4. You SHOULD NOT send an ACK for every incoming packet just because there is one gap in your entire ACK frame.
5. Sending ACKs less frequently than TCP's 'every two packets' can have real benefits in high bandwidth use cases.

Example: *If you receive packet 10 before receiving 9, but 10 is an ACK only packet, DO NOT send an ack (see Rule#1)*

Acknowledgements: When to stop sending old ones

Transport Draft advises waiting until an acknowledgement has been received before you stop sending it.

- This is very conservative
- Minimizes spurious retransmits
- Requires a small amount of state to track acks of acks

Other alternatives may make more sense in different environments

- Send each ACK block for an RTT
- Fixed number of blocks (ie: QUIC in hardware)

Recovery Principles

1. QUIC avoids declaring packets lost until a larger packet has been acked
2. QUIC does not retransmit packets, or even typically frames.
3. Information in lost packets may be re-sent in any order…
    a. Or not at all in the case of stream data
4. The sender decides what to send
    a. The receiver can provide useful information via the transport(ie: STOP_SENDING) or application(ie: Priorities)

WARNING: Issues ahead

Timestamps and ECN (#774, #804, #698)

Growing consensus these should be negotiated extensions to the ack frame

Both exist to improve congestion control

Placing them in the same frame is nice because it:

- Reduces framing overhead
- Simplifies congestion control
- Avoids complex issues of separate frames
  - ie: What if I get ECN marks, but no new acknowledgments?

Dealing with large packet gaps (#432, #613)

Changing connection IDs is going to create relatively large packet number gaps.

The gaps are likely to be larger than the 255 gap size in the QUIC ACK Frame

Gaps larger that 2^16 are impossible in a single ACK frame today

Suggested Solution: Specify when to send multiple ACK frames in a packet

> *ie: Recommend sending one ACK frame per connection ID?*

Alternate Solution: Change the ACK frame format to allow large gaps

Alternate Solution #2: Do both

ACK Framing (#507, probably others)

Goals

1. Efficiently convey acknowledgements for typical receipt patterns
2. Be fast and easy to serialize and parse
3. Allow extensions for ECN and Timestamps

Options

1. Simplify the number of codepoints (16 currently, 1/2/4/8 for both)
2. Make it fixed size (ie: 8 byte largest acked and 1 byte everything else)
3. Varint encoding for everything

NOTE: Option #3 probably should impact all other frames