

# 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. Frames are never retransmitted, but the information within them is
  - a. ACKs ESPECIALLY!
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. **MUST NOT** acknowledge packets that have not been received, decrypted and processed
5. ACK frames **MUST** be included in packets that have equal or greater packet protection than the packets they acknowledge

# 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 unless there is a recently received gap.
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?

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

## Proposal:

- Only use basic ack in the handshake and 0RTT
  - Timestamps already prohibited
- Negotiate use of other ack formats or basic format
- All acks post-handshake are expected to have the negotiated format

**Basic ACK:** Ack blocks

**ACK + TS:** Status quo, timestamp users should propose improvements

**ACK + ECN:** ECN users should propose a format