W3C WebRTC WG Meeting

August 23, 2016 1 PM PDT

Chairs: Harald Alvestrand

Stefan Hakansson

Erik Lagerway

W3C WG IPR Policy

- This group abides by the W3C patent policy https://www.w3.org/Consortium/Patent-Policy-20040205
- Only people and companies listed at https://www.w3.org/2004/01/pp-impl/47318/status are allowed to make substantive contributions to the WebRTC specs

Welcome!

- Welcome to the interim meeting of the W3C WebRTC WG!
- During this meeting, we hope to make progress on some outstanding issues before transition to CR
- Editor’s Draft update to follow meeting

About this Virtual Meeting

Information on the meeting:

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/August_23_2016#Virtual_Interim
- Link to Slides has been published on WG wiki
- Scribe? IRC http://irc.w3.org/ Channel: #webrtc
- The meeting is being recorded.
- WebEx info here

For Discussion Today

- **WebRTC 1.0 API**
  - **Pull Requests**
    - Issue 685/PR 759: Receipt of Multiple RTP Encodings (Bernard)
    - Issue 714/PR 740: STUN/TURN OAuth token parameter (misi/Bernard)
    - Issue 720/PR 738: Getting the fingerprint of an RTCCertificate (Fippo/Bernard)
    - Issue 726/PR 757: Add ufrag attribute to RTCIceCandidate (Ekr/Fluffy)
    - Issue 732/PR 758: replaceTrack with the previous one as ended (mparis/Bernard)
  - **Issues**
    - Issue 678: Support assertions that identify the recipient (Cullen Jennings)
    - Issue 692: Meaning of “Liveness Checks Have Failed” (Taylor)
    - Issue 700: An event for when a circuit breaker is triggered (Varun Singh)
    - Issue 729: RTCStats timestamp source ambiguous (Randell Jesup)

WebRTC PC

- **Pull Requests**
  - **Issue 685**/**PR 759**: Receipt of Multiple RTP Encodings (Bernard)
  - **Issue 714**/**PR 740**: STUN/TURN OAuth token parameter (misi/Bernard)
  - **Issue 720**/**PR 738**: Getting the fingerprint of an RTCCertificate (Fippo/Bernard)
  - **Issue 726**/**PR 757**: Add ufrag attribute to RTCIceCandidate (Ekr/Fluffy)
  - **Issue 732**/**PR 758**: replaceTrack with the previous one as ended (mparis/Bernard)

Issue 685/PR 759: Receipt of Multiple RTP Encodings

- In WebRTC 1.0 Section 5.1 it says:

'If sendEncodings is set, then subsequent calls to `createOffer` will be configured to send multiple RTP encodings as defined in [JSEP] (section 5.2.2. and section 5.2.1.). When`setRemoteDescription` is called with a corresponding remote description that is able to receive multiple RTP encodings as defined in [JSEP], the `RTCRtpSender` may send multiple RTP encodings and the parameters retrieved via the transceiver's `sender.getParameters()` will reflect the encodings negotiated.'

- Questions:
  - JSEP reference to receiving multiple encodings (based on draft-ietf-mmusic-rid)? (IETF Issue)
    - Issue Link: https://github.com/rtcweb-wg/jsep/issues/285
    - Proposed Issue PR: https://github.com/rtcweb-wg/jsep/pull/289/files
  - Does WebRTC 1.0 enable browsers to receive as well as send multiple encodings and if so, how?

Issue 685/PR 759: Receipt of Multiple RTP Encodings

- Proposal in PR 759:
  - No receiveEncodings or RTCRtpReceiver.setParameters(), so calls to createOffer() cannot be configured to receive multiple RTP encodings.
  - If setRemoteDescription() is called with a remote description that is able to send multiple RTP encodings:
    - If the RTCRtpReceiver can receive multiple RTP encodings, the parameters retrieved via the transceiver's receiver.getParameters() reflect the encodings negotiated.
    - If a browser implementation is not capable of receiving multiple RTP encodings, then multiple encodings are not negotiated.
- Comments?

Issue 714/PR 740: STUN/TURN OAuth Token Parameter

- Filed by misi: How is RFC 7635 (STUN Extension for OAuth 2.0) supported within RTCIceServer? Currently, we have:

```
dictionary RTCIceServer {
    required (DOMString or sequence<DOMString>) urls;
             DOMString                          username;
             DOMString                          credential;
             RTCIceCredentialType               credentialType = "password";
};

enum RTCIceCredentialType {
    "password",
    "token"
};
```

Issue 714/PR 740 STUN/TURN OAuth Token (cont'd)

- RFC 7635 Appendix B example of a token credential:

```
{
   "access_token":
"U2FsdGVkX18qJK/kkWmRcnfHglrVTJSpS6yU32kmHmOrfGyI3m1gQj1jRPsr0uBb
HctuycAgsfRX7nJW2BdukGyKMXSiNGNnBzigkAofP6+Z3vkJ1Q5pWbfSRroOkWBn",
   "token_type":"pop",
   "expires_in":1800,
   "kid":"22BIjxU93h/IgwEb",
   "key":"v51N62OM65kyMvfTI08O"
   "alg":HMAC-SHA-256-128
 }
```

```
dictionary TokenCredential {
    required DOMString access_token;
    required DOMString token_type;
    required DOMString expires_in;
    required DOMString kid;
    required DOMstring key;
    required DOMString alg;
};
```

Issue 714/PR 740 STUN/TURN Token Proposals

- Proposal #1 (by misi): Extend RTCIceServer with two new attributes:

```
    dictionary RTCIceServer {
    required (DOMString or sequence<DOMString>) urls;
             DOMString                          username;
             DOMString                          credential;
             DOMString                          accesstoken;
             DOMTimeStamp                       expiry;
             RTCIceCredentialType               credentialType = "password";
};
```

- Comments
  - Harald: Adding more attributes doesn’t seem like the ideal solution.

Issue 714/PR 740 STUN/TURN Token Proposals (cont'd)

- Proposal #2 (by misi): Define new dictionaries and change credential type to PasswordCredential or TokenCredential

```
dictionary TokenCredential {
    required DOMString kid;
    required DOMString key;
    required DOMString alg;
   required DOMString access_token;
  DOMTimeStamp expiry;
};

dictionary PasswordCredential {
             DOMString                                  username;
             DOMString                                  password;
};

dictionary RTCIceServer {
    required (DOMString or sequence<DOMString>) urls;
             (PasswordCredential or TokenCredential)    credential;
             RTCIceCredentialType                       credentialType = "password";
};
```

- Comments
    - Don't you also need token_type in the TokenCredential dictionary?
    - Removal of RTCIceServer.username a backward compatibility issue?

Issue 720/PR 738: Getting the fingerprint of an RTCCertificate

- Filed by Fippo:
  - RTCCertificate doesn't have a way to get the fingerprint of the generated certificate
- Proposal in PR 738:

```
partial interface RTCCertificate {
    readonly attribute RTCDtlsFingerprint fingerprint;
};

dictionary RTCDtlsFingerprint {
    DOMString algorithm;
    DOMString value;
};
```

- Comments
  - Martin: what if we want two hash functions?
  - Fluffy: Would be more comfortable not comparing hash from two different algorithms… would prefer the hash algorithm/fingerprint be returned… nice to support multiple hash algorithms.

Issue 726/PR 757: Add ufrag attribute to RTCIceCandidate

- Filed by ekr:
  - We should have the ufrag included with ICE candidates so you could disambiguate candidates and end-of-candidates from different ICE generations (restarts). We need to:
    - Add ufrag to RTCIceCandidate
    - Add ufrag to one of the things serializer emits
    - Modify “done gathering marker” to be {mid, ufrag, null} (Justin)
- Proposal in PR 757:

```
partial interface RTCIceCandidate {
    readonly attribute DOMString? candidate;
    readonly attribute DOMString ufrag;
    Serializer = {candidate, sdpMid, sdpMLineIndex, ufrag };
}
```

- Comments
  - Fluffy: In the current spec, the end of candidates are indicated by a NULL, this changes this to return a map with the ufrag. It is not clear to me how to make this backwards compatible.

[Issue 732]/[PR 758]: replaceTrack with previous one ended

- Filed by mparis:
  - What should happen if we try to replaceTrack of an ended track?
    - Section 4.8.1: "Internal changes within the implementation can also result in the connection being marked as needing negotiation. For example, if a MediaStreamTrack enters the ended state because its source device became unavailable."
    - Section 5.2: "replaceTrack: Attempts to replace the track being sent with another track provided, without renegotiation."
- Comments
  - Bernard: Can we avoid setting the negotiation-needed flag when *sender.track* enters the ended state?
  - Jesup: Should be up to the application… the track is still attached… the application could call replaceTrack() … or just leave it.
  - Harald: What happens to *receiver.track*?
    - Bernard: If negotiation-needed flag is not set when *sender.track* enters the ended state, application can call *sender.replaceTrack* which leaves *receiver.track* on the remote peer unaffected. Or application can call *transceiver.stop*, which sets the negotiation-needed flag. After negotiation, transceivers on both peers are stopped, ending *receiver.track*.

Issue 732/PR 758: replaceTrack with previous one ended (cont'd)

- Proposal in PR 758:
  - Remove text in Section 4.8.1 setting the negotiation-needed flag when *sender.track* is ended.
  - Add text to Section 5.2 stating that replaceTrack can be used to replace a track that is ended.

WebRTC PC

- **Issues**
  - **Issue 678**: Support assertions that identify the recipient (Cullen Jennings)
  - **Issue 692**: Meaning of “Liveness Checks Have Failed” (Taylor)
  - **Issue 700**: An event for when a circuit breaker is triggered (Varun Singh)
  - **Issue 729**: RTCStats timestamp source ambiguous (Randell Jesup)

Issue 678: Support Assertions that Identify the Recipient

- Agreed to add at previous meeting
- Martin has action to make a PR

Issue 692: Meaning of “Liveness Checks” Have Failed

- RTCIceTransportState’s “disconnected” state definition says:

  ”Liveness checks have failed. This is more aggressive than `failed`, and may trigger intermittently (and resolve itself without action) on a flaky network.”

- Questions
  - What is a liveness check? Is it a STUN consent check?
  - "Checks" is plural.
    - So how many checks must fail before entering `disconnected`?
    - Does the state return to `connected` when the next check succeeds?
  - If the implementation has another way of knowing that connectivity is lost (such as a network interface going down), does it still need to wait for liveness checks to fail?

Issue 692: Meaning of “Liveness Checks” Have Failed (cont’d)

- Recommendation
  - Change “Liveness checks have failed” to: “The ICE agent believes connectivity is currently lost. The criteria is implementation-defined; examples include losing the network interface for the currently active candidate pairs, or successively failing to perform STUN consent checks [RFC7675].”
  - This matches how “disconnected” has been interpreted up until now.
  - Also offers flexibility to browsers to account for differences in ICE implementations.
    - For example, Chrome (currently) sends consent checks more frequently than the recommended 5 seconds. So it can afford to wait for a few to fail before transitioning to “disconnected.”
  - We could also add an informative note: “Applications should use this state as a hint that an ICE restart may restore connectivity.” This further clarifies the purpose of the state.

Issue 700: An Event for when a Circuit Breaker is Triggered

- Question:
  - Has any browser implemented circuit breakers or are there plans to implement circuit breakers?
- If “yes”, how does the implementation behave?
  - draft-ietf-avtcore-rtp-circuit-breakers Section 4.5:
    - What it means to cease transmission depends on the application.
    - This could mean
      - Stopping a single RTP flow, OR
      - that multiple bundled RTP flows are stopped.

[Issue 700](): An Event for when a Circuit Breaker is Triggered (cont'd)

The intention is that the application will stop sending RTP data packets on a particular 5-tuple (transport protocol, source and destination ports, source and destination IP addresses), until whatever network problem that triggered the RTP circuit breaker has dissipated. RTP flows halted by the circuit breaker SHOULD NOT be restarted automatically unless the sender has received information that the congestion has dissipated, or can reasonably be expected to have dissipated.

Section 8: gives guidance for layered media, flows with different qos markings.

[Issue 700](#): An Event for when a Circuit Breaker is Triggered (cont'd)

- Questions:
  - Is the effect of triggering the circuit breaker visible to the application?
    - Via an event?
    - Via change in the value of an attribute?
  - Can the implementation automatically resume sending?
    - When congestion abates?
    - When the ICE selected pair changes?
  - Is there something the application can or should do on triggering the circuit breaker (e.g. ICE restart)?

Issue 729: RTCStats timestamp source ambiguous

- If source is local (most stats), no problem
- If source is remote (RTCP), what is the time?
  - Arrival time of information @local?
  - Calculated from NTP timestamp, RTP timestamp, RTT?
  - Not all reports have the same timing info in them.
- Proposed resolution
  - Stats timestamp is time of arrival at local system
  - Additional field “remote timestamp” gives remote NTP timestamp translated to “normal time” format, no fixups

Thank you

Special thanks to:

W3C/MIT for WebEx

WG Participants, Editors & Chairs