WEBTRANS WG
IETF 111

Virtual Meeting

Friday, July 30, 2021

12:00 - 14:00 Pacific Time

Session I, Room 1

Mailing list: webtransport@ietf.org
Jabber Room: webtrans@jabber.ietf.org
MeetEcho: https://ws.conf.meetecho.com/conference/?group=webtrans

IETF 111 Meeting Tips

https://www.ietf.org/how/meetings/111
https://datatracker.ietf.org/meeting/agenda

This session is being recorded

- **IETF 111 registration and a datatracker login required to attend**
- **No need to manually fill in blue sheets, it's automatic.**
- **Join the session Jabber room via IETF Datatracker Meeting agenda**
- **Please use headphones when speaking to avoid echo.**
- **Please state your full name before speaking.**

IETF 111 Meeting Tips

https://www.ietf.org/how/meetings/111
https://datatracker.ietf.org/meeting/agenda

This session is being recorded

- Enter the queue with , leave with
- When you are called on, you need to enable your audio to be heard.
- Audio is enabled by unmuting and disabled by muting
- Video can also be enabled, but it is separate from audio.
- Video is encouraged to help comprehension but not required.

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-111-webtrans/
- CodiMD (for notes): https://codimd.ietf.org/notes-ietf-111-webtrans
- Jabber Room: webtrans@jabber.ietf.org
- Secretariat: mtd@jabber.ietf.org
- WG Chairs: Bernard Aboba & David Schinazi
- Jabber Scribe:
- Note Takers:

Agenda

- Preliminaries, Chairs (10 minutes)
  - Note Well, Virtual Bluesheets
  - Jabber Scribe, Note Takers
  - Speaking Queue Manager (David Schinazi)
  - Agenda Bash
- W3C WebTransport Update, Will Law, (10 minutes)
- WebTransport over HTTP/3, Victor Vasiliev (40 minutes)
  - https://tools.ietf.org/html/draft-ietf-webtrans-http3
- WebTransport using HTTP/2, Eric Kinnear (40 minutes)
  - https://tools.ietf.org/html/draft-ietf-webtrans-http2
- Hums, Wrap up and Summary, Chairs & ADs (20 minutes)

W3C WebTransport Update

W3C WebTransport WG progress since May 20th

- Decisions & PRs
  - Defined milestone for “minimum viable ship” to support initial browser implementations.
  - Secure context only for API surface.
  - Defined datagram back-pressure control attributes (MaxAge, HighWaterMark)
  - Defined clean-up procedure
  - Removed .state and .onstatechange attributes,
  - Removed custom methods & promises for STOP_SENDING and RESET_STREAM, now errors the stream instead.
  - Progress on a new WebTransportError DOMException w/#316

WebTransport servers & clients

- aioquic -0.9.15+, a python library https://github.com/aiortc/aioquic . The demo server runs a WebTransport echo service at /wt. This server will be used to service web platform tests for W3C.
- Twitch has a closed source server implementation that implements WT over HTTP/3.
- Chrome has signaled a Q4 release for WT over HTTP/3. Status: https://www.chromestatus.com/feature/4854144902889472
  - “Intent to Experiment” (M91 - M95): https://groups.google.com/a/chromium.org/g/blink-dev/c/aaLFxzw5zL4
- Mozilla are working on an implementation in Firefox.

```
let transport = new WebTransport('https://localhost:4433/wt');
await transport.ready;

let stream = await transport.createBidirectionalStream();
let reader = stream.readable.getReader();
let writer = stream.writable.getWriter()

await writer.write(new Uint8Array([65, 66, 67]));
let received = await reader.read();
await transport.close();

console.log('received', received);
```

New streaming protocol proposals

- There are multiple current initiatives being developed for next-gen video built directly on top of QUIC:
  - https://datatracker.ietf.org/doc/html/draft-engelbart-rtp-over-quic
  - https://www.ietf.org/id/draft-kpugin-rush-00.html
  - https://datatracker.ietf.org/doc/html/draft-sharabayko-srt-over-quic
- We encourage these projects to consider rebasing their implementations, if practical, on WebTransport, for the following benefits:
  - Automatic H2 fallback so you do not have to roll your own
  - The same bi/uni-directional streams and datagram support with minimal code changes to your project.
  - W3C browser implementations will give compatibility with billions of clients.

RTP/RTCP over QUIC Local Interface Requirements (from AVTCORE WG presentation by J. Ott)

QUIC Implementation

- Unreliable QUIC DATAGRAM frames MUST be supported
- ACKs/Loss of DATAGRAM frames MUST be signalled to the application
- RTT statistics MUST be exposed to application

Congestion Controller

- Assuming one of the algorithms proposed by RMCAT for real-time media
- Input: ACKed packets, delay, RTT estimations, optionally ECN
- Output: bandwidth estimation for media encoder
- QUIC internal vs. external CC: What about QUIC streams and non-RTP DATAGRAM frames?

W3C Update (Cont'd)

Request from W3C for IETF to resolve (1/2):

- How to prioritize datagrams vs. streamsand streams versus streams - is this the responsibility of the user-agent implementation or will the core protocol provide prioritization features?
  - Stream prioritization #33 - https://github.com/w3c/webtransport/issues/33
  - Datagram tokens as a mechanism of datagram vs stream prioritization #62 - https://github.com/w3c/webtransport/issues/62

W3C Update (Cont'd)

Request from W3C for IETF to resolve (2/2):

- A list of all available error code spaces that this WG feels need to be covered, as well as an outline of any reserved error codes, would be helpful to developing WebTransportError with more attributes. #213:
    - "Seems like we should reserve an error code (out of 0-255) for when streams error without an application-provided WebTransport application error code. @vasilvv mentioned there is an IETF issue for that (link?)
    - Would also like clarity from IETF on whether 0 means no error"

WebTransport over HTTP/3 (40 minutes)

Presentation End: 13:00

Victor Vasiliev

https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-http3

Updates since last IETF

- draft-01
  - Has fixes for all issues we had consensus for at IETF 110, notably:
    - Using https as the URI scheme
    - Consensus for using type-value frames without length
    - Buffering streams until they can be associated with a session
  - Available as an origin trial in Chrome 91-94.

Discussed at the interim

- Issue 39: server waiting for settings from the client
  - PR 56 requires server to wait for settings
- Issue 42: accept all 2xx status codes
  - PR 58 changes 200 to 2xx
- Issue 48: framing for unidirectional streams
  - Agreement to use new framing

Other minor issues

- Issue 10: consistent views of stream IDs across peers.
  - Close as WONTFIX?
- Issue 32: handle bad session IDs
  - PR 55

Issue discussion

https://github.com/ietf-wg-webtrans/draft-ietf-webtrans-http3/labels/next%20meeting

Issues 31, 40: RESET_STREAM

- Issue 31: RESET_STREAM is not reliable
- Issue 40: mapping RESET_STREAM error codes
- Proposal: 8-bit reset error code (remapped into HTTP/3 error space), mostly best effort
  - PR #59

Issue 50: connection throttling

Should we add considerations for limiting the number of connections to the text?

- Important for one-connection-per-session cases, as many connections could lead to resource exhaustion.
- Less relevant when pooling is enabled.

Issue 41: error codes for sessions

Proposal: introduce a capsule to close a WebTransport session.

64-bit error code

Error string (up to 1024 bytes)

If closed without capsule, both are omitted.

Issue 54: context IDs

draft-ietf-h3-datagram defines datagram context IDs.

It is possible to set no context ID.

Options:

1. Require servers to support context IDs.
2. Only allow no-context-ID registration, require extension for everything else.

Issue 27: GOAWAYs

- How does WebTransport interact with HTTP/3 GOAWAY? Does it prevent creating new data streams?
- Do we want a separate end-to-end mechanism for draining? Or should this be deferred to the application?

Issue 34: pooling negotiation

We had a long discussion about this at the previous IETF.

Current proposal: MAX_WEBTRANSPORT_SESSIONS server setting.

Backup discussion slide

WebTransport using HTTP/2 (40 minutes)

Presentation End: 13:40

Eric Kinnear

https://datatracker.ietf.org/doc/html/draft-ietf-webtransport-http2

Layering

WebTransport over HTTP/2 provides a mapping to TCP when HTTP/3 is not available

Layering


WEBTRANSPORT
HTTP/3
Uni-Stream
WT_STREAM
WT_STREAM
H3 DATAGRAM
H3 DATAGRAM
QUIC
Uni-STREAM
Bidi-STREAM
Bidi-STREAM
DATAGRAM
DATAGRAM
UDP
UDP
UDP
UDP
UDP

Layering

WEBTRANSPORT

Uni-Stream | WT_STREAM | WT_STREAM | DATAGRAM | DATAGRAM

HTTP/2

HEADERS/DATA | HEADERS/DATA | HEADERS/DATA | HEADERS/DATA | HEADERS/DATA

TLS

TCP

Layering


WEBTRANSPORT
Uni-Stream
Bidi-Stream
Bidi-Stream
Datagram
Datagram
HTTP/2
WT_STREAM/DATA
WT_STREAM/DATA
WT_STREAM/DATA
WT_DATAGRAM
WT_DATAGRAM
TLS
TCP

Layering

WT_DATAGRAM

WT_STREAM

WT_RST_STREAM

WT_STOP_SENDING

Layered vs Integrated

Per Martin's framing on the list, current h2 draft is "integrated" with h2

- Requires developing "in-stack" in order to function as a client, server or intermediary

Should WT over H2 move to a layered model?

- Can use any vanilla H2 (or H1) stack to implement WT

Layering

WT_DATAGRAM

WT_STREAM

WT_RST_STREAM

WT_STOP_SENDING

DATAGRAM

STREAM

RESET_STREAM

STOP_SENDING

PADDING

MAX_DATA

MAX_STREAM_DATA

MAX_STREAMS

Layering

HTTP/2

WT_DATAGRAM

WT_STREAM

WT_RST_STREAM

WT_STOP_SENDING

QUIC Transport over TCP

DATAGRAM

STREAM

RESET_STREAM

STOP_SENDING

PADDING

MAX_DATA

MAX_STREAM_DATA

MAX_STREAMS

DATA_BLOCKED

STREAM_DATA_BLOCKED

STREAMS_BLOCKED

Datagrams #25

Sending DATAGRAMS over HTTP/2 is being defined in MASQUE

WebTransport should use whatever scheme has consensus there

Since Monday, that means “layered”:

- Body of CONNECT stream will be a sequence of CAPSULEs (conveyed in DATA frames)
- DATAGRAM CAPSULE carries datagrams

What about Streams?

The CAPSULE over message body semantic is used as a fallback when the underlying transport doesn't support a construct natively

- HTTP/3 with H3 DATAGRAM extension does not use DATAGRAM capsule, uses the native feature instead

HTTP/2 has native streams

Should WebTransport streams use native HTTP/2 streams?

Requirements?

Is HTTP/1 support a requirement?

Is proxying through a generic HTTP/2 intermediary a requirement?

What's different

End-to-End vs. Hop-by-Hop

- Native streams are hop-by-hop (like WebTransport over HTTP/3's use of QUIC streams)

Unidirectionality

- HTTP/2 streams are not natively unidirectional
- Current draft adds them as an extension with minor tweaks
- Also adds new frames for unidirectional resets of bidirectional streams

Resource management

- Do HTTP/2 flow control and stream limits match the resource management semantics we want for WebTransport?
- Session state management?

Pros and Cons

Building WebTransport (QUIC) streams over a single stream (layered) requires more effort

- Who wants to write a new stream manager and flow controller?
- Assuming you have an HTTP/2 stack you can modify

But layered has other advantages

- Works with HTTP/1
- End-to-End by design
- Potentially reusable in other contexts where QUIC is not available

Layered vs Integrated

WebTransport over HTTP/2 provides a mapping to TCP when HTTP/3 is not available

Do we have additional use cases for <Thing> over QUIC over TCP?

CONNECT

WebTransport and MASQUE are both using CONNECT

Extended-CONNECT

CONNECT-UDP

CONNECT-IP

CONNECT-<Next Thing Here>

Discuss

Hums, Wrap-up, and Summary

Session End: 14:00

Bernard Aboba

David Schinazi

Thank you

Special thanks to:

The Secretariat, WG Participants & ADs