

JSEP



IETF 94

ekr@rtfm.com

juberti@google.com

fluffy@cisco.com

Changes since IETF 93

- Filled in setLocal and setRemote (*)
- Clarify ICE default candidates during gathering (*)
- Clarify downscaling and upscaling rules
- Update SDP modification rules
- Updated to latest datachannel SDP.
- Allow multiple fingerprint lines
- Dummy candidates use IPv4 (again)

Applying local descriptions (S 5.7)

- Verify no inappropriate SDP modifications
- Loop over m= sections
 - ICE: if new gather, candidates; if ufrag/password changed, trigger restart
 - Process RTCP mux/demux
 - Build RTCP header extension->URL table
 - Build MID demux table
 - Build payload type->media format table
 - Build rtx payload->primary payload table
 - Prepare to send/receive based on direction attributes

Applying remote descriptions (S 5.8)

- ICE
 - If offer & ufrag/password changed, record restart needed
 - Configure ICE components with ufrag/password
 - Pair up ICE candidates
 - Start connectivity checking with any pairs
- RTP
 - Build payload type->media format table
 - Build rtx payload->primary payload table
 - Enable matching fmtp parameters
 - Enable matching RTCP feedback mechanism
 - Set max bandwidth based on TIAS (or AS if no TIAS)
 - If audio set up ptimes for each PT

Applying answers (S 5.9)

- If m= was rejected (port==0) stop transmitting/receiving
- If DTLS fingerprint changed, tear down connection
- Configure DTLS if not already set up
- RTCP
 - If RTCP-muxing, disable RTCP component
 - RTCP-mux to RTP-non-mux transitions are an error
 - Configure reduced-mode RTCP if enabled
- RTP:
 - If sendrecv/sendonly, prepare for transmission once transport set up
- SCTP
 - Prepare for SCTP handshake once transport setup
- Discard ICE components for non-primary bundled m= lines
- Set canTrickle property

Default Candidates

- RFC 5245 Section 4.3 requires a “default” candidate
- With trickle we currently use a “dummy” candidate
 - But what about after some gathering has happened (post setLocal CreateOffers)
- Proposed rules
 - If no candidates gathered, use dummy [existing]
 - If some candidates gathered use “best” [not in draft yet]
 - If any candidate pairs have completed, use the one in use [new]
 - Once ICE is completed, use selected pair [RFC 5245]

Default mux policy

- Text has bundle policy but no mux policy
- Obviously this is some kind of oversight
- Option 1
 - Fix a default.
 - Our recommendation is "require" (PR#183)
- Option 2
 - Just forbid non-MUX entirely
 - Some discussion of this but not clear what WG direction is
- Discuss

#162: RFC 5888 (AGAIN!!!!)

- Proposal at IETF 93:
 - Answerer has to match (media type, sync group) tuple when matching tracks to m= lines
 - This allows offerer and answerer to differ in their views on sync
 - Sync groups maintained on reoffers

#162: RFC 5888 Example

- **Caller (wants to sync its outgoing streams); offer**
m=audio (mid:a)
m=video (mid:v)
a=group:LS a v
- **Callee (no sync); answer**
m=audio (mid:a) recvonly
m=video (mid:v) recvonly
a=group:LS a v (same as caller)
- **Callee reoffer**
m=audio (mid:a) recvonly
m=video (mid:v) recvonly
m=audio (mid:a2) sendonly
m=video (mid:v2) sendonly
a=group:LS a v
- Does this look right?
- Will legacy equipment handle such a reoffer properly?

Report from TPAC: Object Model

- Initial version of API had `addStream`, `setLocal`, `setRemote`
- Then we added `addTrack`, `RtpSender`, `RtpReceiver`
 - Provides per-track control over sending and receiving
- Latest update adds a `RtpTransceiver` object, which models a `m=` section
 - Properties:
 - `MID`
 - `RtpSender`
 - `RtpReceiver`
- App can use this to choose which tracks go in which `m=` sections
 - Or can let browser pick by calling `addTrack`
 - `addTrack` picks the first compatible* `m=` section

Report from TPAC: Remove OfferToReceive*

- Transceivers give direct control of recv-only m= lines in offer
 - OfferToReceive is comparatively clunky compared to addTransceiver
- Consensus: remove OfferToReceive from spec
 - Browsers can continue to support for backcompat
 - Expected to deprecate then remove

Report from TPAC: Rollback and addTrack

- Issue from TPAC: what to do with setRemote(offer), addTrack(), rollback sequence?
 - Problem: what do you do with the track?
 - WG settled on option that keeps the transceiver for the track, but not bound to anything
- What about MIDs?
 - Remember they need to be symmetrical
- Resolution: MIDs can exist in three states
 - **null**: addTrack called but...
 - setLocal not called
 - no paired remote tracks
 - **provisional**: appear in createOffer but setLocal hasn't been called
 - **committed**: either createOffer/setLocal was called or paired with remote track
- Rollback blows away the MID state

Recycling Rules

- The current recycling rules are super-aggressive
 - addTrack() will use an existing m= line if possible
 - Note: transceivers give tight control; **this is only about default behavior**
- Should half-dead m= lines be recycled?
 - addTrack()/createOffer()/setLocal()/setRemote(): **1 sendrecv m= line**
 - removeTrack()/createOffer()/setLocal()/setRemote(): **1 recvonly m= line (half-dead)**
 - addTrack()/createOffer(): **??? m= lines**
- Current JSEP pretty clearly says the answer is yes
 - But this means that previously negotiated parameters (codecs, fmt, b=) are still in-place
 - ... which makes the reoffer weird
 - Would it be better to keep half-dead half-dead?
 - **Proposal: addTrack fills the first compatible port=0 (offerer) or newly proposed (answerer) m= line, otherwise adds one**

Recycling and MIDs

- RTP transceiver objects are uniquely identified by MID.
- When a transceiver is stopped (i.e. m= line rejected), the object is permanently disabled.
- Ergo, when a rejected m= line is recycled, a new transceiver is created, and it (and the m= line) must have a new MID.
- 5888 hints at this being ok, but never spells it out:
 - ... *subsequent offers (e.g., in a re-INVITE) SHOULD use the same "mid" value for **already existing** media streams.*
- Any concerns?

Report from TPAC: IP Address Leakage

- Proposed resolution: four levels of behavior
 - Everything [with consent]
 - Restricted gathering I (default host candidate [+ RFC 1918?]) [default behavior]
 - Restricted gathering II (no host candidates) [via pref, extension]
 - Proxy only [via pref, extension]
- Open issue: what to gather in restricted I
 - Option 1: Just the default host candidate
 - Option 2: Default + all 1918 candidates
 - Pair 1918s with like
 - Still need rules for IPv6 link-locals, etc.
 - Waiting for measurements for Option 1 versus Option 2

Other topics?