QUIC Packet Receive Timestamps

draft-smith-quic-receive-ts

QUIC @ IETF 123

Connor Smith, *NVIDIA*, **Ian Swett**, *Google*, Joseph Beshay, *Meta*, Sharad Jaiswal, *Meta*

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) ...,
}
```

Changes since draft-smith-quic-receive-ts-02

- Always include Receive Timestamps in the existing ACK frame once negotiated (ie: In 1RTT packets). PR #8
  - Removes Extended Ack proposal from IETF122
- Support reporting out-of-order packet number ranges. PR #15
- Add example of ACK format. PR #18
- Proposal for co-existence of multiple Ack Extensions. PR #17

Open Issue #19 : Decide how multiple ACK Frame extensions coexist

Current text:

> *"Multiple extensions can alter the ACK Frame or define new codepoints for variations on the ACK frame, such as {{?MP-QUIC}}. 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."*

Is this what the WG wants?

Related: Issue #12 - Ensure format is compatible with MP-QUIC

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 (-01 Implemented, updates in progress)
  - Google quiche (-01 Mostly complete, -00 support removed)