HTTP Header Compression over QUIC

Alan Frindell
Facebook
June 2017

Current Draft - Serialized HPACK

- HPACK frames are sent on individual streams, with a sequence number
- HPACK decoder must decode frames in order
- Potential unnecessary HOL blocking

QPACK

- Encoder has explicit control of header table
- Table modifications happen on their own stream
- HOL Blocking *may* occur if a reference arrives before an insert
- Some special reference counting for evictions

QCRAM

- Very similar to HPACK

- Only encode a reference if
  - Encoder has received an ack for the insert
  - The insert is in the same QUIC packet
- Otherwise encode a literal

- Evictions must be processed in order, may cause HOL blocking

Comparing Implementations

- I built all implementations of all three schemes, and a simulator

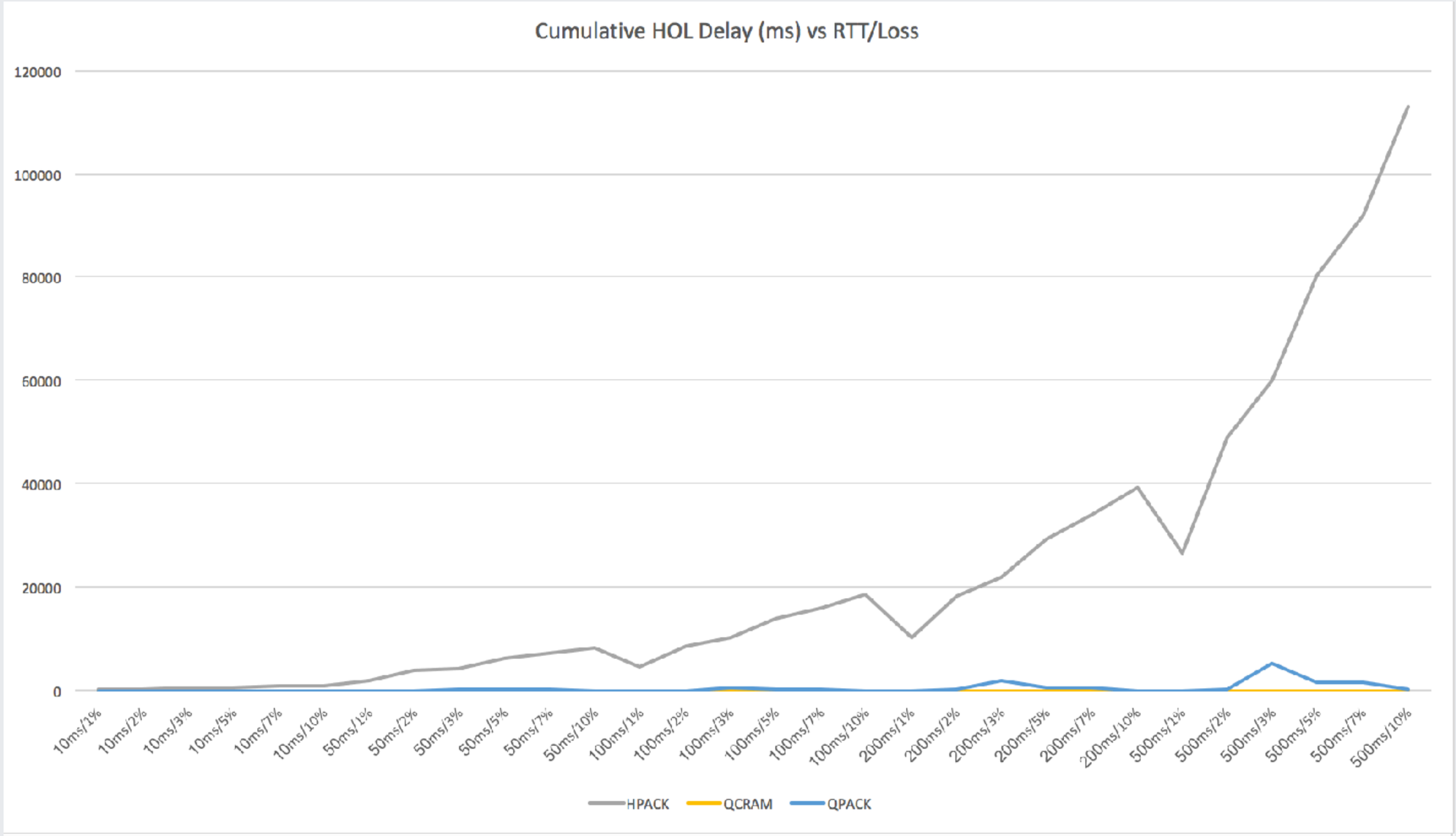
- Simulator has two knobs
  - RTT
  - Loss rate (loss treated as independent events)

- Input is a HAR file containing HTTP request headers and request start timing

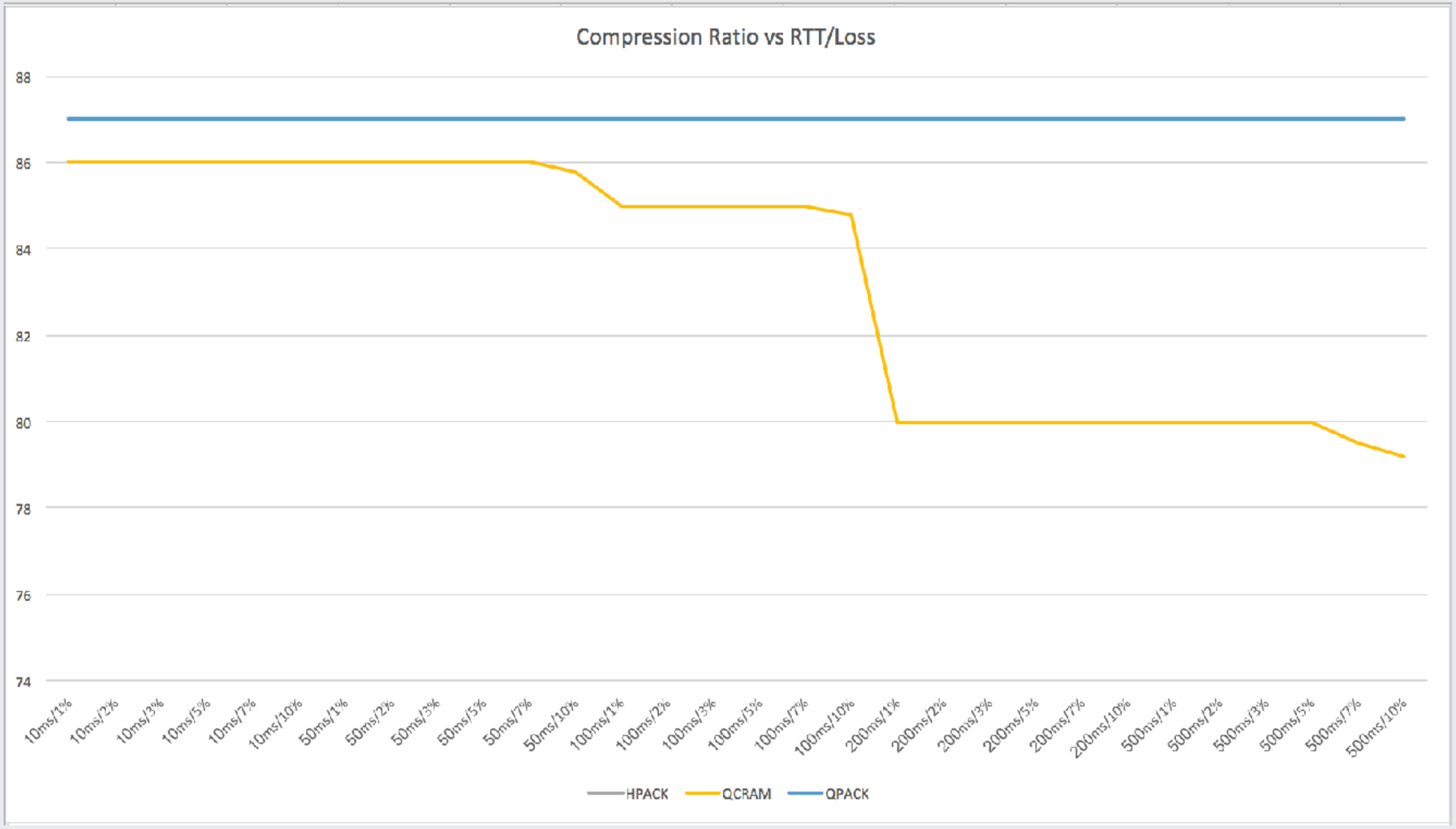
Experiment Setup

- Load Facebook news feed with a forced 100ms RTT
- 227 requests to 5 domains
  - All facebook.com and fbcdn.net requests were coalesced
- Total elapsed time ~15 seconds
- Varied RTT from 10 - 500ms
- Varied Loss from 0 - 10%

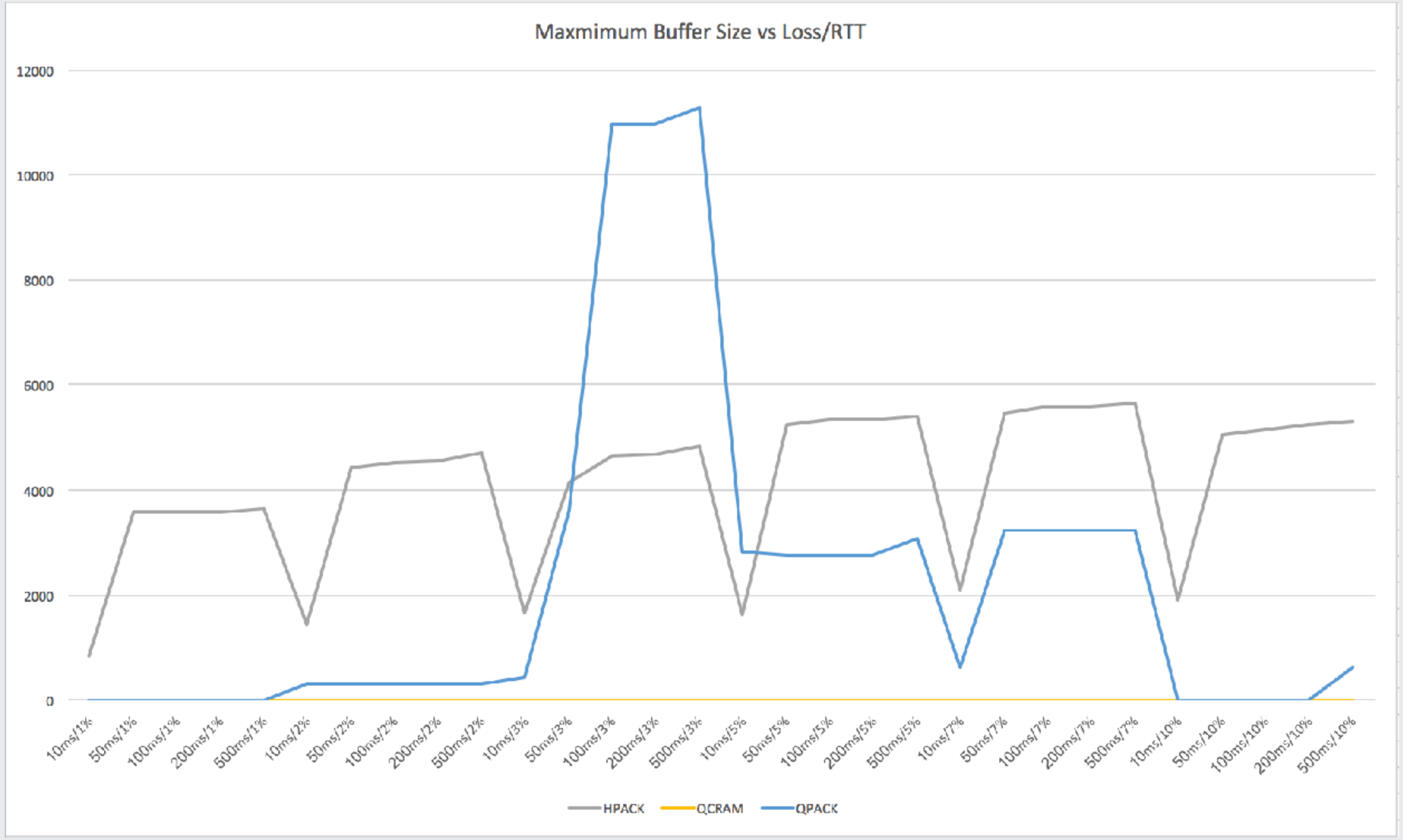
HOL Delay


Cumulative HOL Delay (ms) vs RTT/Loss
120000
100000
80000
60000
40000
20000
0
10ms/1%
10ms/2%
10ms/3%
10ms/5%
10ms/7%
10ms/10%
50ms/1%
50ms/2%
50ms/3%
50ms/5%
50ms/7%
50ms/10%
100ms/1%
100ms/2%
100ms/3%
100ms/5%
100ms/7%
100ms/10%
200ms/1%
200ms/2%
200ms/3%
200ms/5%
200ms/7%
200ms/10%
500ms/1%
500ms/2%
500ms/3%
500ms/5%
500ms/7%
500ms/10%
HPACK
QCRAM
QPACK

Compression Ratio



HOL Buffering



Observations

- Serialized HPACK is not great in the presence of loss
- QCRAM has 0 HOL blocking (if there are no evictions) at the expense of compression ratio
- QPACK has identical compression ratio to HPACK, with some HOL blocking

Implementation Notes

- **QCRAM**
  - Easier to start, because of its similarity to HPACK
  - Full implementation requires deeper integration with transport
  - How to know when to duplicate an un-acked index?

- **QPACK**
  - Mostly a rewrite of the header table
  - Requires two-pass decoding to prevent buffering partially decoded header blocks
  - Has some nice flexibility

Next Steps

- Run the simulation with more varied input
  - Third-party CDN, apps instead of browser

- More runs with lowish loss rates (0 - 2.5%) and also high rates (up to 20%)

- Consider hybrid implementations
  - QPACK + insert acks

- Thanks to Mike Bishop and Buck Krasic for input
- More discussion can be found on the IETF QUIC mailing list