WEBTRANS WG
IETF 118

Hybrid Meeting

Monday, November 6, 2023

15:30 - 17:00 Prague Time

06:30 - 08:00 Pacific Time

Session III, Berlin 1/2

Mailing list: webtransport@ietf.org

MeetEcho: [webtrans (ietf.org)]

Notes: https://notes.ietf.org/notes-ietf-118-webtrans

This session is being recorded

IETF 118 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

This session is being recorded

IETF 118 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 118

- Information about IETF 118
  https://www.ietf.org/how/meetings/118
- 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-118-webtrans/
- Notes: https://notes.ietf.org/notes-ietf-118-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 (25 minutes)
  - https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-http2
- WebTransport over HTTP/3, Victor Vasiliev (25 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 July 27, 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
  - Nov 30, 2023: Candidate for Recommendation - requires stability in API
  - January 2023: Proposed Recommendation - requires two independent implementations per our charter.
  - February 2023: Call for Review of a Proposed Recommendation
  - April 2024: Publication by W3C as a Recommendation after AC review
- **Milestone** status
  - Candidate Recommendation (76% complete, 12 open (12 ready-for-PR), 38 closed)
- **Annual TPAC meeting** held Tuesday Sept 12.

W3C WebTransport Update (2)

Major decisions and updates since last IETF report (July 27):

- Changes to stats
    - ***Remove stats numOutgoingStreamsCreated & numIncomingStreamsCreated #545***
    - ***Add stat bytesLost #546*** **-** to provide symmetry with packetsLost.
    - ***Remove timestamp from all stats objects #552*** - use performance.now() as a replacement

- ***Add SendGroup, and make sendOrder no longer nullable.*** #548
  SendGroup defines a number spaces to preserve fairness between flows using the sendOrder mechanism

```
const sendGroupA = wt.createSendGroup();
const sendGroupB = wt.createSendGroup();
const writableA1 = await wt.createUnidirectionalStream({sendGroup: sendGroupA});
const writableB1 = await wt.createBidirectionalStream({sendGroup: sendGroupB, sendOrder: i--});
for await (const {writable} of wt.incomingBidirectionalStreams) {
  writable.sendGroup = sendGroupB;
  writable.sendOrder = i--;
}
```

W3C WebTransport Update (3)

Major decisions and updates since last IETF report (July 27):

- ***Reject create(Uni|Bi)directionalStream() on stream ID exhaustion. #528*** - if stream IDs exceed MAX_STREAMS limit, stream creation attempt will reject immediately with a `QuotaExceededError`. A constructor option to opt-in to the old behavior (block) is being considered in issue #446.

- ***Add writer.atomicWrite() method.* #551 (not yet merged)**
  Sends a transactional piece of data on a stream only if it can be done entirely without blocking on flow control.

```
async function sendTransactionalData(wt, bytes) {
  const writable = await wt.createUnidirectionalStream();
  const writer = writable.getWriter();
  await writer.ready;
  try {
    await writer.atomicWrite(bytes);
  } catch (e) {
    if (e.name != "AbortError") throw e;
    // rejected to avoid blocking on flow control. Writable remains un-errored unlike with regular writes
  } finally {
    writer.releaseLock();
  }
```

W3C WebTransport Update (4)

Browser support as of Nov 1st, 2023 (FF for Android should also be green)

WebTransport - WD

Usage: % of all users
Global: 73.54%

Protocol framework to send and receive data from servers using HTTP3. Similar to WebSockets but with support for multiple streams, unidirectional streams, out-of-order delivery, and reliable as well as unreliable transport.

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-90																
91-94																
95-96	12-97		2-113	10-82				4-17.0								
97-117	98-117	3.1-17.0	114-117	83-102	6-10		3.2-17.0	18.0-21		12-12.1		2.1-4.4.4				2.5
118	118	17.1	118	103	11	118	17.1	22	all	73	15.5	118	118	13.1	13.18	3.1
119-121		17.2-TP	119-121				17.2									

Safari is coming along: *261007 – Begin adding abstractions for implementing WebTransport (webkit.org)*
260810 – Add IDL skeleton for WebTransport (webkit.org)

W3C WebTransport Update (5)

Current issues of debate (1 of 2):

1. **Quality of a bandwidth estimate #559** - current API provides a stats called estimatedSendRate, defined as "*The estimated rate at which queued data will be sent by the user agent, in bits per second*". There might be utility in exposing some kind of "quality" indication of a bandwidth estimate, indicating how close it is to the "real" bandwidth of the channel.

   There are few signals that congestion controller could expose as an API such as

   - Has there ever been a bandwidth sample that is not application-limited?
   - Is the connection out of slow start?
   - Is the estimate the full bandwidth or a minimum of what's been observed?

   Questions:

   - Is there a utility in enhancing the information reported for the send rate?
   - If so, which signals most benefit an application? Do the ones suggested here make sense? Are there more useful signals?
   - Are such signals expensive for the UA to generate?

W3C WebTransport Update (6)

Current issues of debate (2 of 2):

2. Improve Server→Client Stream Performance by Allowing Customizable Concurrency Limits in WebTransport #544

Some apps which are about to receive lots of streams, need a way to assist the user agent in determining the MAX_STREAM_LIMIT to negotiate with the server, to avoid slow ramp-up over multiple round-trips.
An API shape like below is being considered:

```
const wt = new WebTransport({
  maxConcurrentIncomingUnidirectionalStreams: 10000,
  maxConcurrentIncomingBidirectionalStreams: 10000
});
```

Questions:

i. These are hints to the user agent. Does this make sense?
ii. What should a user agent use for default values?
iii. What are reasonable inputs? E.g. throw outside of [0, 100,000] ?
iv. Is it important to be precise? Would `"low"`, `"medium"`, `"high"` suffice?

WebTransport over HTTP/2

Eric Kinnear

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

#95 Drain WebTransport Session

- HTTP/2 only has WT_STOP_SENDING, which is not bidirectional
- HTTP/3 has DRAIN_WEBTRANSPORT_SESSION capsule
- Want a compatible interface with HTTP/3

#95 Drain WebTransport Session

```
DRAIN_WEBTRANSPORT_SESSION Capsule {
  Type (i) = DRAIN_WEBTRANSPORT_SESSION,
  Length (i) = 0
}
```

After sending or receiving either a DRAIN_WEBTRANSPORT_SESSION capsule or a HTTP/3 GOAWAY frame, an endpoint MAY continue using the session and MAY open new streams. The signal is intended for the application using WebTransport, which is expected to attempt to gracefully terminate the session as soon as possible.

WebTransport as a Generic Transport

There may be more cases like this one…

WebTransport as a Generic Transport

WebTransport should be transport agnostic.

#94 Flow Control Violations

The server MUST NOT close the connection if the client opens sessions exceeding this limit, as the client and the server do not have a consistent view of how many sessions are open due to the asynchronous nature of the protocol; instead, it MUST reset all of the CONNECT streams it is not willing to process with the `REFUSED_STREAM` error code ({{Section 8.7 of HTTP2}}).

#94 Flow Control Violations

- Within WebTransport (and QUIC), we didn’t intend to allow going backwards for certain flow control values
  - e.g. WT_MAX_STREAMS
- In HTTP/2, you can update SETTINGS, which are then ACKed so both endpoints agree on when the new values have been applied
- SETTINGS_WEBTRANSPORT_MAX_SESSIONS limits the number of sessions that can be open at any one time

#94 Flow Control Violations

- SETTINGS_WEBTRANSPORT_MAX_SESSIONS limits the number of sessions that can be open at any one time
- If you lower that limit, both sides agree on what the limit currently is, so we can allow the peer to gracefully close the currently open sessions

- In practice, enforce the limit when a new session is opened
- If you want a new session, you need to be under the limit

WebTransport Overview
WebTransport over HTTP/3

Victor Vasiliev

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

#137 Subprotocol Negotiation

- MoQ and others need ALPN-style functionality from WebTransport
- Proposal: Client sends ALPN header from RFC7639

```
ALPN                = 1#protocol-id
protocol-id         = token ; percent-encoded ALPN identifier
```

- Client can send multiple, server sends one in response
- Optional to send for the client

#85 Flow Control

- Problem: when pooling is enabled, one WebTransport session may starve out another
- Observation: this is a sender-side side allocation problem, not a safety mechanism for the peer
  - The peer can already limit total resource usage per entire QUIC connection
  - If one session starves another, this is not a security issue, since they share origin
  - We do need to give the sender enough information to allocate resources correctly

#85 Flow Control (continued)

Model (for client-initiated bidi streams):

```
max_concurrent_streams =
    max_streams_per_session * max_sessions
     + max_requests
```

max_sessions is known, but we don't know the rest

#85 Flow Control (continued)

Proposal:

WEBTRANSPORT_MAX_BIDI_STREAMS_HINT
MAX_TOTAL_BIDI_STREAMS_HINT

Not enforced, but gives the peer enough information to allocate resources between sessions

#85 Flow Control (continued)

Worked example:

MAX_TOTAL_BIDI_STREAMS_HINT = 100
WEBTRANSPORT_MAX_BIDI_STREAMS_HINT = 30
WEBTRANSPORT_MAX_SESSIONS = 3

Suggests 30 streams per WebTransport session, with 10 streams of headroom for HTTP requests

#85 Flow Control (continued)

Alternatives explored:

- QUIC-style flow control per WT session
  - Makes the peer do the allocation
  - Unclear if it actually solves the problem
- QUIC-level stream namespaces
  - Solves the problem by removing total limit
  - Should work, but requires a lot of QUIC-level changes

Waiting until SETTINGS

(issues 135, 139, 140)

Proposal:

- Client MUST wait until it has server SETTINGS
- Server MUST NOT read client bidi streams until it has client settings

Rationale: required to know that peer supports WT

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