

Using HTTP/2 as a Transport for Arbitrary Bytestreams

draft-kinnear-httpbis-http2-transport-00

Eric Kinnear & Tommy Pauly
HTTPBIS
IETF 103, November 2018, Bangkok

Use Cases

Generic transport for secure, arbitrary bytestreams

Multiplexed streams

- Low setup cost for new streams

- Single congestion and recovery context

Can be peer-to-peer

Ability to tunnel traffic as necessary

Example: Remote IPC

Document Motivation

HTTP/2 provides framing layer with many of the desired transport features

- Configuration exchange

- Multiplexed streams

- Flow control

- Stream relationships

Separate this from HTTP semantics, provide as a generic transport

Potential Strategies

1. New ALPN token
 - Conceptually a new protocol
2. Allow empty HEADERS frames
3. Introduce STREAM frame
 - Negotiated protocol extension

Potential Strategies

New ALPN Token

Something like “h2t”

Allows negotiation alongside h2

Potential to fall back between implementations depending on server support

Conceptually a new protocol

Public information during negotiation

Potential Strategies

Allow empty HEADERS frames

No change to wire format, minimal text changes

Easy for implementations to adopt

Can introduce a SETTING for graceful negotiation

Conflates ideas in an unclean way

Potential Strategies

Negotiated extension

Use the extension mechanism to negotiate the use of a new frame type: STREAM

Negotiate with SETTINGS

Conceptually a STREAM frame is the same as an empty HEADERS frame

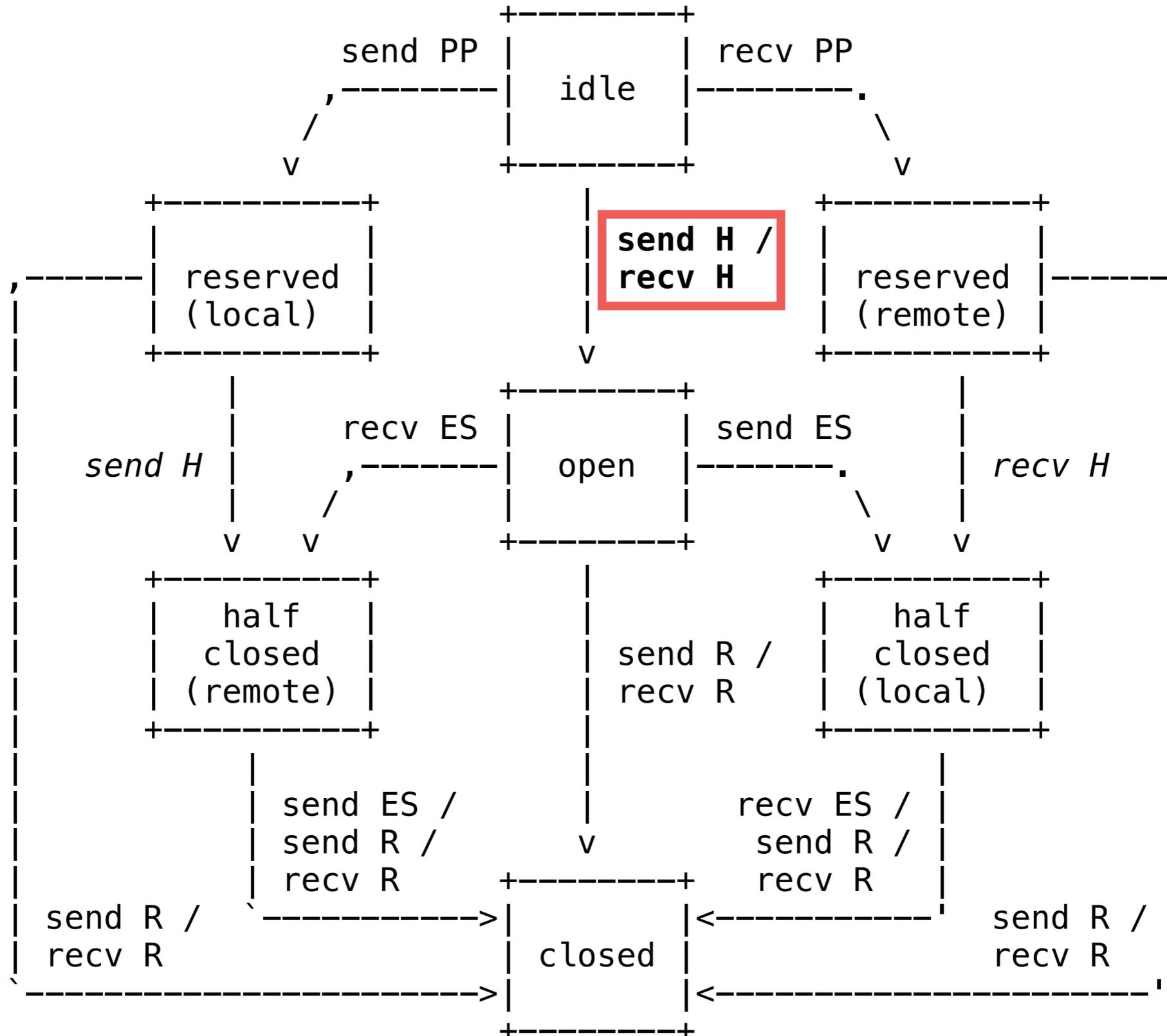
Described today in the document

STREAM Frame Details

A STREAM frame is conceptually an empty HEADERS frame:

- It modifies the stream state in the same manner
- It is allowed at the same times, causes the same errors
- Streams opened with STREAM frames behave in the same manner as before
- Stream limits from SETTINGS apply in the same way as to streams opened with the headers frame

Stream States



Feedback

Additional use cases?

Other options beyond the three discussed?

Any preference towards an option other than the third (STREAM frame)?

