MMUSIC WG R. Even
Internet-Draft Huawei Technologies
Intended status: Informational J. Lennox
Expires: October 13, 2014 Vidyo
Q. Wu
Huawei Technologies
April 11, 2014

The Session Description Protocol (SDP) Application Token Attribute

draft-even-mmusic-application-token-03.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 signaling descriptions associated with particular RTP media streams.

This document defines a mechanism to provide the mapping between the SSRCs of RTP streams and the SDP m-line description by defining extensions to RTP and RTCP messages.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Terminology 4
3. Proposal for Application ID 4
  3.1. appId token 5
    3.1.1. RTCP SDES message 7
    3.1.2. RTP Header Extension 8
    3.1.3. recv-appId 8
4. Using Application ID token in Offer / Answer 9
5. Other Considerations 10
6. Acknowledgements 12
7. IANA Considerations 12
8. Security Considerations 12
9. References 12
  9.1. Normative References 12
  9.2. Informative References 13
Authors' Addresses 14

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 map an RTP media stream to its SDP m-line description.

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. An [RFC3264] offerer may receive an RTP media stream before the SDP answer, and if the same payload type number is used in multiple bundled m-lines, the offerer will not be able to associate incoming media using that pt number to a specific m-line.

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 selective forwarding middlebox as described in RTP topologies section 3.7 [topologies] may project the RTP stream from the source to the 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 description. 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 configuration in an m-line 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.

There is also a need for a token that will allow the offerer to correlate a received RTP stream to an SDP m-line 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.

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.

Note: the name appId does not represent the token functionality very well. We are looking for a better name (SSID source stream ID was proposed on the mailing list but is also well know for wifi networks)

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.

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]
```

AppId attributes are not defined in this specification but are provided for future extensibility. (TODO: define an IANA registry for them.)

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
```

In this example, the SDP signaling does not provide enough information to establish which appId value will be used for the H.264 encoded stream, and which will be used for the RTX retransmission stream. However, it does establish the relationships among the streams identified by appId values, allowing the receiver to properly associate the related streams once it receives them.

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:AppId

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 answerer should use the recv-appId as the token in the RTCP SDES and RTP header extension for the RTP stream sent to the offerer.

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.)
```

An example of an offer using bundle with two video streams using the same payload type number:

```
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
a=recv-appId:4
m=video 49200 RTP/AVP 99
a=rtpmap:99 H264/90000
a=mid:m2
a=content:alt
a=appId:3
a=recv-appId:5
```

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

An offerer sending a BUNDLE group MUST indicate at least one recv-appId for every RTP m=line in the group on which it could receive media (i.e., for every m-line which is marked a=sendrecv (possibly implicitly) or a=recvonly. On m-lines whose configuration is such that multiple packet streams are expected to be sent simultaneously (e.g., one with rtx or fec payload types configured), the offerer MUST specify as many recv-appId values as the number of simultaneous packet streams.

Additionally, an offerer sending a BUNDLE group MUST indicate at least one appId for every m= line on which it expects to send media (i.e., every a=sendrecv or a=sendonly m= line), and MUST send multiple appIds for m= lines on which it expects to send multiple packet streams simultaneously.

An answerer, receiving an offer containing appId or recv-appId attributes, MUST respond with mirrored recv-appId and appId values for the subset of m= lines and packet streams indicated in the answer, maintaining the same recv-appId and appId values.

In subsequent offers, appId and recv-appId values SHOULD be maintained per m=line unless the offer is recycling the m=line for a fundamentally new purpose, in which case new appId and recv-appId values SHOULD be used. AppId and recv-appId values MUST NOT be reused, in the same session, for a different m= line than the one to which they were originally assigned, unless at least two times the maximum segment lifetime (MSL) has passed since the previous offer/answer exchange in which the values were used.

5. Other Considerations

During the work on the document we identified that there are two different problems that we were trying to address. One has to do with mapping the RTP stream to an m-line addressed in the current version of the document. The other problem is to provide semantics to an RTP stream description in the SDP description for the cases

where the SSRC of the RTP stream is not known to the SDP signaling layer and we also identified issues not addressed by the SSRC attribute [RFC5576].

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..

Note: These issues will be addressed in a separate document based on the feedback from the working group that even though these are real issues they should have a separate solution in order to differentiate between the token and the semantics needed.

6. Acknowledgements

Place Holder

7. IANA Considerations

TBD

8. Security Considerations

TBD.

9. References

9.1. 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.

9.2. 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-01 (work in progress), February 2014.

[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-06 (work in progress), April 2014.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-09 (work in progress), February 2014.

[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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[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