WEBTRANS WG
IETF 107

Virtual Meeting

Friday, March 30, 2020

22:10 - 00:10 UTC

15:10 - 17:10 Pacific Time

Mailing list: webtransport@ietf.org
Jabber Room: webtrans@jabber.ietf.org

IETF 107 Meeting Tips

https://www.ietf.org/how/meetings/107

https://datatracker.ietf.org/meeting/agenda

This session is being recorded

- WebEx Link – **Make sure your video is off.**
- **Mute your microphone unless you are speaking, use headphones.**
- **Use Webex chat only to join the mic queue.**
  - "+q" adds you to the queue
  - "-q" removes you from the queue
- **Add your name and affiliation to the virtual blue sheet in the session Etherpad via IETF Datatracker Meeting agenda**
- **Join the session Jabber room via IETF Datatracker Meeting agenda**
- **Please state your full name before speaking**

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- BCP 9 (Internet Standards Process)
- BCP 25 (Working Group processes)
- BCP 25 (Anti-Harassment Procedures)
- BCP 54 (Code of Conduct)
- BCP 78 (Copyright)
- BCP 79 (Patents, Participation)
- https://www.ietf.org/privacy-policy/(Privacy Policy)

About this meeting

- The meeting will be using WebEx. To join the meeting: https://ietf.webex.com/ietf/j.php?MTID=m730d59fbafc6d18db1f05b3d9b759a18
  - Meeting number (access code): 642 394 286
  - Meeting password: 3idGtEPsQ27
- Agenda: https://datatracker.ietf.org/meeting/107/agenda
- Etherpad: https://etherpad.ietf.org:9009/p/notes-ietf-107-webtrans?useMonospaceFont=true
- Jabber Room: webtrans@jabber.ietf.org
- Secretariat: mtd@jabber.ietf.org
- WG Chairs: Bernard Aboba & David Schinazi

A Public Service Message

Please mute your microphones and turn off your cameras. Don’t startle the iguanas.

'Falling iguana' alert issued in Florida due to cold temperatures

Agenda

- 15:10 – 15:30 Preliminaries, Chairs (20 minutes)
  - Note Well, Virtual Bluesheets
  - Jabber Scribe (Lucas Pardue), Etherpad Note Takers ()
  - Speaking Queue Manager (David Schinazi)
  - Agenda Bash
  - W3C WebTransport Update
- 15:30 - 15:50 WebTransport Overview and Requirements, Victor Vasiliev (20 minutes)
  - https://tools.ietf.org/html/draft-vvv-webtransport-overview
- 15:50 - 16:05 WebTransport using HTTP/2, Eric Kinnear (15 minutes)
  - https://tools.ietf.org/html/draft-kinnear-webtransport-http2
- 16:05 - 16:20 WebTransport over QUIC, Victor Vasiliev (15 minutes)
  - https://tools.ietf.org/html/draft-vvv-webtransport-quic
- 16:20 - 16:35 WebTransport over HTTP/3, Victor Vasiliev (15 minutes)
  - https://tools.ietf.org/html/draft-vvv-webtransport-http3
- 17:00 - 17:10 Wrap up and Summary, Chairs & ADs (10 minutes)

W3C WebTransport Update

https://github.com/w3c/webtransport-charter

Dominique Hazael-Massieux (dom@w3.org)

WebTransport Overview and Requirements (20 minutes)

Presentation End: 15:50

Victor Vasiliev

https://tools.ietf.org/html/draft-vvv-webtransport-overview

Goal of this document

“To assist in the coordination with owners of the WebTransport API, the group will initially develop an overview document containing use cases and requirements in order to clarify the goals of the effort. The requirements will include those arising from the WebTransport API.”

(from the charter)

Goal of this presentation

- Overview of what the draft covers
- Overview of open issues

WebTransport Overview

The WebTransport API is a web interface for client-server communication over unidirectional streams, bidirectional streams, and datagrams.

The WebTransport Protocol provides mappings of that API over popular transport and application protocols, such as QUIC and HTTP.

Target applications

Anything that wants one of the following:

- “WebSockets for UDP”
- “WebSockets without head-of-line blocking”

We’ve reached out to a wide range of web developers, and there is plenty of interest in this in following domains:

- Web games
- Live streaming
- Cloud gaming
- Remote desktop
- Web chat

Bidirectional Communication on the Web (proposed)

	Client-Server	**Peer-to-peer**
Reliable and ordered	WebSocket (also WebTransport!)	RTCDataChannel
Reliable but unordered	WebTransport	
Unreliable and unordered		

Requirements

- MUST use TLS or equivalent
- MUST maintain consent to send continuously
- MUST have congestion control
- MUST ensure that the server is aware the client is a WebTransport client
- MUST let server filter connections by origin (CORS)
- MUST let server be represented as a URI (for CSP, etc)
- MUST let same client and server have multiple sessions at the same time

Common features

- Streams
  - Arbitrary sized
  - Reliable
  - Independent (when possible)
  - Cancellable (when possible)
- Datagrams
  - MTU-sized
  - Unreliable (when possible)

Streams

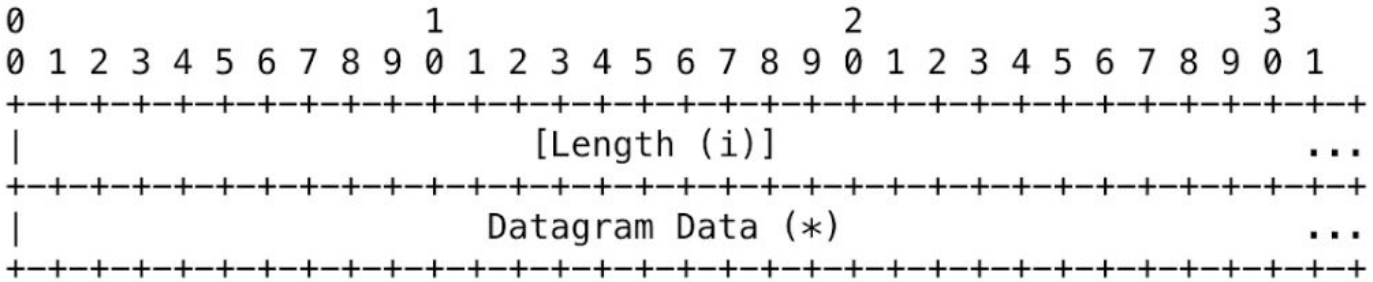
- Both bidirectional and unidirectional
- Initiated by either client or server
- How does the state machine look like?
- How do we reset bidirectional streams?
  - TCP/HTTP2 way: reset closes both halves.
  - QUIC way: reset closes one half.
- Should we expose stream IDs as an API surface?

QUIC Datagrams

https://tools.ietf.org/html/draft-ietf-quic-datagram

DATAGRAM frame (`0x30` and `0x31`)

Length field is optional, determined by least significant bit



Negotiated via `max_datagram_frame_size` transport parameter

Terminology

Lots of open issues here.

- We tend to refer to things as “a transport”, but this is ambiguous
- “Transport session” is an instance of a WebTransport
- “Transport protocol” is also ambiguous (does it refer to TCP/QUIC, or QuicTransport/HttpTransport?)
- “Stream” vs “Message”
  - Are they the same thing?
  - Which one do we use?

Priorities

“A QUIC implementation SHOULD provide ways in which an application can indicate the relative priority of streams”.

Options:

- Deem this out of scope as a pure API issue.
- Define something
- Wait for HTTP/3 priorities and use them as-is.

Discussion

WG Adoption of
draft-vvv-webtransport-overview?

Other Questions?

WebTransport using HTTP/2 (15 minutes)

Presentation End: 16:05

Eric Kinnear

https://tools.ietf.org/html/draft-kinnear-webtransport-http2

WebTransport using HTTP/2

draft-kinnear-webtransport-http2

Alan Frindell, Eric Kinnear, Tommy Pauly,
Victor Vasiliev, Guowu Xie

WEBTRANS

IETF 107, March 2020, Virtual

Since IETF 106

Merging of HTTP transport documents

- draft-xie-bidirectional-messaging-02
- draft-kinnear-httpbis-http2-transport
- draft-vvv-webtransport-http3

Shared architecture, mapped onto HTTP/2 and HTTP/3

Why HTTP/2?

HTTP/2 provides framing layer with many desired transport features

- Configuration exchange
- Multiplexed streams
- Shared congestion control and loss recovery state
- Flow control
- Stream relationships and priorities
- Traverses the internet

Does not have unreliable or unordered data

Shared Concepts

Mapping of WebTransport over HTTP/2

- Bidirectional communication over HTTP using TLS/TCP

Negotiation and session establishment phase

Data exchange phase

Negotiation

New SETTINGS value: SETTINGS_ENABLE_WEBTRANSPORT

Extended CONNECT allows connecting to server itself

Combine SETTINGS_ENABLE_CONNECT_PROTOCOL and SETTINGS_ENABLE_WEBTRANSPORT

New token, “`webtransport`”, for use with the `:protocol` pseudo-header

Session Establishment

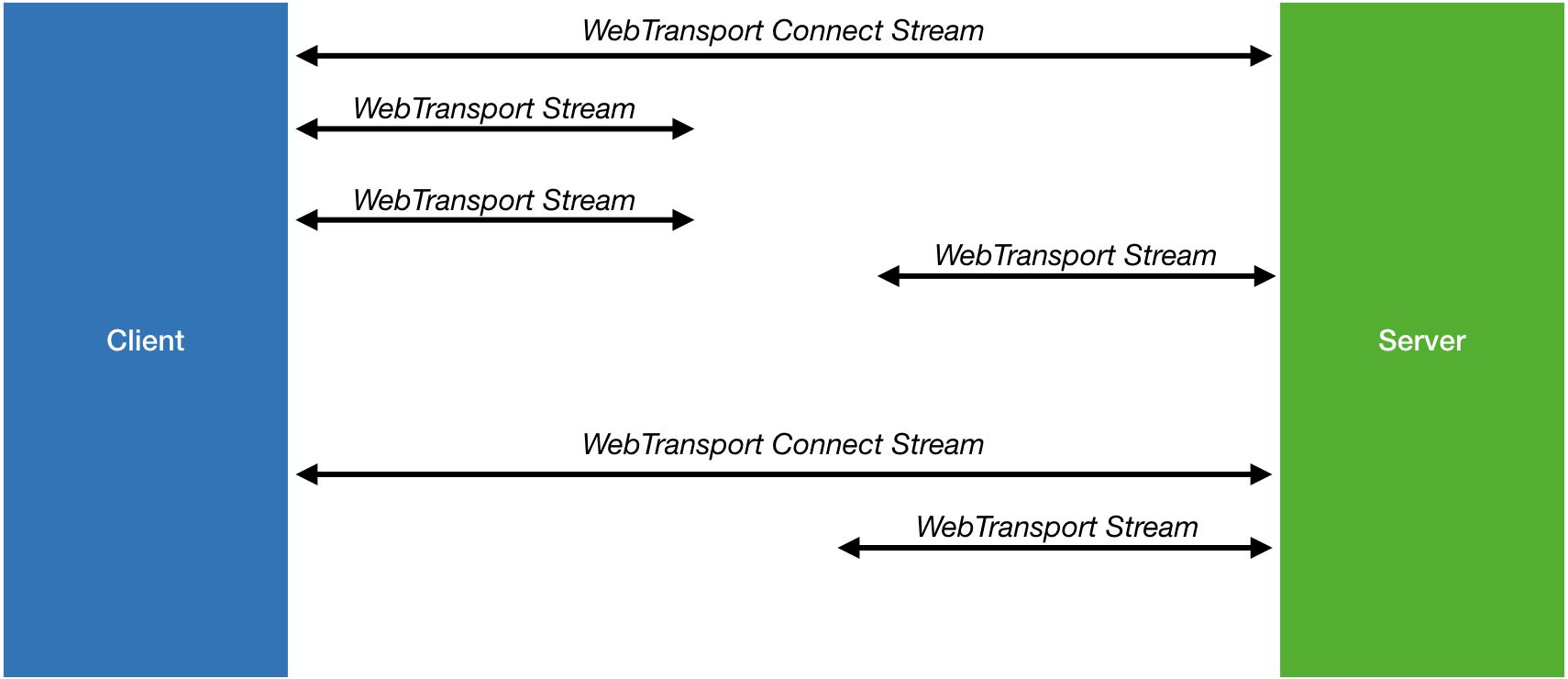
Http2Transport servers are defined by pair of authority value and path value

WebTransport Connect Stream established by using the extended CONNECT method with `:protocol` token “`webtransport`”

Connect Stream used only for WebTransport session, no data exchange

- WebTransport Stream lifetime tied to that of the session

WebTransport Stream Types



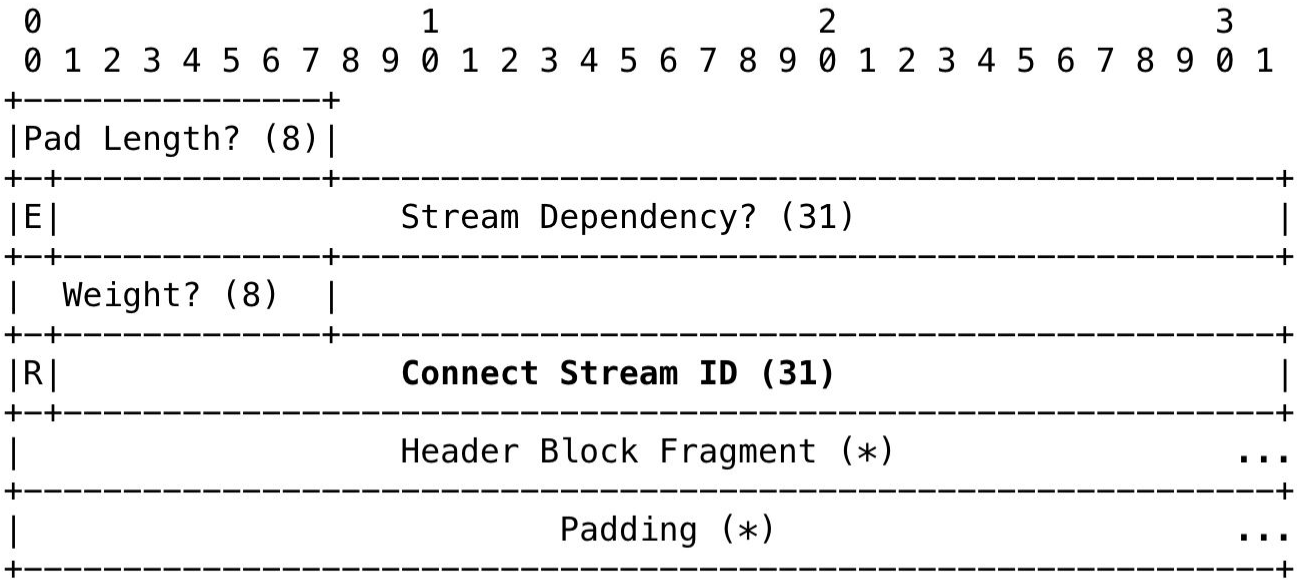
Data Exchange

WebTransport Streams established using new frame, WTHEADERS

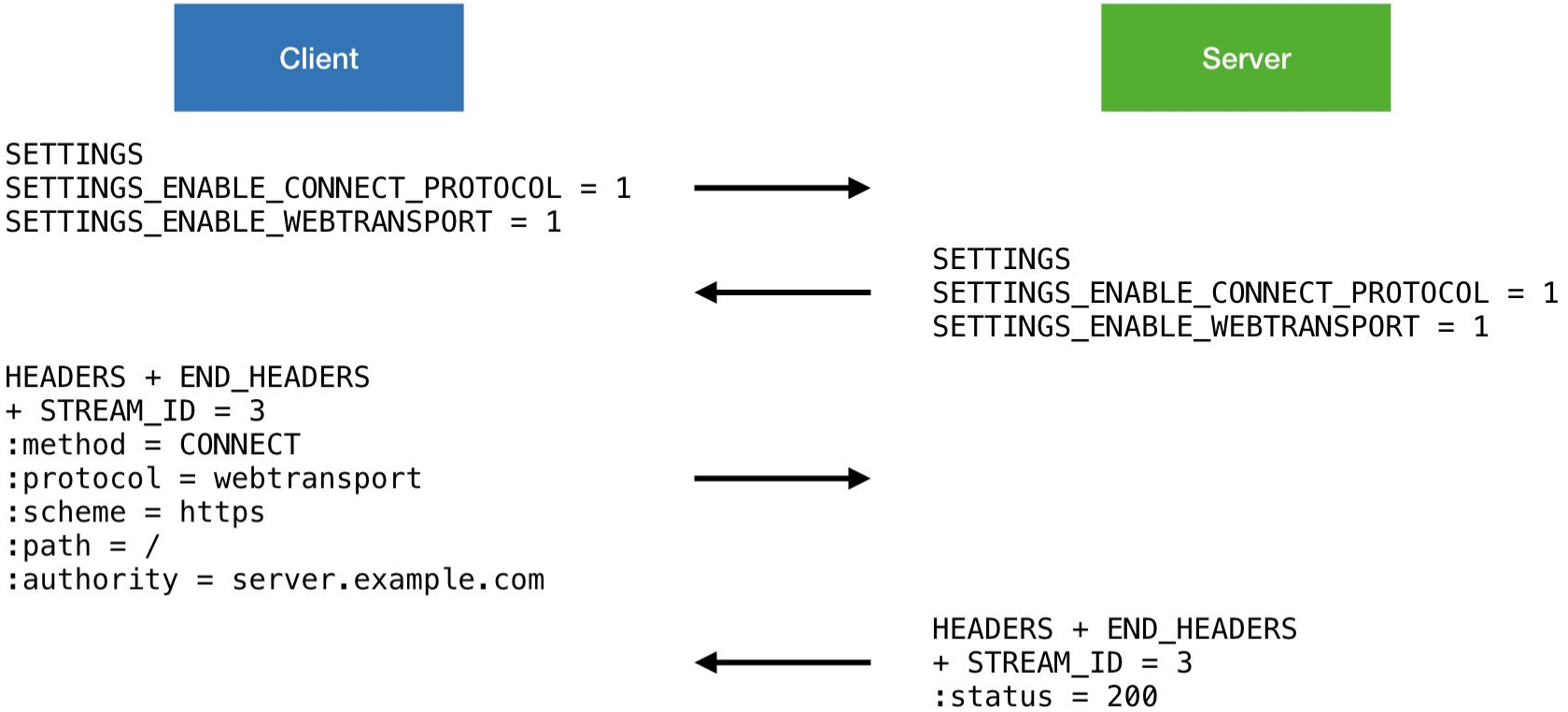
Same as a regular HEADERS frame, but bidirectional and references the WebTransport Connect Stream for its session

Corresponding Connect Stream provides forwarding information to intermediaries

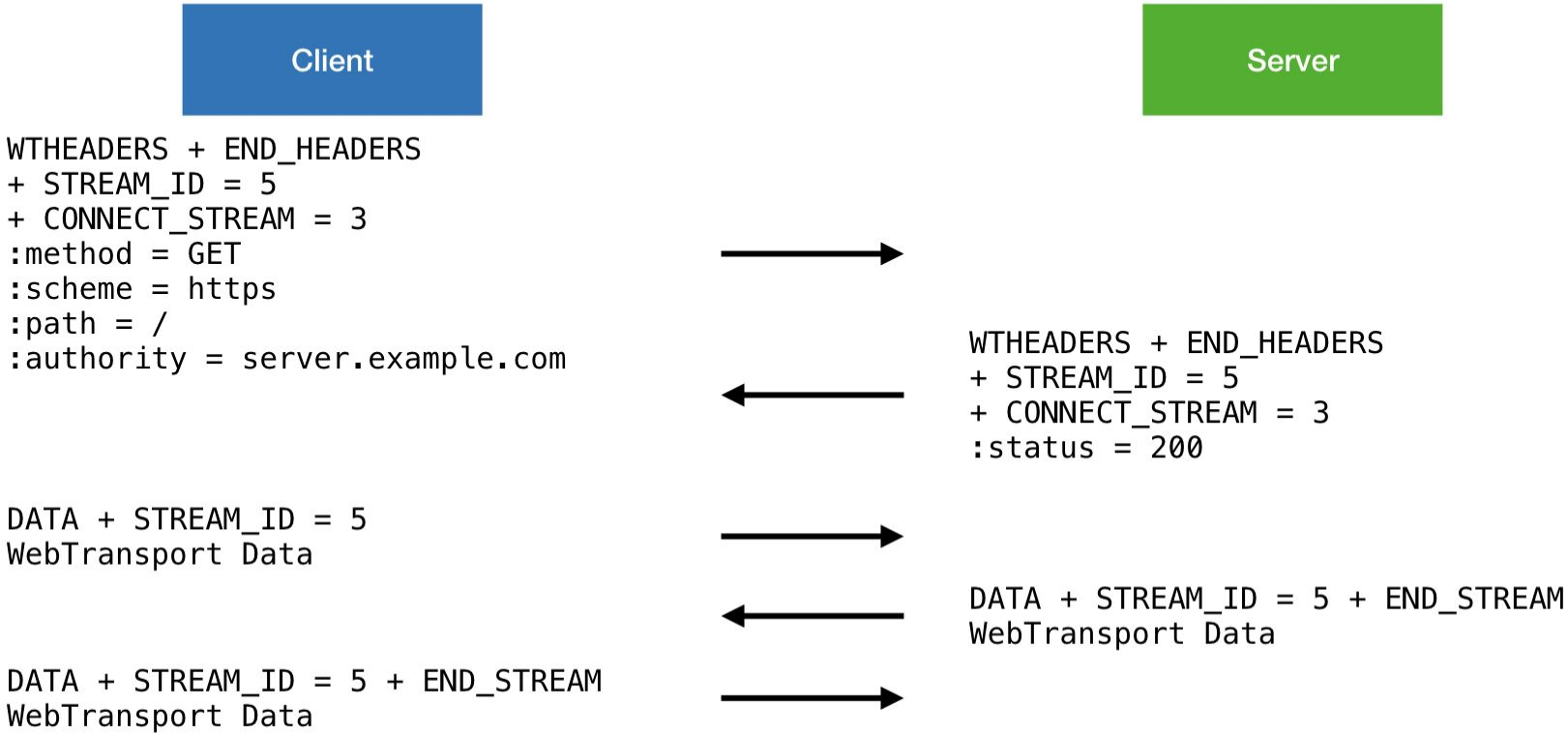
WTHEADERS Frame



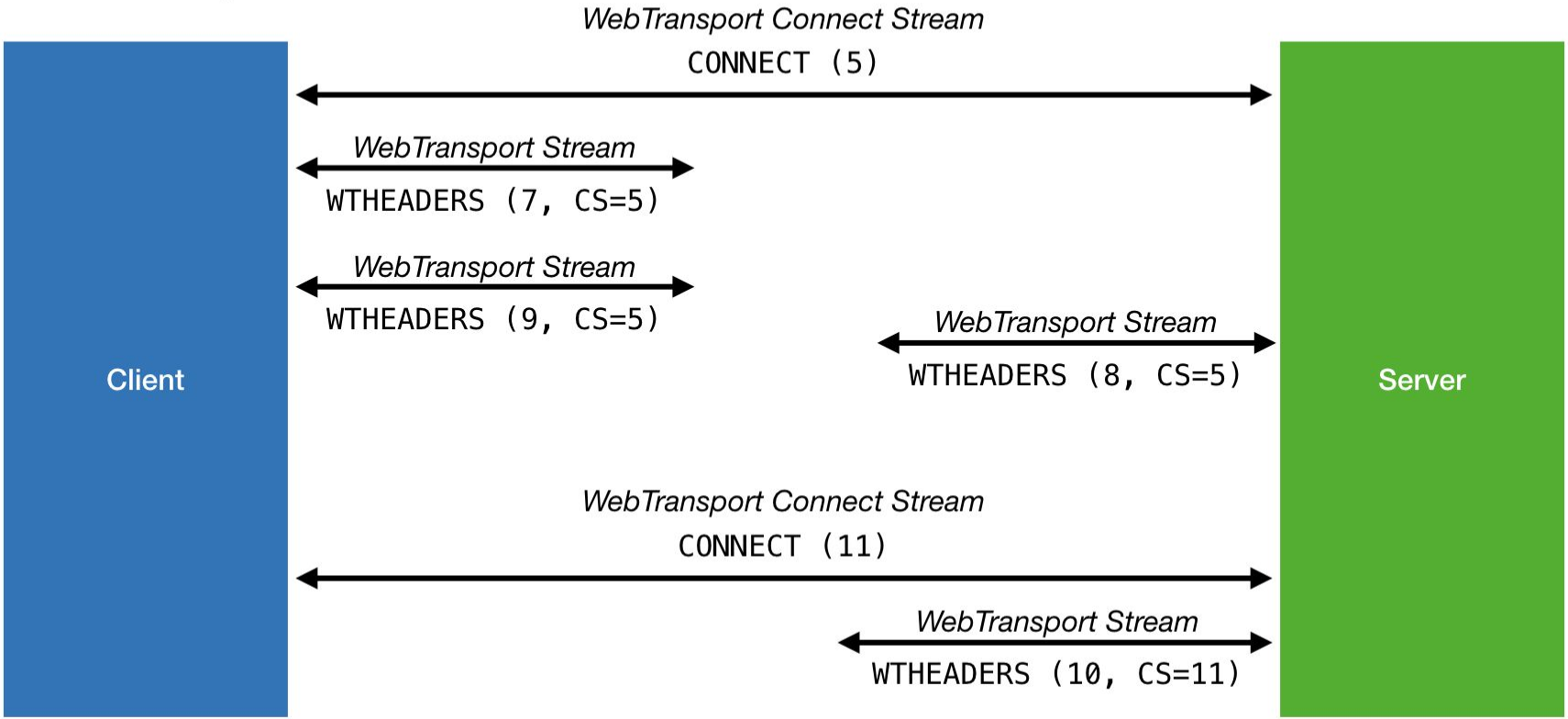
Example


Client
Server
SETTINGS
SETTINGS_ENABLE_CONNECT_PROTOCOL = 1
SETTINGS_ENABLE_WEBTRANSPORT = 1
SETTINGS
SETTINGS_ENABLE_CONNECT_PROTOCOL = 1
SETTINGS_ENABLE_WEBTRANSPORT = 1
HEADERS + END_HEADERS
+ STREAM_ID = 3
:method = CONNECT
:protocol = webtransport
:scheme = https
:path = /
:authority = server.example.com
HEADERS + END_HEADERS
+ STREAM_ID = 3
:status = 200

Example


Client
Server
WTHEADERS + END_HEADERS
+ STREAM_ID = 5
+ CONNECT_STREAM = 3
:method = GET
:scheme = https
:path = /
:authority = server.example.com
WTHEADERS + END_HEADERS
+ STREAM_ID = 5
+ CONNECT_STREAM = 3
:status = 200
DATA + STREAM_ID = 5
WebTransport Data
DATA + STREAM_ID = 5 + END_STREAM
WebTransport Data
DATA + STREAM_ID = 5 + END_STREAM
WebTransport Data

Example


WebTransport Connect Stream
CONNECT (5)
WebTransport Stream
WTHEADERS (7, CS=5)
WebTransport Stream
WTHEADERS (9, CS=5)
WebTransport Stream
WTHEADERS (8, CS=5)
Client
Server
WebTransport Connect Stream
CONNECT (11)
WebTransport Stream
WTHEADERS (10, CS=11)

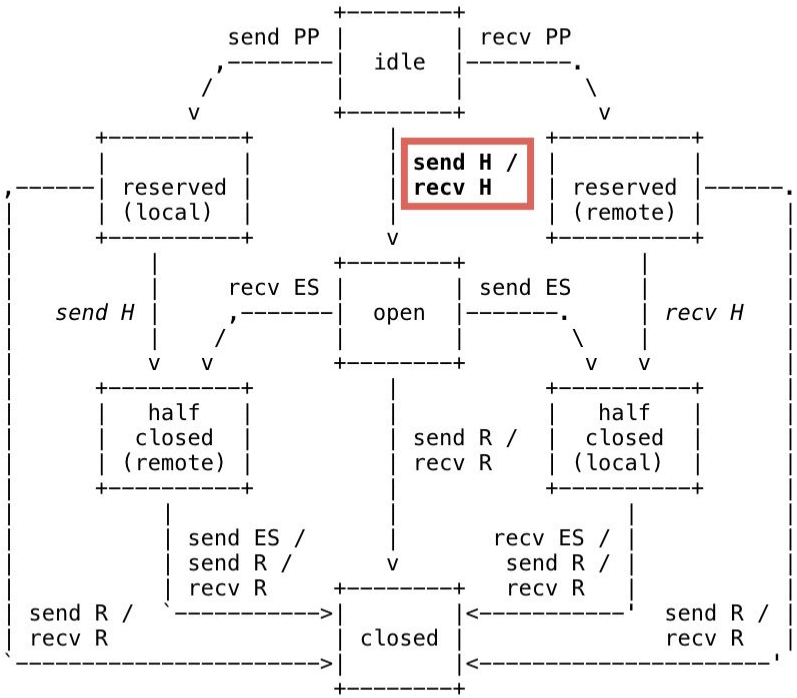
Lifecycle

Stream states remain the same

Client initiates Connect Streams

Either endpoint initiates WebTransport Streams

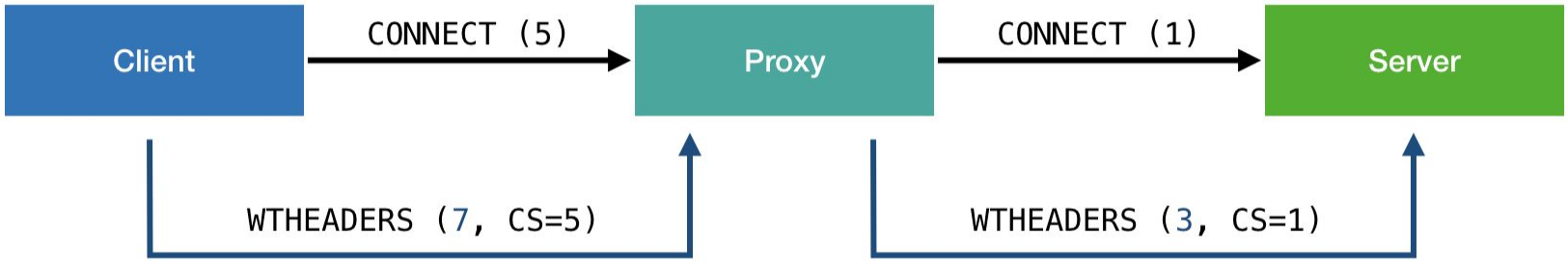
If the Connect Stream closes, all WebTransport Streams associated with that Connect Stream close



Intermediaries

WebTransport Streams are routed on the same connection as their corresponding WebTransport Connect Streams by any segment which has negotiated the use of WebTransport

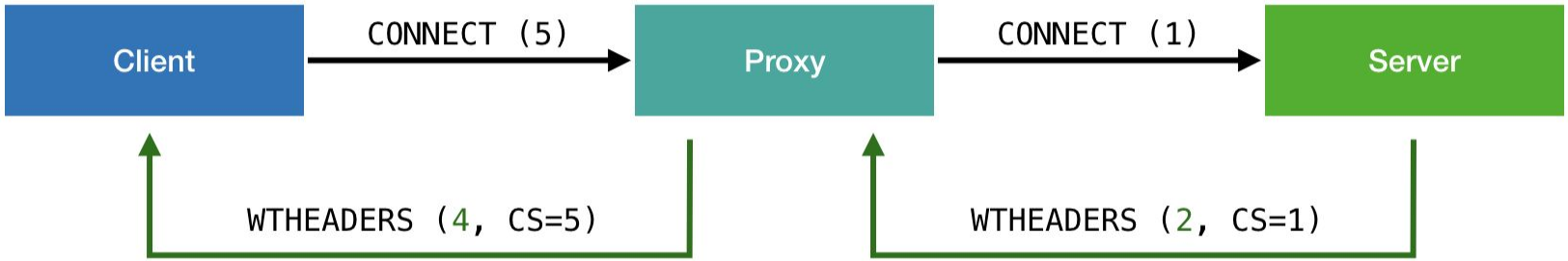
Example: *Client initiates stream through proxy*



Intermediaries

WebTransport Streams are routed on the same connection as their corresponding WebTransport Connect Streams by any segment which has negotiated the use of WebTransport

Example: *Server initiates stream through proxy*



Summary

HTTP/2 extension, negotiated with new SETTINGS value

WebTransport sessions are established with extended CONNECT

WebTransport Streams carry data, reference WebTransport Connect Streams

Either endpoint can establish WebTransport Streams

Intermediaries use Connect Streams to route all associated streams

Questions?

WebTransport over HTTP/3
WebTransport over QUIC
(30 minutes)

Presentation End: 16:35

Victor Vasiliev

https://tools.ietf.org/html/draft-vvv-webtransport-http3

Http3Transport

...is like Http2Transport, but over HTTP/3!

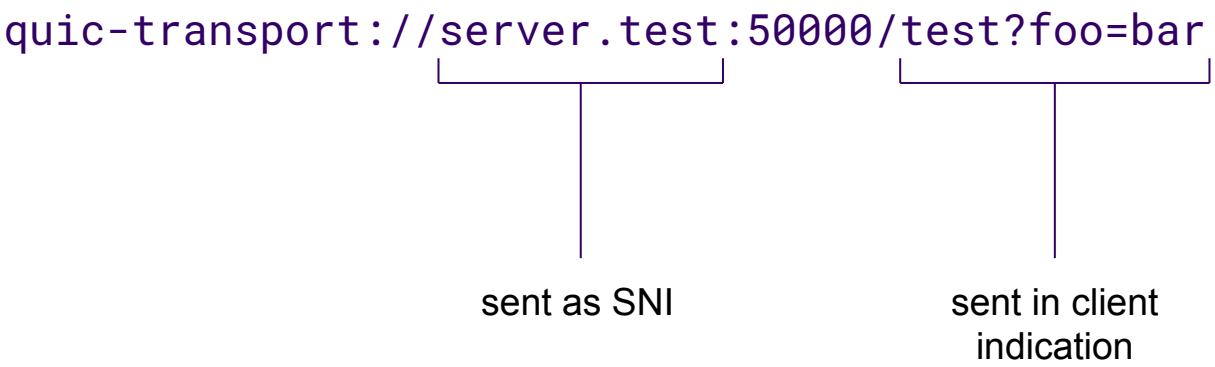
- Datagram support using draft-schinazi-quic-h3-datagram-03
- Draft is currently in process of being converged towards design choices outlined in draft-kinnear-webtransport-http2-00:
  - SETTINGS-based negotation
  - Using stream IDs to associate WebTransport streams with a Connect stream
  - WebTransport streams can have optional headers and trailers

QuicTransport

Minimal protocol on top of QUIC

- ALPN value (“wq”)
- URI scheme
- Client indication (special stream with metadata)
  - Contains origin of the initiating webpage
  - Contains the path from the URI
- One dedicated QUIC connection per transport session

QuicTransport URI scheme



The Great Transport Zoo

Transports proposed so far

- QuicTransport
  A QUIC connection with minimal additions required to make it work with Web security model.
- Http2Transport
  Virtual multiplexed transport inside an HTTP/2 connection.
- Http3Transport
  Virtual multiplexed transport inside an HTTP/3 connection.
- FallbackTransport (no draft currently)
  Simulation of multiplexed streams on top of WebSocket protocol

Which ones do we actually need?

Overview of proposed transports

	Dedicated	**Pooled**
QUIC-based	QuicTransport	Http3Transport
TCP-based (fallback)	FallbackTransport	Http2Transport

Axis: underlying transport protocol

- QuicTransport and Http3Transport use QUIC
- Http2Transport and FallbackTransport use TCP

We need at least one for QUIC and one for TCP, as QUIC may be blocked on some networks.

Axis: dedicated vs shared

- QuicTransport has a dedicated connection per transport session.
- Http2Transport and Http3Transport multiplex all sessions into a single connection when possible.

As a general principle, we want to multiplex connections as often as possible.

Reasons to want a dedicated connection

- Dedicated congestion control context
  - Congestion control data can be used to provide bandwidth estimate
  - Useful for media and other situation where the rate of application data can be adjusted based on network conditions
- Connection-level statistics (packet loss, etc)
- Connection-level TLS configuration
  - Client cert authentication
  - Custom server certificates

Some other considerations

- HTTP-based transports have advantage of easy integration with existing HTTP infrastructure (load balancers, CDNs, web frameworks)
- HTTP solves a lot of practical problems (e.g. redirects) that QuicTransport chose not to solve
- QuicTransport allows existing protocols on top of QUIC to be adapted into a Web-usable version “for free”

Some options

1. Adopt both QuicTransport and Http{2,3}Transport
2. Adopt Http{2,3}Transport and consider making a dedicated version of that
   (note: this is not as easy as it sounds)
3. Adopt Http{2,3}Transport and punt dedicated version until we're certain it is needed

Questions?

Wrapup and Summary (10 minutes)

Session End: 17:10

Bernard Aboba

David Schinazi

Thank you

Special thanks to:

The Secretariat, WG Participants & Chairs

The iguanas

WEBTRANS WG
IETF 107

Virtual Meeting

Friday, March 30, 2020

22:10 - 00:10 UTC

15:10 - 17:10 Pacific Time

Mailing list: webtransport@ietf.org
Jabber Room: webtrans@jabber.ietf.org