

JSEP

7/23/2015

Changes since -09 (Interim)

- Define imageattr behavior
- Clarify BUNDLE behavior
 - with non-BUNDLE endpoints
 - Balanced
- Stop overriding RTP-usage for CNAME behavior
- Define LS behavior
- Explicitly don't check "c=" line
- Require parsing "b=" line

#162: LS

- Answerer can't add m= lines to LS groups
 - S 9.2 of RFC 5888 prohibits this; can only return a subset
- Problem when caller/callee disagree on sync
 - e.g. if callee wants a/v synced, but caller doesn't
- Could add new m= line in this case, but this complicates m= line assignment algorithm
- **Proposal:** this is an edge case; just document that this is currently unsupported. The callee can accept what caller wanted or reject the call

#158 (imageattr and CVO)

- How do you indicate resolutions for multiple orientations?
 - I support 640x360 in landscape, but if I rotate I support 360x640 in portrait
 - This is what CVO is for, and handles it well
 - But what if I don't have CVO?
- Magnus looked into this from 3GPP angle and you're supposed to interpret a single imageattr as supporting both orientations
- **Proposal:** use a single imageattr always

#122: AS->TIAS conversion

- We previously decided to use TIAS whenever possible, but accept AS.
- Need to define how a JSEP impl should convert an AS value to a TIAS equivalent
- **Proposal:**
$$\text{TIAS} = \text{AS} * 0.95 - \text{packet_rate} * \text{hdr_size}$$
$$\text{packet_rate} = 50$$
$$\text{hdr_size} = 40$$
- If you want more accuracy, set TIAS :-)

#125: BUNDLE and SDP mangling

Can one unbundle things that are bundled by createOffer (i.e. before setLD)?

Technically this would work, but requires up to 3 changes:

- a. change a=group:BUNDLE
- b. remove a=bundle-only
- c. change m=foo 9 to m=foo 1

Proposal: Do not support this

#155: Selecting between multiple aspect ratios

- JSEP clients generate one imageattr value
- Other clients may generate multiple with different $q=$ values; how do we select?
 - e.g. with different aspect ratio requests
- Should we try to avoid all scaling? Preserve aspect ratio? Could get complicated...
- **Proposal:** Implementations SHOULD pick imageattr with highest $q=$ value that doesn't require upscaling

#15: Early Transport Warmup

- Want to get ICE/DTLS started while call ringing; requires answer with ICE/DTLS attribs
- We discourage use of PRANSWER; it leaves the state machine in a non-stable state (S 4.1.4.1)
- Currently recommended mechanism:
 - offer + a=inactive answer to start DTLS during ringing
 - when accepted, a=sendrecv offer to start media
- However, we received comments that this could lead to signaling/media race + clipping
- Is this true? (a=inactive prohibits receiving)

#15: Early Transport Warmup (2)

- Question: can callee send caller media at this before recving second answer?
- Yes, if callee used a=sendonly in initial answer
- But, this would cause callee to start sending media, so callee would have to give a=inactive to setLD, but send a=sendonly to caller
- This is suboptimal

#15: Early Transport Warmup (3)

- **Alternate proposal:**

- a=sendonly in initial answer
 - accomplished by setting
 `RTCRtpReceiver.active = false`
- Once call accepted, start transmitting media
 - accomplished by setting
 `RTCRtpEncodingParams.active = false`
- and ask for incoming media
 - accomplished by setting
 `RTCRtpReceiver.active = true`, and reoffer

#143: Can you call CreateOffer in remote-pranswer

- Argument for
 - Not necessary for createOffer to enforce this state; you can't call setLocal() anyway
 - Behavior of createOffer is unambiguous (same as stable); prohibiting this requires more state checks
- Argument against
 - You can't call setLocal(), and this violates 3264 semantics
- **Proposal:** Simplest option - just allow this

#149: SSRC and clock rate

- Streams are signaled with their SSRCs (a=ssrc)
- But RFC 7190 prohibits changing clock rate within an SSRC
 - Like between Opus and G.711
 - One could re-offer here, but we don't want to re-offer every time we want to change the send codec
- **Proposal:** document the limitation and move on
 - i.e. note that when changing between codecs with different clock rates, you will get wacky RTP timestamps
 - Can revisit moving away from a=ssrc in the future once MID and ESID behaviors are better understood

#169: Multiple a=fingerprint lines

- For hash agility, endpoints may have have multiple certificates with different hash algorithms
- This means we need to generate and verify multiple a=fingerprint lines
- **Proposal:**
 - Generate a=fingerprint line for each certificate
 - Ensure received certificate matches a=fingerprint line with corresponding hash