

WEBTRANS WG

IETF 117

Hybrid Meeting

Thursday, July 27, 2023

13:00 - 15:00 Pacific Time

Session II, Plaza A

Mailing list: webtransport@ietf.org

MeetEcho: <https://meetecho.ietf.org/conference/?group=webtrans>

Notes: <https://notes.ietf.org/notes-ietf-117-webtrans>

IETF 117 Meeting Tips

In-person participants





- Make sure to sign into the session using Meetecho (usually the “Onsite tool” client) from the Datatracker agenda
- Use Meetecho to join the mic queue
- *Keep audio and video off if not using the onsite version*



Remote participants

- Make sure your audio and video are off unless you are chairing or presenting during a session
- Use of a headset is strongly recommended

IETF 117 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.

Resources for IETF 117

- Information about IETF 117
<https://www.ietf.org/how/meetings/117>
- Agenda
<https://datatracker.ietf.org/meeting/agenda>
- If you need technical assistance, see the Reporting Issues page:
<http://www.ietf.org/how/meetings/issues/>

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



- Agenda:
<https://datatracker.ietf.org/doc/agenda-117-webtrans/>
- Notes: <https://notes.ietf.org/notes-ietf-117-webtrans>
- WG Chairs: Bernard Aboba & David Schinazi
- Zulip Scribe: David Schinazi
- Note Takers: ?

Agenda



- Preliminaries, Chairs (15 minutes)
 - Note Well(s), Note Takers, Participation hints
 - Agenda Bash
- W3C WebTransport Update, Will Law, (15 minutes)
- WebTransport over HTTP/2, Eric Kinnear (40 minutes)
 - <https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-http2>
- WebTransport over HTTP/3, Victor Vasiliev (40 minutes)
 - <https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-http3>
- Wrap up and Summary, Chairs & ADs (10 minutes)

W3C WebTransport Update (1)



W3C WebTransport WG progress since March 29, 2023

- **Status:** Published a [Working Draft](#) - latest version July 12 2023
- **Charter** current [charter](#) will expire Dec 31, 2023. We will need another extension.
- **Timetable** for year
 - Sept 30, 2023: Candidate for Recommendation - requires stability in API
 - October 2023: Proposed Recommendation - requires two independent implementations per our charter.
 - December 2023: Call for Review of a Proposed Recommendation
 - February 2024: Publication by W3C as a Recommendation after AC review
- **Milestone** status
 - [Candidate Recommendation](#) (74% complete, 13 open (13 ready-for-PR), 37 closed)
- **Annual TPAC meeting** planned for Tuesday Sept 12, 17:00 - 18:30 CEST

W3C WebTransport Update (2)



Major decisions and updates since last IETF report (March 29):

- **Add [estimatedSendRate](#) to stats. [#494](#)** - the estimated rate at which queued data will be sent by the user agent, in bits per second. This rate applies to all streams and datagrams that share a WebTransport session and is calculated by the congestion control algorithm. Now also available for pooled connections [#497](#) (no pooling across origins)
- **Specify BFCache interaction [#500](#)** - allow pages using WebTransport to enter the BFCache, but close the connection when navigating away.
- **Networking privacy considerations [#485](#)** - improved the spec language on [fingerprinting](#) and tracking
- **Support BYOB Readers for Datagrams [#487](#)** - this change adds support for BYOB readers for datagrams in the spec, similar to WebTransportReceiveStreams.
- **Make streamErrorCode 32-bit long [#509](#)** - 'nuff said
- **Add `sendStream.sendOrder = n` attribute. [#510](#)** - allows setting sendOrder of incoming BiDi streams and modifying stream sendOrder after creation.
- **UA SHOULD divide bandwidth fairly between all streams that aren't starved [#521](#)** - User agents are strongly encouraged to divide bandwidth fairly between all streams that are actively sending (e.g. not starved by having lower send order numbers)

W3C WebTransport Update (3)



WebTransport is now in Firefox 114 general release!

This means we now have two independent implementations of WebTransport.

- Firefox 114 supports WebTransport, including datagrams and BYOB
 - Congestion control is CUBIC
 - Protocol implementation is largely in Rust as part of our neqo http3 support
 - Passes 551 / 594 [web-platform tests](#)
A few features haven't landed yet: sendOrder support, getStats
- Demo with <https://jsfiddle.net/jib1/y05av6xz/>
 - select file to send to server and back
- Demo with MOQ demo server (<https://moq.streaming.university/>)
 - Playback remains blocked on upcoming [webcodecs in Firefox](#).



moz://a

W3C WebTransport Update (4)



Current issue of debate:

1. **Priority Groups #515** - prior work added the ability to opt send streams into strict-ordering by setting a non-null int64 `sendOrder`. This was designed with per-media-segment streams in mind. But an application with two or more such media flows would need to create streams at a lock-step rate with identical highest `sendOrder` for this to work. This seems too limiting, e.g. when mixing stream-per-frame and stream-per-segment flows. It also didn't address datagrams. **Proposal**: introduce explicit priority groups in the API to separate flows and their `sendOrder` number-spaces.
 - a. Every writable in the session (streams and the datagram queue) gets a priority group. There is a default group.
 - b. All groups have equal (implementation-defined) "fairness" from a send priority standpoint.
 - c. Within groups, every stream has an int64 `sendOrder` that's strict and *0 by default*, near null (simpler).

```
const group = new WebTransportSendPriorityGroup(wt); // or const group = wt.createSendPriorityGroup();
const writable = await wt.createUnidirectionalStream({sendPriorityGroup: group});
const {writable} = await wt.createBidirectionalStream({sendPriorityGroup: group});
for await (const {writable, readable} of wt.incomingBidirectionalStreams) {
  writable.sendPriorityGroup = group;
}
```

Questions:

- d. Is this group construct sufficient to meet expected media (and non-media) transmission needs?
- e. Currently fairness is implied between groups. Is explicit weighting needed for groups?
- f. Should datagrams be in their own group? If so, what should their relative default weight be?
- g. Are groups actually "Flows"? When applied to datagrams, this would allow sending of parallel flows.
- h. If flows were implemented on the send side, is there any mechanism in the transport to replicate these flows on the receive side?
- i. Does such an application-level priority group construct match|||match any WebTransport layer construct around prioritization and fairness?

W3C WebTransport Update (5)

Current issues of debate continued:

2. **Retransmissions and send order #523** - should new data on higher sendOrder stream preempt retransmissions of data lost on a lower sendOrder stream?

Questions:

1. Should there be a Boolean API to toggle whether retransmissions inherit stream sendOrder?
2. Should there be an API to specify a time window within which retransmissions be given the highest send priority?
3. This may not be a problem if retransmissions are rare and short lived.
4. This may not be a problem because lower priority streams can be aborted by the sender.

WebTransport over HTTP/2

Eric Kinnear

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

Changes since IETF 116

-06

Adds initial flow control limits

Settings

Defaults for new sessions on that connection

Header field

Defaults for each session

Changes since IETF 116

-06

Updated examples

What's left?

Error handling ([#44](#))

A few new issues from implementation experience

Error Handling (#44)

Matching HTTP/3, but...

If a fatal error is discovered, reset the HTTP/2 stream

Final Size (#87)

```
WT_RESET_STREAM Capsule {  
    Type (i) = 0x190B4D39,  
    Length (i),  
    Stream ID (i),  
    Application Protocol Error Code (i),  
}
```

```
QUIC RESET_STREAM Frame {  
    Type (i) = 0x04,  
    Stream ID (i),  
    Application Protocol Error Code (i),  
    Final Size (i),  
}
```

Final Size (#87)

Unlike the equivalent QUIC frame, this capsule does not include a Final Size field. In-order delivery of WT_STREAM capsules ensures that the amount of session-level flow control consumed by a stream is always known by both endpoints.

Final Size (#87)

Unlike the equivalent QUIC frame, this capsule does not include a Final Size field. In-order delivery of WT_STREAM capsules ensures that the amount of session-level flow control consumed by a stream is always known by both endpoints.

Is that sufficient? Sometimes you want to know how much was consumed

Flow Control Terminology ([#88](#))

You cannot lower the max number of sessions after advertising a higher value

Stream limits are cumulative, perhaps with an example
You need to keep updating with a higher value if you want more streams (i.e. they are not about the concurrent number of streams)

WebTransport overview

WebTransport over HTTP/3

(40 minutes)

Victor Vasiliev

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

Key exporters (#116)

There is a proposal for how to derive key exporters for a given WebTransport session

Issue: do we need this?

Protocol negotiation (#130)

SETTINGS_WEBTRANSPORT_MAX_SESSIONS

is currently required in both directions

Every WebTransport version has its own codepoint

Version used =

$$\max(\max(\text{client_offered}), \max(\text{server_offered}))$$

Proposal: only send the setting on the server-side,
rotate codepoints of frames on breaking changes

Flow Control (#85)

draft-thomson-webtrans-session-limit

Could also include WebTransport-Init

Flow Control (#85)

WT_MAX_DATA

WT_MAX_STREAMS

~~WT_MAX_STREAM_DATA~~

WT_DATA_BLOCKED

WT_STREAMS_BLOCKED

Wrap-up, and Summary (15 minutes)

Bernard Aboba

David Schinazi

Thank you

Special thanks to:

The Secretariat, WG Participants & ADs