

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

Roni Even

Jonathan Lennox

Qin Wu



Introduction

- RFC3550 allows multiplexing of multiple RTP streams in a single RTP session.
 - SDP support for multiplexing is specified in MMUSIC.
- The RTP fixed header includes the payload type number and the SSRC values of the RTP stream.
- There are proposals that allow describing one or more RTP streams in a single SDP m-line. The SDP a=ssrc and a=max-ssrc attributes can be used to describe such cases.
- Some applications require more information about the usage of the RTP streams. For example, RTP streams from different cameras that need to be identified by the application in order to render them correctly.

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 (Source may be a single RTP stream identified by a unique SSRC). 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.

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 or for a specific SSRC
 - a=appID 2
 - Declare that this appID is associated with this m-line.
 - a=appID:2 imageattr:98 send [x=480,y=320] recv *
 - Associate appID with application attribute in SDP.

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.
 - 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.
- In some use cases, (e.g., speaker switching) this can be frequent – don't want to require an Offer/Answer exchange every time.

appID in CLUE

- If MMUSIC decides to go with unified plan (one source per m-line):
 - Encodings are represented in SDP as m-lines, as in draft-hansen-clue-sdp-interaction and draft-kyzivat-clue-signaling-03.
 - M-lines are bundled, and can be disambiguated by a receiver by SSRC or by App-ID.
- In O/A the receiver defines the appID
- appID identifies an SDP m-line
- SDP m-line (identified by label attribute) is mapped to CLUE encoding ID (individual encode) in advertisement.
- Consumer's configuration maps the provider's captures to its encodings as normal.
- Provider can change the SSRC for a switched capture just by attaching the appropriate appID to a new source.

appID in CLUE

Example of the provider's SDP

```
a=group:BUNDLE mid1 mid2
a=extmap:1 urn:ietf:params:rtp-hdext:app-id
m=video
a=sendonly
a=mid:mid1
a=label:video1
m=video
a=sendonly
a=mid:mid2
a=label:video2
```

appID in CLUE

The consumer answers with matching SDP, adding a=app-id values:

```
a=group:BUNDLE mid1 mid2
a=extmap:1 urn:ietf:params:rtp-hdrext:app-id
m=video
a=recvonly
a=mid:mid1
a=app-id:ID1
m=video
a=recvonly
a=mid:mid2
a=app-id:ID2
```