WEBTRANS WG
IETF 108

Virtual Meeting

Monday, July 27, 2020

14:10 - 15:50 UTC

7:10 - 8:50 AM Pacific Time

Virtual Room 1

Mailing list: webtransport@ietf.org
Jabber Room: webtrans@jabber.ietf.org
MeetEcho link: http://www.meetecho.com/ietf108/webtrans/

IETF 108 Meeting Tips

https://www.ietf.org/how/meetings/108
https://datatracker.ietf.org/meeting/agenda
https://www.ietf.org/media/documents/Documentation-Meetecho-IETF.pdf

This session is being recorded

- **IETF 108 registration and a datatracker account is required to join the meeting.**
- **Your name will be automatically added to the attendee list based on your datatracker login.**
- **Join the session Jabber room via IETF Datatracker Meeting agenda**
- **Please use headphones when speaking.**
- **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

- Agenda: https://datatracker.ietf.org/doc/agenda-108-webtrans/
- CodiMD (for notes): https://codimd.ietf.org/notes-ietf-108-webtrans
- Jabber Room: webtrans@jabber.ietf.org
- Secretariat: mtd@jabber.ietf.org
- WG Chairs: Bernard Aboba & David Schinazi
- Jabber Scribe:
- Note takers:

Agenda

- 14:10 – 14:20 Preliminaries, Chairs (10 minutes)
    - Note Well, Virtual Bluesheets
    - Jabber Scribe, Etherpad Note Takers
    - Speaking Queue Manager (David Schinazi)
    - Agenda Bash
    - W3C WebTransport Update
- 14:20 - 14:30 WebTransport Use Cases, Will Law (10 minutes)
- 14:30 - 14:50 WebTransport Overview and Requirements, Victor Vasiliev (20 minutes)
    - https://tools.ietf.org/html/draft-vvv-webtransport-overview
- 14:50 - 15:05 WebTransport using HTTP/2, Eric Kinnear (15 minutes)
    - https://tools.ietf.org/html/draft-kinnear-webtransport-http2
- 15:05 - 15:20 WebTransport over QUIC, Victor Vasiliev (15 minutes)
    - https://tools.ietf.org/html/draft-vvv-webtransport-quic
- 15:20 - 15:35 WebTransport over HTTP/3, Victor Vasiliev (15 minutes)
    - https://tools.ietf.org/html/draft-vvv-webtransport-http3
- 15:35 - 15:50 Wrap up and Summary, Chairs & ADs (10 minutes)

W3C WebTransport Update

- The WorkGroup Charter has been published at https://w3c.github.io/webtransport-charter/charter.html
- Charter process is underway and voting is expected to complete today (July 27), with next steps in WG creation in August
- Timeline
    - September 2020: First teleconference
    - January 2021: FPWD for WebTransport
    - Q2 2022: CR for WebTransport
    - Q? 2023: Expected completion
- Use-case document being developed, to drive both W3C API and guide IETF development.
- Two Co-Chairs have been nominated in anticipation of WG establishment

Jan-Ivar Bruaroey
Mozilla

Will Law
Akamai

WebTransport Use Cases (10 minutes)

Presentation End: 14:30

Will Law

Many cited use cases over the past two years

- https://github.com/w3c/webtransport-charter/issues/10#issuecomment-642200384]
- https://github.com/w3c/media-and-entertainment/issues/25
- https://docs.google.com/presentation/d/1njF8fMo9WJ5G-U4VYPvI2XBtCqEFQm1I5Gk1b00V6_o/edit#slide=id.g78f93d0334_0_20
- https://docs.google.com/presentation/d/1cr_4jtUCYTkoJQl4jEWT9LKJfZkkYM1W3MW8s2bz47Y
- https://web.dev/quictransport/
- https://docs.google.com/presentation/d/1njF8fMo9WJ5G-U4VYPvI2XBtCqEFQm1I5Gk1b00V6_o
- https://github.com/w3c/webrtc-nv-use-cases/issues/50
- https://github.com/w3c/webrtc-nv-use-cases/issues/51
- https://docs.google.com/presentation/d/1rHXd5vxv_ME5r2TgpLPhJlD2iq0lcxQCsYbuT6fg0dc/edit?ts=5ef67d9f#slide=id.g8b79acbdb7_10_116

Collected these for organisation in to a temporary home at https://docs.google.com/document/d/1ns2NhdqRQsARLy54ilkVXAlnm0y6mcaPj6sEphuzIsM/edit?usp=sharing

This will be moved via PR to https://github.com/WICG/web-transport **for curation and debate.**

Use-cases

(Ordinality of listing does not imply priority)

- Machine learning - data IO to cloud ML processing
  - Speech translation/emotion analysis - sending audio/video data from client to server for analysis and receiving translated data/text/audio in return.
  - Security camera analysis - data and/or video sent to cloud service for analysis. Service may return data instructions.
  - QuicTransport use case

- Multiplayer Gaming - web and consoles
  - Game play instructions sent from client to cloud based game engine. Some instructions are time sensitive (such as location data) , others are stateful (avatar selection). Dataflow is bi-directional.
  - Mixture of client-server and p2p data flows.
  - AR gaming requires real-world interaction, including virtual theatre - geo-separate actors with virtual backgrounds.
  - QuicTransport use case

Use-cases
(Continued)

- Low-latency live streaming
  - Unidirectional Broadcast - one to many millions - sports events, news, wagering, latency < 1000ms to preempt social media and quality to support UHD, HDR, HFR, DRM.
  - Bi-directional few-to-few video chats via server, reduced connection time/complexity compared to WebRTC. Example - Apple Facetime™
  - QuicTransport or Htt3Transport?

- Cloud Game Streaming
  - Server-side game rendering (such as Google Stadia™) transmitted to thin client with low latency (<60ms)
  - Bi-directional Game play instructions (both server and p2p).
  - QuicTransport use case.

Use-cases
(Continued)

- Server-based video conferencing
  - Simpler session establishment
  - Censorship circumvention - preventing fingerprinting and identification during session establishment.
  - QuicTransport or Htt3Transport?

- Remote desktop
  - Transmission of screen capture/sharing and control instructions.
  - Collaborative work on a shared screen.
  - Including scaling to very large audiences.
  - Online document sharing - synched mouse location + edit state
  - Remote assistance temporarily "taking over" control of a system
  - Client/server or P2P
  - QuicTransport Use Case

Use-cases (Continued)

- Time Synchronized Multimedia Web communications
  - Combining geo-separate singing and/or instruments together online with precise time synchronization.
  - QuicTransport or Htt3Transport?

- IOT sensor and analytics data transfer
  - Efficient and intermittent transmission of data. For example - sending a 1 bit flag, GPS position updates, mouse clicks on site etc.
  - Sensor data upload - including filters, aggregation, triggers.
  - QuicTransport Use Case

- PubSub Models - avoid long-polling
  - Social feeds - Twitter™ , financial tickers etc.
  - Messaging platforms, including Enterprise messaging infrastructure
  - Http3Transport Use Case

Use-Case issues

- Which of these use-cases
  - can be solved sufficiently well using existing technologies?
  - can be solved by extending existing technologies (websocket, WebRTC)?
  - warrant the development of a new technology?
  - are best handled via QuicTransport? Http3Transport?
- Who curates goals and non-goals between IETF and W3C?
- Encourage WebTransport to do a few things really well

We look forward to fruitful collaboration between IETF and W3C WG on WebTransport development.

WebTransport Overview and Requirements (20 minutes)

Presentation End: 14:50

Victor Vasiliev

https://tools.ietf.org/html/draft-ietf-webtrans-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)

Updates since last IETF

- Draft adopted
- Can now file issues at <https://github.com/ietf-wg-webtrans/draft-ietf-webtrans-overview/issues>

Issue #1: stream IDs

Need a consistent model for all transports.

Current text:

“Every stream within a transport has a unique 64-bit number identifying it. Both unidirectional and bidirectional streams share the number space. The client and the server have to agree on the numbering, so it can be referenced in the application payload. WebTransport does not impose any other specific restrictions on the structure of stream IDs, and they should be treated as opaque 64-bit blobs.”

Issue #1: stream IDs

Why are stream IDs hard?

- Proxying
  - Have to preserve stream IDs in case new streams are opened out of order
  - In multiplexed transports (H2/H3) 1:1 correspondence is impossible
- Information disclosure
  - When multiple origins accessed over same connection, HTTP-level stream IDs reveal their state

Issue #1: stream IDs

What is the use case for stream IDs?

Developers who asked for it care mostly about knowing the ordering between streams, rather than using them as on-the-wire reference.

Issue #2: stream resets

- In HTTP/2, resetting a stream resets both halves
- In QUIC, resetting a stream causes the write half being closed, STOP_SENDING causes read half being closed
- Options:
    - Require WebTransport over QUIC/H3 automatically close other half
    - Port STOP_SENDING to HTTP/2

Issue #3: streams, messages

WebTransport uses streams of bytes as a primitive, since that’s what QUIC and HTTP/{2,3} use.

Problem: WebSocket/RTCDataChannel use streams of messages.

Do we want to provide that as an additional primitive?

Other TODOs in the draft

- Currently missing sections:
  - Explicit state machine description
    - Depending on resolution of #2, this would look either like QUIC or like HTTP/2
  - Missing section on priorities

Discussion

WebTransport using HTTP/2 (15 minutes)

Presentation End: 15:05

Eric Kinnear

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

Http2Transport

draft-kinnear-webtransport-http2-01

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

WEBTRANS

IETF 108, July 2020, Virtual

Since IETF 107

Some great feedback, thank you!

Moved many of these to GitHub issues

Updated to -01

Concepts

Http2Transport provides bidirectional streams over HTTP/2

- Either endpoint can initiate a new stream
- QUIC can do this today

This is over HTTP and can traverse intermediaries

- But only ones that support WebTransport

Coexistence with existing HTTP traffic

Concepts

HTTP/2 is missing a few things from HTTP/3

- Unidirectional streams
- Datagrams
  - Especially unreliability

Aside

Eventually, we need to decide if we want to specify a QUIC equivalent over TCP without HTTP/2

Put another way, do we need “QUIC, the multistreaming transport without HTTP/3” to have an equivalent over TCP?

#3 New streams without additional roundtrips

> Each new stream would require a WebSocket handshake to agree on application protocol used, meaning that it would take at least one RTT to establish each new stream before the client can write to it.

HTTP/2 can guarantee in-order delivery of data, do we need to be able to open a new Connect Stream (routing stream) and then Stream?

HTTP/2 may not be present across every hop

Can we provide an equivalent for each mapping?

#5 Unidirectional streams

> All transport protocols MUST provide datagrams, unidirectional and bidirectional streams in order to make the transport protocols easily interchangeable.

Do we want to use half-closed (local | remote) streams?

Or simply bar endpoints from sending data in one direction for a unidirectional stream

Establishing different types of streams requires metadata, does this impact our choices for HTTP/3 mapping vs. QUIC equivalent?

#6 Datagrams

> The WebTransport sender is not expected to retransmit datagrams, though it may if it is using a TCP-based protocol or some other underlying protocol that requires reliable delivery.

Applications need to know what they requested vs. what they got

Dedicated datagram stream?

New frame?

WTHEADERS per datagram?

#6 Datagrams

> WebTransport datagrams are not expected to be flow controlled, meaning that the receiver might drop datagrams if the application is not consuming them fast enough.

New frame makes this somewhat easier to reason about

Interactions with other streams, ability to fully consume every flight

Reliable link may drop some data

Would we use this elsewhere over HTTP/2?

A brief note about stream IDs

So far, have avoided exposing anything about the underlying HTTP/2 stream IDs to the client

Each WebTransport Session has a stream ID space

From earlier: should we expose stream IDs as an API surface?

Consistency is key

Questions?

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

Presentation End: 15:35

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-http2-01:
  - SETTINGS-based negotiation
  - 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

$$\texttt{quic-transport://}\underbrace{\texttt{server.test}}_{\text{sent as SNI}}\texttt{:50000/}\underbrace{\texttt{test?foo=bar}}_{\text{sent in client indication}}$$

QuicTransport origin trial

Available in Chrome 84-86!

https://web.dev/quictransport/

Implements QUIC draft-27 (draft-29 starting Chrome 85).

The Great Transport Zoo
Season 2

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

QuicTransport vs Http3Transport

QuicTransport:

- Dedicated connection
  - Transport fine-tuning on server
  - More stats exposed to client
- No HTTP/3 dependency
- Target applications:
  - Machine learning
  - Video games on the Web
  - Live streaming
  - Real-time media
  - IoT

Http3Transport:

- Pooled with other HTTP/3 traffic
- HTTP features:
  - Use HTTP load balancing and routing
  - Headers for metadata
- Target applications:
  - General web applications
  - Web chats
  - Push notifications

Advantages of HTTP transports

- Multiplexing support
  - Support for multiplexing Http3Transport and HTTP/3
  - Reduces number of QUIC/TCP sockets in use
  - Lower startup cost for new transports
  - Traffic appears identical to HTTP to the network

  Note that multiplexing being supported does not automatically imply it being required
    - When dedicated connection is beneficial, this could be controlled at the API layer

Advantages of HTTP transports

- Shared metadata format
  - Can reuse HTTP headers and status codes
  - Standard headers that are useful:
    - Origin
    - Location
    - Forwarded
    - :path/:authority/:scheme
  - Custom headers can be reused as-is
  - Counterpoint: similarity of HTTP can lead to wrong expectations

Disadvantages of HTTP

- Implementation complexity
  - Most comes from HPACK/QPACK
  - Can be solved by making header compression support negotiable
  - Multiplexing is harder to implement in browsers
- Design complexity
  - Have to define interaction with existing HTTP mechanisms
  - Pooling and flow control can lead to DoS
  - Things like stats are easier to define with dedicated connections

Implementation experience

- QuicTransport
  - Implemented in Chrome
  - Various server implementations
  - Easy to implement on top of an existing QUIC library (100-200 lines for Python)
- HttpTransport
  - Variations implemented at Facebook and Apple
  - No in-browser support for the clients currently

Use cases

- Both options satisfy the WebTransport requirements, notably unreliable datagrams and streams without head-of-line blocking
  - Those two properties are key for satisfying the enumerated use cases
- Other aspects may make individual transports a better fit for specific use cases:
  - HTTP-based options are more attractive to the operators of large server setups
  - Raw QUIC-based option is more attractive to people who would want to implement this from scratch (e.g. game developers)

Beyond wire protocol

- What URL scheme would the resulting resources be represented by?
  - Determines whether WebTransport and HTTP are same-origin
- Handshake-level concerns
  - Do we send cookies?
  - HTTP auth, TLS client certs, etc
  - Alt-Svc and socket pool integration

Next steps

- The current discussion is between two options: “only QUIC” and “only HTTP”
- Need more input from wider audience of Web developers
- Current plan: continue discussion on the mailing list
- Potential focus of an interim meeting?

Discussion

Wrapup and Summary (15 minutes)

Session End: 15:50

Bernard Aboba
David Schinazi

Thank you

Special thanks to:

The Secretariat, WG Participants & Chairs

WEBTRANS WG
IETF 108

Virtual Meeting

Monday, July 27, 2020

14:10 - 15:50 UTC

7:10 - 8:50 AM Pacific Time

Virtual Room 1

Mailing list: webtransport@ietf.org
Jabber Room: webtrans@jabber.ietf.org
MeetEcho link: http://www.meetecho.com/ietf108/webtrans/