Unbound DATA frames in HTTP/3

Optional optimization to reduce encapsulation complexity

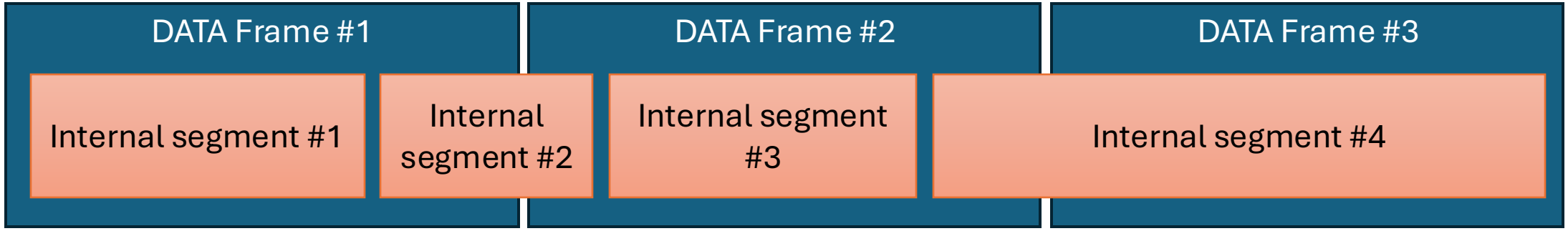
`draft-rosomakho-httpbis-h3-unbound-data-00`

Yaroslav Rosomakho David Schinazi

HTTPBis
IETF124, November 2025, Montreal

DATA frames in HTTP/3

- Can indicate where streamed data ends and trailers HEADERS begin
- Allow future extensions in HTTP/3
- Inner segments inside DATA frames do not have to align with the data frames!



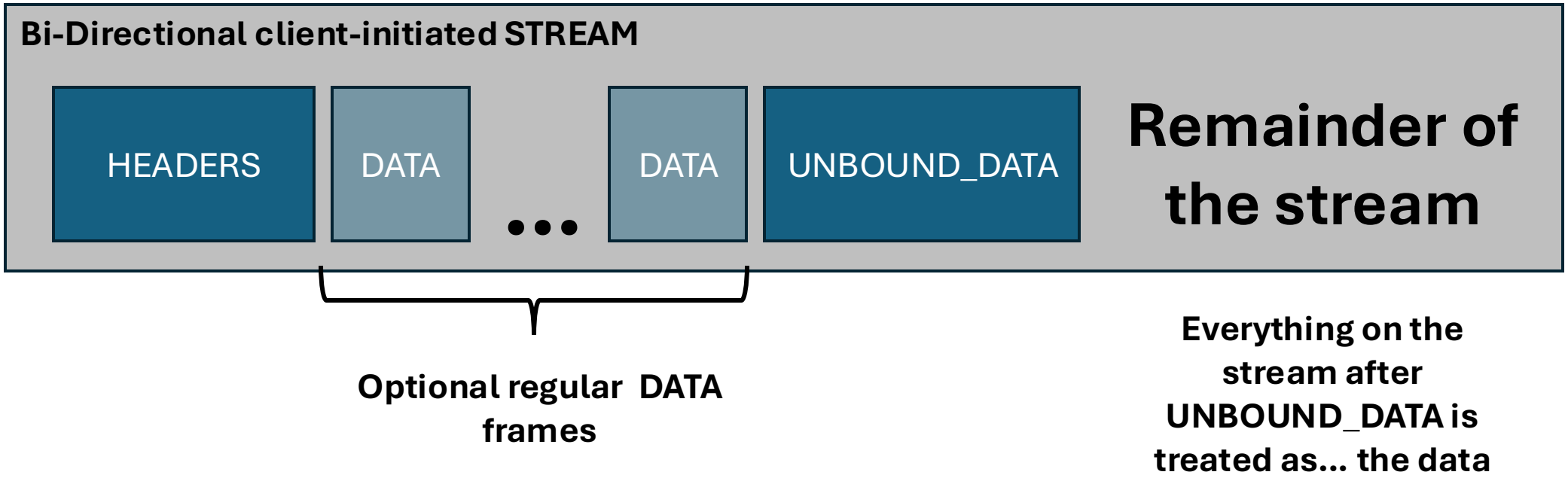
Drawbacks of DATA frames

- Useless if trailers are not used
- Add overhead
- Add complexity
- Create unnecessary state
- Inner segments must still carry their own length

Proposal: UNBOUND_DATA frame

- A simple 0-length indicator that the rest of the QUIC stream is data
- An optional capability negotiated through SETTINGS
  - Don’t want to use it? Don’t negotiate it!
  - Planning to send a trailer? Don’t use it!
- Can be sent at any point in time after the HEADERS

UNBOUND_DATA on the stream

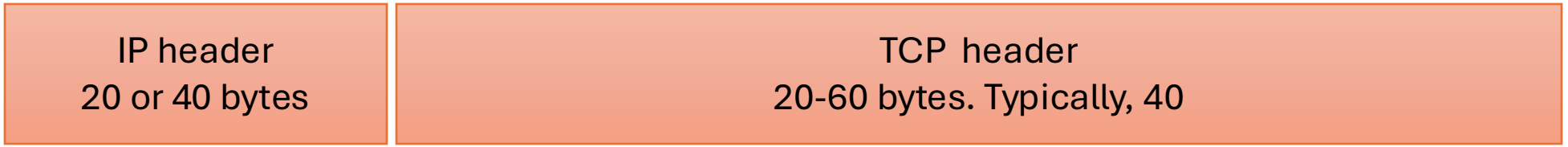


How much does it save?

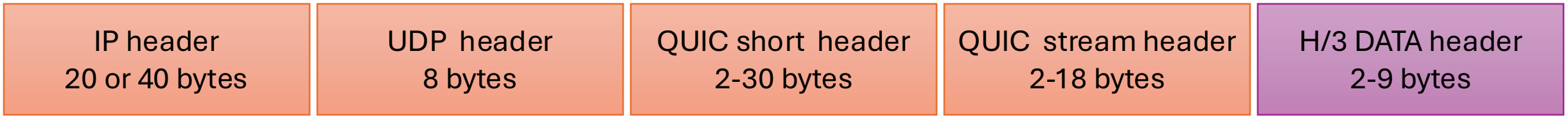
- It depends… could be as little as 3 bytes per frame, but could help avoid unnecessary memory copy
- Can be particularly efficient when tunneling things

Practical example: zero copy of TCP data to CONNECT stream

Original TCP packet



Encapsulated data in HTTP/3 CONNECT



Depending on connection ID and various varints addition of HTTP/3 DATA frame header may require a memory allocation and a copy

FAQ

- Why not CAPSULE frame type?
  - Not all DATA is Capsules (for example TCP data, Websockets to name a few)
  - CAPSULE frames would require transformation from regular Capsules
  - There is a proposal for unbound Capsules…
- Does it contradict RFC9114?
  - “Once the CONNECT method has completed, only DATA frames are permitted to be sent on the stream. Extension frames MAY be used if specifically permitted by the definition of the extension.”
- Why not 0-length DATA as a marker?
  - Some (mis-)use 0 length DATA frames as a keepalive

Thank you!

Thoughts? Suggestions? Comments?