MMUSIC WG
Internet-Draft
Intended status: Informational
Expires: July 7, 2014

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

January 3, 2014

The Session Description Protocol (SDP) Application Token Attribute
draft-even-mmusic-application-token-02.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 as conveyed in the signaling.

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.

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.ietf-avtext-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 retransmission (RTX) and Forward Error Correction (FEC) are used to protect media, or simulcast or 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 scalable video stream base layer, an enhancement layer and a FEC stream.

Multiple SDP m-lines can be multiplexed to a single RTP session using [I-D.ietf-mmusic-sdp-bundle-negotiation]. The same payload type number can be used in multiple bundled m-lines.

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 (i.e. simulcast [I-D.westerlund-avtcore-rtp-simulcast]).

A single RTP media stream can be identified in SDP by using the SSRC attribute [RFC5576]. Relations between RTP streams in the same session can be specified using the ssrc-group attribute [RFC5576]. Using the SSRC to identify RTP streams in an SDP session assumes that this information is available to the SDP signaling layer. The SSRC is RTP layer information and may change in the RTP layer during a session.

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

The following 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 layer S2 are 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.

This example uses the SSRC and SSRC-GROUP attributes which requires the pre-knowledge of the SSRCs that are RTP layer values.

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 rtpmap line corresponds to each of the a=ssrc lines, e.g. which rtpmap line will be sent using ssrc = 2110. The answerer can deduce the information based on analyzing the ssrc-group information but there can be case that it will not be possible..

There are cases where the SSRCs of the RTP streams may not be available.

A selective forwarding middlebox as described in RTP topologies section 3.7 [topologies] may project the RTP stream from the source to destination and forward new SSRCs without any signaling.

A three camera telepresence system may send two video media stream of the two recent active speakers to a system with two monitors. In this case it may send first the video from the left and center camera (this will cause the video from the center camera to be displayed on the right) and later the video from the center and right camera (this will cause the video from the center camera to be displayed on the left). The SSRC of the video stream from the center camera will remain the same but the mapping to the stream description will change.

As discussed in [I-D.roach-mmusic-unified-plan] during call establishment, circumstances may arise under which an endpoint can send an offer to receive a new stream, and begin receiving media for that media stream prior to receiving the SDP that correlates its SSRC to the m-line. For such cases, the endpoint will not know how to handle the media, and will most probably be forced to discard it.

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 introduction, the SSRC of the RTP stream which should be created by the RTP layer is used by the SDP in the offer to map an RTP stream description to the SDP or application logic. There are cases where the SSRCs of the RTP streams may not be available or do not provide all the required information.

There is a need to have a token that will allow the mapping between a single RTP stream in an m-line to the application logic and to the actual RTP media stream. For example, stream1 is the RTP stream from the left camera and stream2 is the RTP stream from the right camera and stream3 is the FEC stream that protects both streams.

This token can be used for defining group semantics inside an SDP m-line. There is also a need for a token that will allow the offerer to correlate a received RTP stream to the application logic before receiving the answer from the remote side.

In order to address these requirements this document defines an SDP token attribute appId that provides a level of indirection for the binding. The actual binding is done in RTP by associating the appId with an SSRC using a new RTCP SDES message and a new RTP header extension that define the mapping from appId to a specific SSRC. Having the binding in RTP/RTCP allows the RTP layer to change the SSRC of a media stream by sending a new binding message (SDES an RTP header extension) without a need to have an SDP level offer/answer exchange.

The document also defines an appId-group attribute that has similar semantics to SSRC-group but uses the appId instead of SSRC to specify the different RTP streams in the group.

For the case when the offerer receives an RTP stream before the SDP answer, we define a new optional attribute recv-appId to be used for correlating this received RTP stream.

3.1. appId token

AppId is a general-purpose token associated with an RTP stream, allowing the binding of the SDP representation to an SSRC. This allows the semantics of the stream with the token to be defined by the application and mapped to an RTP stream without having to know its SSRC in the application.

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 starts sending, even if the receiver has not seen any RTP packets from this source (as in a multipoint conference).

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

o 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.
o 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-line. 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 appIds MUST be unique in an SDP session.

Editor Note (is this required?): It is preferable that they will be unique in an RTP session which is not a problem in a point to point call or in a multipoint conference with a central signaling point.

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. 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 definition is

a=appId:token

a=appId:token <attribute>

a=appId:token <attribute>:<value>

The formal representation of the appId token is:

```
appId-attribute = "appId:" token *[WSP attribute]
attribute =/ appId-attr
; The base definition of "attribute" is in [RFC4566].
; (It is the content of "a=" lines.)
; WSP and DIGIT defined in [RFC5234]
; token defined in [RFC4566]
```

Examples:

The SDP offer specifies an appId that will be used for mapping to

specific SSRCs. The example shows two RTP streams having different content [RFC4796] with the same payload type number. The appId can be used to identify the content of the RTP stream.

```
a=group:BUNDLE m1 m2
m=video 49200 RTP/AVP 99
a=rtpmap:99 H264/90000
a=mid:m1
a=content:main
a=appId:2
m=video 49200 RTP/AVP 99
a=rtpmap:99 H264/90000
a=mid:m2
a=content:alt
a=appId:3
```

The second example is when the application usage of the RTP stream is specified using SDP to specify different content [RFC4796] , and each RTP stream has a Retransmission stream. The media receiver can map the received SSRC of the RTP stream or the retransmission to the specific content based on the appId.

```
a=group:BUNDLE m1 m2
m=video 49200 RTP/AVP 97,98
a=rtpmap:98 H264/90000
a=mid:m1
a=content:main
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=98;rtx-time=3000
a=appId:2
a=appId:3
m=video 49200 RTP/AVP 97,98
a=rtpmap:98 H264/90000
a=mid:m2
a=content:alt
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=98;rtx-time=3000
a=appId:4
a=appId:5
```

3.1.1. RTCP SDES message

This document specifies a new RTCP SDES message

```
0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    AppId = XXX    |     length    |AppId token
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   ....
```

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

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

Editor Note: Need guidance on how often the SDES message should be sent.

3.1.2. RTP Header Extension

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

Support 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 then contain a header extension as shown below

```
0                      1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  ID=1             |   Len-1              |    AppId
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  AppID   .............       |
     +-+-+-+-+-+-+-+-+-+
```

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.)
```

3.2. The "appId-group" media attribute

a=appId-group:<semantics> <appId> ...

The SDP media attribute "appId-group" expresses a relationship among several media sources specified in the same SDP m-line. It is analogous to the "group" session-level attribute [RFC3388], which expresses a relationship among media streams in an SDP multimedia session (i.e., a relationship among several logically related RTP sessions). As media sources are already identified by their appId, no analogous property to the "mid" attribute is necessary.

Editor note: Since the appId is unique in an SDP session the app-Id group can be used also at the session level - do we want it?

The <semantics> parameter is taken from the specification of the "group" attribute [RFC3388]. The initial semantic values defined for the "appId-group" attribute are FID (Flow Identification) [RFC3388] and FEC (Forward Error Correction) [RFC4756]. In each case, the relationship among the grouped sources is the same as the relationship among corresponding sources in media streams grouped using the SDP "group" attribute.

Though the "appId-group" semantic values follow the same syntax as "group" semantic values, they are defined independently. All "appId-group" semantic values MUST be registered with IANA, using the registry defined in Section 6

The "appId-group" attribute indicates the sources in a group by listing the appIds of the sources in the group. It MUST list at least one appId for a group and MAY list any number of additional ones. Every appId listed in an "appId-group" attribute MUST be

defined by a corresponding "appId" line in the same media description.

3.3. appId attributes

3.3.1. The "pt" appId attribute

The SDP offer example in the introduction demonstrated that when there are multiple RTP streams in the offer each have a different pt number it it is not clear which SSRC specified using a=ssrc: is correlated to each of the rtpmap lines. In order to provide the mapping we define an appId attribute "pt".

```
a=appId:token pt:value

appId-attrib = "pt:" pt-value

pt-value = 1*3DIGIT
```

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.

A simple case is when each SDP m-line describes one RTP stream and the m-lines are bundled . The recv-appId is offered so when the offerer sees an RTP stream with appId token value 10 it knows it is the main video.

The offer is:

```
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 99
a=rtpmap:99 H264/90000
a=mid:m2
a=content:alt
a=appId:3
a=recv-appId:20
```

A second example is using the same case as in section one (SVC with FEC) This example shows how to use the appId optional pt parameter to map to a specific stream description.

```
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=mid:m1
a=rtpmap:0 PCMU/8000
a=rtpmap:109 opus/48000
a=appId:10
m=video 56602 RTP/AVPF 100 101 110 111 - Main camera
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=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
```

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.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

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

8.2. Informative References

[I-D.ietf-avtext-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-ietf-avtext-rtp-grouping-taxonomy-00 (work in progress), November 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-05 (work in progress), October 2013.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-08 (work in progress), September 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-03 (work in progress), October 2013.

[RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

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

[RFC4756] Li, A., "Forward Error Correction Grouping Semantics in Session Description Protocol", RFC 4756, November 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