JSEP

IETF 95

ekr@rtfm.com
juberti@google.com
fluffy@cisco.com

Changes since IETF 94 (1 of 2)

- Added discussion of RtpTransceivers + RtpSenders + RtpReceivers, and how they interact with createOffer/createAnswer.
- Removed OfferToReceiveX options.
- Explained how addIceCandidate can be used for end-of-candidates.
- Clarified which SDP lines can be ignored.
- Clarified how to handle various received attributes.
- Revised how attributes should be generated for bundled m= lines.
- Remove text advocating use of unilateral PTs.
- Trigger an ICE restart even if the ICE candidate policy is more strict.
- Remove the 'public' ICE candidate policy.

Changes since IETF 94 (2 of 2)

Split local/remote description accessors into current/pending.
Clarify a=imageattr handling.
Add more detail on VoiceActivityDetection handling.
Reference draft-shieh-rtcweb-ip-handling.
Make it clear when an ICE restart should occur.
Remove MSID semantics.
ice-options are now at session level.
Default RTCP mux policy is now 'require'.

Added RtpTransceivers, RtpSenders, RtpReceivers

Big change, but pretty much as we agreed upon before

Transceivers encapsulate m= lines; apps can now control what tracks go into which m= lines

Changes in transceivers cause changes in the SDP and callbacks for renegotiation (e.g. changing of direction attribute)

Senders allow precise control of track encodings without SDP munging

Removed OfferToReceiveX options

Removed this because the new RtpTransceivers, RtpSenders, and RtpReceiver objects allow applications to create whatever m= lines they needed; richer control than this option offered

General Bandwidth Handling

Handle RR and RS at media level or session level

Handle CT at session level, ignore CT at media level

Ignore AS at session level, handle at media level

How addIceCandidate can be used for end-of-candidates

This is signalled by calling addIceCandidate(null)

Bug in w3c spec - need to change for this
https://github.com/w3c/webrtc-pc/issues/569

Clarified which SDP lines can be ignored

Don’t use values of

"i=", "u=", "e=", "p=", "t=", "r=", "z=", and "k="

Clarified how to handle various received attributes

Updated text for various attributes not covered in jsep-12:

a=rtcp-rsize a=msid a=max-message-size

a=candidate a=remote-candidates a=end-of-candidates

a=imageattr a=rid a=simulcast

"a=rtcp" (value ignored)

How attributes should be generated for bundled m= lines

Per latest BUNDLE spec:

- Attributes are attached to the first m= line in the bundle group
- Omitted for all other m= sections in the bundle group
- This means they may be on different m= lines for offer and answer
    - If the answerer rejects the first m= line

ICE restart even if the ICE policy is more strict

We used to restart only if ICE policy added new potential candidates

Now we trigger a restart with any policy change

This is simpler, and avoids any confusion

Remove the 'public' ICE candidate policy

This is something the browser sets, not the site

Reference draft-rtcweb-ietf-ip-handling.

Clarify that ICE Policy filtering happens on top of any filtering browser applied per draft-rtcweb-ietf-ip-handling

Clarify a=imageattr handling.

Clarified how we handle

- multiple a=imageattr for different codecs (use appropriate one)
- multiple a=imageattr for same codec (use highest q=)
- constraints other than resolution (ignore)

Add more detail on VoiceActivityDetection handling.

Previously, the createOffer VoiceActivityDetection option could be set separately for either direction of the call. To ensure consistent behavior, the answerer can now only set VoiceActivityDetection if it was offered.

In addition, the document now states more clearly that this option only affects the negotiated codecs, and does not affect things such as the VAD bit in client-to-mixer-audio-levels.

Issue #206 covers an API to control dynamic use of VAD.

No longer required to support RTCP MUX ‘negotiate’

Browser no longer needs to implement support for normal RTCP and RTP (not multiplexed)

Browser no longer needs to support the RTCP Mux option of “negotiate”

And the default RTCP mux policy is now 'required'.

Open Issues

#235 Nail down specifics of configurable SDP params

JSEP says that the SDP can be changed between createOffer and calling setLocal, with a list of what is allowed in Section 6. To simplify the document, we considered either allowing all logically possible modifications or removing this functionality.

It's no longer clear that this functionality is needed:

- New APIs have been added to control things that previously required SDP munging (e.g. reordering codecs)
- We want to encourage these APIs instead of SDP munging
- Applications that want to go "low-level" can drop down and use WebRTC NV APIs.
- As such, we recommend removing this functionality. Note that remote SDP will of course continue to be changeable.

#72 Maintain DTLS role during SDP renegotiation

Discussion from IETF 92 was that in reoffers, a=setup should contain **active** or **passive**, as appropriate, instead of **actpass**.

Implementations should send active/passive but accept actpass, for compatibility.

Need to add dtls-association-id to generated SDP as well.

#149 Limitation on SSRC change is in violation with RFC7160 regarding RTP PT change

With MIDs + RIDs, we now have the ability to move away from SSRC. We will add a new section to JSEP to indicate how non-SSRC demux should occur.

#223 Specify demux algorithm for RTP

The port, RID, and MID on an incoming RTP packet are used to find the RTP Receiver that uniquely matches this set

An incoming RTP packet may not have some of these values or the SDP may not have specified some of the values; need to document exactly how demux works in these cases.

#162 LipSync groups violating RFC5888

The new RTP Transceiver, Sender, Receiver objects allow an application to explicitly control which things are synchronized, by creating new m= lines if needed.

Applications that receive an offer with two streams that are not lip synced can answer with no streams, and then in a re-offer, create new m= lines that are synced with a LS group.

#206 support explicitly disabling VAD in RTPSender

Even if VAD support is signaled, it may not always be appropriate to use.
This is a similar situation to FEC.
Suggestion is to add a new API to handle this, similar to what was done for FEC, e.g. `RtpEncodingParameters.vad`

#247 Document what should happen when there are no matching codecs in answer

Problem seen in interop between VP8-only and H.264-only implementations.

Proper behavior should be that the specific m= line is rejected by the answerer.

#249 Clarify a=rtcp processing

Since ICE is used, a=rtcp is typically superfluous.

However, we need to understand the consensus for rtcp-mux-exclusive; if we use a=rtcp:0 instead, we need to indicate the meaning of that.

#250 explain how to roll back an ICE restart

This seems complex. Should we even support this?

There could be wire collisions with ICE restarts on both ends, but perhaps there is a simpler solution for this case.

#239 When is the SDP sess-version in o= line updated

One of 3 options:

- Every time createOffer is called
- Every time createOffer is called after a setLocal is ca
- Every time the SDP changes (including when candidates are added by the ICE Agent)

- Proposal: Each time a create offer / answer is called and the SDP is not identical to current local description, the sess-version is changed

#261 document API points for RtpSender/RtpReceiver

The states associated with these APIs are discussed in Section 5, but the APIs themselves need to be documented in the API section (Section 4).