

QUIC Packet Receive Timestamps

draft-ietf-quic-receive-ts
QUIC @ IETF 124

Joseph Beshay (Meta), Ian Swett (Google)

Overview of Draft

Enable the peer to report packet receive timestamps for some or all ACKed packets. The timestamp is connection local and not intended to be absolute.

Additional fields added at the end of the ACK frame:

```
Timestamp Range {  
    Delta Largest Acknowledged (i),  
    Timestamp Delta Count (i),  
    Timestamp Delta (i) ...,  
}
```

draft-smith-quick-receive-ts-03 → draft-ietf-quick-receive-ts-00

- Draft adopted by WG
- Trimmed editor/author list ([#25](#))
- One minor editorial change to the Multipath extension reference ([#24](#))
- No open PRs

Open Issue ([#23](#)): Make this extension independent from the ACK_FRAME

Reasoning:

- Timestamps should be independent of ACKs
- Easier extensibility

The editors view:

- Timestamps are closely tied to ACKs, just like ECN counts.
- Separating them introduces unnecessary implementation complexity. Existing use-cases expect the data to be available with ACKs:
 - Google CC
 - SQP CCA
- We can define how this coexists with other extensions (discussed more later)

Open Issue ([#19](#)): Decide how multiple ACK Frame extensions coexist

“Multiple extensions can alter the ACK Frame or define new codepoints for variations on the ACK frame, such as `{{?MULTIPATH}}`. Each extension defines how it co-exists with past extensions. If multiple extensions add more information to the ACK Frame, as this receive timestamp extension does, the additional extensions are appended at the end of the ACK Frame in the order of their RFC number, unless otherwise specified.”

Based upon prior discussions, the editors intend to drop the specification of using the RFC number. The timestamps go at the end of the frame. Future extensions can specify more elaborate coexistence mechanisms if needed.

Open Issues ([#14](#), [#21](#)): Best practices for writing Timestamps

#14: Emphasize that timestamps are best effort within the current ACK packet and should not result in a whole new packet.

#21: If we can't report all timestamps, should we prefer the latest received or the highest packet number?

Open Issue #9: [Write Security Considerations](#)

Ideas

Ensure manipulated timestamps can't produce unrealistic bandwidth (ie: BBR)

Other Ideas?

What's next?

- What's left to Add/Fix/etc in the draft?
 - It seems like the right featureset to the authors, does the WG agree?
 - Please read for clarity and editorial suggestions.
- Implementations:
 - Mvfst (adopted-00 updates in progress)
 - Google quiche (individual-01 Mostly complete)