

**The Session Description Protocol (SDP)
Application Token Attribute
draft-even-mmusic-application-token-00
IETF 87**

Roni Even

Jonathan Lennox

Qin Wu



Introduction

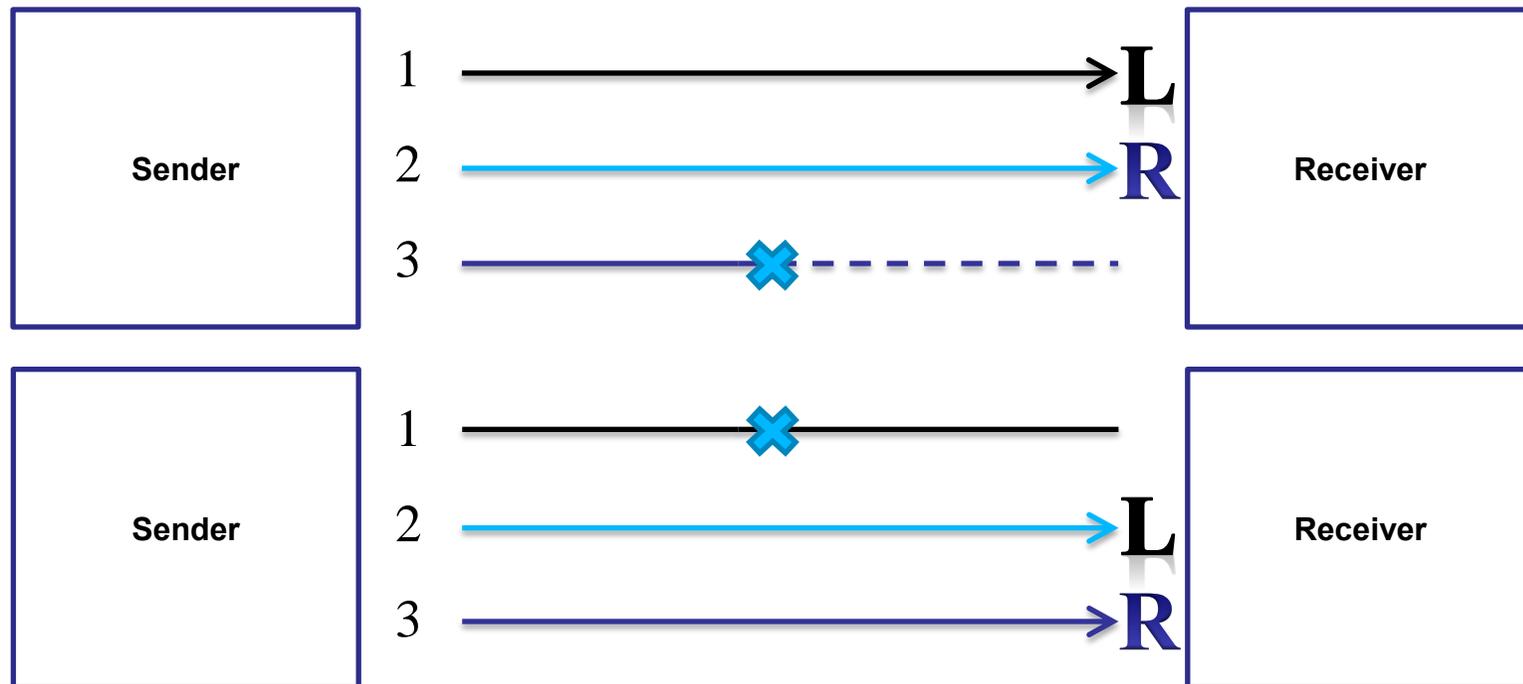
- There is a need to have a token that will allow the mapping between a single source (identified by an SSRC) in an m-line to the application logic.
 - For example, SSRC1 is the RTP stream from the left camera and SSRC2 is the RTP stream from the right camera. Both can be specified in a single SDP m-line (or multiple m-lines bundled into a single RTP session) and may have the same PT number.
- A new RTP stream can replace an existing one (taking over its appID) or be a new application purpose with a new appID, for example a new presentation stream.
 - Support changing which RTP stream maps to an appID without a need for a new RFC3264 offer/answer

Proposal

- Define a token “appID” associated with an RTP stream, allowing the semantics of the stream with a token to be defined by the application.
- The token can be sent using SDP, RTCP SDES messages, or an RTP header extension.
- An application may receive a new RTP stream replacing an existing RTP stream having the same appID, or a new RTP stream with a new appID.
- The appID can be used for an m-line (Plan A / unified)
 - a=appID 2
 - Declare that this appID is associated with this m-line.
- In SDP the appID can be tied to a specific attribute
 - a= appID:1 imageattr:98 send [x=480,y=320]
- Associate an appID with an SSRC using RTCP SDES and/or RTP header extension.

Dynamic appID-to-source mapping

- Example:
 - Sender has three cameras, but receiver has two screens.
 - Sender dynamically switches between sending Cameras 1+2 and sending Cameras 2+3 for receiver's Left and Right.



Dynamic appID-to-source mapping

- In appID terms, this means sometimes we have ssrc:appID as {1:L, 2:R}, sometimes as {2:L, 3:R}. SSRC 2's appID changes.
- Different SSRCs map to an appID, reflecting the source's current application usage.
 - I.e., “render on left display.”
- SSRC still defines synchronization, decoder context, etc., as always.
- In some use cases, (e.g., speaker switching) changes can be frequent – don't want to require an Offer/Answer exchange every time.

Open issues

- When mapping a specific SSRC to an appID in SDP, need to keep consistency when the mapping is changed using SDES or RTP header extension.
 - Propose that RTP always wins (SDES / RTP Header extension)
 - Other option – never use a=SSRC:value appID:token, just use a=appID:token attribute:value example:
 - a=appID 1 imageattr:98 send[x=800,y=640,sar=1.1,q=0.6]
 - a=appID 3 imageattr:98 send [x=480,y=320]
 - Alternate syntax, in Plan A / Unified semantic (single concurrent source per m-line): just declare a single appID per m-line
 - a=appID 1
 - No a=ssrc attribute in this case

Open issues

- Is the appID defined by the sender (as in the current text) or by the receiver (unified plan)?
 - Receiver-chosen fixes an early-media race condition.

Open issue: are header extensions/RTCP reliable?

- Argument's been made that we need SDP as backup, because header extensions and RTCP might be dropped.
- Is this really possible?
 - Possible to place header extensions such that if packets carrying them are lost, stream is useless anyway (e.g. on I-Frames).
 - Or just always send them, if you're paranoid.
 - Middleboxes that strip header extensions and RTCP – but don't otherwise interfere in RTP – seem very unlikely.
 - Remember these are multi-SSRC sessions, and probably SRTP encrypted.
 - Support for this mechanism is negotiated, so can always be negotiated off. If middleboxes participate in signaling, there's no problem.

Open issue: multiple appIDs for the same stream?

- Receiver requested (e.g.) a view of the loudest speaker, and a view of Jonathan
- Right now Jonathan is the loudest speaker
- Do we want to support sending Jonathan's RTP stream with two appID values (while he is speaking) to indicate that this stream satisfies both requests?
 - Might require tweaks to how we represent appID in header extensions and/or SDES values.