# The Session Description Protocol (SDP) Application Token Attribute
# draft-even-mmusic-application-token-02
# IETF 89

Roni Even

Jonathan Lennox

Qin Wu

# Introduction

- SSRC identifies an RTP stream in an RTP session.
    - SSRC as an RTP parameter is managed in the RTP layer.
    - SSRC may change in the RTP during a session.
    - SSRC is chosen by sender.
- In SDP media streams are represented in m-lines.
    - An m-line can represent a media source (unified) with multiple packet streams (e.g. simulcast, FEC).
    - Multiple m-lines can be bundled, to allow multiplexing of multiple media sources in a single RTP session.
- Need to be able to disambiguate packet streams within RTP session, to map received packet streams to bundled m-lines and packet streams within an m-line.
    - But using SSRC can be problematic, due to its characteristics above.

# Application token

- Define a token “appId” providing a layer of indirection between signaling and RTP layer.
- Values for appId can be chosen by either sender or receiver (in offer/answer negotiation).
- The binding between appId and SSRC is done using an RTP header extension and RTCP SDES. May also be done in SDP, but RTP always wins.
- AppId-to-SSRC association can change dynamically, indicating that a new SSRC (packet stream) now fills the role previously provided by a different one, without needing new signaling.
- AppId SDP syntax is essentially the same as a=ssrc syntax, with a few extensions (that a=ssrc would probably need anyway).
- The name “appId” is poor (confusing, too generic) – better suggestions welcome!

# Application token: SDP syntax

- The appId can be used for an m-line
  - a=appId:2
    - Declare that this appId is associated with this m-line, with no further semantics
- The appId can describe characteristics of the packet streams it’s associated with
  - a=appId:1 imageattr:98 send [x=480,y=320]
  - a=appId:2 imageattr:98 send [x=240,y=160]
- An SDP endpoint can dictate the appId to be used by its offer/answer peer
  - a=recv-appId:3
- Relationships among appId-associated packet streams can be expressed
  - a=appId-group:FEC-FR 1000 2000
- Can indicate that an appId will only use a subset of m-line’s payload types
  - a=appId: 1 pt=97
  - Useful for Unified: e.g., where m-line indicates both primary media and repair

# Application token: Bundled example

```
a=group:BUNDLE m1 m2
m=video 49200 RTP/AVP 98
a=rtpmap:98 H264/90000
a=mid:m1
a=content:main
a=appId:2
a=recv-appId:10
m=video 49200 RTP/AVP 98
a=rtpmap:98 H264/90000
a=mid:m2
a=content:alt
a=appId:3
a=recv-appId:20
```

# Application token: SVC with FEC example

```
m=video 56602 RTP/AVPF 100 101 110 111 - Main camera
 a=rtpmap:100 H264/90000 - Base layer
 a=rtpmap:101 H264-SVC/90000 - Enhancement layer.
 a=depend:101 lay L1:100 - dependencies
 a=rtpmap:110 1d-interleaved-parityfec/90000
 a=fmtp:110 L=5; D=10; repair-window=200000
 a=rtpmap:111 1d-interleaved-parityfec/90000
 a=fmtp:111 L=10; D=10; repair-window=400000
 a=appId:1000 pt=100
 a=appId:1010 pt=101
 a=appId:2110 pt=110
 a=appId:2120 pt=111
 a=appId-group:FEC-FR 1000 2110
 a=appId-group:FEC-FR 1000 1010 2120
 a=appId-group:DDP 1000 1010
```

# Advantages of application token

- Leaves SSRC values to the RTP stack.
  - Robust to SSRC collision
  - Keeps protocol layering cleaner – don’t need to know SSRC when making an offer
- Avoids early-media race conditions
  - SSRC values can only be specified by a sender
- Allows dynamic mappings between sources
  - E.g., loudest-speaker switching
  - AppId moves from one source to another
  - E.g., “Selective Forwarding Middlebox” RTP topology

# Relationship between AppId and MSID

- None – they are orthogonal, solving different problems
- MSID identifies a media source: there will be only a single MSID per m-line in the Unified plan.
- AppId identifies a packet stream: a single m-line in Unified can have multiple appId values.

# Next steps

- New name for appId
- Propose adopting as WG document.