Stream: Internet Engineering Task Force (IETF)
RFC: 8830
Category: Standards Track
Published: August 2020
ISSN: 2070-1721
Author: H. Alvestrand
Google

RFC 8830
WebRTC MediaStream Identification in the Session Description Protocol

Abstract

This document specifies a Session Description Protocol (SDP) grouping mechanism for RTP media streams that can be used to specify relations between media streams.

This mechanism is used to signal the association between the SDP concept of "media description" and the Web Real-Time Communication (WebRTC) concept of MediaStream/MediaStreamTrack using SDP signaling.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Terminology

This document uses terminology from [RFC8825]. In addition, the following terms are used as described below:

RTP stream: A stream of RTP packets containing media data [RFC7656].

MediaStream: An assembly of MediaStreamTracks [W3C.CR-mediacapture-streams]. One MediaStream can contain multiple MediaStreamTracks, of the same or different types.

MediaStreamTrack: Defined in [W3C.CR-mediacapture-streams] as a unidirectional flow of media data (either audio or video, but not both). Corresponds to the [RFC7656] term "source stream". One MediaStreamTrack can be present in zero, one, or multiple MediaStreams.

Media description: Defined in [RFC4566] as a set of fields starting with an "m=" field and terminated by either the next "m=" field or the end of the session description.

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Structure of This Document

This document adds a new Session Description Protocol (SDP) [RFC4566] mechanism that can attach identifiers to the RTP streams and attach identifiers to the groupings they form. It is designed for use with WebRTC [RFC8825].

Section 1.3 gives the background on why a new mechanism is needed.

Section 2 gives the definition of the new mechanism.

Section 3 gives the necessary semantic information and procedures for using the "msid" attribute to signal the association of MediaStreamTracks to MediaStreams in support of the WebRTC API [W3C-WebRTC].

1.3. Why a New Mechanism Is Needed

When media is carried by RTP [RFC3550], each RTP stream is distinguished inside an RTP session by its Synchronization Source (SSRC); each RTP session is distinguished from all other RTP sessions by being on a different transport association (strictly speaking, two transport associations, one used for RTP and one used for the RTP Control Protocol (RTCP), unless RTP/RTCP multiplexing [RFC5761] is used).

SDP [RFC4566] gives a format for describing an SDP session that can contain multiple media descriptions. According to the model used in [RFC8829], each media description describes exactly one media source. If multiple media sources are carried in an RTP session, this is signaled using BUNDLE [RFC8843]; if BUNDLE is not used, each media source is carried in its own RTP session.

The SDP Grouping Framework [RFC5888] can be used to group media descriptions. However, for the use case of WebRTC, there is the need for an application to specify some application-level information about the association between the media description and the group. This is not possible using the SDP Grouping Framework.

1.4. The WebRTC MediaStream

The W3C WebRTC API specification [W3C-WebRTC] specifies that communication between WebRTC entities is done via MediaStreams, which contain MediaStreamTracks. A MediaStreamTrack is generally carried using a single SSRC in an RTP session, forming an RTP stream. The collision of terminology is unfortunate. There might possibly be additional SSRCs, possibly within additional RTP sessions, in order to support functionality like forward error correction or simulcast. These additional SSRCs are not affected by this specification.

MediaStreamTracks are unidirectional; they carry media in one direction only.

In the RTP specification, RTP streams are identified using the SSRC field. Streams are grouped into RTP sessions and also carry a CNAME. Neither CNAME nor RTP session corresponds to a MediaStream. Therefore, the association of an RTP stream to MediaStreams need to be explicitly signaled.

WebRTC defines a mapping (documented in [RFC8829]) where one SDP media description is used to describe each MediaStreamTrack, and the BUNDLE mechanism [RFC8843] is used to group MediaStreamTracks into RTP sessions. Therefore, the need is to specify the identifier (ID) of the MediaStreamTrack and its associated MediaStream for each media description, which can be accomplished with a media-level SDP attribute.

This usage is described in Section 3.

2. The MSID Mechanism

This document defines a new SDP [RFC4566] media-level "msid" attribute. This new attribute allows endpoints to associate RTP streams that are described in separate media descriptions with the right MediaStreams, as defined in [W3C-WebRTC]. It also allows endpoints to carry an identifier for each MediaStreamTrack in its "appdata" field.

The value of the "msid" attribute consists of an identifier and an optional "appdata" field.

The name of the attribute is "msid".

The value of the attribute is specified by the following ABNF [RFC5234] grammar:

```
  msid-value = msid-id [ SP msid-appdata ]
  msid-id = 1*64token-char ; see RFC 4566
  msid-appdata = 1*64token-char  ; see RFC 4566
```

An example "msid" value for a group with the identifier "examplefoo" and application data "examplebar" might look like this:

```
  msid:examplefoo examplebar
```

The identifier is a string of ASCII characters that are legal in a "token", consisting of between 1 and 64 characters.

Application data (msid-appdata) is carried on the same line as the identifier, separated from the identifier by a space.

The identifier ("msid-id") uniquely identifies a group within the scope of an SDP description.

There may be multiple "msid" attributes in a single media description. This represents the case where a single MediaStreamTrack is present in multiple MediaStreams; the value of "msid-appdata" **MUST** be identical for all occurrences.

Multiple media descriptions with the same value for "msid-id" and "msid-appdata" are not permitted.

Endpoints can update the associations between RTP streams as expressed by "msid" attributes at any time.

The "msid" attributes depend on the association of RTP streams with media descriptions but do not depend on the association of RTP streams with RTP transports. Therefore, their Mux Category (as defined in [RFC8859]) is NORMAL; the process of deciding on "msid" attributes doesn't have to take into consideration whether or not the RTP streams are bundled.

3. Procedures

This section describes the procedures for associating media descriptions representing MediaStreamTracks within MediaStreams, as defined in [W3C-WebRTC].

In the Javascript API described in that specification, each MediaStream and MediaStreamTrack has an "id" attribute, which is a DOMString.

The value of the "msid-id" field in the MSID consists of the "id" attribute of a MediaStream, as defined in the MediaStream's WebIDL specification [WEBIDL]. The special value "-" indicates "no MediaStream".

The value of the "msid-appdata" field in the MSID, if present, consists of the "id" attribute of a MediaStreamTrack, as defined in the MediaStreamTrack's WebIDL specification.

When an SDP session description is updated, a specific "msid-id" value continues to refer to the same MediaStream, and a specific "msid-appdata" to the same MediaStreamTrack. There is no memory apart from the currently valid SDP descriptions; if an MSID "identifier" value disappears from the SDP and appears in a later negotiation, it will be taken to refer to a new MediaStream.

If the "msid" attribute does not conform to the ABNF given here, it **SHOULD** be ignored.

The following is a high-level description of the rules for handling SDP updates. Detailed procedures are located in Section 3.2.

- When a new MSID "identifier" value occurs in a session description, and it is not "-", the recipient can signal to its application that a new MediaStream has been added.
- When a session description is updated to have media descriptions with an MSID "identifier" value, with one or more different "appdata" values, the recipient can signal to its application that new MediaStreamTracks have been added and note to which MediaStream they have been added. This is done for each different MSID "identifier" value, including the special value "-", which indicates that a MediaStreamTrack has been added with no corresponding MediaStream.
- If an MSID "identifier" value with no "appdata" value appears, it means that the sender did not inform the recipient of the desired identifier of the MediaStreamTrack, and the recipient will assign the "id" value of the created MediaStreamTrack on its own. All MSIDs in a media section that do not have an "appdata" value are assumed to refer to the same MediaStreamTrack.
- When a session description is updated to no longer list any "msid" attribute on a specific media description, the recipient can signal to its application that the corresponding MediaStreamTrack has ended.

In addition to signaling that the track is ended when its "msid" attribute disappears from the SDP, the track will also be signaled as being ended when all associated SSRCs have disappeared by the rules of [RFC3550], Sections 6.3.4 (BYE packet received) and 6.3.5 (timeout), or when the corresponding media description is disabled by setting the port number to zero. Changing the direction of the media description (by setting "sendonly", "recvonly", or "inactive" attributes) will not end the MediaStreamTrack.

The association between SSRCs and media descriptions is specified in [RFC8829].

3.1. Handling of Nonsignaled Tracks

Entities that do not use the mechanism described in this document will not send the "msid" attribute and thus will not send information allowing the mapping of RTP packets to MediaStreams. This means that there will be some incoming RTP packets for which the recipient has no predefined MediaStream ID value.

Note that the handling described below is triggered by incoming RTP packets, not SDP negotiation.

When communicating with entities that use the MSID mechanism, the only time incoming RTP packets can be received without an associated MediaStream ID value is when, after the initial negotiation, a negotiation is performed where the answerer adds a MediaStreamTrack to an already established connection and starts sending data before the answer is received by the offerer. For initial negotiation, packets won't flow until the Interactive Connectivity Establishment (ICE) candidates and fingerprints have been exchanged, so this is not an issue.

The recipient of those packets will perform the following steps:

- When RTP packets are initially received, it will create an appropriate MediaStreamTrack based on the type of the media (carried in PayloadType) and use the MID RTP header extension [RFC8843] (if present) to associate the RTP packets with a specific media section.
- If the connection is not in the RTCSignalingState "stable", it will wait at this point.
- When the connection is in the RTCSignalingState "stable", it will assign ID values.

The following steps are performed to assign ID values:

- If there is an "msid" attribute, it will use that attribute to populate the "id" field of the MediaStreamTrack and associated MediaStreams, as described above.
- If there is no "msid" attribute, the identifier of the MediaStreamTrack will be set to a randomly generated string, and it will be signaled as being part of a MediaStream with the WebIDL "label" attribute set to "Non-WebRTC stream".
- After deciding on the "id" field to be applied to the MediaStreamTrack, the track will be signaled to the user.

The process above may involve a considerable amount of buffering before the "stable" state is entered. If the implementation wishes to limit this buffering, it **MUST** signal to the user that media has been discarded.

It follows from the above that MediaStreamTracks in the "default" MediaStream cannot be closed by removing the "msid" attribute; the application must instead signal these as closed when the SSRC disappears, either according to the rules of Sections 6.3.4 and 6.3.5 of [RFC3550] or by disabling the media description by setting its port to zero.

3.2. Detailed Offer/Answer Procedures

These procedures are given in terms of sections recommended by [RFC3264]. They describe the actions to be taken in terms of MediaStreams and MediaStreamTracks; they do not include event signaling inside the application, which is described in the JavaScript Session Establishment Protocol (JSEP) [RFC8829].

3.2.1. Generating the Initial Offer

For each media description in the offer, if there is an associated outgoing MediaStreamTrack, the offerer adds one "a=msid" attribute to the section for each MediaStream with which the MediaStreamTrack is associated. The "identifier" field of the attribute is set to the WebIDL "id" attribute of the MediaStream. If the sender wishes to signal identifiers for the MediaStreamTracks, the "appdata" field is set to the WebIDL "id" attribute of the MediaStreamTrack; otherwise, it is omitted.

3.2.2. Answerer Processing of the Offer

For each media description in the offer and each "a=msid" attribute in the media description, the receiver of the offer will perform the following steps:

- Extract the "appdata" field of the "a=msid" attribute, if present.
- If the "appdata" field exists: Check if a MediaStreamTrack with the same WebIDL "id" attribute as the "appdata" field already exists and is not in the "ended" state. If such a MediaStreamTrack is not found, create it.
- If the "appdata" field does not exist, and a MediaStreamTrack is not associated with this media section, create a MediaStreamTrack and associate it with this media section for future use.
- Extract the "identifier" field of the "a=msid" attribute.
- Check if a MediaStream with the same WebIDL "id" attribute already exists. If not, create it.
- Add the MediaStreamTrack to the MediaStream.
- Signal to the user that a new MediaStreamTrack is available.

3.2.3. Generating the Answer

The answer is generated in exactly the same manner as the offer. "a=msid" values in the offer do not influence the answer.

3.2.4. Offerer Processing of the Answer

The answer is processed in exactly the same manner as the offer.

3.2.5. Modifying the Session

On subsequent exchanges, precisely the same procedure as for the initial offer/answer is followed, but with one additional step in the parsing of the offer and answer:

- For each MediaStreamTrack that has been created as a result of previous offer/answer exchanges, and is not in the "ended" state, check to see if there is still an "a=msid" attribute in the present SDP whose "appdata" field is the same as the WebIDL "id" attribute of the track.
- If no such attribute is found, stop the MediaStreamTrack. This will set its state to "ended".

3.3. Example SDP Description

The following SDP description shows the representation of a WebRTC PeerConnection with two MediaStreams, each of which has one audio and one video track. Only the parts relevant to the MSID are shown.

Line wrapping, empty lines, and comments are added for clarity. They are not part of the SDP.

```
# First MediaStream - id is 4701...
m=audio 56500 UDP/TLS/RTP/SAVPF 96 0 8 97 98
a=msid:47017fee-b6c1-4162-929c-a25110252400
       f83006c5-a0ff-4e0a-9ed9-d3e6747be7d9

m=video 56502 UDP/TLS/RTP/SAVPF 100 101
a=msid:47017fee-b6c1-4162-929c-a25110252400
       b47bdb4a-5db8-49b5-bcdc-e0c9a23172e0

# Second MediaStream - id is 6131....
m=audio 56503 UDP/TLS/RTP/SAVPF 96 0 8 97 98
a=msid:61317484-2ed4-49d7-9eb7-1414322a7aae
       b94006c5-cade-4e0a-9ed9-d3e6747be7d9

m=video 56504 UDP/TLS/RTP/SAVPF 100 101
a=msid:61317484-2ed4-49d7-9eb7-1414322a7aae
       f30bdb4a-1497-49b5-3198-e0c9a23172e0
```

4. IANA Considerations

4.1. Attribute Registration in Existing Registries

IANA has registered the "msid" attribute in the "att-field" (media level only) registry within the "Session Description Protocol (SDP) Parameters" registry, according to the procedures of [RFC4566].

The "msid" registration information is as follows:

Contact name, email: IETF, contacted via mmusic@ietf.org, or a successor address designated by IESG

Attribute name: msid

Attribute syntax:

```
            msid-value = msid-id [ SP msid-appdata ]
            msid-id = 1*64token-char ; see RFC 4566
            msid-appdata = 1*64token-char  ; see RFC 4566
```

Attribute semantics: Described in RFC 8830

Attribute value: msid-value

Long-form attribute name: MediaStream Identifier

Usage level: media

Subject to charset: The attribute value contains only ASCII characters and is therefore not subject to the charset attribute.

Purpose: The attribute can be used to signal the relationship between a WebRTC MediaStream and a set of media descriptions.

O/A Procedures: Described in RFC 8830

Appropriate values: The details of appropriate values are given in RFC 8830 (this document).

Mux Category: NORMAL

The Mux Category is defined in [RFC8859].

5. Security Considerations

An adversary with the ability to modify SDP descriptions has the ability to switch around tracks between MediaStreams. This is a special case of the general security consideration that modification of SDP descriptions needs to be confined to entities trusted by the application.

If implementing buffering as mentioned in Section 3.1, the amount of buffering should be limited to avoid memory exhaustion attacks.

Careless generation of identifiers can leak privacy-sensitive information. [W3C.CR-mediacapture-streams] recommends that identifiers be generated using a Universally Unique IDentifier (UUID) class 3 or 4 as a basis, which avoids such leakage.

No other attacks have been identified that depend on this mechanism.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8829] Uberti, J., Jennings, C., and E. Rescorla, Ed., "JavaScript Session Establishment Protocol (JSEP)", RFC 8829, DOI 10.17487/RFC8829, July 2020, <https://www.rfc-editor.org/info/rfc8829>.

[RFC8859] Nandakumar, S., "A Framework for Session Description Protocol (SDP) Attributes When Multiplexing", DOI 10.17487/RFC8859, RFC 8859, July 2020, <https://www.rfc-editor.org/info/rfc8859>.

[W3C-WebRTC] Jennings, C., Boström, H., and J-I. Bruaroey, "WebRTC 1.0: Real-time Communication Between Browsers", W3C Candidate Recommendation, 13 December 2019, <https://www.w3.org/TR/2019/CR-webrtc-20191213/>.

[W3C.CR-mediacapture-streams] Burnett, D., Bergkvist, A., Jennings, C., Narayanan, A., Aboba, B., Bruaroey, J.-I., and H. Boström, "Media Capture and Streams", W3C Candidate Recommendation, 2 July 2019, <https://www.w3.org/TR/2019/CR-mediacapture-streams-20190702/>.

6.2. Informative References

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <https://www.rfc-editor.org/info/rfc5761>.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <https://www.rfc-editor.org/info/rfc5888>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

[RFC8825] Alvestrand, H., "Overview: Real-Time Protocols for Browser-Based Applications", RFC 8825, DOI 10.17487/RFC8825, July 2020, <https://www.rfc-editor.org/info/rfc8825>.

[RFC8843] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", RFC 8843, DOI 10.17487/RFC8843, July 2020, <https://www.rfc-editor.org/info/rfc8843>.

[WEBIDL] Chen, E. and T. Gu, "Web IDL", W3C Editor's Draft, August 2020, <https://heycam.github.io/webidl/>.

Appendix A. Design Considerations, Rejected Alternatives

One suggested mechanism has been to use CNAME instead of a new attribute. This was abandoned because CNAME identifies a synchronization context; one can imagine both wanting to have tracks from the same synchronization context in multiple MediaStreams and wanting to have tracks from multiple synchronization contexts within one MediaStream (but the latter is impossible, since a MediaStream is defined to impose synchronization on its members).

Another suggestion has been to put the "msid" value within an attribute of RTCP SR (sender report) packets. This doesn't offer the ability to know that you have seen all the tracks currently configured for a MediaStream.

A suggestion that survived for a number of drafts of this document was to define MSID as a generic mechanism, where the particular semantics of this usage of the mechanism would be defined by an "a=wms-semantic" attribute. This was removed in April 2015.

Acknowledgements

This note is based on sketches from, among others, Justin Uberti and Cullen Jennings.

Special thanks to Flemming Andreassen, Ben Campbell, Miguel Garcia, Martin Thomson, Ted Hardie, Adam Roach, Magnus Westerlund, Alissa Cooper, Sue Hares, and Paul Kyzivat for their work in reviewing this document, with many specific language suggestions.

Author's Address

Harald Alvestrand
Google
Kungsbron 2
SE-11122 Stockholm
Sweden
Email: harald@alvestrand.no