JSEP

IETF 97

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

Changes since IETF 96

- Lots - but it's in WGLC, go read it from start to finish instead of just looking at what changed

Open Issues

#382 Section 4.2.3: setDirection

- **RtpTransceiver.direction** indicates the 'preferred' direction of a transceiver, i.e. whether the local side wants to send or receive
- However, even when **.direction** == 'sendrecv', the transceiver may not be sending or receiving, due to the direction expressed in the remote offer

- Proposal: add a new **RtpTransceiver.currentDirection** property to indicate the current transceiver state
- **.direction** affects what comes out of createOffer/createAnswer
- **.currentDirection** changes based on the direction attributes given to setLocal/setRemote

#381 Need to detail the effects of changing the candidate pool size mid-session.

- What happens if you change the pool size after setLocal()
  a. iceCandidatePoolSize=4
  b. setLocal(with two m= lines)
  c. You get two candidates
  d. iceCandidatePoolSize=4
  e. What now?
- Proposal:
  a. Setting pool size after setLocal() is a no-op
  b. Optional: actually causes failure (but why?)

#380 MID and RID construction

- Magnus raise concerns about privacy leakage from these IDs
- Existing text:

An "a=mid" line, as specified in [RFC5888], Section 4. When generating mid values, it is RECOMMENDED that the values be 3 bytes or less, to allow them to efficiently fit into the RTP header extension defined in I-D.ietf-mmusic-sdp-bundle-negotiation], Section 11.

- Proposal: tell people to generate MIDs safely (counter, random, etc.)
  a. (same for RID).

#368 Section 3.7: Receiving Simulcast

- Spec currently covers sending Simulcast with open path to future extensions to define receiving. This simplified work and enabled a bunch of the use cases
- Some old text says that simulcast state is not surfaced in API but reporting of the simulcast streams is actually possible via sender/receiver.getParameters().
   a. Will clean up text in draft
- Should we add a way to respond with simulcast to an offer to send simulcast ?
   a. The reason we discouraged this in the current text was that we didn't have a way for the app to get the desired simulcast sizes, but Bernard thinks we might be able to get this info from getParameters()

#367 Section 6: RTP/RTCP routing rules

- Problems observed with PT-based SSRC latching; will be revised
- Bernard pointed out that the current text doesn't handle SSRCs that are encapsulated in RFC5104 FIR/TMMBR FCI messages (note: not the SSRC in the base RTPFB message)
- Unclear whether this is important - can't the RTPFB SSRC be used to route the packet to the right RtpSender?

#366 Need rules to prevent applying an old offer/answer

- We are forbidding changing SDP after createOffer
- However, we're not preventing apps from passing an old offer to setLD
- Apps might try to use this to re-offer resources

Proposal: invalidate any outstanding offers when setLD is called

createOffer already keeps a version count; setLD can store this count and fail any offers than have v= less than the current count

#364 Considering layering issue in RTP/RTCP processing and demux algorithm (Section 6)

- Propose that we add text to explain incoming RTP packets are groups into streams identified by SSRC. Then refactor text here to talk about what happens to stream. Algorithm stays roughly the same, but aligns terminology with RTP.

- Should we move most of section 6 that describes how to de-multiplex the RTP to the bundle draft?
- Pros:
  a. Makes it clear how that other things that do bundle also use this
- Cons:
  a. Scatters text of how to handle this across multiple specification