

WEBTRANS WG

Virtual Interim

February 22, 2024

13:00 - 15:00 Pacific Time





Mailing list: webtransport@ietf.org

Meeting info:

<https://datatracker.ietf.org/meeting/interim-2024-webtrans-02/session/webtrans>

Notes: <https://notes.ietf.org/notes-ietf-interim-2024-webtrans-00-webtrans>

Virtual Interim Remote Meeting Tips

- 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.
- Keep audio and video off unless you are chairing or presenting.
- Use of a headset is strongly recommended.

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)

Note really well

- IETF meetings, virtual meetings, and mailing lists are intended for professional collaboration and networking, as defined in the [IETF Guidelines for Conduct](#) (RFC 7154), the [IETF Anti-Harassment Policy](#), and the [IETF Anti-Harassment Procedures](#) (RFC 7776). If you have any concerns about observed behavior, please talk to the [Ombudsteam](#), who are available if you need to confidentially raise concerns about harassment or other conduct in the IETF.
- The IETF strives to create and maintain an environment in which people of many different backgrounds are treated with dignity, decency, and respect. Those who participate in the IETF are expected to behave according to professional standards and demonstrate appropriate workplace behavior.
- IETF participants must not engage in harassment while at IETF meetings, virtual meetings, social events, or on mailing lists. Harassment is unwelcome hostile or intimidating behavior -- in particular, speech or behavior that is aggressive or intimidates.
- If you believe you have been harassed, notice that someone else is being harassed, or have any other concerns, you are encouraged to raise your concern in confidence with one of the Ombudspersons.

About this meeting



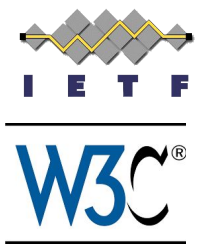
- Meeting info: <https://datatracker.ietf.org/meeting/interim-2024-webtrans-02/session/webtrans>
- Notes: <https://notes.ietf.org/notes-ietf-interim-2024-webtrans-00-webtrans>
- Secretariat: mtd@jabber.ietf.org
- WG Chairs: Bernard Aboba & David Schinazi
- Zulip Scribe: David Schinazi
- Note taker: TBD

Agenda



- Preliminaries, Chairs (10 minutes)
 - Note Well(s), Note Takers, Participation hints
 - Agenda Bash
- W3C WebTransport Update, Jan Ivar Bruaroey (10 minutes)
- Issue 85: MAX_STREAMS and Flow Control (80 minutes)
 - <https://github.com/ietf-wg-webtrans/draft-ietf-webtrans-http3/issues/85>
 - <https://datatracker.ietf.org/doc/html/draft-thomson-webtrans-session-limit/>
 - <https://datatracker.ietf.org/doc/html/draft-vvv-quic-namespaces/>
 - <https://datatracker.ietf.org/doc/html/draft-seeman-quic-stream-groups/>
- Wrap up and Summary, Chairs & ADs (15 minutes)

W3C WebTransport Update (1 of 3)



We will present a full update at IETF #119 brisbane, but for this interim would like to focus on one PR and one issue on which we have an IETF dependencies.

W3C WebTransport Update (2 of 3)



Issue [#411](#): expose a TLS Keying Material Exporter / unique value for the TLS session

- It would be useful to expose a TLS Key Exporter (RFC 5707), or at least a value that's derived from the TLS master secret, and therefore unique to the TLS session.
- We would like to extend our use of WebTransport to use the web PKI. Here, being able to derive a unique value would allow us to bind the TLS session to the Noise handshake, similar to how we use serverCertificateHashes

This is a useful building block for protocols running on top of TLS. Strong support from multiple authors detailed in the issue.

Victor reopened IETF WT WG issue [#116](#): Keying material exporters

Request to IETF: resolve #116 so that W3C can create an API, similar to

```
.exportKeyMaterial(label, context, len)
```


W3C WebTransport Update (3 of 3)



Issue [#580](#): Remove one dependency on "Data Recvd"

1. Wait for `|stream|.{{WebTransportSendStream/[[InternalStream]]}}` to enter the "Data Recvd" state. `[[!QUIC]]`

- Background as discussed in [ietf-wg-webtrans/draft-ietf-webtrans-http2#103](#)
- Useful for aborting a stream to create "partial reliable" sending (to abandon data you already sent)
- May need to keep a signal in HTTP/3 and not HTTP/2 (or add it, not useful there though)
- PR Remove dependency on "Data Recvd" state [#581](#) was merged.

@vasilvv to open an issue on the IETF side to update overview about the definition of streams to include "some sort of emitted state".

Issue 85: MAX_STREAMS and Flow Control

Victor Vasiliev, Eric Kinnear, Martin Thomson

<https://github.com/ietf-wg-webtrans/draft-ietf-webtrans-http3/issues/85>

<https://datatracker.ietf.org/doc/html/draft-thomson-webtrans-session-limit/>

<https://datatracker.ietf.org/doc/html/draft-vvv-quick-namespaces/>

<https://datatracker.ietf.org/doc/html/draft-seeman-quick-stream-groups/>

WebTransport Flow Control

And The Perils of Nesting QUIC Connections

Why Flow Control?



Flow control lets the receiver control how much data the sender can force it to buffer.

Flow control is essential since acknowledging data in QUIC means that the receiver commits to buffering it until it is processed by the application; no flow control would mean the receiver would have to buffer infinite amounts of data!

This is obviously not true for WebTransport, since WebTransport over HTTP/3 is subject to QUIC's flow control mechanisms.

So what do we mean by “WebTransport flow control”?

Problem: clients sharing the same connection

JavaScript code in different tabs can open connections to the same server without being aware of what the other tabs do; that is the explicit goal of pooling.

This means that a single JS client can open as many streams as possible, potentially using up all of the flow control credits that the server was willing to provide.

This situation is symmetric: a server may open too many streams since it does not have visibility with respect to resource usage to the other applications sharing the connection.

Counterpoints (why this might not be a problem)

Browsers already isolate connections to different websites when pooling.

- See <https://fetch.spec.whatwg.org/#network-partition-keys>
- This means that one website cannot use up the resource of another.

So while this is not a security issue, it is a robustness issue:

- Web applications in general are not aware of things going on in other tabs connected to the same website (unless they explicitly go out of their way)
- Even if two paths share a domain, they are not necessarily operated by the same people.

#85 MAX_STREAMS and Flow Control



- Also HTTP/2 #102
 - An intermediary doesn't currently have a good mapping between H2 and H3 for flow control

Options

- A. Do nothing
- B. Add “hints” suggesting flow control breakdown
- C. Add flow control to groups of QUIC streams
- D. Add flow control to WebTransport over HTTP/3

Options

- A. **Do nothing**
- B. Add “hints” suggesting flow control breakdown
- C. Add flow control to groups of QUIC streams
- D. Add flow control to WebTransport over HTTP/3

Options

- A. Do nothing
- B. Add “hints” suggesting flow control breakdown**
- C. Add flow control to groups of QUIC streams
- D. Add flow control to WebTransport over HTTP/3

Proposal B: make the sender enforce the limits

If the sender knows how much it has in terms of limits, it can limit the local senders on individual sessions itself.

With the way QUIC flow control works, the sender does not have a reliable way to know that: `MAX_STREAMS=400` indicates that the sender can open 100 streams initially, but this is not an indication that 100 streams will always be available.

Solution: give the sender enough information to allocate resources between sessions.

(we are going to refer to those numbers as “hints”, since the sender can choose to ignore those, at the cost of letting some of its sessions potentially starve)

Proposal B: overall idea

Let:

- `max_total` be the total maximum of allowed streams on the HTTP/3 connection (known by the receiver)
- `max_streams_per_wt` be the maximum number of allowed streams per individual WebTransport session
- `max_wt_sessions` be the maximum number of WebTransport sessions per connection (known by both parties via SETTINGS)
- `max_http_streams` is the number of non-WT HTTP/3 streams

$$\text{max_total} = \text{max_streams_per_wt} * \text{max_wt_sessions} + \text{max_http_streams}$$

The client needs to determine `max_streams_per_wt` and `max_http_streams` to perform local resource allocation.

Proposal B: two approaches to hinting

Approach 1. Hint the sender what is the expected `max_total` is.

- The sender now knows the total number of streams it can expect to be able to open, and the total number of sessions among which the resources are to be allocated.
- It can determine its own allocation of `max_streams_per_wt` and `max_http_streams`.

Approach 2. Hint the sender what `max_streams_per_wt` and `max_http_streams` to use.

Proposal B: advantages and disadvantages

Advantages:

- Simple in terms of wire format
 - Only requires implementing extra SETTINGS parameters
- Simple in terms of implementation
 - Fixed numeric limit that is enforced locally

Disadvantages:

- Assumes all sessions should get equal allocation of resources
 - QUIC has precedents for that (e.g. initial flow control window is same for all streams)
- Every session has to potentially be ready to handle the max flow control parameters for the entire connection
 - Consequence of the limits being expressed as a hint as opposed to a hard requirement.

Options

- A. Do nothing
- B. Add “hints” suggesting flow control breakdown
- C. Add flow control to groups of QUIC streams**
- D. Add flow control to WebTransport over HTTP/3

Proposal C: solve the problem on the QUIC layer

The most common approach appears to be making QUIC stack aware of individual WebTransport sessions and which groups belong to which sessions.

Proposed multiple times by several times by different people:

- [draft-seemann-quic-stream-groups](#)
- [draft-ppv-quic-namespaces](#)
- mvfst supports grouping streams on API level

This is similar in functionality to proposal D, except it is done at the transport level.

Proposal C: advantages and disadvantages

Advantages:

- Potentially useful for non-WebTransport use cases
- Reuses flow control logic already implemented in QUIC

Disadvantages:

- Blocks on QUIC working group
- *A lot* of space for bikeshedding
- The exact shape and usefulness is unclear

Options

- A. Do nothing
- B. Add “hints” suggesting flow control breakdown
- C. Add flow control to groups of QUIC streams
- D. Add flow control to WebTransport over HTTP/3**

Proposal D: General Idea

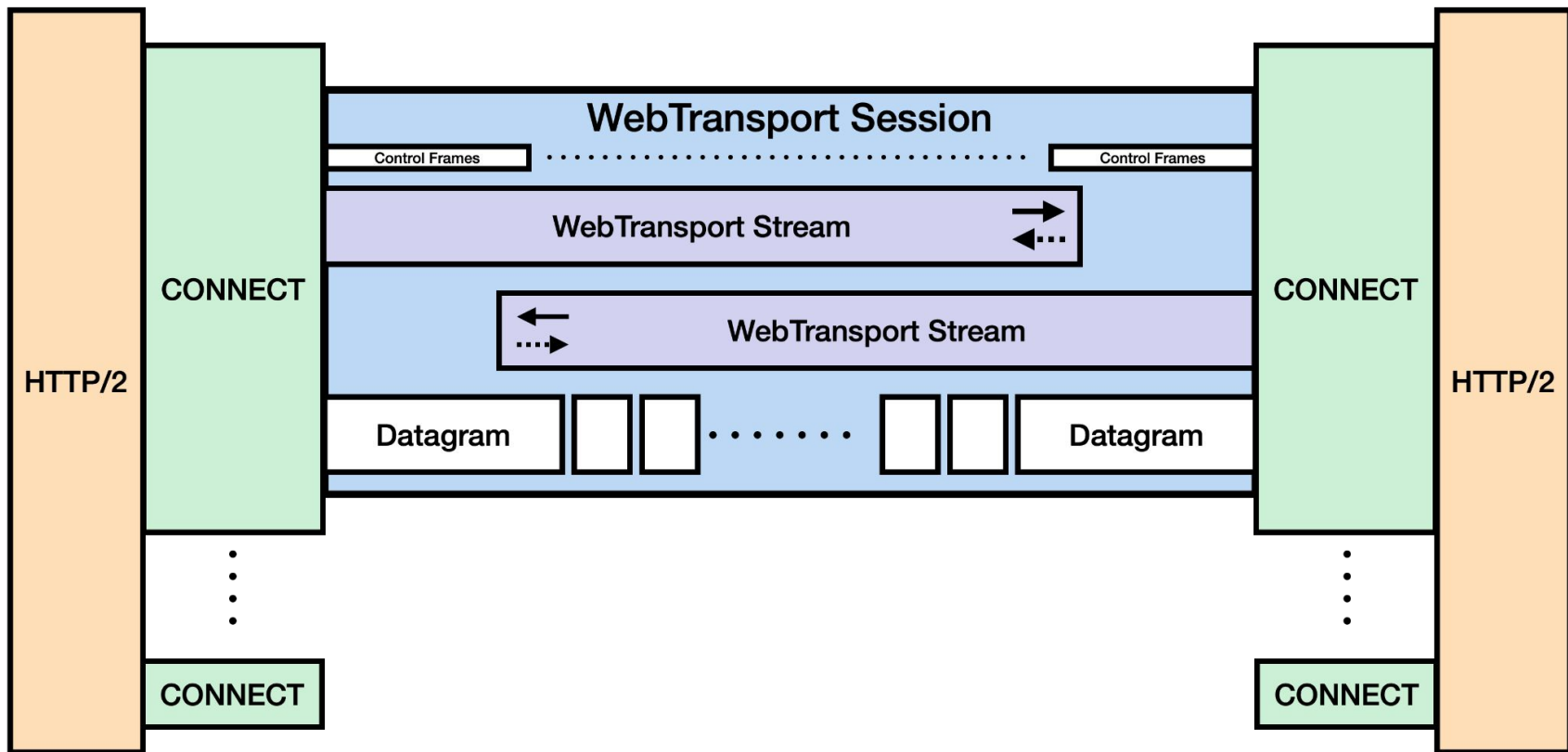
WebTransport over HTTP/2 uses QUIC-style flow control ~~frames~~ capsules

Proposal D: General Idea

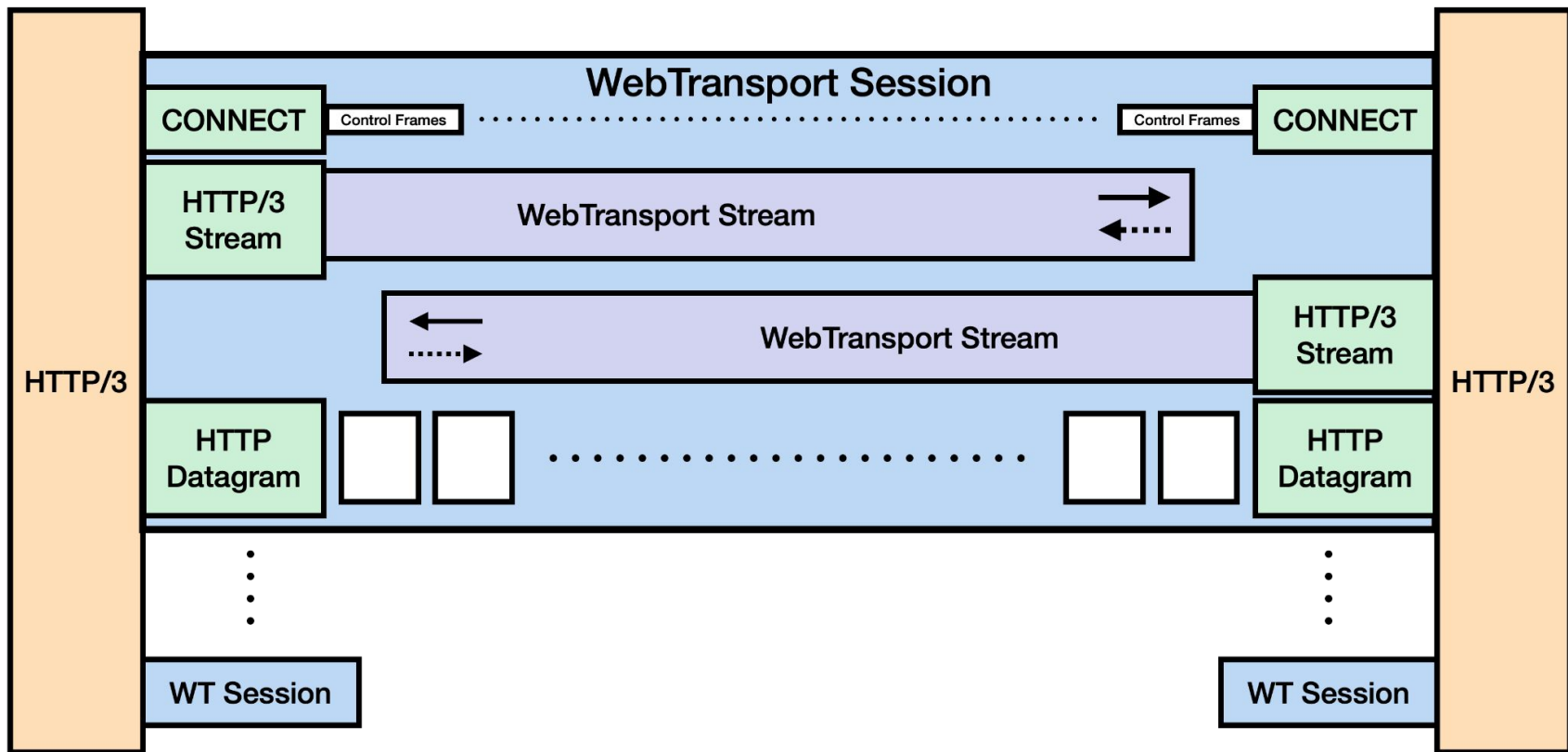
WebTransport over HTTP/2 uses QUIC-style flow control ~~frames~~ capsules

Use the same capsules consistently across all WebTransport

HTTP/2



HTTP/3



Capsules

WT_STREAM

DATAGRAM

PADDING

WT_RESET_STREAM

WT_STOP_SENDING

WT_MAX_STREAMS

WT_MAX_STREAM_DATA

WT_MAX_DATA

DRAIN_WEBTRANSPORT_SESSION

CLOSE_WEBTRANSPORT_SESSION

WT_STREAMS_BLOCKED

WT_STREAM_DATA_BLOCKED

WT_DATA_BLOCKED

HTTP/3

DRAIN_WEBTRANSPORT_SESSION

Both

DATAGRAM

CLOSE_WEBTRANSPORT_SESSION

HTTP/2

WT_STREAM

WT_MAX_STREAMS



ADDING

WT_MAX_STREAM_DATA

WT_RESET_STREAM

WT_MAX_DATA

WT_STOP_SENDING

WT_STREAMS_BLOCKED

WT_STREAM_DATA_BLOCKED

WT_DATA_BLOCKED

Both

WT_STREAM

DATAGRAM

PADDING

CLOSE_WEBTRANSPORT_SESSION

WT_RESET_STREAM

WT_STOP_SENDING

WT_MAX_STREAM_DATA

WT_STREAM_DATA_BLOCKED

HTTP/3

DRAIN_WEBTRANSPORT_SESSION

HTTP/2

WT_MAX_STREAMS

WT_MAX_DATA

WT_STREAMS_BLOCKED

WT_DATA_BLOCKED

Both

WT_STREAM

DATAGRAM

PADDING

CLOSE_WEBTRANSPORT_SESSION

WT_MAX_STREAMS

WT_MAX_DATA

WT_RESET_STREAM

WT_STOP_SENDING

WT_MAX_STREAM_DATA

WT_STREAM_DATA_BLOCKED

WT_STREAMS_BLOCKED

WT_DATA_BLOCKED

HTTP/3

DRAIN_WEBTRANSPORT_SESSION

HTTP/2

Flow Control Capsules

WT_MAX_DATA

WT_MAX_STREAMS

WT_DATA_BLOCKED

WT_STREAMS_BLOCKED

Proposal D: Details

[draft-thomson-webtrans-session-limit-01](#)

Options

- A. Do nothing
- B. Add “hints” suggesting flow control breakdown
- C. Add flow control to groups of QUIC streams
- D. Add flow control to WebTransport over HTTP/3

Timeliness, and The Possible Courses of Action

Four approaches:

1. Adopt a flow control mechanism now; make it always mandatory.
2. Adopt a flow control mechanism now; make it mandatory for pooling (browsers won't pool if the mechanism is not supported).
3. Defer flow control; ship the current version of the draft, ship an extension that would be prerequisite for pooling support in browsers later (gives us more time).
4. Do nothing (existing QUIC flow control is sufficient).

Discussion

- A. Do nothing
- B. Add “hints” suggesting flow control breakdown
- C. Add flow control to groups of QUIC streams
- D. Add flow control to WebTransport over HTTP/3

Wrap-up, and Summary

Bernard Aboba

David Schinazi

Next steps

- Merge some PRs
- Write some Code
- Editorialize some Drafts
- WGLC all the Things
- Profit

Thank you

Special thanks to:

The Secretariat, WG Participants & ADs