HTTP/QUIC

IETF 103

Notable Changes since Montreal:

No more flags
Priority placeholders
Simpler settings
Types for unidirectional streams
Settings assumed in 0-RTT

You Are Here

Levels of SETTINGS Synchronization

Full Negotiation

- Client MUST know server's settings before generating a SETTINGS frame
- Server MUST process client's SETTINGS before any other data
- Enables full offer-select negotiation

Declarations

- Settings are unilateral declarations of capability
- Each endpoint MUST know the other side's settings before generating data on other streams
- Negotiation is an extrapolation from the sent+received values

Defaults

- Data can be sent/processed using reasonable assumptions before SETTINGS arrives
- New capabilities found in SETTINGS can be opportunistically used later

You Are Here

Levels of SETTINGS Synchronization

Full Negotiation

- Client MUST know server’s settings before generating a SETTINGS frame
- Server MUST process client’s SETTINGS before any other data
- Enables full offer-select negotiation

Head-of-Line Blocking

Declarations

- Settings are unilateral declarations of capability
- Each endpoint MUST know the other side’s settings before generating data on other streams
- Negotiation is an extrapolation from the sent+received values

Defaults

- Data can be sent/processed using reasonable assumptions before SETTINGS arrives
- New capabilities found in SETTINGS can be opportunistically used later

ⓧ You Are Here

Levels of SETTINGS Interlocking

Full Negotiation

- Client MUST know server's settings before generating a SETTINGS frame
- Server MUST process client's SETTINGS before any other data
- Enables full offer-select negotiation

Head-of-Line Blocking

Declarations

- Settings are unilateral declarations of capability
- Each endpoint MUST know the other side's settings before generating data on other streams
- Negotiation is an extrapolation from the sent+received values

Lack of Synch Point

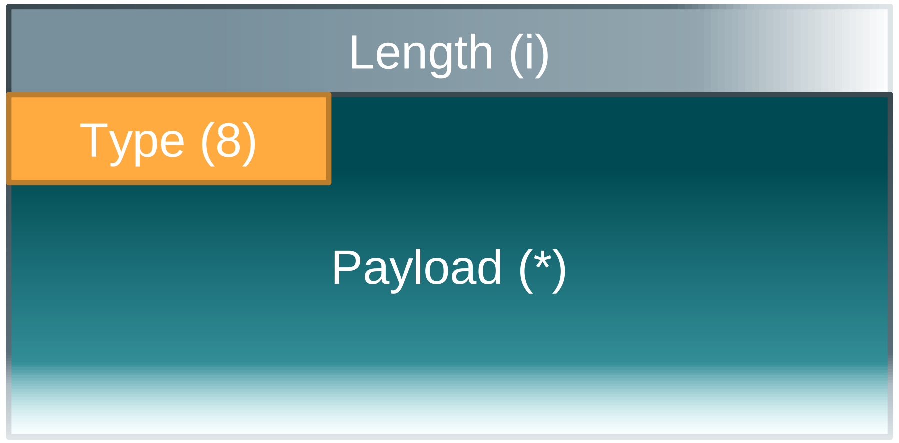
Defaults

- Data can be sent/processed using reasonable assumptions before SETTINGS arrives
- New capabilities found in SETTINGS can be opportunistically used later

QUIC

Length-Prefixed Frames Considered Irksome

- Endpoints often generate output in chunks
- Annoying to length-prefix each chunk when all remaining chunks will share the same type



Idea: What if Length=0 means "to end of stream"?

Levels of Remainder Framing

Do Nothing

- We’re late in the process
- This is a non-critical new feature

You Are Here

DATA frames are special

- DATA in server responses is most common place to use this

New Framing Rule

- Consistent processing across frame types
- Likely to be used by HEADERS on requests w/o body

Initial Prioritization

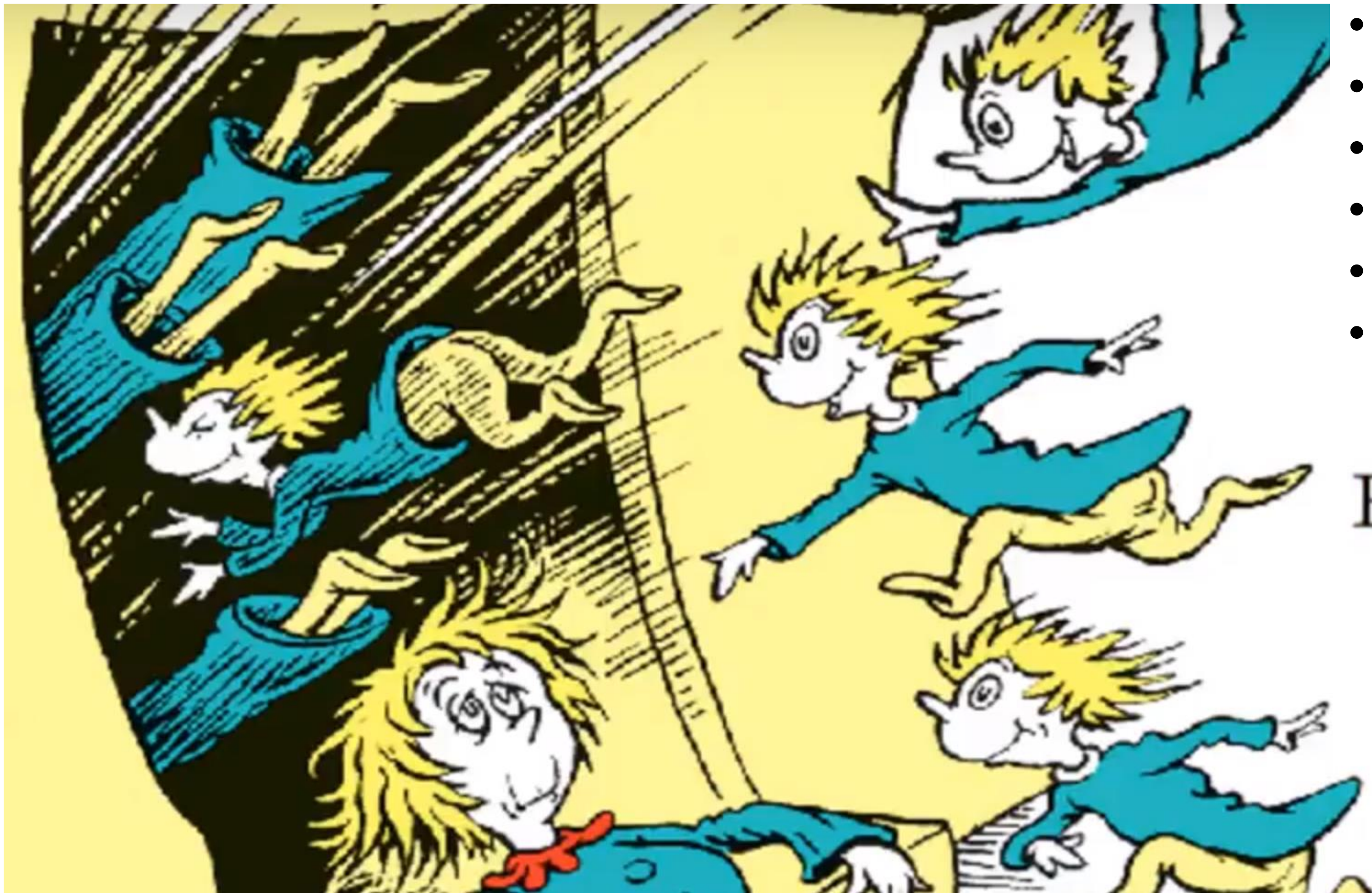
- Reordering makes priority hairy
  - HTTP/2 included priority within HEADERS, then updated with PRIORITY if needed
  - HTTP/QUIC eliminates the embedded priority to ensure consistent ordering
- Leaves a gap: Requests can arrive, be processed before the priority information

What’s in a Name?

QUIC

Instead of 23 “Daves,” Mrs. McCave wishes she’d named her kids…

- Bodkin Van Horn
- Hoos-Foos
- Snimm
- Hot-Shot
- Sunny Jim
- Shadrack
- Blinkey
- Stuffy
- Stinkey
- Putt-Putt
- Moon Face
- Marvin O'Gravel Balloon Face
- Ziggy
- Soggy Muff
- Buffalo Bill
- Biffalo Buff
- Sneepy
- Weepy Weed
- Paris Garters
- Harris Tweed
- Sir Michael Carmichael Zutt
- Oliver Boliver Butt
- Zanzibar Buck-Buck McFate



But she didn’t do it.
And now it’s too late.

“Too Many Daves,” from *The Sneeches and Other Stories*, by Dr. Seuss

Naming HTTP/QUIC

- HTTP/QUIC doesn’t clearly relate to HTTP/1.1, HTTP/2
- This really isn’t “HTTP/2 over QUIC”
  - …so please stop calling it that!
- The mapping isn’t “QUIC”
  - …so please stop calling it that!
- HTTP WG declined to backport changes from HTTP/QUIC to HTTP/2 extensions
  - Because “QUIC is the future”
  - (…we think)

Naming HTTP/QUIC

- HTTP/QUIC doesn't clearly relate to HTTP/1.1, HTTP/2
- This really isn't "HTTP/2 over QUIC"
  - ...so please stop calling it that!
- The mapping isn't "QUIC"
  - ...so please stop calling it that!
- HTTP WG declined to backport changes from HTTP/QUIC to HTTP/2 extensions
  - Because "QUIC is the future"
  - (...we think)

QUIC

Go ye forth and implement!

QUIC