

JSEP Update

Justin Uberti & Cullen Jennings
IETF 88

Updates and Open Issues

- Updates
 - Behavior of createOffer/Answer specified
 - Examples of SDP generated by these methods
 - Mandatory-to-use features specified
- Open Issues
 - Re-offer handling
 - a=connection attribute
 - CNAMEs
 - BUNDLE control
 - m= section recycling

Example SDP

```
v=0
o=- 4962303333179871722 1 IN IP4 0.0.0.0
s=-
t=0 0
a=msid-semantic:WMS
a=group:BUNDLE audio video data
m=audio 56500 UDP/TLS/RTP/SAVPF 111 0 8 126
c=IN IP4 192.0.2.1
a=rtcp:56501 IN IP4 192.0.2.1
a=candidate:3348148302 1 udp 2113937151 192.0.2.1 56500 typ host generation 0
a=candidate:3348148302 2 udp 2113937151 192.0.2.1 56501 typ host generation 0
a=ice-frag:ETEnlv9DoTMB9J4r
a=ice-pwd:OtSK0WpNtpUjky4+86js7ZQl
a=ice-options:trickle
a=mid:audio
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=sendrecv
a=rtcp-mux
a=rtcp-rsize
a=fingerprint:sha-1 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70
a=setup:actpass
a=rtpmap:111 opus/48000/2
a=fmtp:111 minptime=10
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:126 telephone-event/8000
a=maxptime:60
a=ssrc:1732846380 cname:EocUG1f0fcg/yvY7
a=msid:47017fee-b6c1-4162-929c-a2511025240 f83006c5-a0ff-4e0a-9ed9-d3e6747be7d9
```

Mandatory-to-Use Features

The document now specifies features that are must-use, in addition to those that are must-implement. Current list is simple:

- ICE
- DTLS-SRTP

Q1: Re-offer Handling

Once the local and remote client agree on session parameters, these don't need to be renegotiated on each offer/answer. Namely:

- Codecs
- RTP Header Extensions
- RTCP Feedback Mechanisms
- BUNDLED m= sections

Proposal: Only re-offer all parameters in new m= sections. For full renegotiation, use a new PeerConnection.

Q2: a=connection Attribute

Current SDP specifies the following attributes for configuring DTLS:

```
a=fingerprint:sha-1 19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04:BB:05:2F:70  
a=setup:actpass
```

Do we also need to specify **a=connection:new** or **a=connection:existing**?

- Could be used to force renegotiation
- RFC 4572 indicates yes (Section 3.4)
- RFC 5763 doesn't show it in its examples

Proposal: Add it; assume 'existing' if absent

Q3: CNAMEs

- CNAMEs are randomly generated, a la RFC 7022, and signaled using `a:ssrc X cname:Y`
- `MediaStreamTracks` in the same `MediaStream` are synchronized, and therefore should share the same CNAME
- How should multiple `MediaStreams` be synchronized, i.e. be told to share a common CNAME?

Proposal: ???

Q4: BUNDLE control

- Want to get maximum benefit from BUNDLE with minimal incompatibility
- Need a way to control which MediaStreamTracks are BUNDLE-only
- **Proposal:** Try to BUNDLE all tracks; for BUNDLE-only, add the BundleOnlyPolicy constraint to PC, which can be either:
 - same-media (port for each media type, **default**)
 - all (all streams on one port)
 - none (each stream on its own port)
- No need to control use of BUNDLE itself

Q5: m= section recycling

Action	Local Description	Remote Description
Initial call, no video	No video m= line	No video m= line
Local side adds video, remote side rejects	m=video X, a=sendonly	m=video 0
Local side adds video remote side accepts	m=video Y, a=sendonly	m=video Y, a=recvonly
Remote side adds video, local side accepts	m=video Y, a=sendrecv	m=video Y, a=sendrecv [what about BUNDLE?]
Local side removes video	m=video Y, a=recvonly	m=video Y, a=sendonly
Remote side removes video	m=video 0	m=video 0
Local side adds video	m=video Z, a=sendonly	m=video Z, a=recvonly