HTTP/QUIC

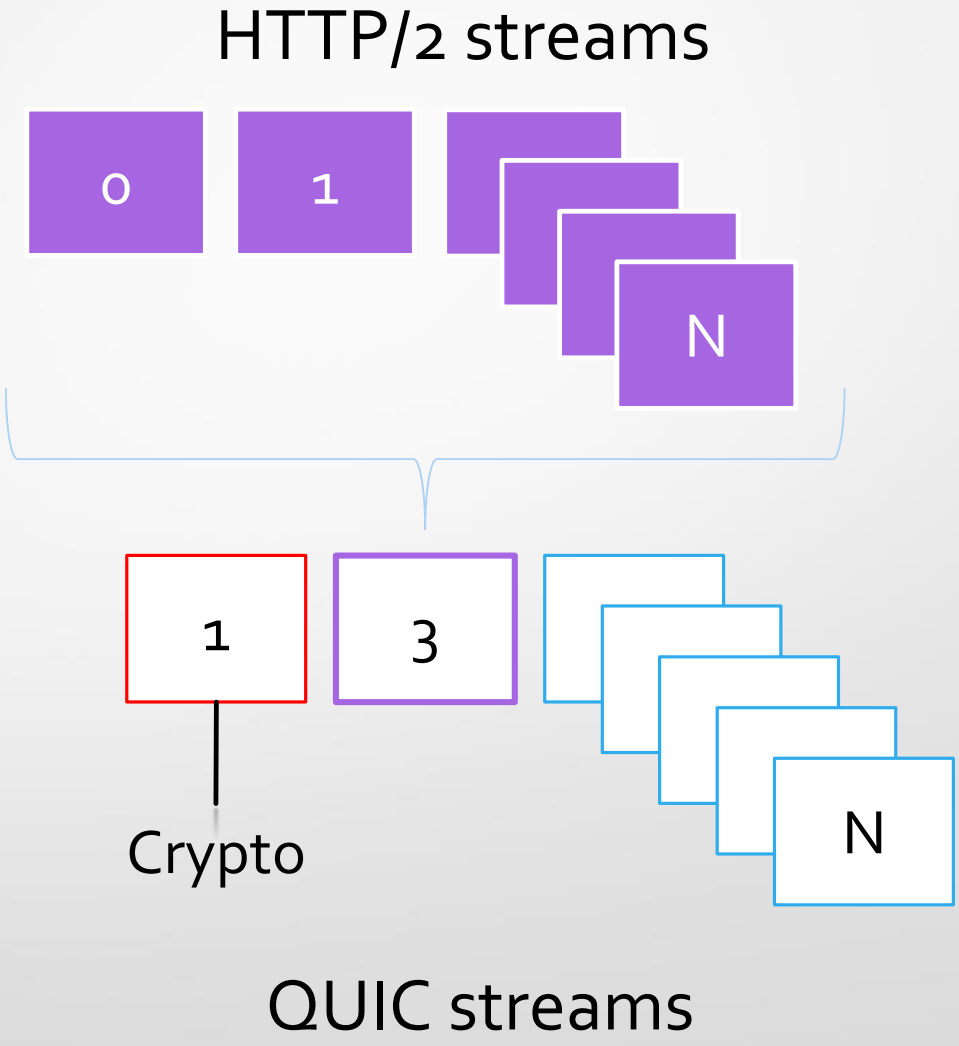
draft-ietf-quic-http-01

Notable Changes

- Requests now have a pair of streams to themselves
  - HPACK
- ALPN and Alt-Svc reworked
- Adopted different SETTINGS format

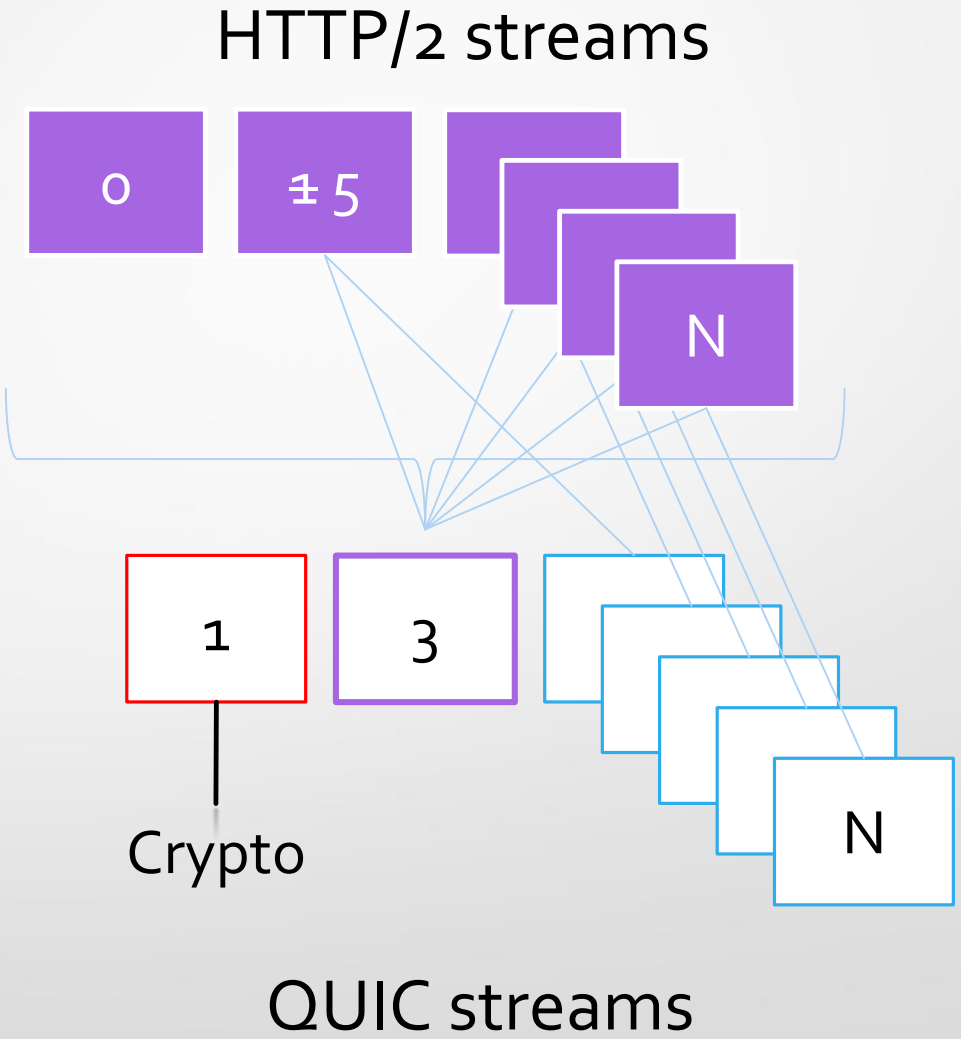
-00 Stream Usage

- Stream 1 reserved for crypto
- Stream 3 reserved for abridged HTTP/2 session
  - Reflects migration path from TCP to QUIC
  - Functionality added to QUIC is removed from HTTP/2
    - PING
    - GOAWAY
    - Flow Control



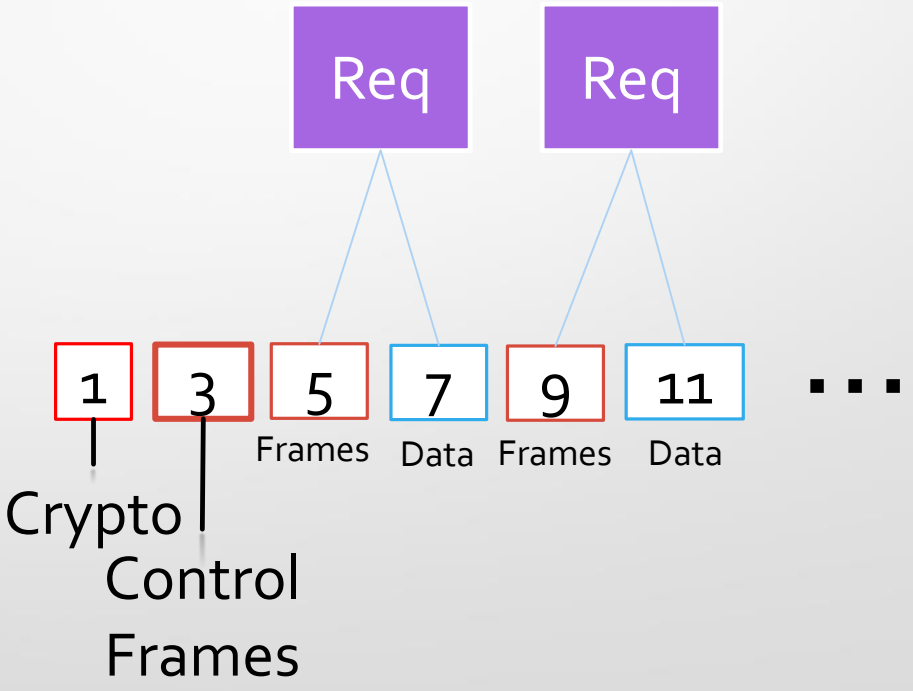
-00 Stream Usage

- Stream 1 reserved for crypto
- Stream 3 reserved for abridged HTTP/2 session
- HTTP/2 streams straddle QUIC Stream 3 *and* another QUIC stream
  - H2 Stream 0 is only on QUIC Stream 3
  - Other QUIC streams replace DATA frames
  - All other frames (HPACK) on QUIC Stream 3

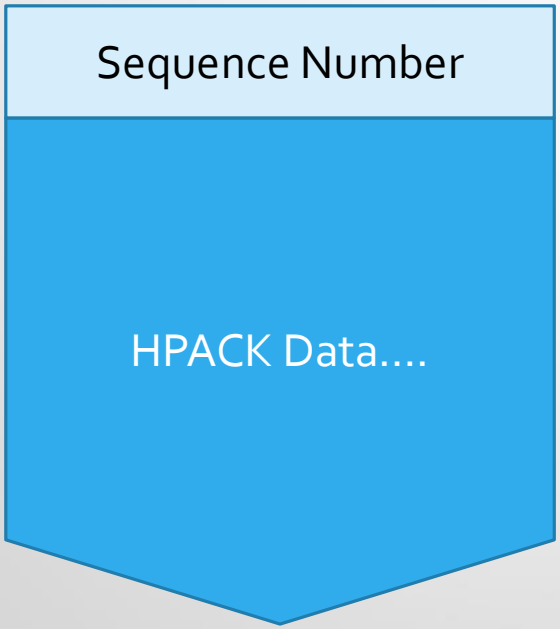


-01 Stream Usage

- Stream 3 – Connection Control Stream
  - Carries session-wide info (SETTINGS, PRIORITY)
- Each request occupies two streams
  - Message control stream – HEADERS, etc.
  - Unframed data stream carries message payload
- No muxing in HTTP-layer framing
  - Still uses frames



Shoehorning HPACK



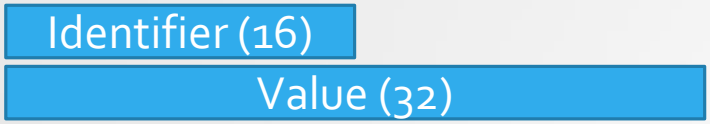
- HTTP/QUIC -01 still uses HPACK
- Adds a counter on HPACK frames
  - Requires decoder process frames in encode-order
- No *more* HOLB than before, but no less

Connection Negotiation

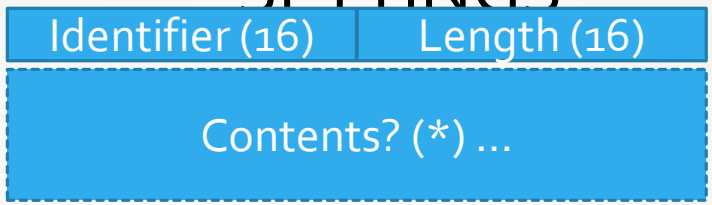
- Needed an ALPN token; was "quic"
  - Defined hq for RFC
  - Defined hq-xx for drafts (e.g. hq-01)
- HTTP/QUIC support still detected by use of Alt-Svc
- Alt-Svc "v" parameter as version negotiation hint
  - Format of version changed
    - Was given as list of numbers; implied prefix of ASCII Q plus three-digit number in ASCII
      - 37 magically understood as version "Q037"
    - Now permits two representations:
      - x + hex: Accommodates "x1" for final value
        - "xffoooonn" for draft versions
      - c + 4-char token: Accommodate "Q037" for existing styles
        - No %xx: If it's not a token, use the other one
  - List usage modified to better match draft-kamp-httpbis-structure
    - Repeated v= options instead of v=value,value,value

Changes to SETTINGS

HTTP/2 SETTINGS



HTTP/QUIC SETTINGS



- Based on draft-bishop-httpbis-extended-settings
  - HttpBis feedback: Save for protocol rev, not an HTTP/2 extension with separate identifier space
- Borrows heavily from RFC7540 SETTINGS text
- Values are length-prefixed blobs
  - Currently static 16-bit length; do we need this much?
- ~~ACK works differently:~~
  - ~~Sender sets flag if ACK is desired~~
  - ~~Recipient sends back SETTINGS_ACK frame listing the values which it did not understand~~

Optimization for Boolean values

If length=0, use the value of a flag in the header

QPACK

draft-bishop-quic-http-and-qpack

Order-Independent Header Compression

HPACK

- Inserts always append to the dynamic table
- Table size managed implicitly
  - If the table overflows, drop oldest value
  - Indices change over time

QPACK

- Inserts are to an explicit index
- Table size managed by explicitly deleting entries
  - If the table overflows, kill the connection
  - Indices are consistent over time

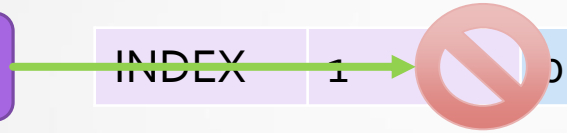
Index	Use Count	Del Count	Key	Value
…	…	…	…	…

How to be Order-Independent

Index	Use Count	Delete Count	Key	Value
…	…	…	…	…
Insert KEY=VALUE at INDEX	INDEX	1	0	KEY	VALUE
Insert KEY=VALUE at INDEX	INDEX	2	0	KEY	VALUE
Reference INDEX	INDEX	3	0	KEY	VALUE
Delete INDEX (uses=3)		0	0		

Insert Reordering

Index	Use Count	Del Count	Key	Value
…	…	…	…	…
Reference INDEX	INDEX	1	0	
Insert KEY=VALUE at INDEX	INDEX	2	0	KEY	VALUE
Insert KEY=VALUE at INDEX	INDEX	3	0	KEY	VALUE
Delete INDEX (uses=3)		0	0		



- References block if the field isn't defined yet
- Inserting the same value multiple times is permitted to reduce the chance for this.

Delete Reordering

Index	Use Count	Del Count	Key	Value
…	…	…	…	…
INDEX	1	0	KEY	VALUE
INDEX	2	0	KEY	VALUE
INDEX	2	3	KEY	VALUE
	0	0		

Insert KEY=VALUE at INDEX

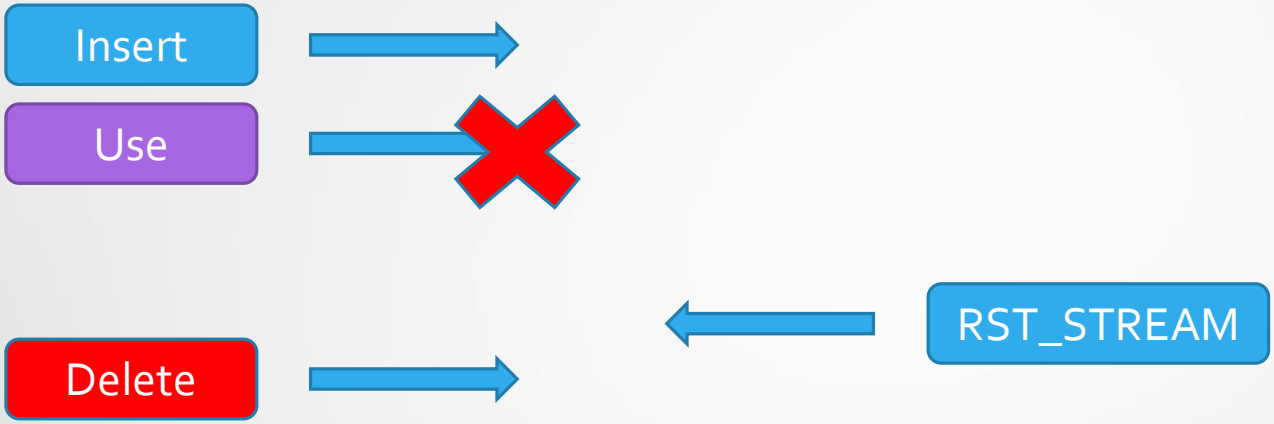
Insert KEY=VALUE at INDEX

Delete INDEX (uses=3)

Reference INDEX

- Deletes include the number of uses
- If not all references have arrived yet, deletes deferred until the last one arrives

The fatal flaw….



Packet loss + no retransmission =
references that never arrive

(Equally true for HPACK)

More generally

- Suppose an application protocol requires some shared state
- How can it safely establish/maintain this shared state?
  - Make some stream(s) special; if it/they are ever RST, kill the connection
    - We need language for this around Streams 1 and 3
  - Make some data on each stream special and get the transport to guarantee delivery
  - Shove it into the transport somehow special and have QUIC guarantee delivery