MMUSIC WG
Internet-Draft
Intended status: Informational
Expires: March 19, 2014

R. Even
Huawei Technologies
J. Lennox
Vidyo
Q. Wu
Huawei Technologies

September 15, 2013

The Session Description Protocol (SDP) Application Token Attribute
draft-even-mmusic-application-token-01.txt

Abstract

The RTP fixed header includes the payload type number and the SSRC values of the RTP stream. RTP defines how to de-multiplex streams within an RTP session, but in some use cases applications need further identifiers in order to identify the application semantics associated with particular streams within the session.

This document defines a mechanism to provide the mapping between the SSRCs of RTP streams and the application semantics by defining extensions to RTP and RTCP messages.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 19, 2014.

Table of Contents

1. Introduction

The RTP [RFC3550] header includes the payload type number and the SSRC values of the RTP stream. RTP defines how to de-multiplex streams within an RTP session, but in some use cases, applications need further identifiers in order to identify semantics associated with particular streams within the session.

SDP [RFC4566]can be used to describe multiple RTP media streams in one or more m-lines that define a single SSRC multiplexed RTP session (as specified in [RFC3550]). This addresses the WebRTC architecture [I-D.ietf-rtcweb-overview].

A Unified Plan for Using SDP with Large Numbers of Media Flows [I-D.roach-mmusic-unified-plan] proposes that each m-line will represent a media source [I-D.lennox-raiarea-rtp-grouping-taxonomy]. In the simple case a media source will be one video or audio RTP stream. Media source description becomes more complicated when for robust applications, techniques like RTX and FEC are used to protect media. Also simulcast/layered coding can be used to provide support to heterogeneous receivers. In these cases a media source may send more than one RTP stream, for example, a video stream and a FEC stream.

Some applications may 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, or a source that can send multiple versions of the same stream in different resolutions (Simulcast [I-D.westerlund-avtcore-rtp-simulcast]).

SDP provides in [RFC4574] a "label" attribute that contains a token defined by an application and is used in its context. "Label" can be attached to m-lines in multiple SDP documents allowing the application to logically identify the media streams across SDP sessions when necessary. The "label" attribute is a token and does not provide any information about the content of the stream. [RFC4796] defines the "content" attribute providing information about the content of the stream, currently there is a small set of values for the content attribute.

Both "label" and "content" attribute are SDP media-level attributes, so when an SDP m-line supports multiple RTP streams, this value is applicable to all RTP streamsdescribed by the SDP m-line.

There is a need to have a token that will allow the mapping between a single RTP streams(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 specifiedand SSRC3 is the FEC stream that protect both streams. Note that there are cases where the SSRCs of the RTP streams are not known or may change during the call..

Support of FEC, SVC and simulcast bring more requirements as explained using the following examples.

The first example is of a unified plan [I-D.roach-mmusic-unified-plan] offer of one audio source and one video source. The video source includes two SVC RTP streams a base layer and an enhancement layer. There are also two FEC options:

>Base layer S1 is protected by FEC repair stream R1

Base Layer S1 and Enhancement S2 layers protected by FEC repair stream R2.

This enables the answer to select the base layer with R1 or the Base + enhancement layers both protected by R2.

SDP Offer:

```
v=0

o=- 20518 0 IN IP4 198.51.100.1

s=FEC Grouping Semantics for SSRC Multiplexing

t=0 0

c=IN IP4 203.0.113.1

a=group:BUNDLE m1 m2

m=audio 56600 RTP/SAVPF 0 109

a=msid:ma ta

a=mid:m1

a=ssrc:53280

a=rtpmap:0 PCMU/8000

a=rtpmap:109 opus/48000

m=video 56602 RTP/AVPF 100 101 110 111 - Main camera

a=msid:ma tb

a=mid:m2

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=ssrc:1000 cname:MSTFEC@example.com

a=ssrc:1010 cname:MSTFEC@example.com

a=ssrc:2110 cname:MSTFEC@example.com

a=ssrc:2120 cname:MSTFEC@example.com

a=ssrc-group:FEC-FR 1000 2110

a=ssrc-group:FEC-FR 1000 1010 2120

a=ssrc-group:DDP 1000 1010
```

In this case all video streams are from the same source and can be described using a single m-line. The grouping relations are specified using the SSRCs values that need to be available in the offer. It is also not clear based on the offer which SSRC is mapped to each of the PT numbers.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119[RFC2119] and indicate requirement levels for compliant RTP implementations.

3. Proposal for Application ID

As we saw in the previous section, there are tokens defined that could be used for the mapping, but they have existing usages and semantics, and tend to apply at media-level or session level rather than stream-level. In order to avoid overload of existing attributes, it is better to have a new token attribute that can identify a specific RTP stream corresponding to the application. This document defines such new token, "AppID".

[I-D.roach-mmusic-unified-plan] describes a use case where for early media it is important that the offer will include a token allowing the media receiver to associate it with the correct m-line. This requires that the appID will be the token of the received RTP stream to be used by the sending side. On the other hand to specify the

appIds of the source RTP stream and the protecting RTP streams there may be a need to specify the sent appId since the relations between the source and repair streams are for the send side and the protection may not be symmetrical. Similar issue may exist for the simulcast use case. This requires having a second optional attribute for the recv-appId to be used for early media.

3.1. appID token

AppID is a general-purpose token associated with an RTP stream, allowing the semantics of the stream with a token to be defined by the application. This token may also be mapped, for example, to a FEC stream , or to a specific resolution in a simulcast application described in the SDP .

The token is chosen by the sender, and represents the RTP stream that will be sent to the receiver.

The proposed token can be sent using SDP, RTCP SDES messages [RFC3550], or an RTP header extension [RFC5285]

The SSRC mapping may be available to the receiver when receiving the RTP stream through the RTP header extension, but may also be available ahead of time via an RTCP SDES message conveyed before the source started sending, even if the receiver has not seen any RTP packets from this source like in a multipoint conference or in the SDP description.

The receiver can receive new sources that may be of two kinds.

- A new RTP stream replacing an existing RTP stream, in which case the AppID of the replaced RTP stream will be assigned to the new SSRC.

- A new RTP stream requiring a different AppID, for example, when adding a presentation stream to an existing call with two video cameras from a room.

The solution supports an RTP session as described using SDP. The RTP session may use Bundle [I-D.ietf-mmusic-sdp-bundle-negotiation]with more than one m-lines. In this case, if the SSRCs of all RTP streams are not known in advance, the AppIDs associated with each m-line need to be available to the media receiver in order to map each SSRC to a specific m-line configuration.

The document defines a new SDP media level attribute a=appID that can be used to list all the appIDs that an application may use.

The appID syntax provides a token identifier and optional SDP attributes that describe the application usage if exists in SDP. Application usage in SDP may be, for example, an image attribute describing a simulcast application usage [I-D.westerlund-avtcore-rtp-simulcast] or a FEC stream that protects multiple RTP streams.

Each value of the AppID maps to one SSRC at a time. When a new SSRC is mapped to an existing AppID using an RTP header extension or SDES message, it replaces the previous RTP stream for this application usage.

The formal representation of the appID token is:

```
appid-attribute = "appID:" token [SP attribute]

; The base definition of "attribute" is in [RFC4566].

; (It is the content of "a=" lines.)
```

Examples:

The SSRC of the stream is not known when the SDP offer is sent, an appID is specified and can be used for mapping to specific SSRCs in the application.

```
m=video 49200 RTP/AVP 99

a=rtpmap:99 H264/90000

a=appID:2
```

The second example is when the application usage of the RTP steam is specified using SDP to provide different image resolutions. The media receiver can map the received SSRC to the specific resolution based on the appId.

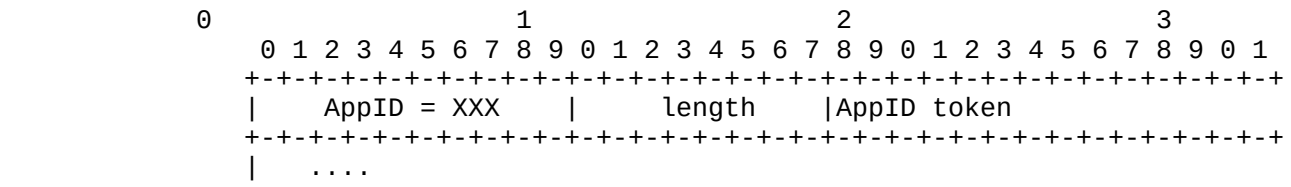
Note:This example is using a separate m-line for each offered resolution on the send direction grouped using SCR option [I-D.westerlund-avtcore-rtp-simulcast] It uses the same msid for all grouped image attribute. Other options will be added based on the work done on [I-D.westerlund-avtcore-rtp-simulcast]

```
a=group:SCR 1 2

m=video 49200 RTP/AVP 98

a=rtpmap:98 H264/90000
```

```
imageattr:98 send [x=640,y=360]  recv[[x=640,y=360] [x=320,y=180]

a=msid:ma ta

a=appID:2

a=mid:1

m=video 49200 RTP/AVP 99

a=rtpmap:99 H264/90000

imageattr:99 send [x=320,y=180]

a=msid:ma ta

a=appID:3

a=mid:2

a=sendonly
```

3.1.1. RTCP SDES message

The document specify a new RTCP SDES message



This AppID is the same token as defined in the new SDP attribute and will also be used in the RTP header extension.

This SDES message MAY be sent in a compound RTCP packet based on the application need.

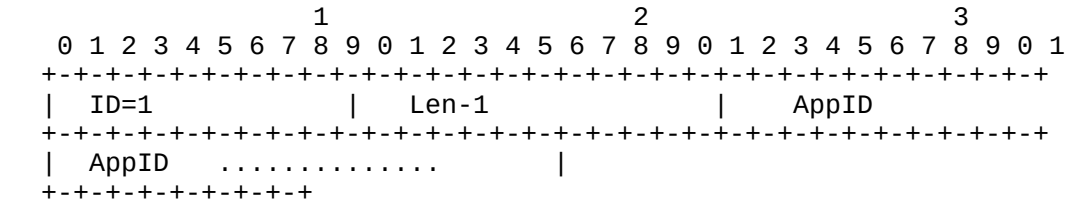
3.1.2. RTP Header Extension

The Application ID could be carried within the RTP header extension field, using [RFC5285] two bytes header extension.

This is negotiated within the SDP i.e.

```
a=extmap:1 urn:ietf:params:rtp-hdrext:App-ID
```

Packets tagged by the sender with the AppID will then contain a header extension as shown below



To add or modify the AppID by an intermediary can be an expensive operation, particularly if SRTP is used to authenticate the packet. Modification to the contents of the RTP header requires a re-authentication of the complete packet, and this could prove to be a limiting factor in the throughput of a multipoint device.

There is no need to send the AppID header extension with all RTP packets. Senders MAY choose to send it only when a new SSRC is sent, or when an SSRC changes its association to an AppID. If such a mode is being used, the header extension SHOULD be sent in the first few RTP packets to reduce the risk of losing it due to packet loss. For codecs with decoder refresh points (such as I-Frames in video codecs), senders also SHOULD send the AppID header extension along with the packets carrying the decoder refresh.

3.1.3. recv-appID

An offer may include a recv-appID attribute allowing the offerer to request from the answerer to use this token for the RTP stream sent from the answerer for a sendrecv or recvonly RTP stream. This is important in order to support early media from the answerer that may be received by the offerer before the answer SDP arrives.

The formal representation of the appID token is:

```
appid-attribute = "recv-appID:" token

; The base definition of "attribute" is in [RFC4566].

; (It is the content of "a=" lines.)
```

4. Using Application ID token in Offer / Answer

The appId may be used in offer answer. Some use cases are provided. They only show part of the SDP that can demonstrate the usage.

the simple case is when each media source describes one RTP stream. In this case the SSRC may be used for the mapping if known but having appId address the case where the SSRC changes. The recv-appID is offered to allow for early media synchronization.

The offer is:

```
m=video 49200 RTP/AVP 99

a=rtpmap:99 H264/90000

a=appID 2

a=recv-appId 10

a=ssrc:20010 CNAME:v1@example.com

m=video 49200 RTP/AVP 100

a=rtpmap:100 H264/90000

a=appID 3

a=recv-appId 20

a=ssrc:20010 CNAME:v2@example.com
```

In this example a three camera system sending three RTP streams protected by a single FEC stream. (note that the full offer may also include a FEC stream for each of the three RTP streams and the answerer may choose which FEC scheme he prefers).

This is the SDP offer for the video sources:

```
v=0

o=- 20518 0 IN IP4 198.51.100.1

s=FEC Grouping Semantics for SSRC Multiplexing

t=0 0

c=IN IP4 203.0.113.1
```

```
a=group:BUNDLE m1 m2 m3 m4 R1
a=group:FEC-FR m2 m3 m4 R1
m=audio 56600 RTP/SAVPF 109
a=mid:m1
a=msid:ma ta
a=appID 1
a=ssrc:53280
a=rtpmap:109 opus/48000
m=video 56602 RTP/AVPF 100 - left camera
a=mid:m2
a=msid:ma tb
a=appID 2
a=rtpmap:100 H264/90000
a=ssrc:1000 cname:MSTFEC@example.com
m=video 56602 RTP/AVPF 101- Middle camera
a=mid:m3
a=msid:ma tc
a=appID 3
a=rtpmap:101 H264/90000
a=ssrc:1010 cname:MSTFEC@example.com
m=video 56602 RTP/AVPF 102 - Right camera
a=mid:m4
a=msid:ma td
a=appID 4
```

```
a=rtpmap:102 H264/90000

a=ssrc:1020 cname:MSTFEC@example.com

m=video 56602 RTP/AVP 110

a=rtpmap:110 1d-interleaved-parityfec/90000

a=fmtp:110 L=5; D=10; repair-window=200000

a=mid:R1

a=appID 5
```

The FEC stream is specified in a separate SDP m-line even though it is not a media source but it does not have any msid so it is not a media stream track. The appID is used to identify this stream as the FEC stream

In the CLUE WG case the mapping is from a media source represented by an SDP m-line to a CLUE Capture encoding specified in the CLUE framework [I-D.ietf-clue-framework]. The mapping may be done using the label attribute.

Example of an offer that offers three CLUE individual encodes. The CLUE config message can be used to map an individual encode to a CLUE media capture [I-D.kyzivat-clue-signaling]. The label value is used in the CLUE protocol to identify CLUE individual encodes. The appId is used to identify the stream by the receiver:

```
a=group:CLUE 4 5 6

...

m=video 6002 RTP/AVP 96

a=rtpmap:96 H264/90000

a=fmtp:96 profile-level-id=42e016;max-mbps=108000;max-fs=3600

a=sendrecv

a=mid:2

a=appID 9

...
```

```
m=video 6002 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42e016
a=sendonly
a=mid:4
a=appID 8
a=label:enc1
m=video 6002 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42e016
a=sendonly
a=mid:5
a=appID 7
a=label:enc2
m=video 6002 RTP/AVP 96
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42e016
a=sendonly
a=mid:6
a=appID 6
a=label:enc3
```

5. Acknowledgements

Place Holder

6. IANA Considerations

TBD

7. Security Considerations

TBD.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, July 2008.

8.2. Informative References

[I-D.ietf-clue-framework] Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-10 (work in progress), May 2013.

[I-D.ietf-mmusic-msid] Alvestrand, H., "Cross Session Stream Identification in the Session Description Protocol", draft-ietf-mmusic-msid-00 (work in progress), February 2013.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-bundle-negotiation-04 (work in progress), June 2013.

[I-D.ietf-rtcweb-overview] Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-06 (work in progress), February 2013.

[I-D.kyzivat-clue-signaling] Kyzivat, P., Xiao, L., Groves, C., and R. Hansen, "CLUE Signaling", draft-kyzivat-clue-signaling-04 (work in progress), July 2013.

[I-D.lennox-raiarea-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., and G. Salgueiro, "A Taxonomy of Grouping Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-lennox-raiarea-rtp-grouping-taxonomy-01 (work in progress), July 2013.

[I-D.roach-mmusic-unified-plan]
Roach, A., Uberti, J., and M. Thomson, "A Unified Plan for Using SDP with Large Numbers of Media Flows", draft-roach-mmusic-unified-plan-00 (work in progress), July 2013.

[I-D.westerlund-avtcore-rtp-simulcast]
Westerlund, M., Lindqvist, M., and F. Jansson, "Using Simulcast in RTP Sessions", draft-westerlund-avtcore-rtp-simulcast-02 (work in progress), February 2013.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.

[RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", RFC 4796, February 2007.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

Authors' Addresses

Roni Even
Huawei Technologies
Tel Aviv
Israel

Email: roni.even@mail01.huawei.com

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Qin Wu
Huawei Technologies

Email: bill.wu@huawei.com