

Calibrating Minimum RTT Under Low ACK Frequency

draft-li-quic-minimum-rtt-estimation-01

Tong Li, Ke Xu, Bo Wu, Youjian Zhao

IETF-125, QUIC WG, March, 2026

Overview of Standards on ACK Mechanism

Reducing ACK frequency is standardizing in QUIC WG



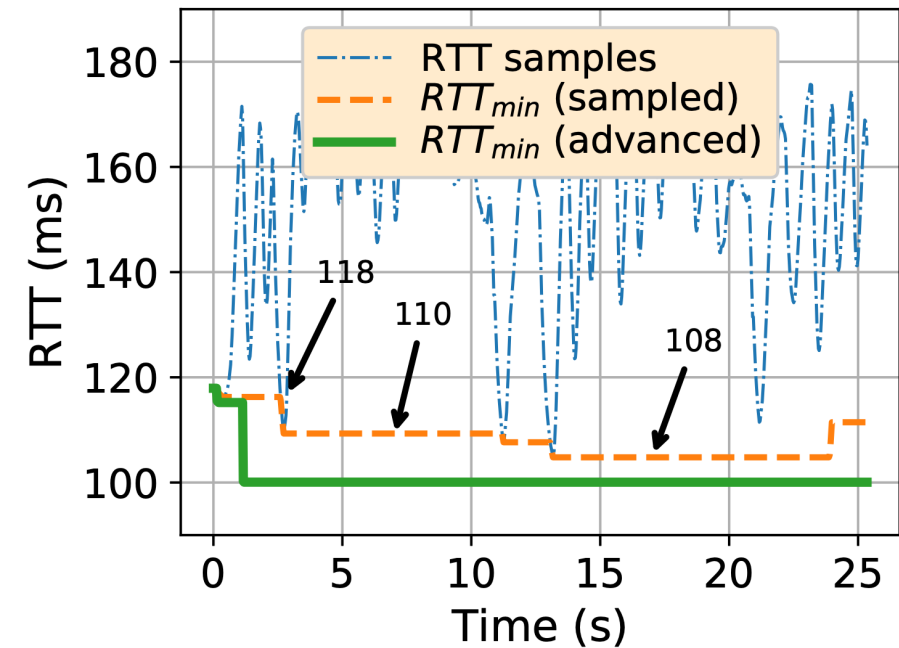
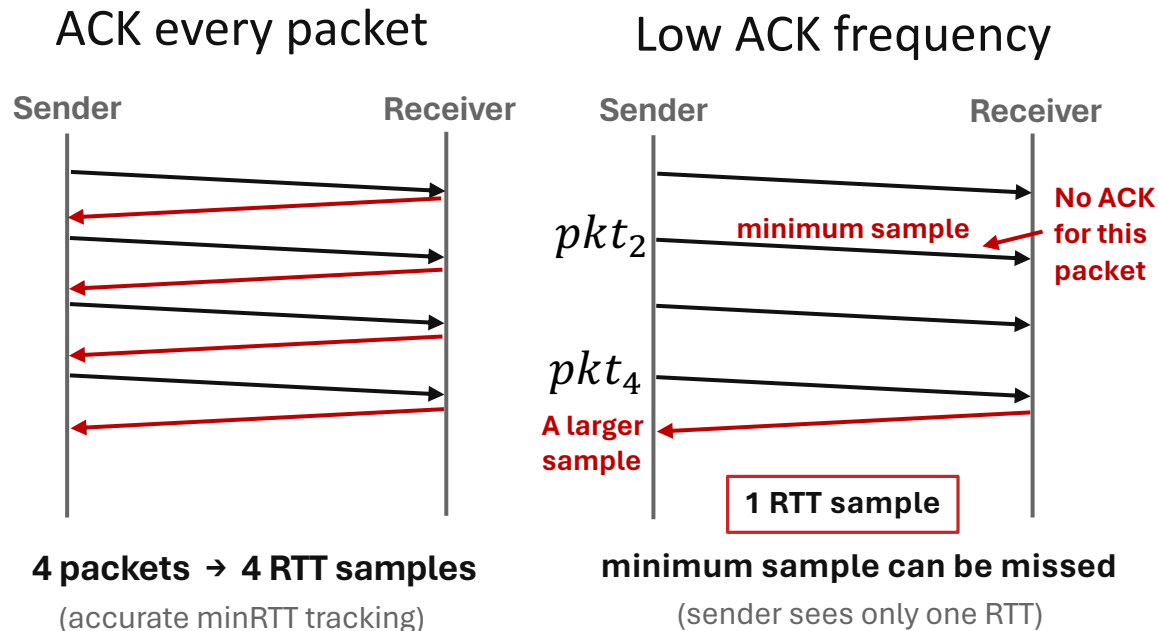
[1] TACK: Improving Wireless Transport Performance by Taming Acknowledgments. ACM SIGCOMM, 2020.

[2] Jana Iyengar, Ian Swett, Mirja Kühlewind. QUIC Acknowledgment Frequency. Work in Progress Internet-Draft, draft-ietf-quic-ack-frequency-14, February 2026

[3] Ian Swett, Joseph Beshay, QUIC Extended Acknowledgement for Reporting Packet Receive Timestamps, draft-ietf-quic-receive-ts-00

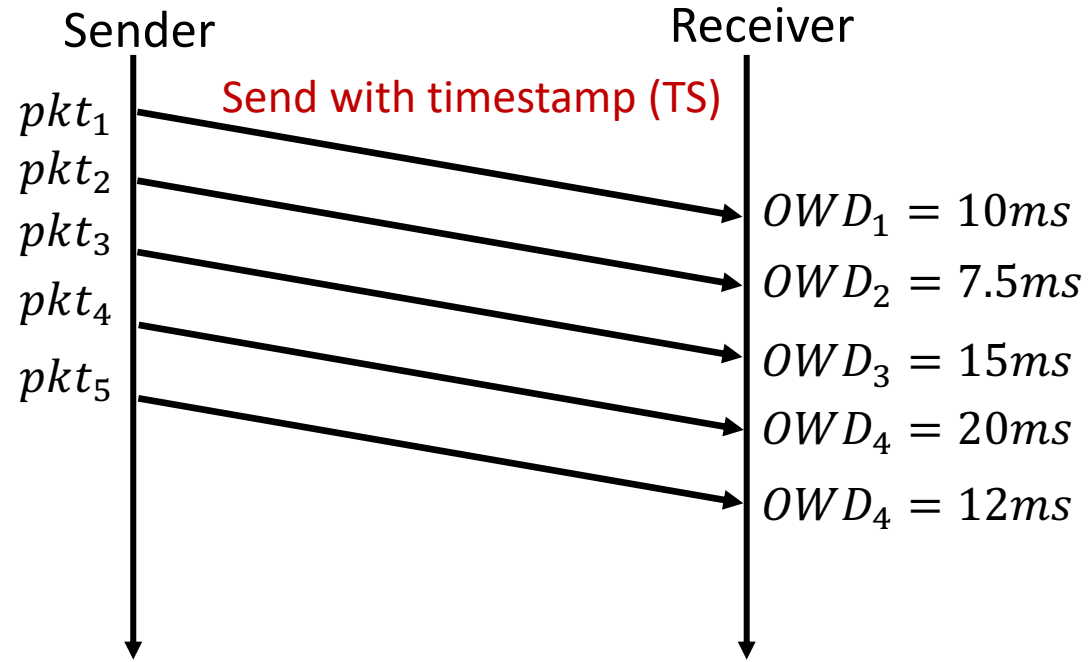
Issues Under Low ACK Frequency

- ACK for every packet $\Rightarrow$ Per-packet RTT samples $\Rightarrow$ Accurate minRTT
- Fewer ACKs $\Rightarrow$ One RTT sample for many packets $\Rightarrow$ Biased minRTT
- Experiments in Wi-Fi links report **8%–18%** larger minRTT estimates
- *Worse in satellite links...*



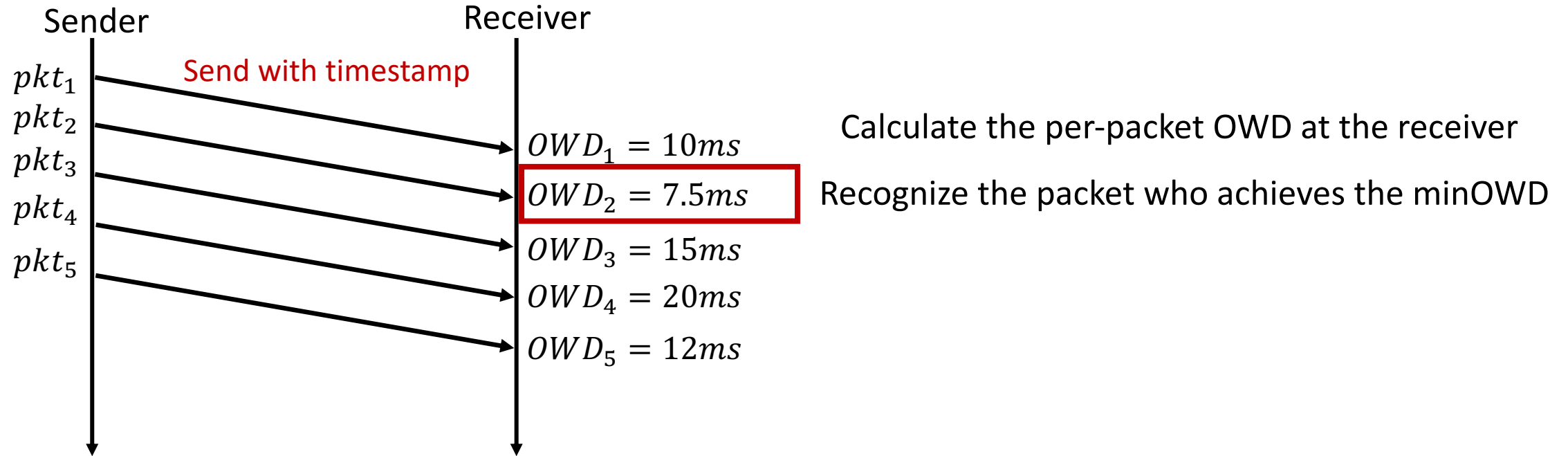
How to calibrate MinRTT?

One-Way Delay (OWD) to the rescue!



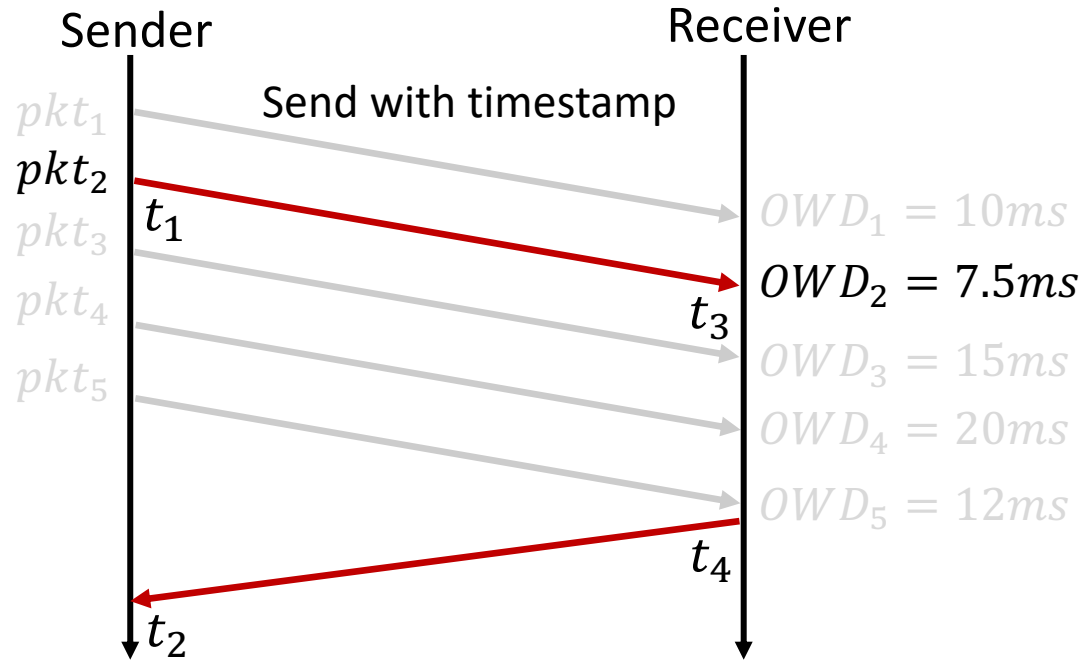
How to calibrate MinRTT?

One-Way Delay (OWD) to the rescue!



How to calibrate MinRTT?

One-Way Delay (OWD) to the rescue!



Once we need to send an ACK, the ACK Delay and Departure TS of pkt_2 other than pkt_5 should be reported!

$$minRTT_{sample} = t_2 - t_1 - ACK\ Delay$$

$$ACK\ Delay = t_4 - t_3$$

- t_1 : Departure TS of packet that achieves minOWD
- t_2 : Arrival TS of ACK frame
- t_3 : Arrival TS of packet that achieves minOWD
- t_4 : Departure TS of ACK frame

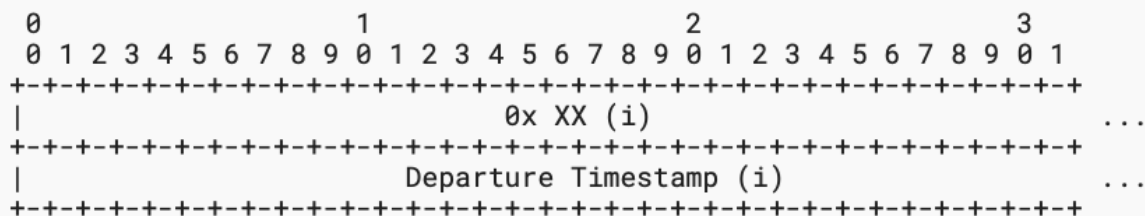
What's new for QUIC protocol

1. Transport Parameter: timestamp-support

A new field named timestamp-support should be added for negotiation between both parties on whether to sync packet departure timestamps in QUIC connection.

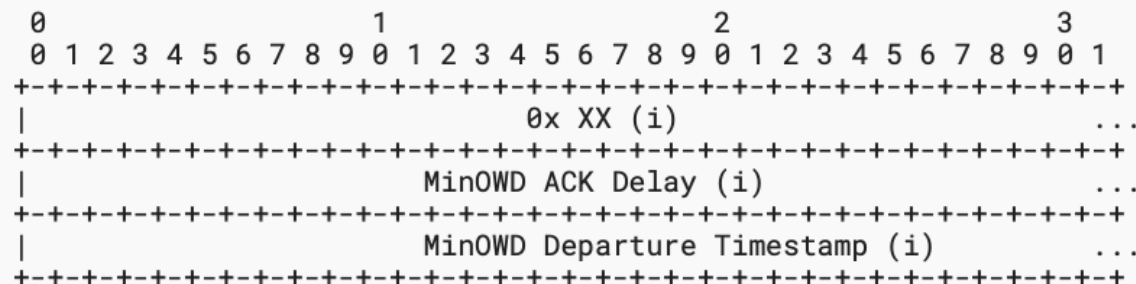
2. TIMESTAMP Frame

A new frame is needed to exchange per-packet departure timestamps, and its format can reuse the TIMESTAMP frame proposed in an existing QUIC draft [4].



3. MINOWD-ACK Frame

A new frame to report the ACK Delay and Departure Timestamp of the packet who achieves the minimum one-way delay.



[4] Christian Huitema, QUIC Timestamps for Measuring One-Way Delays, Work in Progress, Internet-Draft, draft-huitema-
quic-ts-08, August 2022. Progress Internet-Draft, draft-huitema-quic-ts-08

Conclusion

1. Low ACK frequency biases minRTT Estimation

2. New method to calibrate minRTT estimation

- Receiver calculates per-packet OWD, reports timestamp selectively
- Sender calculates RTT samples according to timestamps

3. QUIC Modification

- Transport parameter: Timestamp-support
- Recall draft-huitema-quic-ts-08: TIMESTAMP frame
- Add a new MINOWD-ACK frame: Report ACK Delay and Departure TS of pkt_minOWD

Any comments welcome

FAQs

Q1: Why minRTT estimation **important**?

A2: minRTT is a fundamental state variable in transmission control

- Used by BBR to estimate congestion window [CARDWELL-BBR]
- Used by QUIC loss detection to reject implausibly small RTT samples [RFC9002]
- Used to update the ACK frequency in [SIGCOMM 20]

Q2: Why use **OWD** to calibrate minRTT?

A2: OWD change reflects RTT change in most cases. OWD can deliberately identify the packet with the minRTT instead of hoping legacy ACK sampling happens to catch it.

Q3: Need clock synchronization for one-way delay?

A3: No, it only relies on **comparing relative OWDs** using the receiver's local clock and reporting the packet that achieves the minimum OWD.

Q4: Why receiver report **departure timestamp** instead of **packet number** of pkt_ minOWD?

A4: Sender can compute minRTT sample according to departure timestamp or packet number of the packet that achieves minOWD. However, departure timestamp feedback **avoids sender-side table lookup**.