INTERNET-DRAFT
Intended Status: Standard Track
October 19, 2015

R. Huang
Huawei

Multi-view streams in SDP and RTP Sessions
draft-huang-mmusic-multiview-00

Abstract

This document analyses the streaming options of multi-view applications,and describes the required SDP signaling for them.

Status of this Memo

Copyright and License Notice

Table of Contents

1 Introduction

Multi-view video consists of multiple views that are taken by multiple cameras from different positions and angles. 3D video and free viewpoint video are two typical use cases. The first offers a 3D depth impression of the observed scenery, while the second allows for interactive selection of viewpoint and direction within a certain operating range as known from computer graphics. Streaming of such multi-view applications on Internet is usually offered at varying speeds and costs over a variety of physical infrastructures. However, since the multi-view video consists of multiple video sequences, the traffic is several times larger than traditional multimedia, which brings the dramatic increase in the bandwidth requirement. For streaming of multi-view representations two operating options are usually used: streaming all views in a highly compressed MVC bit-stream with little possibility of random access or encoding all views independently and streaming only required views.

This document analyses the streaming options of multi-view applications,and describes the required SDP signaling for them.

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 RFC 2119 [RFC2119].

This document uses the following terms:

Multi-view Video Compression: Using efficient compression techniques to transmit multi-view video. Usually, inter-view similarity between adjacent views and temporal similarity between temporally successive images of each video are exploited.

MVC (Multiview Video Coding): An extension of the AVC standard, standardised in 2009, covers a wide range of 3D video applications including 3D video streaming, free-viewpoint video. It is inherently backward compatible with AVC which serves as the base-view that can be decoded independently in the absence of the MVC decoder. Any additional views are referred to as enhancement views and are typically coded using interview prediction within the same bitstream.

3. Use Cases

Multi-view video communications can be applied on a wide variety of use cases. In this section, several of the most likely usage scenarios are introduced.

3.1. 3D Channel of IPTV

The multi-view video content is sent to a group of end users by multicast, which is widely applied in 3D channels of IPTV. Different views can be separately transmitted in different multicast group or just be transmitted in one multicast group in MVC.

3.2 Multi-view video conference

Two or more users engage in a remote video conversation using mobile or desktop terminals, which have stereoscopic capture and display capabilities. The multi-view video stream requires the transmission of 2 or more views or, alternately, of one view and a depth map. The views or depth maps may be transmitted as separate transport streams, or together,depending on the choice of multi-view video transmissions. In the case of a multiparty session, an intermediate media server should be used for mixing. Video mixing for the multi-view video stream needs to be aware of the additional views or depth maps.

3.2.3 Tele-education with Free Viewpoint Video

A teacher intends to give a real-time lecture to students in one or more remote sites. The teacher’s site is equipped with a multi-view capture setup, and has a 2D display to show feedback from the students, who are equipped with regular 2D cameras and displays. Multi-view video content is transmitted so that the student sites are capable of selecting perspectives of the captured scene within a certain operating range, and rendering them in real time, thus some signaling interaction between students and the teacher may be required.

3.2.4 Immersive Telepresence with Multi-view Video

A group of users wants to conduct a meeting through a telepresence system, connecting to the session from several sites, and with each terminal supporting multiple users. Participants are arranged around a shared virtual table, and each remote participant is shown in a separate screen, which is autostereoscopic and displays two different views for each user in the room. At each site, several autostereoscopic displays and a multi-view capture setup are deployed. The viewpoint of these views is adjusted during the session so as to match the position of observing users. The use of multi-view video makes telepresence systems having perfect eye contact and spatial faithfulness.

4. Multi-view Video Transmission

Multi-view video, which has two or more views, increases the encoding complexity and bandwidth requirement for transmission. There are several ways that can be used to transmit these views:

* Multi-view simulcast: encode each view and/or depth map independently using a monocular video codec, which enables streaming each view over separate channels; and clients can requests as many views as their displays require without worrying about inter-view dependencies. While this method has its benefits, it does not exploit the redundancies that are preset in between the views.

* Multi-view video compression: encode views using specific technique to decrease the overall bit rate by exploiting the inter-view redundancies. However, although it exploits the similarities that are present between the views, it increases the effect of transmission errors. And the inter-view efficient compression techniques make the views depend on each others. In order to decode a frame correctly, the frames it depend on must be decoded at first, which will bring more unnecessary transmission and delay if the views should be displayed is far away from the reference view. MVC is a typical and often used video compression format applied to multi-view videos.

* Adaptive Multi-view video transmission: Adaptive streaming of multi-view video is also considered when functionalities such as rate scalability, resolution scalability, view scalability, and packet-loss resilience should be offered. In such a case, simulcast or Multi-view video compression will be used together with adaptive streaming techniques, e.g., SVC.

* Combination transmission: Sending multi-view video by using a combination of simulcast and multi-view video compression techniques may be also a good option for some specific scenarios. For example, if one multi-view video compression technique works well for closely related views but not for widely differing views, sending several multi-view video compression streams simultaneously can solve the problem.

5. SDP Signaling Requirements for Multi-view Video

The following requirements need to be met to support streaming multi-view video in previous sections:

REQ-1: It must be possible to signal whether multi-view simulcast or multi-view video compression is used.

REQ-2: It must be possible to signal adaptive multi-view video transmission, e.g., multi-view video simulcast is used together with adaptive simulcast.

REQ-3: It must be possible to signal combination transmission where multi-view video compression is used together with simulcast.

REQ-4: Bundled [I-D.ietf-mmusic-sdp-bundle-negotiation] usage must be considered.

REQ-5: It must be possible to signal multi-view video related decoder constraints, e.g., maximum number of view streams that can be provided at the sender and maximum number of view streams that can be received at the receiver.

REQ-6: It must be possible to support both declarative SDP and SDP offer/answer.

REQ-7: When multi-view simulcast is used, it must be possible to have some ways to allow receivers ask for required view streams that they wish to receive.

REQ-8: It must be compatible with existing other mechanisms, e.g., RTP retransmission [RFC4588], Forward Error Correction [RFC5109].

6. Gap Analysis

6.1. RFC5583

This specification defines a SDP mechanism to signaling the decoding dependency of different media descriptions with the same media type [RFC5583]. It can be used to signal the use of SVC or MVC. However, it cannot differentiate the usages between multi-view simulcast and MVC, not mention when multi-view transmission is used together with simulcast or SVC.

6.2. Simulcast

[I.d-ietf-mmusic-sdp-simulcat] describes simulcast as the scenarios where sending multiple differently encoded versions of the same media source in different RTP streams. It is mainly used for the same video source encoded with different video encoder types or image resolutions. It still can not deal with the case when multi-view transmission is used together with simulcast or SVC.

6.3. RID

[I.d-pthatcher-mmusic-rid] defines a framework to identify Source RTP streams with constraints on its payload format in SDP. It can effectively identify the source RTP stream within a RTP session, which is quite useful for simulcast. Thus, it may be also helpful for multi-view video signaling. It need to be considered further for potential problems and issues.

6.4. CLUE

CLUE is dedicated for telepresence systems which provide high definition, high quality audio/video enabling a "being-there" experience. It involves multiple devices like multiple cameras, displays, microphones, and loudspeakers. It specifies spatial relationship of these devices, viewpoint, field of view/capture for these devices, and related information [I.d-ietf-clue-signal]. However, the usages of 3D or free view point video in CLUE are not considered in current CLUE scope. Thus, the supporting multi-view view of CLUE needs to be considered further.

6.5. 3D Signaling

There were some work in MMUSIC to propose some 3D signaling solutions. [I.d-greevenbosch-mmusic-signal-3d-format] and [I.d-greevenbosch-mmusic-sdp-parallax] introduce new SDP attributes to provide format description and depth position signaling in 3D applications. [I.d-capelastegui-mmusic-3dv-sdp] introduces a mechanism to describe 3D video streams composed of multiple video video views, or of a combination of views and depth maps. They are not directly multi-view, but should be evaluated when proposing possible solutions.

7. Possible Solutions

TBD.

8. Security Considerations

TBD.

9. IANA Considerations

TBD.

10. Acknowledgments

11. References

11.1 Normative References

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

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R., Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-23 (work in progress), July 2015.

[I.d-ietf-mmusic-sdp-simulcat] Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in SDP and RTP Sessions", draft-ietf-mmusic-sdp-simulcast-02 (work in progress), October 6, 2015.

[I.d-pthatcher-mmusic-rid] Thatcher, P., Zanaty, M., Nandakuma, S., Roach, A., Burman, B., and B. Campen, "RTP Payload Format Constraints", draft-pthatcher-mmusic-rid-01 (work in progress), October 2015.

[I.d-ietf-clue-signal] Kyzivat, P., Xiao, L., Groves, C., and R., Hansen, "CLUE Signaling", draft-ietf-clue-signaling-06, August 2015.

[I.d-greevenbosch-mmusic-signal-3d-format] Greevenbosch, B., and Y., Hui, "Signal 3D format", draft-greevenbosch-mmusic-sdp-3d-format-01, October 2012.

[I.d-greevenbosch-mmusic-sdp-parallax] Greevenbosch, B., and Y., Hui, "SDP attribute to signal parallax", draft-greevenbosch-mmusic-sdp-parallax-01, October 2012.

[I.d-capelastegui-mmusic-3dv-sdp] Capelastegui, P., "3D Video in the Session Description Protocol (SDP)", draft-capelastegui-mmusic-3dv-sdp-00, April 2012.

11.2 Informative References

Authors' Addresses

Rachel Huang
Huawei
101 Software Avenue
Nanjing, China

EMail: rachel.huang@huawei.com

Using the SDP Offer/Answer Mechanism for DTLS
draft-ietf-mmusic-dtls-sdp-01.txt

Abstract

This draft defines the SDP offer/answer procedures for negotiating and establishing a DTLS association. The draft also defines the criteria for when a new DTLS association must be established.

This draft defines a new SDP media-level attribute, 'dtls-connection'.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC5763] defines SDP Offer/Answer procedures for SRTP-DTLS. This draft defines the SDP Offer/Answer [RFC3264] procedures for negotiation DTLS in general, based on the procedures in [RFC5763].

This draft also defines a new SDP attribute, 'dtls-connection'. The attribute is used in SDP offers and answers to explicitly indicate whether a new DTLS association is to be established.

As defined in [RFC5763], a new DTLS association MUST be established when transport parameters are changed. Transport parameter change is not well defined when Interactive Connectivity Establishment (ICE)

[RFC5245] is used. One possible way to determine a transport change is based on ufrag change, but the ufrag value is changed both when ICE is negotiated and when ICE restart [RFC5245] occurs. These events do not always require a new DTLS association to be established, but currently there is no way to explicitly indicate in an SDP offer or answer whether a new DTLS association is required. To solve that problem, this draft defines a new SDP attribute, 'dtls-connection'. The attribute is used in SDP offers and answers to explicitly indicate whether a new DTLS association is to be established/re-established. The attribute can be used both with and without ICE.

2. Abbreviations

TBD

3. Conventions

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].

4. Establishing a new DTLS Association

4.1. General

A new DTLS association MUST be established in the following cases:

- o The DTLS roles change;
- o The fingerprint (certificate) value changes; or
- o The establishment of a new DTLS association is explicitly signaled;

NOTE: The first two items list above are based on the procedures in [RFC5763]. This draft adds the support for explicit signaling.

The sections below describe typical cases where a new DTLS association needs to be established.

4.2. Change of Local Transport Parameters

If an endpoint modifies its local transport parameters (IP address and/or port), and if the modification requires a new DTLS association, the endpoint MUST either change its DTLS role, its fingerprint value and/or use the SDP 'dtls-connection' attribute with a 'new' value Section 5.

4.3. Change of ICE ufrag value

If an endpoint uses ICE, and modifies a local ufrag value, and if the modification requires a new DTLS association, the endpoint MUST either change its DTLS role, its fingerprint value and/or use the SDP 'dtls-connection' attribute with a 'new' value Section 5.

4.4. Multiple SDP fingerprint attributes

It is possible to associate multiple SDP fingerprint attribute values to an 'm-' line. If any of the attribute values associated with an 'm-' line are removed, or if any new attribute values are added, it is considered a fingerprint value change.

5. SDP DTLS-Connection Attribute

5.1. General

The SDP 'connection' attribute [RFC4145] was originally defined for connection-oriented protocols, e.g. TCP and TLS. This section defines a similar attribute, 'dtls-connection', to be used with DTLS.

A 'dtls-connection' attribute value of 'new' indicates that a new DTLS association MUST be established. A 'dtls-connection' attribute value of 'existing' indicates that a new DTLS association MUST NOT be established.

Unlike the SDP 'connection' attribute for TLS, there is no default value defined for the 'dtls-connection' attribute. Implementations that wish to use the attribute MUST explicitly include it in SDP offers and answers. If an offer or answer does not contain an attribute, other means needs to be used in order for endpoints to determine whether an offer or answer is associated with an event that requires the DTLS association to be re-established.

The SDP Offer/Answer [RFC3264] procedures associated with the attribute are defined in Section 6

5.2. ABNF

The ABNF [RFC5234] grammar for the SDP 'dtls-connection' attributes is:

```
dtls-connection-attr   = "a=dtls-connection:" conn-value
conn-value             = "new" / "existing"
```

6. SDP Offer/Answer Procedures

6.1. General

This section defines the SDP offer/answer procedures for using the SDP 'dtls-connection' attribute for DTLS. The section also describes how the usage of the SDP 'setup' attribute and the SDP 'fingerprint' attribute [RFC4572] is affected.

The procedures in this section are based on the procedures for SRTP-DTLS [RFC5763], with the addition of usage of the SDP 'dtls-connection' attribute.

6.2. Generating the Initial SDP Offer

When the offerer sends the initial offer, and the offerer wants to establish a DTLS association, it MUST insert an SDP 'dtls-connection' attribute with a 'new' value in the offer. In addition, the offerer MUST insert an SDP 'setup' attribute according to the procedures in [RFC4145], and an SDP 'fingerprint' attribute according to the procedures in [RFC4572], in the offer.

Unlike for TCP and TLS connections, in case of DTLS associations the SDP 'setup' attribute 'holdconn' value MUST NOT be used.

6.3. Generating the Answer

If an answerer receives an offer that contains an SDP 'dtls-connection' attribute with a 'new' value, the answerer MUST insert a 'new' value in the associated answer. The same applies if the answerer receives an offer that contains an SDP 'dtls-connection' attribute with a 'new' value, but the answerer determines (based on the criteria for establishing a new DTLS association) that a new DTLS association is to be established. In addition, the answerer MUST insert an SDP 'setup' attribute according to the procedures in [RFC4145], and an SDP 'fingerprint' attribute according to the procedures in [RFC4572], in the answer.

If the answerer does not accept the establishment of the DTLS association, it MUST reject the "m=" lines associated with the suggested DTLS association [RFC3264].

If an answerer receives an offer that contains a 'dtls-connection' attribute with an 'existing' value, and if the answerer determines that a new DTLS association does not need to be established, it MUST insert a connection attribute with an 'existing' value in the associated answer. In addition, the answerer MUST insert an SDP 'setup' attribute with a value that does not change the previously negotiated DTLS roles, and an SDP 'fingerprint' attribute with a value that does not change the fingerprint, in the answer.

If the answerer receives an offer that does not contain an SDP 'dtls-connection' attribute, the answerer MUST NOT insert a 'dtls-connection' attribute in the answer.

If a new DTLS association is to be established, and if the answerer becomes DTLS client, the answerer MUST initiate the procedures for establishing the DTLS association. If the answerer becomes DTLS server, it MUST wait for the offerer to establish the DTLS association.

6.4. Offerer Processing of the SDP Answer

When an offerer receives an answer that contains an SDP 'dtls-connection' attribute with a 'new' value, and if the offerer becomes DTLS client, the offerer MUST establish a DTLS association. If the offerer becomes DTLS server, it MUST wait for the answerer to establish the DTLS association.

If the answer contains an SDP 'dtls-connection' attribute with an 'existing' value, the offerer will continue using the previously established DTLS association. It is considered an error case if the answer contains a 'dtls-connection' attribute with an 'existing' value, and a DTLS association does not exist.

6.5. Modifying the Session

When the offerer sends a subsequent offer, and the offerer wants to establish a new DTLS association, the offerer MUST insert an SDP 'dtls-connection' attribute with a 'new' value in the offer. In addition, the offerer MUST insert an SDP 'setup' attribute according to the procedures in [RFC4145], and an SDP 'fingerprint' attribute according to the procedures in [RFC4572], in the offer.

when the offerer sends a subsequent offer, and the offerer does not want to establish a new DTLS association, if a previously established DTLS association exists, the offerer MUST insert an SDP 'dtls-connection' attribute with an 'existing' value in the offer. In addition, the offerer MUST insert an SDP 'setup' attribute with a value that does not change the previously negotiated DTLS roles, and an SDP 'fingerprint' attribute with a value that does not change the fingerprint, in the offer.

7. ICE Considerations

An ICE restart [RFC5245] does not by default require a new DTLS association to be established.

As defined in [RFC5763], each ICE candidate associated with a component is treated as being part of the same DTLS association. Therefore, from a DTLS perspective it is not considered a change of local transport parameters when an endpoint switches between those ICE candidates.

8. SIP Considerations

When the Session Initiation Protocol (SIP) [RFC3261] is used as the signal protocol for establishing a multimedia session, dialogs [RFC3261] might be established between the caller and multiple callees. This is referred to as forking. If forking occurs, separate DTLS associations MUST be established between the caller and each callee.

It is possible to send an INVITE request which does not contain an SDP offer. Such INVITE request is often referred to as an 'empty INVITE', or an 'offerless INVITE'. The receiving endpoint will include the SDP offer in a response associated with the response. When the endpoint generates such SDP offer, it MUST assign an SDP connection attribute, with a 'new' value, to each 'm-' line that describes DTLS protected media. If ICE is used, the endpoint MUST allocate a new set of ICE candidates, in order to ensure that two DTLS association would not be running over the same transport.

9. RFC Updates

Here we will add the RFC updates that are needed.

10. Security Considerations

This draft does not modify the security considerations associated with DTLS, or the SDP offer/answer mechanism. The draft simply clarifies the procedures for negotiating and establishing a DTLS association.

11. IANA Considerations

11.1. Registration of New SDP Attribute

This document updates the "Session Description Protocol Parameters" registry as specified in Section 8.2.2 of [RFC4566]. Specifically, it adds the SDP attributes in Section 11.1 to the table for SDP media level attributes.

Attribute name: dtls-connection
Type of attribute: media-level
Subject to charset: no
Purpose: TBD
Appropriate Values: see Section X
Contact name: Christer Holmberg

12. Acknowledgements

Thanks to Justin Uberti, Martin Thomson, Paul Kyzivat and Jens Guballa for providing comments and suggestions on the draft.

13. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-dtls-00

- o - SDP ’connection’ attribute replaced with new ’dtls-connection’ attribute.
- o - IANA Considerations added.
- o - E-mail regarding ’dtls-connection-id’ attribute added as Annex.

Changes from draft-holmberg-mmusic-sdp-dtls-01

- o - draft-ietf-mmusic version of draft submitted.
- o - Draft file name change (sdp-dtls -> dtls-sdp) due to collision with another expired draft.
- o - Clarify that if ufrag in offer is unchanged, it must be unchanged in associated answer.
- o - SIP Considerations section added.
- o - Section about multiple SDP fingerprint attributes added.

Changes from draft-holmberg-mmusic-sdp-dtls-00

- o - Editorial changes and clarifications.

14. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

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

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <http://www.rfc-editor.org/info/rfc4145>.

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

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <http://www.rfc-editor.org/info/rfc4572>.

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

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <http://www.rfc-editor.org/info/rfc5763>.

Appendix A. Design Considerations

A.1. dtls-connection versus dtls-connection-id

The text below is from an e-mail sent by Roman to the MMUSIC mailing list, 1st October 2015. It is intended to serve as background reading when discussing the way forward regarding the SDP attribute.

The "dtls-ufrag" has little to do with ICE and exists in a completely different layer. We can call this attribute "dtls-connection-id" if this will makes it less spooky. The problem that I am trying to resolve with new attribute is related to when new DTLS association needs to be established. I would argue that original intent was, that new DTLS association needs to be established on change of one of the end points or DTLS association setup attributes (setup role or fingerprint).

Originally, end point change was detected based on transport 5-tuple change. This, of cause, does not work for ICE, where 5-tuple is not known in advance and all 5-tuples associated with the same ICE component should be treated as the same connection. One option was to detect end point change when ICE is used based on ICE ufrag change, but this does not work either since ufrag can change due to ICE restart, but the same endpoints will continue to communicate.

I would also argue that setting up new DTLS association on 5-tuple change does not always work for non-ICE case either, since we can have an end point which can initiate a re-INVITE when it detects the local IP changes due to DHCP lease expiration or any other reason. This transport change does not necessarily require DTLS association change, and new DTLS handshake is undesirable since it will delay the media flow re-establishment but several network round trips.

So, we need to detect when two new end-points are communicating and new DTLS association needs to be setup. What we originally proposed is that end point will simply tell that it is setting up a new session by using SDP connection attribute or some renamed version of it.

What I am saying here is that end point cannot always identify if it needs to setup a new DTLS association. The problem arises when new offer is generated in response to an offerless INVITE. In such case, an end point does not know if it is continuing to communicate with the same end-point or if this offer is intended to be sent to a new end point.

There are two solution possible to this:

1. We specify that if an end points generates an offer in response to an offer-less INVITE it should always assume it is communicating with a new end point, it MUST add "connection:new" and MUST make sure that none of the existing transports can be possibly reused for this new DTLS association by allocating new IP:port for non ICE or a complete new set of ICE candidates in case of ICE. This will work, but it is wasteful when offer-less INVITE re-establishes connection between two existing end points. In such cases additional ports will be consumed, TURN tunnels will be allocated, and time spent on creating a DTLS session when all of this can be simply reused.

2. Instead of asking the end point which generates the offer to determine if it is establishing a new DTLS association, we will ask the end point to identify itself. So, instead of SDP connection attribute, an end point will provide some sort of randomly generated end point identifier in the new attribute (dtls-ufrag or dtls-connection-id). When the connection ID pair stays the same, the existing DTLS association continues to run over the negotiated transport. If one of the connection IDs changes, this would mean new DTLS association would need to be established. This nicely uncouples end point change identification from transport and makes negotiation follow the original intent.

In case of response to an offer-less INVITE, an offer with the existing connection ID will be generated. If this offer is sent to a new end point, both end points will detect that new DTLS association is required due to connection ID change of the answering end point. If this offer will be sent to an end point which is already a part of the existing

DTLS association, no new DTLS association will be necessary, since both connection IDs will stay the same.

This also gives us path to a more "strategic" solution in the future. DTLS handshake can be extended to include the connection ID. Each DTLS handshake can negotiate a association identifier similar to SSRC which can be used in the all subsequent DTLS messages for this association. This way multiple DTLS associations can be multiplexed over the single transport and each of them can be tied to an m= line in offer/answer. This, of cause, is not part of the current draft and is outside of MMUSIC chapter, but does provide a natural extension path for DTLS in the future.

In general Christer and I are trying to understand if there is interest in formalizing the dtls-connection-id option (more complex) or if we should stick with SDP connection:new/existing attribute and force new DTLS association always be established in response to offer-less INVITE (simpler option but can waste resources).

Please let us know if these options need further clarification or if you have any additional questions or opinions.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Roman Shpount
TurboBridge
4905 Del Ray Avenue, Suite 300
Bethesda, MD 20814
USA

Phone: +1 (240) 292-6632
Email: rshpount@turbobridge.com

MMUSIC Working Group C. Holmberg
Internet-Draft Ericsson
Updates: 3264 (if approved) H. Alvestrand
Intended status: Standards Track Google
Expires: January 21, 2016 C. Jennings
Cisco
July 20, 2015

Negotiating Media Multiplexing Using the Session Description Protocol (SDP)
draft-ietf-mmusic-sdp-bundle-negotiation-23.txt

Abstract

This specification defines a new Session Description Protocol (SDP) Grouping Framework extension, 'BUNDLE'. The extension can be used with the SDP Offer/Answer mechanism to negotiate the usage of a single address:port combination (BUNDLE address) for receiving media, referred to as bundled media, associated with multiple SDP media descriptions ("m=" lines).

To assist endpoints in negotiating the use of bundle this specification defines a new SDP attribute, 'bundle-only', which can be used to request that specific media is only used if bundled.

There are multiple ways to correlate the bundled RTP packets with the appropriate media descriptions. This specification defines a new Real-time Transport Protocol (RTP) source description (SDES) item and a new RTP header extension that provides an additional way to do this correlation by using them to carry a value that associates the RTP/RTCP packets with a specific media description.

Status of This Memo

This Internet-Draft will expire on January 21, 2016.

Copyright Notice

Table of Contents

1. Introduction

This specification defines a way to use a single address:port combination (BUNDLE address) for receiving media associated with multiple SDP media descriptions ("m=" lines).

This specification defines a new SDP Grouping Framework [RFC5888] extension called 'BUNDLE'. The extension can be used with the Session Description Protocol (SDP) Offer/Answer mechanism [RFC3264] to negotiate the usage of a BUNDLE group. Within the BUNDLE group, a BUNDLE address is used for receiving media associated with multiple "m=" lines. This is referred to as bundled media.

The offerer and answerer [RFC3264] use the BUNDLE extension to negotiate the BUNDLE addresses, one for the offerer (offerer BUNDLE address) and one for the answerer (answerer BUNDLE address), to be used for receiving the bundled media associated with a BUNDLE group. Once the offerer and the answerer have negotiated a BUNDLE group, they assign their respective BUNDLE address to each "m=" line in the BUNDLE group. The BUNDLE addresses are used to receive all media associated with the BUNDLE group.

The use of a BUNDLE group and a BUNDLE address also allows the usage of a single set of Interactive Connectivity Establishment (ICE) [RFC5245] candidates for multiple "m=" lines.

This specification also defines a new SDP attribute, 'bundle-only', which can be used to request that specific media is only used if kept within a BUNDLE group.

As defined in RFC 4566 [RFC4566], the semantics of assigning the same port value to multiple "m=" lines are undefined, and there is no grouping defined by such means. Instead, an explicit grouping mechanism needs to be used to express the intended semantics. This specification provides such an extension.

This specification also updates sections 5.1, 8.1 and 8.2 of RFC 3264 [RFC3264]. The update allows an answerer to assign a non-zero port value to an "m=" line in an SDP answer, even if the "m=" line in the associated SDP offer contained a zero port value.

This specification also defines a new Real-time Transport Protocol (RTP) [RFC3550] source description (SDES) item and a new RTP header extension that can be used to carry a value that associates RTP/RTCP packets with a specific media description. This can be used to correlate a RTP packet with the correct media.

SDP bodies can contain multiple BUNDLE groups. A given BUNDLE address MUST only be associated with a single BUNDLE group. The procedures in this specification apply independently to a given BUNDLE group. All RTP based media flows associated with a single BUNDLE group belong to a single RTP session [RFC3550].

The BUNDLE extension is backward compatible. Endpoints that do not support the extension are expected to generate offers and answers without an SDP 'group:BUNDLE' attribute, and are expected to assign a unique address to each "m=" line within an offer and answer, according to the procedures in [RFC4566] and [RFC3264]

2. Terminology

"m=" line: SDP bodies contain one or more media descriptions. Each media description is identified by an SDP "m=" line.

5-tuple: A collection of the following values: source address, source port, destination address, destination port, and transport-layer protocol.

Unique address: An IP address and port combination that is assigned to only one "m=" line in an offer or answer.

Shared address: An IP address and port combination that is assigned to multiple "m=" lines within an offer or answer.

Offerer BUNDLE-tag: The first identification-tag in a given SDP 'group:BUNDLE' attribute identification-tag list in an offer.

Answerer BUNDLE-tag: The first identification-tag in a given SDP 'group:BUNDLE' attribute identification-tag list in an answer.

Offerer BUNDLE address: Within a given BUNDLE group, an IP address and port combination used by an offerer to receive all media associated with each "m=" line within the BUNDLE group.

Answerer BUNDLE address: Within a given BUNDLE group, an IP address and port combination used by an answerer to receive all media associated with each "m=" line within the BUNDLE group.

BUNDLE group: A set of "m=" lines, created using an SDP Offer/Answer exchange, which uses the same BUNDLE address for receiving media.

Bundled "m=" line: An "m=" line, whose identification-tag is placed in an SDP 'group:BUNDLE' attribute identification-tag list in an offer or answer.

Bundle-only "m=" line: A bundled "m=" line with an associated SDP 'bundle-only' attribute.

Bundled media: All media associated with a given BUNDLE group.

Initial offer: The first offer, within an SDP session (e.g. a SIP dialog when the Session Initiation Protocol (SIP) [RFC3261] is used to carry SDP), in which the offerer indicates that it wants to create a given BUNDLE group.

Subsequent offer: An offer which contains a BUNDLE group that has been created as part of a previous offer/answer exchange.

Identification-tag: A unique token value that is used to identify an "m=" line. The SDP 'mid' attribute [RFC5888], associated with an "m=" line, carries an unique identification-tag. The session-level SDP 'group' attribute [RFC5888] carries a list of identification-tags, identifying the "m=" lines associated with that particular 'group' attribute.

3. Conventions

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 BCP 14, RFC 2119 [RFC2119].

4. Applicability Statement

The mechanism in this specification only applies to the Session Description Protocol (SDP) [RFC4566], when used together with the SDP offer/answer mechanism [RFC3264]. Declarative usage of SDP is out of scope of this document, and is thus undefined.

5. SDP Grouping Framework BUNDLE Extension

This section defines a new SDP Grouping Framework extension [RFC5888], 'BUNDLE'. The BUNDLE extension can be used with the SDP Offer/Answer mechanism to negotiate the usage of a single address:port combination (BUNDLE address) for receiving bundled media.

A single address:port combination is also used for sending bundled media. The address:port combination used for sending bundled media MAY be the same as the BUNDLE address, used to receive bundled media, depending on whether symmetric RTP [RFC4961] is used.

All media associated with a BUNDLE group share a single 5-tuple, i.e. in addition to using a single address:port combination all bundled media MUST be transported using the same transport-layer protocol (e.g. UDP or TCP).

The BUNDLE extension is indicated using an SDP 'group' attribute with a "BUNDLE" semantics value [RFC5888]. An identification-tag is assigned to each bundled "m=" line, and each identification-tag is listed in the SDP 'group:BUNDLE' attribute identification-tag list. Each "m=" line whose identification-tag is listed in the identification-tag list is associated with a given BUNDLE group.

SDP bodies can contain multiple BUNDLE groups. Any given bundled "m=" line MUST NOT be associated with more than one BUNDLE group.

Section 8 defines the detailed SDP Offer/Answer procedures for the BUNDLE extension.

6. SDP 'bundle-only' Attribute

This section defines a new SDP media-level attribute [RFC4566], 'bundle-only'.

Name: bundle-only

Value:

Usage Level: media

Charset Dependent: no

Example:

a=bundle-only

In order to ensure that an answerer that does not support the BUNDLE extension always rejects a bundled "m=" line, the offerer can assign a zero port value to the "m=" line. According to [RFC4566] an answerer will reject such "m=" line. By associating an SDP 'bundle-only' attribute with such "m=" line, the offerer can request that the answerer accepts the "m=" line if the answerer supports the Bundle extension, and if the answerer keeps the "m=" line within the associated BUNDLE group.

NOTE: Once the offerer BUNDLE address has been selected, the offerer does not need to include the 'bundle-only' attribute in subsequent offers. By assigning the offerer BUNDLE address to an "m=" line of a subsequent offer, the offerer will ensure that the answerer will either keep the "m=" line within the BUNDLE group, or the answerer will have to reject the "m=" line.

The usage of the 'bundle-only' attribute is only defined for a bundled "m=" line with a zero port value, within an offer. Other usage is unspecified.

Section 8 defines the detailed SDP Offer/Answer procedures for the 'bundle-only' attribute.

7. SDP Information Considerations

7.1. General

This section describes restrictions associated with the usage of SDP parameters within a BUNDLE group. It also describes, when parameter and attribute values have been associated with each bundled "m=" line, how to calculate a value for the whole BUNDLE group.

7.2. Connection Data (c=)

The "c=" line nettype value [RFC4566] associated with a bundled "m=" line MUST be 'IN'.

The "c=" line addrtype value [RFC4566] associated with a bundled "m=" line MUST be 'IP4' or 'IP6'. The same value MUST be associated with each "m=" line.

NOTE: Extensions to this specification can specify usage of the BUNDLE mechanism for other nettype and addrtype values than the ones listed above.

7.3. Bandwidth (b=)

An offerer and answerer MUST use the rules and restrictions defined in [I-D.mmusic-sdp-mux-attributes] for when associating the SDP bandwidth (b=) line with bundled "m=" lines.

7.4. Attributes (a=)

An offerer and answerer MUST use the rules and restrictions defined in [I-D.mmusic-sdp-mux-attributes] for when associating SDP attributes with bundled "m=" lines.

8. SDP Offer/Answer Procedures

8.1. General

This section describes the SDP Offer/Answer [RFC3264] procedures for:

- o Negotiating and creating of a BUNDLE group;
- o Selecting the BUNDLE addresses (offerer BUNDLE address and answerer BUNDLE address);
- o Adding an "m=" line to a BUNDLE group;
- o Moving an "m=" line out of a BUNDLE group; and
- o Disabling an "m=" line within a BUNDLE group.

The generic rules and procedures defined in [RFC3264] and [RFC5888] also apply to the BUNDLE extension. For example, if an offer is rejected by the answerer, the previously negotiated SDP parameters and characteristics (including those associated with a BUNDLE group) apply. Hence, if an offerer generates an offer in which the offerer wants to create a BUNDLE group, and the answerer rejects the offer, the BUNDLE group is not created.

The procedures in this section are independent of the media type or "m=" line proto value represented by a bundled "m=" line. Section 10 defines additional considerations for RTP based media. Section 6 defines additional considerations for the usage of the SDP 'bundle-only' attribute. Section 11 defines additional considerations for the usage of Interactive Connectivity Establishment (ICE) [RFC5245] mechanism .

SDP offers and answers can contain multiple BUNDLE groups. The procedures in this section apply independently to a given BUNDLE group.

8.2. Generating the Initial SDP Offer

8.2.1. General

When an offerer generates an initial offer, in order to create a BUNDLE group, it MUST:

- o Assign a unique address to each "m=" line within the offer, following the procedures in [RFC3264], unless the media line is a 'bundle-only' "m=" line (see below);
- o Add an SDP 'group:BUNDLE' attribute to the offer;
- o Place the identification-tag of each bundled "m=" line in the SDP 'group:BUNDLE' attribute identification-tag list; and
- o Indicate which unique address the offerer suggests as the offerer BUNDLE address [Section 8.2.2].

If the offerer wants to request that the answerer accepts a given bundled "m=" line only if the answerer keeps the "m=" line within the BUNDLE group, the offerer MUST:

- o Associate an SDP 'bundle-only' attribute [Section 8.2.2] with the "m=" line; and
- o Assign a zero port value to the "m=" line.

NOTE: If the offerer assigns a zero port value to an "m=" line, but does not also associate an SDP 'bundle-only' attribute with the "m=" line, it is an indication that the offerer wants to disable the "m=" line [Section 8.5.5].

[Section 17.1] shows an example of an initial offer.

8.2.2. Suggesting the offerer BUNDLE address

In the offer, the address assigned to the "m=" line associated with the offerer BUNDLE-tag indicates the address that the offerer suggests as the offerer BUNDLE address.

8.3. Generating the SDP Answer

8.3.1. General

When an answerer generates an answer that contains a BUNDLE group, the following general SDP grouping framework restrictions, defined in [RFC5888], also apply to the BUNDLE group:

- o The answerer MUST NOT include a BUNDLE group in the answer, unless the offerer requested the BUNDLE group to be created in the associated offer; and
- o The answerer MUST NOT include an "m=" line within a BUNDLE group, unless the offerer requested the "m=" line to be within that BUNDLE group in the associated offer.

If the answer contains a BUNDLE group, the answerer MUST:

- o Select an Offerer BUNDLE Address [Section 8.3.2]; and
- o Select an Answerer BUNDLE Address [Section 8.3.3];

The answerer is allowed to select a new Answerer BUNDLE address each time it generates an answer to an offer.

If the answerer does not want to keep an "m=" line within a BUNDLE group, it MUST:

- o Move the "m=" line out of the BUNDLE group [Section 8.3.4]; or
- o Reject the "m=" line [Section 8.3.5];

If the answerer keeps a bundle-only "m=" line within the BUNDLE group, it follows the procedures (assigns the answerer BUNDLE address to the "m=" line etc) for any other "m=" line kept within the BUNDLE group.

If the answerer does not want to keep a bundle-only "m=" line within the BUNDLE group, it MUST reject the "m=" line [Section 8.3.5].

The answerer MUST NOT associate an SDP 'bundle-only' attribute with any "m=" line in an answer.

NOTE: If a bundled "m=" line in an offer contains a zero port value, but the "m=" line does not contain an SDP 'bundle-only' attribute, it is an indication that the offerer wants to disable the "m=" line [Section 8.5.5].

8.3.2. Answerer Selection of Offerer Bundle Address

In an offer, the address (unique or shared) assigned to the bundled "m=" line associated with the offerer BUNDLE-tag indicates the address that the offerer suggests as the offerer BUNDLE address [Section 8.2.2]. The answerer MUST check whether that "m=" line fulfils the following criteria:

- o The answerer will not move the "m=" line out of the BUNDLE group [Section 8.3.4];
- o The answerer will not reject the "m=" line [Section 8.3.5]; and
- o The "m=" line does not contain a zero port value.

If all of the criteria above are fulfilled, the answerer MUST select the address associated with the "m=" line as the offerer BUNDLE address. In the answer, the answerer BUNDLE-tag represents the "m=" line, and the address associated with the "m=" line in the offer becomes the offerer BUNDLE address.

If one or more of the criteria are not fulfilled, the answerer MUST select the next identification-tag in the identification-tag list, and perform the same criteria check for the "m=" line associated with that identification-tag. If there are no more identification-tags in the identification-tag list, the answerer MUST NOT create the BUNDLE group. In addition, unless the answerer rejects the whole offer, the answerer MUST apply the answerer procedures for moving an "m=" line out of a BUNDLE group [Section 8.3.4] to each bundled "m=" line in the offer when creating the answer.

[Section 17.1] shows an example of an offerer BUNDLE address selection.

8.3.3. Answerer Selection of Answerer BUNDLE Address

When the answerer selects a BUNDLE address for itself, referred to as the answerer BUNDLE address, it MUST assign that address to each bundled "m=" line within the created BUNDLE group in the answer.

The answerer MUST NOT assign the answerer BUNDLE address to an "m=" line that is not within the BUNDLE group, or to an "m=" line that is within another BUNDLE group.

[Section 17.1] shows an example of an answerer BUNDLE address selection.

8.3.4. Moving A Media Description Out Of A BUNDLE Group

When an answerer wants to move an "m=" line out of a BUNDLE group, it MUST first check the following criteria:

- o In the associated offer, the "m=" line contains a shared address (e.g. a previously selected offerer BUNDLE address); or

- o In the associated offer, if an SDP 'bundle-only' attribute is associated with the "m=" line, and if the "m=" line contains a zero port value.

If either criteria above is fulfilled, the answerer MUST reject the "m=" line [Section 8.3.5].

Otherwise, if in the associated offer the "m=" line contains a unique address, the answerer MUST assign a unique address to the "m=" line in the answer (the answerer does not reject the "m=" line).

In addition, in either case above, the answerer MUST NOT place the identification-tag, associated with the moved "m=" line, in the SDP 'group' attribute identification-tag list associated with the BUNDLE group.

8.3.5. Rejecting A Media Description In A BUNDLE Group

When an answerer rejects an "m=" line, it MUST assign an address with a zero port value to the "m=" line in the answer, according to the procedures in [RFC4566].

In addition, the answerer MUST NOT place the identification-tag, associated with the rejected "m=" line, in the SDP 'group' attribute identification-tag list associated with the BUNDLE group.

8.4. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answer contains a BUNDLE group, the offerer MUST check that any bundled "m=" line in the answer was indicated as bundled in the associated offer. If there is no mismatch, the offerer MUST use the offerer BUNDLE address, selected by the answerer [Section 8.3.2], as the address for each bundled "m=" line.

NOTE: As the answerer might reject one or more bundled "m=" lines, or move a bundled "m=" line out of a BUNDLE group, each bundled "m=" line in the offer might not be indicated as bundled in the answer.

If the answer does not contain a BUNDLE group, the offerer MUST process the answer as a normal answer.

8.5. Modifying the Session

8.5.1. General

When an offerer generates a subsequent offer, it MUST assign the previously selected offerer BUNDLE address [Section 8.3.2], to each bundled "m=" line (including any bundle-only "m=" line), except if:

- o The offerer suggests a new offerer BUNDLE address [Section 8.5.2];
- o The offerer wants to add a bundled "m=" line to the BUNDLE group [Section 8.5.3];
- o The offerer wants to move a bundled "m=" line out of the BUNDLE group [Section 8.5.4]; or
- o The offerer wants to disable the bundled "m=" line [Section 8.5.5].

In addition, the offerer MUST select an offerer BUNDLE-tag [Section 8.2.2] associated with the previously selected offerer BUNDLE address, unless the offerer suggests a new offerer BUNDLE address.

8.5.2. Suggesting a new offerer BUNDLE address

When an offerer generates an offer, in which it suggests a new offerer BUNDLE address [Section 8.2.2], the offerer MUST:

- o Assign the address (shared address) to each "m=" line within the BUNDLE group; or
- o Assign the address (unique address) to one bundled "m=" line.

In addition, the offerer MUST indicate that the address is the new suggested offerer BUNDLE address [Section 8.2.2].

NOTE: Unless the offerer assigns the new suggested offerer BUNDLE address to each bundled "m=" line, it can assign unique addresses to any number of bundled "m=" lines (and the previously selected offerer BUNDLE address to any remaining bundled "m=" line) if it wants to suggest multiple alternatives for the new offerer BUNDLE address.

8.5.3. Adding a media description to a BUNDLE group

When an offerer generates an offer, in which it wants to add a bundled "m=" line to a BUNDLE group, the offerer MUST:

- Assign a unique address to the "m=" line;
- Assign the previously selected offerer BUNDLE address to the "m=" line; or
- If the offerer assigns a new (shared address) suggested offerer BUNDLE address to each bundled "m=" line [Section 8.5.2], also assign that address to the added "m=" line.

In addition, the offerer MUST extend the SDP 'group:BUNDLE' attribute identification-tag list with the BUNDLE group [Section 8.2.2] by adding the identification-tag associated with the added "m=" line to the list.

NOTE: Assigning a unique address to the "m=" line allows the answerer to move the "m=" line out of the BUNDLE group [Section 8.3.4], without having to reject the "m=" line.

If the offerer assigns a unique address to the added "m=" line, and if the offerer suggests that address as the new offerer BUNDLE address [Section 8.5.2], the offerer BUNDLE-tag MUST represent the added "m=" line [Section 8.2.2].

If the offerer assigns a new suggested offerer BUNDLE address to each bundled "m=" line [Section 8.5.2], including the added "m=" line, the offerer BUNDLE-tag MAY represent the added "m=" line [Section 8.2.2].

[Section 17.3] shows an example where an offerer sends an offer in order to add a bundled "m=" line to a BUNDLE group.

8.5.4. Moving A Media Description Out Of A BUNDLE Group

When an offerer generates an offer, in which it wants to move a bundled "m=" line out of a BUNDLE group it was added to in a previous offer/answer transaction, the offerer:

- MUST assign a unique address to the "m=" line; and

- o MUST NOT place the identification-tag associated with the "m=" line in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group.

NOTE: If the removed "m=" line is associated with the previously selected BUNDLE-tag, the offerer needs to suggest a new BUNDLE-tag [Section 8.2.2].

NOTE: If an "m=" line, when being moved out of a BUNDLE group, is added to another BUNDLE group, the offerer applies the procedures in [Section 8.5.3] to the "m=" line.

[Section 17.4] shows an example of an offer for moving an "m=" line out of a BUNDLE group.

8.5.5. Disabling A Media Description In A BUNDLE Group

When an offerer generates an offer, in which it wants to disable a bundled "m=" line (added to the BUNDLE group in a previous offer/ answer transaction), the offerer:

- o MUST assign an address with a zero port value to the "m=" line, following the procedures in [RFC4566]; and

- o MUST NOT place the identification-tag associated with the "m=" line in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group.

[Section 17.5] shows an example of an offer for disabling an "m=" line within a BUNDLE group.

9. Protocol Identification

9.1. General

Each "m=" line within a BUNDLE group MUST use the same transport-layer protocol. If bundled "m=" lines use different protocols on top of the transport-layer protocol, there MUST exist a publicly available specification which describes a mechanism, for this particular protocol combination, how to associate received data with the correct protocol.

In addition, if received data can be associated with more than one bundled "m=" line, there MUST exist a publicly available specification which describes a mechanism for associating the received data with the correct "m=" line.

This document describes a mechanism to identify the protocol of received data among the STUN, DTLS and SRTP protocols (in any combination), when UDP is used as transport-layer protocol, but does not describe how to identify different protocols transported on DTLS. While the mechanism is generally applicable to other protocols and transport-layers protocols, any such use requires further specification around how to multiplex multiple protocols on a given transport-layer protocols, and how to associate received data with the correct protocols.

9.2. STUN, DTLS, SRTP

Section 5.1.2 of [RFC5764] describes a mechanism to identify the protocol of a received packet among the STUN, Datagram Transport Layer Security (DTLS) and SRTP protocols (in any combination). If an offer or answer includes bundled "m=" lines that represent these protocols, the offerer or answerer MUST support the mechanism described in [RFC5764], and no explicit negotiation is required in order to indicate support and usage of the mechanism.

[RFC5764] does not describe how to identify different protocols transported on DTLS, only how to identify the DTLS protocol itself. If multiple protocols are transported on DTLS, there MUST exist a specification describing a mechanism for identifying each individual protocol. In addition, if a received DTLS packet can be associated with more than one "m=" line, there MUST exist a specification which describes a mechanism for associating the received DTLS packet with the correct "m=" line.

[Section 10.2] describes how to associate a received (S)RTP packet with the correct "m=" line.

10. RTP Considerations

10.1. Single RTP Session

10.1.1. General

All RTP-based media within a single BUNDLE group belong to a single RTP session [RFC3550]. Disjoint BUNDLE groups will form multiple RTP sessions, one per BUNDLE group.

Since a single RTP session is used for each bundle group, all "m=" lines representing RTP-based media in a bundle group will share a single SSRC numbering space [RFC3550].

The following rules and restrictions apply for a single RTP session:

o A specific payload type value can be used in multiple bundled "m=" lines if each codec associated with the payload type number shares an identical codec configuration [Section 10.1.2].

o The proto value in each bundled RTP-based "m=" line MUST be identical (e.g. RTP/AVPF).

o The RTP MID header extension MUST be enabled, by associating an SDP 'extmap' attribute [RFC5285], with a 'urn:ietf:params:rtp-hdrext:sdes:mid' URI value, with each bundled RTP-based "m=" line in every offer and answer.

o A given SSRC MUST NOT transmit RTP packets using payload types that originate from different bundled "m=" lines.

NOTE: The last bullet above is to avoid sending multiple media types from the same SSRC. If transmission of multiple media types are done with time overlap, RTP and RTCP fail to function. Even if done in proper sequence this causes RTP Timestamp rate switching issues [RFC7160]. However, once an SSRC has left the RTP session (by sending an RTCP BYE packet), that SSRC value can later be reused by another source(possible associated with a different bundled "m=" line.

10.1.2. Payload Type (PT) Value Reuse

Multiple bundled "m=" lines might represent RTP based media. As all RTP based media associated with a BUNDLE group belong to the same RTP session, in order for a given payload type value to be used inside more than one bundled "m=" line, all codecs associated with the payload type number MUST share an identical codec configuration. This means that the codecs MUST share the same media type, encoding name, clock rate and any parameter that can affect the codec configuration and packetization. [I-D.mmusic-sdp-mux-attributes] lists SDP attributes, whose attribute values must be identical for all codecs that use the same payload type value.

10.2. Associating RTP/RTCP Packets With Correct SDP Media Description

There are multiple mechanisms that can be used by an endpoint in order to associate received RTP/RTCP packets with a bundled "m=" line. Such mechanisms include using the payload type value carried inside the RTP packets, the SSRC values carried inside the RTP packets, and other "m=" line specific information carried inside the RTP packets.

As all RTP/RTCP packets associated with a BUNDLE group are received (and sent) using single address:port combinations, the local address:port combination cannot be used to associate received RTP packets with the correct "m=" line.

As described in [Section 10.1.2], the same payload type value might be used inside RTP packets described by multiple "m=" lines. In such cases, the payload type value cannot be used to associate received RTP packets with the correct "m=" line.

An offerer and answerer can inform each other which SSRC values they will use for RTP and RTCP by using the SDP 'ssrc' attribute [RFC5576]. To allow for proper association with this mechanism, the 'ssrc' attribute needs to be associated with each "m=" line that shares a payload type with any other "m=" line in the same bundle. As the SSRC values will be carried inside the RTP/RTCP packets, the offerer and answerer can then use that information to associate received RTP packets with the correct "m=" line. However, an offerer will not know which SSRC values the answerer will use until it has received the answer providing that information. Due to this, before the offerer has received the answer, the offerer will not be able to associate received RTP/RTCP packets with the correct "m=" line using the SSRC values.

In order for an offerer and answerer to always be able to associate received RTP and RTCP packets with the correct "m=" line, an offerer and answerer using the BUNDLE extension MUST support the mechanism defined in Section 14, where the remote endpoint inserts the identification-tag associated with an "m=" line in RTP and RTCP packets associated with that "m=" line.

10.3. RTP/RTCP Multiplexing

10.3.1. General

When a BUNDLE group, which contains RTP based media, is created, the offerer and answerer MUST negotiate whether to enable RTP/RTCP multiplexing for the RTP based media associated with the BUNDLE group [RFC5761].

If RTP/RTCP multiplexing is enabled, the same address:port combination will be used for sending all RTP packets and the RTCP packets associated with the BUNDLE group. Each endpoint will send the packets towards the BUNDLE address of the other endpoint. The same address:port combination MAY be used for receiving RTP packets and RTCP packets.

If RTP/RTCP multiplexing is not enabled, separate address:port combinations will be used for sending the RTP packets and the RTCP packets. The same address:port combinations MAY be used for receiving RTP packets and RTCP packets. If the remote endpoint has associated an SDP 'rtcp' attribute with the "m=" line associated with the BUNDLE-tag, the attribute value will be used for sending all RTCP packets associated with the BUNDLE group towards that endpoint.

10.3.2. SDP Offer/Answer Procedures

10.3.2.1. General

This section describes how an offerer and answerer can use the SDP 'rtcp-mux' attribute [RFC5761] and the SDP 'rtcp' attribute [RFC3605] to negotiate usage of RTP/RTCP multiplexing for RTP based media associated with a BUNDLE group.

10.3.2.2. Generating the Initial SDP Offer

When an offerer generates an initial offer, if the offerer wants to negotiate usage of RTP/RTCP multiplexing within a BUNDLE group, the offerer MUST associate an SDP 'rtcp-mux' attribute [RFC5761] with each bundled RTP-based "m=" line (including any bundle-only "m=" line) in the offer.

If the offerer does not want to negotiate usage of RTP/RTCP multiplexing, it MUST NOT associate an SDP 'rtcp-mux' attribute with any bundled "m=" line in the offer.

In addition, the offerer can associate an SDP 'rtcp' attribute [RFC3605] with one or more bundled RTP-based "m=" lines (including any bundle-only "m=" line) in the offer, in order to provide a port for receiving RTCP packets (if the answerer does not accept usage of RTP/RTCP multiplexing, or if the offerer does not want to negotiate usage of RTP/RTCP multiplexing).

In the initial offer, the IP address and port combination for RTCP MUST be unique in each bundled RTP-based "m=" line, similar to RTP.

NOTE: In case the offer wants to receive RTCP packets on the next higher port value, the SDP 'rtcp' attribute is not needed.

10.3.2.3. Generating the SDP Answer

When an answerer generates an answer, if the offerer indicated support of RTP/RTCP multiplexing [RFC5761] within a BUNDLE group in the associated offer, the answerer MUST either accept or reject the usage of RTP/RTCP multiplexing for the whole BUNDLE group in the answer.

If the answerer accepts the usage of RTP/RTCP multiplexing within the BUNDLE group, it MUST associate an SDP 'rtcp-mux' attribute with each bundled RTP-based "m=" line in the answer. The answerer MUST NOT associate an SDP 'rtcp' attribute with any bundled "m=" line in the answer. The answerer will use the port value of the selected offerer BUNDLE address for sending RTP and RTCP packets associated with each RTP-based bundled "m=" line towards the offerer.

If the answerer does not accept the usage of RTP/RTCP multiplexing within the BUNDLE group, it MUST NOT associate an SDP 'rtcp-mux' attribute with any bundled "m=" line in the answer. The answerer will use the RTP and RTCP port values associated with the selected offerer BUNDLE address for sending RTP and RTCP packets associated with each RTP-based bundled "m=" line towards the offerer.

In addition, if the answerer rejects the usage of RTP/RTCP multiplexing within the BUNDLE group, it MAY associate an SDP 'rtcp' attribute, with identical attribute values, with each RTP-based bundled "m=" line in the answer, in order to provide a port value for receiving RTCP packets from the offerer.

NOTE: In case the answerer wants to receive RTCP packets on the next higher port value, the SDP 'rtcp' attribute is not needed.

If the usage of RTP/RTCP multiplexing within a BUNDLE group has been negotiated in a previous offer/answer transaction, and if the offerer indicates that it wants to continue using RTP/RTCP multiplexing in a subsequent offer, the answerer MUST associate an SDP 'rtcp-mux' attribute with each bundled "m=" line in the answer. I.e. the answerer MUST NOT disable the usage of RTP/RTCP multiplexing.

If the usage of RTP/RTCP multiplexing within a BUNDLE group has not been negotiated in a previous offer/answer transaction, and if the offerer indicates that it wants to use RTP/RTCP multiplexing in a subsequent offer, the answerer either accepts or rejects the usage, using the procedures above.

10.3.2.4. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answerer has accepted the usage of RTP/RTCP multiplexing (see Section 10.3.2.3), the answerer follows the procedures for RTP/RTCP multiplexing defined in [RFC5761]. The offerer will use the port value associated with the answerer BUNDLE address for sending RTP and RTCP packets associated with each RTP-based bundled "m=" line towards the answerer.

If the answerer did not accept the usage of RTP/RTCP multiplexing (see Section 10.3.2.3), the offerer will use separate address:port combinations for sending RTP and RTCP packets towards the answerer. If the answerer associated an SDP 'rtcp' attribute with the "m=" line representing the answerer BUNDLE address, the offerer will use the attribute port value for sending RTCP packets associated with each bundled RTP-based "m=" line towards the answerer. Otherwise the offerer will use the next higher port value associated with the answerer BUNDLE address for sending RTCP packets towards the answerer.

10.3.2.5. Modifying the Session

When an offerer generates a subsequent offer, if it wants to negotiate the usage of RTP/RTCP multiplexing within a BUNDLE group, or if it wants to continue the use of previously negotiated RTP/RTCP multiplexing, it MUST associate an SDP 'rtcp-mux' attribute with each RTP-based bundled "m=" line (including any bundled "m=" line that the offerer wants to add to the BUNDLE group), unless the offerer wants to disable or remove the "m=" line from the BUNDLE group.

If the offerer does not want to negotiate the usage of RTP/RTCP multiplexing within the BUNDLE group, or if it wants to disable previously negotiated usage of RTP/RTCP multiplexing, it MUST NOT associate an SDP 'rtcp-mux' and attribute with any bundled "m=" line in the subsequent offer.

In addition, if the offerer does not indicate support of RTP/RTCP multiplexing within the subsequent offer, it MAY associate an SDP 'rtcp' attribute, with identical attribute values, with each RTP-based bundled "m=" line (including any bundled "m=" line that the offerer wants to add to the BUNDLE group), in order to provide a port for receiving RTCP packets.

NOTE: It is RECOMMENDED that, once the usage of RTP/RTCP multiplexing has been negotiated within a BUNDLE group, that the usage is not disabled. Disabling RTP/RTCP multiplexing means that the offerer and answerer need to reserve new ports, to be used for sending and receiving RTCP packets. Similar, if the usage of a specific RTCP port has been negotiated within a BUNDLE group, it is RECOMMENDED that the port value is not modified.

11. ICE Considerations

11.1. General

This section describes how to use the BUNDLE grouping extension together with the Interactive Connectivity Establishment (ICE) mechanism [RFC5245].

The procedures defined in [RFC5245] also apply to usage of ICE with BUNDLE, with the following exception:

o When BUNDLE addresses for a BUNDLE group have been selected for both endpoints, ICE connectivity checks and keep-alives only need to be performed for the whole BUNDLE group, instead of per bundled "m=" line.

Support and usage of ICE mechanism together with the BUNDLE extension is OPTIONAL.

11.2. SDP Offer/Answer Procedures

11.2.1. General

When an offerer assigns a unique address to a bundled "m=" line (excluding any bundle-only "m=" line), it MUST also associate unique ICE candidates [RFC5245] to the "m=" line.

An offerer MUST NOT assign ICE candidates to a bundle-only "m=" line with a zero port value.

NOTE: The bundle-only "m=" line, if accepted by the answerer, will inherit the candidates associated with the selected offerer BUNDLE address. An answerer that does not support BUNDLE would not accept a bundle-only "m=" line.

When an offerer or answerer assigns a shared address (i.e. a previously selected BUNDLE address) to one or more bundled "m=" lines, it MUST associate identical ICE candidates (referred to as shared ICE candidates) to each of those "m=" lines.

11.2.2. Generating the Initial SDP Offer

When an offerer generates an initial offer, it assigns unique or shared ICE candidates to the bundled "m=" lines, according to Section 11.1.

11.2.3. Generating the SDP Answer

When an answerer generates an answer that contains a BUNDLE group, the answerer MUST assign shared ICE candidates to each bundled "m=" line (including "m=" lines that were indicated as bundle-only in the associated offer) in the answer.

11.2.4. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answerer supports and uses the ICE mechanism and the BUNDLE extension, the offerer MUST assign the same ICE candidates, associated with the "m=" line representing the offerer BUNDLE address (selected by the answerer), to each bundled "m=" line.

11.2.5. Modifying the Session

When an offerer generates a subsequent offer, it assigns unique or shared ICE candidates to the bundled "m=" lines, according to (Section 11.1).

12. DTLS Considerations

One or more media streams within a BUNDLE group might use the Datagram Transport Layer Security (DTLS) protocol [RFC6347] in order to encrypt the data, or to negotiate encryption keys if another encryption mechanism is used to encrypt media.

When DTLS is used within a BUNDLE group, the following rules apply:

- o There can only be one DTLS association [RFC6347] associated with the BUNDLE group;

- o Each usage of the DTLS association within the BUNDLE group MUST use the same mechanism for determining which endpoints (the offerer or answerer) becomes DTLS client and DTLS server; and

- o If the DTLS client supports DTLS-SRTP [RFC5764] it MUST include the 'use_srtp' extension [RFC5764] in the DTLS ClientHello message [RFC5764], The client MUST include the extension even if the usage of DTLS-SRTP is not negotiated as part of the multimedia session (e.g. SIP session [RFC3261].

NOTE: The inclusion of the 'use_srtp' extension during the initial DTLS handshake ensures that a DTLS renegotiation will not be required in order to include the extension, in case DTLS-SRTP encrypted media is added to the BUNDLE group later during the multimedia session.

13. Update to RFC 3264

13.1. General

This section replaces the text of the following sections of RFC 3264:

- o Section 5.1 (Unicast Streams).

- o Section 8.2 (Removing a Media Stream).
- o Section 8.4 (Putting a Unicast Media Stream on Hold).

13.2. Original text of section 5.1 (2nd paragraph) of RFC 3264

For recvonly and sendrecv streams, the port number and address in the offer indicate where the offerer would like to receive the media stream. For sendonly RTP streams, the address and port number indirectly indicate where the offerer wants to receive RTCP reports. Unless there is an explicit indication otherwise, reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer indicates that the stream is offered but MUST NOT be used. This has no useful semantics in an initial offer, but is allowed for reasons of completeness, since the answer can contain a zero port indicating a rejected stream (Section 6). Furthermore, existing streams can be terminated by setting the port to zero (Section 8). In general, a port number of zero indicates that the media stream is not wanted.

13.3. New text replacing section 5.1 (2nd paragraph) of RFC 3264

For recvonly and sendrecv streams, the port number and address in the offer indicate where the offerer would like to receive the media stream. For sendonly RTP streams, the address and port number indirectly indicate where the offerer wants to receive RTCP reports. Unless there is an explicit indication otherwise, reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer by default indicates that the stream is offered but MUST NOT be used, but an extension mechanism might specify different semantics for the usage of a zero port value. Furthermore, existing streams can be terminated by setting the port to zero (Section 8). In general, a port number of zero by default indicates that the media stream is not wanted.

13.4. Original text of section 8.2 (2nd paragraph) of RFC 3264

A stream that is offered with a port of zero MUST be marked with port zero in the answer. Like the offer, the answer MAY omit all attributes present previously, and MAY list just a single media format from amongst those in the offer.

13.5. New text replacing section 8.2 (2nd paragraph) of RFC 3264

A stream that is offered with a port of zero MUST by default be marked with port zero in the answer, unless an extension mechanism, which specifies semantics for the usage of a non-zero port value, is used. If the stream is marked with port zero in the answer, the answer MAY omit all attributes present previously, and MAY list just a single media format from amongst those in the offer."

13.6. Original text of section 8.4 (6th paragraph) of RFC 3264

RFC 2543 [10] specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. Its usage for putting a call on hold is no longer recommended, since it doesn't allow for RTCP to be used with held streams, doesn't work with IPv6, and breaks with connection oriented media. However, it can be useful in an initial offer when the offerer knows it wants to use a particular set of media streams and formats, but doesn't know the addresses and ports at the time of the offer. Of course, when used, the port number MUST NOT be zero, which would specify that the stream has been disabled. An agent MUST be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP should be sent to the peer.

13.7. New text replacing section 8.4 (6th paragraph) of RFC 3264

RFC 2543 [10] specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. Its usage for putting a call on hold is no longer recommended, since it doesn't allow for RTCP to be used with held streams, doesn't work with IPv6, and breaks with connection oriented media. However, it can be useful in an initial offer when the offerer knows it wants to use a particular set of media streams and formats, but doesn't know the addresses and ports at the time of the offer. Of course, when used, the port number MUST NOT be zero, if it would specify that the stream has been disabled. However, an extension mechanism might specify different semantics of the zero port number usage. An agent MUST be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP should be sent to the peer.

14. RTP/RTCP extensions for identification-tag transport

14.1. General

SDP Offerers and Answerers [RFC3264] can associate identification-tags with "m=" lines within SDP Offers and Answers, using the procedures in [RFC5888]. Each identification-tag uniquely represents an "m=" line.

This section defines a new RTCP SDES item [RFC3550], 'MID', which is used to carry identification-tags within RTCP SDES packets. This section also defines a new RTP header extension [RFC5285], which is used to carry identification-tags in RTP packets.

The SDES item and RTP header extension make it possible for a receiver to associate received RTCP- and RTP packets with a specific "m=" line, to which the receiver has assigned an identification-tag, even if those "m=" lines are part of the same RTP session. A media recipient informs the media sender about the identification-tag associated with an "m=" line through the use of an 'mid' attribute [RFC5888]. The media sender then inserts the identification-tag in RTCP and RTP packets sent to the media recipient.

NOTE: This text above defines how identification-tags are carried in SDP Offers and Answers. The usage of other signalling protocols for carrying identification-tags is not prevented, but the usage of such protocols is outside the scope of this document.

[RFC3550] defines general procedures regarding the RTCP transmission interval. The RTCP MID SDES item SHOULD be sent in the first few RTCP packets sent on joining the session, and SHOULD be sent regularly thereafter. The exact number of RTCP packets in which this SDES item is sent is intentionally not specified here, as it will depend on the expected packet loss rate, the RTCP reporting interval, and the allowable overhead.

The RTP MID header extension SHOULD be included in some RTP packets at the start of the session and whenever the SSRC changes. It might also be useful to include the header extension in RTP packets that comprise random access points in the media (e.g., with video I-frames). The exact number of RTP packets in which this header extension is sent is intentionally not specified here, as it will depend on expected packet loss rate and loss patterns, the overhead the application can tolerate, and the importance of immediate receipt of the identification-tag.

For robustness purpose, endpoints need to be prepared for situations where the reception of the identification-tag is delayed, and SHOULD NOT terminate sessions in such cases, as the identification-tag is likely to arrive soon.

14.2. RTCP MID SDES Item

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      MID=TBD  |     length    | identification-tag          ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The identification-tag payload is UTF-8 encoded, as in SDP.

The identification-tag is not zero terminated.

[RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.]

14.3. RTP MID Header Extension

The payload, containing the identification-tag, of the RTP MID header extension element can be encoded using either the one-byte or two-byte header [RFC5285]. The identification-tag payload is UTF-8 encoded, as in SDP.

The identification-tag is not zero terminated. Note, that set of header extensions included in the packet needs to be padded to the next 32-bit boundary using zero bytes [RFC5285].

As the identification-tag is included in either an RTCP SDES item or an RTP header extension, or both, there should be some consideration about the packet expansion caused by the identification-tag. To avoid Maximum Transmission Unit (MTU) issues for the RTP packets, the header extension's size needs to be taken into account when the encoding media.

It is recommended that the identification-tag is kept short. Due to the properties of the RTP header extension mechanism, when using the one-byte header, a tag that is 1-3 bytes will result in that a minimal number of 32-bit words are used for the RTP header extension, in case no other header extensions are included at the same time. Note, do take into account that some single characters when UTF-8 encoded will result in multiple octets.

15. IANA Considerations

15.1. New SDES item

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

[RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.]

This document adds the MID SDES item to the IANA "RTCP SDES item types" registry as follows:

```
Value:          TBD
Abbrev.:        MID
Name:           Media Identification
Reference:      RFCXXXX
```

15.2. New RTP Header Extension URI

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document defines a new extension URI in the RTP Compact Header Extensions subregistry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

```
Extension URI: urn:ietf:params:rtp-hdrext:sdes:mid
Description:   Media identification
Contact:       christer.holmberg@ericsson.com
Reference:     RFCXXXX
```

15.3. New SDP Attribute

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document defines a new SDP media-level attribute, 'bundle-only', according to the following data:

```
Attribute name:     bundle-only
Type of attribute:  media
Subject to charset: No
Purpose:            Request a media description to be accepted
                in the answer only if kept within a BUNDLE
                group by the answerer.
Appropriate values: N/A
Contact name:       Christer Holmberg
Contact e-mail:     christer.holmberg@ericsson.com
Reference:          RFCXXXX
```

15.4. New SDP Group Semantics

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document registers the following semantics with IANA in the "Semantics for the "group" SDP Attribute" subregistry (under the "Session Description Protocol (SDP) Parameters" registry:

Semantics	Token	Reference
Media bundling	BUNDLE	[RFCXXXX]

16. Security Considerations

The security considerations defined in [RFC3264] and [RFC5888] apply to the BUNDLE extension. Bundle does not change which information flows over the network but only changes which ports that information is flowing on and thus has very little impact on the security of the RTP sessions.

When the BUNDLE extension is used, a single set of security credentials might be used for all media streams associated with a BUNDLE group.

When the BUNDLE extension is used, the number of SSRC values within a single RTP session increases, which increases the risk of SSRC collision. [RFC4568] describes how SSRC collision may weaken SRTP and SRTCP encryption in certain situations.

17. Examples

17.1. Example: Bundle Address Selection

The example below shows:

- o 1. An offer, in which the offerer assigns a unique address to each bundled "m=" line within the BUNDLE group.

- o 2. An answer, in which the answerer selects the offerer BUNDLE address, and in which selects its own BUNDLE address (the answerer BUNDLE address) and assigns it each bundled "m=" line within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

17.2. Example: BUNDLE Extension Rejected

The example below shows:

- o 1. An offer, in which the offerer assigns a unique address to each bundled "m=" line within the BUNDLE group.

- o 2. An answer, in which the answerer rejects the offered BUNDLE group, and assigns a unique addresses to each "m=" line (following normal RFC 3264 procedures).

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
m=audio 20000 RTP/AVP 0
b=AS:200
a=rtpmap:0 PCMU/8000
m=video 30000 RTP/AVP 32
b=AS:1000
a=rtpmap:32 MPV/90000
```

17.3. Example: Offerer Adds A Media Description To A BUNDLE Group

The example below shows:

o 1. A subsequent offer (the BUNDLE group has been created as part of a previous offer/answer transaction), in which the offerer adds a new "m=" line, represented by the "zen" identification-tag, to a previously negotiated BUNDLE group, assigns a unique address to the added "m=" line, and assigns the previously selected offerer

BUNDLE address to each of the other bundled "m=" lines within the BUNDLE group.

o 2. An answer, in which the answerer assigns the answerer BUNDLE address to each bundled "m=" line (including the newly added "m=" line) within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar zen
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10000 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtpmap:66 H261/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar zen
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
```

```
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtpmap:66 H261/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

17.4. Example: Offerer Moves A Media Description Out Of A BUNDLE Group

The example below shows:

- 1. A subsequent offer (the BUNDLE group has been created as part of a previous offer/answer transaction), in which the offerer moves a bundled "m=" line out of a BUNDLE group, assigns a unique address to the moved "m=" line, and assigns the offerer BUNDLE address to each other bundled "m=" line within the BUNDLE group.

- 2. An answer, in which the answerer moves the "m=" line out of the BUNDLE group, assigns unique address to the moved "m=" line, and assigns the answerer BUNDLE address to each of the remaining bundled "m=" line within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10000 RTP/AVP 31 32
b=AS:1000
a=mid:bar
```

```
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 50000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtpmap:66 H261/90000
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 60000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtpmap:66 H261/90000
```

17.5. Example: Offerer Disables A Media Description Within A BUNDLE Group

The example below shows:

- o 1. A subsequent offer (the BUNDLE group has been created as part of a previous offer/answer transaction), in which the offerer disables a bundled "m=" line within BUNDLE group, assigns a zero port number to the disabled "m=" line, and assigns the offerer BUNDLE address to each of the other bundled "m=" lines within the BUNDLE group.

- o 2. An answer, in which the answerer moves the disabled "m=" line out of the BUNDLE group, assigns a zero port value to the disabled "m=" line, and assigns the answerer BUNDLE address to each of the remaining bundled "m=" line within the BUNDLE group.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 10000 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 0 RTP/AVP 66
a=mid:zen
a=rtpmap:66 H261/90000
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtpmap:0 PCMU/8000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
m=video 20000 RTP/AVP 32
b=AS:1000
a=mid:bar
a=rtpmap:32 MPV/90000
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:mid
```

```
m=video 0 RTP/AVP 66
a=mid:zen
a=rtpmap:66 H261/90000
```

18. Acknowledgements

The usage of the SDP grouping extension for negotiating bundled media is based on a similar alternatives proposed by Harald Alvestrand and Cullen Jennings. The BUNDLE extension described in this document is based on the different alternative proposals, and text (e.g. SDP examples) have been borrowed (and, in some cases, modified) from those alternative proposals.

The SDP examples are also modified versions from the ones in the Alvestrand proposal.

Thanks to Paul Kyzivat, Martin Thomson, Flemming Andreasen, Thomas Stach, Ari Keranen, Adam Roach, Christian Groves, Roman Shpount, Suhas Nandakumar, Nils Ohlmeier, Jens Guballa, Raju Makaraju and Justin Uberti for reading the text, and providing useful feedback.

Thanks to Magnus Westerlund, Colin Perkins and Jonathan Lennox for providing help and text on the RTP/RTCP procedures.

Thanks to Spotify for providing music for the countless hours of document editing.

19. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-22

- - Correction of Ari’s family name

- - Editorial fixes based on comments from Thomas Stach

- - RTP/RTCP correction based on comment from Magnus Westerlund

- -- http://www.ietf.org/mail-archive/web/mmusic/current/msg14861.html

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-21

- - Correct based on comment from Paul Kyzivat o -- 'received packets' replaced with 'received data'

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-20

o - Clarification based on comment from James Guballa

o - Clarification based on comment from Flemming Andreasen

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-19

o - DTLS Considerations section added.

o - BUNDLE semantics added to the IANA Considerations

o - Changes based on WGLC comments from Adam Roach

o -- http://www.ietf.org/mail-archive/web/mmusic/current/msg14673.html

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-18

o - Changes based on agreements at IETF#92

o -- BAS Offer removed, based on agreement at IETF#92.

o -- Procedures regarding usage of SDP "b=" line is replaced with a reference to to draft-ietf-mmusic-sdp-mux-attributes.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-17

o - Editorial changes based on comments from Magnus Westerlund.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-16

o - Modification of RTP/RTCP multiplexing section, based on comments from Magnus Westerlund.

o - Reference updates.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-15

o - Editorial fix.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-14

o - Editorial changes.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-13 o Changes to allow a new suggested offerer BUNDLE address to be assigned to each bundled m- line.

o Changes based on WGLC comments from Paul Kyzivat

o - Editorial fixes

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-12

o Usage of SDP ’extmap’ attribute added

o SDP ’bundle-only’ attribute scoped with "m=" lines with a zero port value

o Changes based on WGLC comments from Thomas Stach

o - ICE candidates not assigned to bundle-only m- lines with a zero port value

o - Editorial changes

o Changes based on WGLC comments from Colin Perkins

o - Editorial changes:

o -- "RTP SDES item" -> "RTCP SDES item"

o -- "RTP MID SDES item" -> "RTCP MID SDES item"

o - Changes in section 10.1.1:

o -- "SHOULD NOT" -> "MUST NOT"

o -- Additional text added to the Note

o - Change to section 13.2:

o -- Clarify that mid value is not zero terminated

o - Change to section 13.3:

o -- Clarify that mid value is not zero terminated

o -- Clarify padding

o Changes based on WGLC comments from Paul Kyzivat

o - Editorial changes:

o Changes based on WGLC comments from Jonathan Lennox

o - Editorial changes:

o - Defintion of SDP bundle-only attribute alligned with structure in 4566bis draft

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-11

o Editorial corrections based on comments from Harald Alvestrand.

o Editorial corrections based on comments from Cullen Jennings.

o Reference update (RFC 7160).

o Clarification about RTCP packet sending when RTP/RTCP multiplexing is not used (http://www.ietf.org/mail-archive/web/mmusic/current/msg13765.html).

o Additional text added to the Security Considerations.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-10

o SDP bundle-only attribute added to IANA Considerations.

o SDES item and RTP header extension added to Abstract and Introduction.

o Modification to text updating section 8.2 of RFC 3264.

o Reference corrections.

o Editorial corrections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-09

o Terminology change: "bundle-only attribute assigned to m= line" to "bundle-only attribute associated with m= line".

o Editorial corrections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-08

o Editorial corrections.

o - "of"->"if" (8.3.2.5).

o - "optional"->"OPTIONAL" (9.1).

o - Syntax/ABNF for 'bundle-only' attribute added.

o - SDP Offer/Answer sections merged.

o - 'Request new offerer BUNDLE address' section added

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-07

o OPEN ISSUE regarding Receiver-ID closed.

o - RTP MID SDES Item.

o - RTP MID Header Extension.

o OPEN ISSUE regarding insertion of SDP 'rtcp' attribute in answers closed.

o - Indicating that, when rtcp-mux is used, the answerer MUST NOT include an 'rtcp' attribute in the answer, based on the procedures in section 5.1.3 of RFC 5761.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-06

o Draft title changed.

o Added "SDP" to section names containing "Offer" or "Answer".

o Editorial fixes based on comments from Paul Kyzivat (http://www.ietf.org/mail-archive/web/mmusic/current/msg13314.html).

o Editorial fixed based on comments from Colin Perkins (http://www.ietf.org/mail-archive/web/mmusic/current/msg13318.html).

o - Removed text about extending BUNDLE to allow multiple RTP sessions within a BUNDLE group.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-05

o Major re-structure of SDP Offer/Answer sections, to align with RFC 3264 structure.

o Additional definitions added.

o - Shared address.

o - Bundled "m=" line.

- - Bundle-only "m=" line.

- - Offerer suggested BUNDLE mid.

- - Answerer selected BUNDLE mid.

- Q6 Closed (IETF#88): An Offerer MUST NOT assign a shared address to multiple "m=" lines until it has received an SDP Answer indicating support of the BUNDLE extension.

- Q8 Closed (IETF#88): An Offerer can, before it knows whether the Answerer supports the BUNDLE extension, assign a zero port value to a 'bundle-only' "m=" line.

- SDP 'bundle-only' attribute section added.

- Connection data nettype/addrtype restrictions added.

- RFC 3264 update section added.

- Indicating that a specific payload type value can be used in multiple "m=" lines, if the value represents the same codec configuration in each "m=" line.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-04

- Updated Offerer procedures (http://www.ietf.org/mail-archive/web/mmusic/current/msg12293.html).

- Updated Answerer procedures (http://www.ietf.org/mail-archive/web/mmusic/current/msg12333.html).

- Usage of SDP 'bundle-only' attribute added.

- Reference to Trickle ICE document added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-02

- Mechanism modified, to be based on usage of SDP Offers with both different and identical port number values, depending on whether it is known if the remote endpoint supports the extension.

- Cullen Jennings added as co-author.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-01

- No changes. New version due to expiration.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-00

- o No changes. New version due to expiration.

Changes from draft-holmberg-mmusic-sdp-multiplex-negotiation-00

- o Draft name changed.
- o Harald Alvestrand added as co-author.
- o "Multiplex" terminology changed to "bundle".
- o Added text about single versus multiple RTP Sessions.
- o Added reference to RFC 3550.

20. References

20.1. 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, <http://www.rfc-editor.org/info/rfc2119>.

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

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

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <http://www.rfc-editor.org/info/rfc5285>.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <http://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, <http://www.rfc-editor.org/info/rfc5888>.

[I-D.mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-08 (work in progress), January 2015.

20.2. Informative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[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, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <http://www.rfc-editor.org/info/rfc3605>.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <http://www.rfc-editor.org/info/rfc4568>.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, DOI 10.17487/RFC4961, July 2007, <http://www.rfc-editor.org/info/rfc4961>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <http://www.rfc-editor.org/info/rfc5576>.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010, <http://www.rfc-editor.org/info/rfc5764>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC7160] Petit-Huguenin, M. and G. Zorn, Ed., "Support for Multiple Clock Rates in an RTP Session", RFC 7160, DOI 10.17487/RFC7160, April 2014, <http://www.rfc-editor.org/info/rfc7160>.

[I-D.ietf-mmusic-trickle-ice] Ivov, E., Rescorla, E., and J. Uberti, "Trickle ICE: Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", draft-ietf-mmusic-trickle-ice-02 (work in progress), January 2015.

Appendix A. Design Considerations

A.1. General

One of the main issues regarding the BUNDLE grouping extensions has been whether, in SDP Offers and SDP Answers, the same port value should be inserted in "m=" lines associated with a BUNDLE group, as the purpose of the extension is to negotiate the usage of a single address:port combination for media associated with the "m=" lines. Issues with both approaches, discussed in the Appendix have been raised. The outcome was to specify a mechanism which uses SDP Offers with both different and identical port values.

Below are the primary issues that have been considered when defining the "BUNDLE" grouping extension:

- 1) Interoperability with existing UAs.
- 2) Interoperability with intermediary B2BUA- and proxy entities.
- 3) Time to gather, and the number of, ICE candidates.
- 4) Different error scenarios, and when they occur.
- 5) SDP Offer/Answer impacts, including usage of port number value zero.

NOTE: Before this document is published as an RFC, this Appendix might be removed.

A.2. UA Interoperability

Consider the following SDP Offer/Answer exchange, where Alice sends an SDP Offer to Bob:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 10000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 10002 RTP/AVP 97
a=rtpmap:97 H261/90000
```

SDP Answer

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
m=audio 20000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 20002 RTP/AVP 97
a=rtpmap:97 H261/90000
```

RFC 4961 specifies a way of doing symmetric RTP but that is an a later invention to RTP and Bob can not assume that Alice supports RFC 4961. This means that Alice may be sending RTP from a different port than 10000 or 10002 - some implementation simply send the RTP from an ephemeral port. When Bob’s endpoint receives an RTP packet, the only way that Bob know if it should be passed to the video or audio codec is by looking at the port it was received on. This lead some SDP implementations to use the fact that each "m=" line had a different port number to use that port number as an index to find the correct m line in the SDP. As a result, some implementations that do support symmetric RTP and ICE still use a SDP data structure where SDP with "m=" lines with the same port such as:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 10000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 10000 RTP/AVP 98
a=rtpmap:98 H261/90000
```

will result in the second "m=" line being considered an SDP error because it has the same port as the first line.

A.3. Usage of port number value zero

In an SDP Offer or SDP Answer, the media associated with an "m=" line can be disabled/rejected by setting the port number value to zero. This is different from e.g. using the SDP direction attributes, where RTCP traffic will continue even if the SDP "inactive" attribute is indicated for the associated "m=" line.

If each "m=" line associated with a BUNDLE group would contain different port values, and one of those port values would be used for a BUNDLE address associated with the BUNDLE group, problems would occur if an endpoint wants to disable/reject the "m=" line associated with that port, by setting the port value to zero. After that, no "m=" line would contain the port value which is used for the BUNDLE address. In addition, it is unclear what would happen to the ICE candidates associated with the "m=" line, as they are also used for the BUNDLE address.

A.4. B2BUA And Proxy Interoperability

Some back to back user agents may be configured in a mode where if the incoming call leg contains an SDP attribute the B2BUA does not understand, the B2BUS still generates that SDP attribute in the Offer for the outgoing call leg. Consider an B2BUA that did not understand the SDP "rtcp" attribute, defined in RFC 3605, yet acted this way. Further assume that the B2BUA was configured to tear down any call where it did not see any RTCP for 5 minutes. In this cases, if the B2BUA received an Offer like:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtcp:53020
```

It would be looking for RTCP on port 49172 but would not see any because the RTCP would be on port 53020 and after five minutes, it would tear down the call. Similarly, an SBC that did not understand BUNDLE yet put BUNDLE in it's offer may be looking for media on the wrong port and tear down the call. It is worth noting that a B2BUA that generated an Offer with capabilities it does not understand is not compliant with the specifications.

A.4.1. Traffic Policing

Sometimes intermediaries do not act as B2BUA, in the sense that they don't modify SDP bodies, nor do they terminate SIP dialogs. Still, however, they may use SDP information (e.g. IP address and port) in order to control traffic gating functions, and to set traffic policing rules. There might be rules which will trigger a session to be terminated in case media is not sent or received on the ports retrieved from the SDP. This typically occurs once the session is already established and ongoing.

A.4.2. Bandwidth Allocation

Sometimes intermediaries do not act as B2BUA, in the sense that they don't modify SDP bodies, nor do they terminate SIP dialogs. Still, however, they may use SDP information (e.g. codecs and media types) in order to control bandwidth allocation functions. The bandwidth allocation is done per "m=" line, which means that it might not be enough if media associated with all "m=" lines try to use that bandwidth. That may either simply lead to bad user experience, or to termination of the call.

A.5. Candidate Gathering

When using ICE, an candidate needs to be gathered for each port. This takes approximately 20 ms extra for each extra "m=" line due to the NAT pacing requirements. All of this gather can be overlapped with other things while the page is loading to minimize the impact.

If the client only wants to generate TURN or STUN ICE candidates for one of the "m=" lines and then use trickle ICE [I-D.ietf-mmusic-trickle-ice] to get the non host ICE candidates for the rest of the "m=" lines, it MAY do that and will not need any additional gathering time.

Some people have suggested a TURN extension to get a bunch of TURN allocation at once. This would only provide a single STUN result so in cases where the other end did not support BUNDLE, may cause more use of the TURN server but would be quick in the cases where both sides supported BUNDLE and would fall back to a successful call in the other cases.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Harald Tveit Alvestrand
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: harald@alvestrand.no

Cullen Jennings
Cisco
400 3rd Avenue SW, Suite 350
Calgary, AB T2P 4H2
Canada

Email: fluffy@iii.ca

Network Working Group B. Burman
Internet-Draft M. Westerlund
Intended status: Standards Track Ericsson
Expires: April 21, 2016 S. Nandakumar
M. Zanaty
Cisco
October 19, 2015

Using Simulcast in SDP and RTP Sessions
draft-ietf-mmusic-sdp-simulcast-03

Abstract

In some application scenarios it may be desirable to send multiple differently encoded versions of the same media source in different RTP streams. This is called simulcast. This document discusses the best way of accomplishing simulcast in RTP and how to signal it in SDP. A solution is defined by making an extension to SDP, and using RTP/RTCP identification methods to relate RTP streams belonging to the same media source. The SDP extension consists of a new media level SDP attribute that expresses capability to send and/or receive simulcast RTP streams. RTP/RTCP identification using either payload types or a separately defined method for RTP stream configuration are defined.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Most of today's multiparty video conference solutions make use of centralized servers to reduce the bandwidth and CPU consumption in the endpoints. Those servers receive RTP streams from each participant and send some suitable set of possibly modified RTP streams to the rest of the participants, which usually have heterogeneous capabilities (screen size, CPU, bandwidth, codec, etc). One of the biggest issues is how to perform RTP stream adaptation to different participants' constraints with the minimum possible impact on both video quality and server performance.

Simulcast is defined in this memo as the act of simultaneously sending multiple different encoded streams of the same media source, e.g. the same video source encoded with different video encoder types or image resolutions. This can be done in several ways and for different purposes. This document focuses on the case where it is desirable to provide a media source as multiple encoded streams over RTP [RFC3550] towards an intermediary so that the intermediary can provide the wanted functionality by selecting which RTP stream(s) to forward to other participants in the session, and more specifically how the identification and grouping of the involved RTP streams are done. From an RTP perspective, simulcast is a specific application of the aspects discussed in RTP Multiplexing Guidelines [I-D.ietf-avtcore-multiplex-guidelines].

This document describes a few scenarios where it is motivated to use simulcast, and also defines the needed SDP signaling for it.

2. Definitions

2.1. Terminology

This document makes use of the terminology defined in RTP Taxonomy [I-D.ietf-avtext-rtp-grouping-taxonomy], RTP Topology [RFC5117] and RTP Topologies Update [I-D.ietf-avtcore-rtp-topologies-update]. In addition, the following terms are used:

RTP Mixer: An RTP middle node, defined in [RFC5117] (Section 3.4: Topo-Mixer), further elaborated and extended with other topologies in [I-D.ietf-avtcore-rtp-topologies-update] (Section 3.6 to 3.9).

RTP Switch: A common short term for the terms "switching RTP mixer", "source projecting middlebox", and "video switching MCU" as discussed in [I-D.ietf-avtcore-rtp-topologies-update].

Simulcast Stream: One Encoded Stream or Dependent Stream from a set of concurrently transmitted Encoded Streams and optional Dependent Streams, all sharing a common Media Source, as defined in [I-D.ietf-avtext-rtp-grouping-taxonomy]. Decoding a Dependent Stream also requires the related (Dependent and) Encoded Stream(s), but in the context of simulcast that is considered a property of the Dependent Stream constituting the simulcast stream. For example, HD and thumbnail video simulcast versions of a single Media Source sent concurrently as separate RTP Streams.

Simulcast Format: Different formats of a simulcast stream serve the same purpose as alternative RTP payload types in non-simulcast SDP, to allow multiple alternative media formats for a given RTP Stream. As for multiple RTP payload types on the m-line, any one of the alternative formats can be used at a given point in time, but not more than one (based on RTP timestamp), and what format is used can change dynamically from one RTP packet to another. For example, if all participants in a group video call can decode H.264 and H.265 video, but only some can encode H.265, both H.264 and H.265 can be kept as alternative formats, and the format may dynamically switch between H.264 and H.265 as different participants become active speaker.

2.2. Requirements Language

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 RFC 2119 [RFC2119].

3. Use Cases

Many use cases of simulcast as described in this document relate to a multi-party communication session where one or more central nodes are used to adapt the view of the communication session towards individual participants, and facilitate the media transport between participants. Thus, these cases targets the RTP Mixer type of topology.

There are two principle approaches for an RTP Mixer to provide this adapted view of the communication session to each receiving participant:

- o Transcoding (decoding and re-encoding) received RTP streams with characteristics adapted to each receiving participant. This often include mixing or composition of media sources from multiple participants into a mixed media source originated by the RTP Mixer. The main advantage of this approach is that it achieves close to optimal adaptation to individual receiving participants.

The main disadvantages are that it can be very computationally expensive to the RTP Mixer and typically also degrades media Quality of Experience (QoE) such as end-to-end delay for the receiving participants.

- o Switching a subset of all received RTP streams or sub-streams to each receiving participant, where the used subset is typically specific to each receiving participant. The main advantages of this approach are that it is computationally cheap to the RTP Mixer and it has very limited impact on media QoE. The main disadvantage is that it can be difficult to combine a subset of received RTP streams into a perfect fit to the resource situation of a receiving participant.

The use of simulcast relates to the latter approach, where it is more important to reduce the load on the RTP Mixer and/or minimize QoE impact than to achieve an optimal adaptation of resource usage.

3.1. Reaching a Diverse Set of Receivers

The media sources provided by a sending participant potentially need to reach several receiving participants that differ in terms of available resources. The receiver resources that typically differ include, but are not limited to:

Codec: This includes codec type (such as SDP MIME type) and can include codec configuration options (e.g. SDP fmtp parameters). A couple of codec resources that differ only in codec configuration will be "different" if they are somehow not "compatible", like if they differ in video codec profile, or the transport packetization configuration.

Sampling: This relates to how the media source is sampled, in spatial as well as in temporal domain. For video streams, spatial sampling affects image resolution and temporal sampling affects video frame rate. For audio, spatial sampling relates to the number of audio channels and temporal sampling affects audio bandwidth. This may be used to suit different rendering capabilities or needs at the receiving endpoints, as well as a method to achieve different transport capabilities, bitrates and eventually QoE by controlling the amount of source data.

Bitrate: This relates to the amount of bits spent per second to transmit the media source as an RTP stream, which typically also affects the Quality of Experience (QoE) for the receiving user.

Letting the sending participant create a simulcast of a few differently configured RTP streams per media source can be a good tradeoff when using an RTP switch as middlebox, instead of sending a single RTP stream and using an RTP mixer to create individual transcodings to each receiving participant.

This requires that the receiving participants can be categorized in terms of available resources and that the sending participant can choose a matching configuration for a single RTP stream per category and media source.

For example, assume for simplicity a set of receiving participants that differ only in that some have support to receive Codec A, and the others have support to receive Codec B. Further assume that the sending participant can send both Codec A and B. It can then reach all receivers by creating two simulcasted RTP streams from each media source; one for Codec A and one for Codec B.

In another simple example, a set of receiving participants differ only in screen resolution; some are able to display video with at most 360p resolution and some support 720p resolution. A sending participant can then reach all receivers by creating a simulcast of RTP streams with 360p and 720p resolution for each sent video media source.

In more elaborate cases, the receiving participants differ both in available sampling and bitrate, and maybe also codec, and it is up to the RTP switch to find a good trade-off in which simulcasted stream to choose for each intended receiver. It is also the responsibility of the RTP switch to negotiate a good fit of simulcast streams with the sending participant.

The maximum number of simulcasted RTP streams that can be sent is mainly limited by the amount of processing and uplink network resources available to the sending participant.

3.2. Application Specific Media Source Handling

The application logic that controls the communication session may include special handling of some media sources. It is for example commonly the case that the media from a sending participant is not sent back to itself.

It is also common that a currently active speaker participant is shown in larger size or higher quality than other participants (the sampling or bitrate aspects of Section 3.1). Not sending the active speaker media back to itself means there is some other participant's media that instead has to receive special handling towards the active speaker; typically the previous active speaker. This way, the previously active speaker is needed both in larger size (to current active speaker) and in small size (to the rest of the participants), which can be solved with a simulcast from the previously active speaker to the RTP switch.

3.3. Receiver Media Source Preferences

The application logic that controls the communication session may allow receiving participants to apply preferences to the characteristics of the RTP stream they receive, for example in terms of the aspects listed in Section 3.1. Sending a simulcast of RTP streams is one way of accommodating receivers with conflicting or otherwise incompatible preferences.

4. Requirements

The following requirements need to be met to support the use cases in previous sections:

REQ-1: Identification. It must be possible to identify a set of simulcasted RTP streams as originating from the same media source:

REQ-1.1: In SDP signaling.

REQ-1.2: On RTP/RTCP level.

REQ-2: Transport usage. The solution must work when using:

REQ-2.1: Legacy SDP with separate media transports per SDP media description.

REQ-2.2: Bundled [I-D.ietf-mmusic-sdp-bundle-negotiation] SDP media descriptions.

REQ-3: Capability negotiation. It must be possible that:

REQ-3.1: Sender can express capability of sending simulcast.

REQ-3.2: Receiver can express capability of receiving simulcast.

REQ-3.3: Sender can express maximum number of simulcast streams that can be provided.

REQ-3.4: Receiver can express maximum number of simulcast streams that can be received.

REQ-3.5: Sender can detail the characteristics of the simulcast streams that can be provided.

REQ-3.6: Receiver can detail the characteristics of the simulcast streams that it prefers to receive.

REQ-4: Distinguishing features. It must be possible to have different simulcast streams use different codec parameters, as can be expressed by SDP format values and RTP payload types.

REQ-5: Compatibility. It must be possible to use simulcast in combination with other RTP mechanisms that generate additional RTP streams:

REQ-5.1: RTP Retransmission [RFC4588].

REQ-5.2: RTP Forward Error Correction [RFC5109].

REQ-5.3: Related payload types such as audio Comfort Noise and/or DTMF.

REQ-6: Interoperability. The solution must be possible to use in:

REQ-6.1: Interworking with non-simulcast legacy clients using a single media source per media type.

REQ-6.2: WebRTC "Unified Plan" environment with a single media source per SDP media description.

5. Overview

As an overview, the above requirements are met by signaling simulcast capability and configurations in SDP [RFC4566]:

- o An offer or answer can contain a number of simulcast streams, separate for send and receive directions.
- o An offer or answer can contain multiple, alternative simulcast streams in the same fashion as multiple, alternative codecs can be offered in a media description.
- o A single media source per SDP media description is assumed, which is aligned with the concepts defined in [I-D.ietf-avtext-rtp-grouping-taxonomy] and will specifically work in a WebRTC context, both with and without BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] grouping.
- o The codec configuration for a simulcast stream can be expressed in two alternative ways, with complementing drawbacks and benefits:

   * Through existing SDP formats (corresponding to RTP payload types), enabling the use of simulcast with a minimum set of additions to existing SDP specifications.

   * Through use of a separately specified RTP-level identification mechanism [I-D.pthatcher-mmusic-rid], which complements and effectively extends the available simulcast stream identification and configuration possibilities provided by using SDP formats.

- o It is possible, but not required to use source-specific signaling [RFC5576] with the proposed solution.

6. Detailed Description

This section further details the overview above (Section 5).

6.1. Simulcast Capability

Simulcast capability is expressed as a new media level SDP attribute, "a=simulcast". For each desired direction (send/recv), the simulcast attribute defines a list of simulcast streams (separated by semicolons), each of which is a list of simulcast formats (separated by commas). The meaning of the attribute on SDP session level is undefined and MUST NOT be used. The ABNF [RFC5234] for this attribute is:

```
sc-attr     = "a=simulcast:" 1*2( WSP sc-str-list ) [WSP sc-pause-list]
sc-str-list = sc-dir WSP sc-id-type "=" sc-alt-list *( ";" sc-alt-list )
sc-pause-list = "paused=" sc-alt-list
sc-dir      = "send" / "recv"
sc-id-type  = "pt" / "rid" / token
sc-alt-list = sc-id *( "," sc-id )
sc-id       = fmt / rid-identifier / token
; WSP defined in [RFC5234]
; fmt, token defined in [RFC4566]
; rid-identifier defined in [I-D.pthatcher-mmusic-rid]
```

Figure 1: ABNF for Simulcast

There are separate and independent sets of parameters for simulcast in send and receive directions. When listing multiple directions, each direction MUST NOT occur more than once on the same line.

Two simulcast stream identification methods are defined; "pt" using RTP payload type (SDP format), and "rid" using an additional RTP-level identification mechanism [I-D.pthatcher-mmusic-rid]. Different identification methods MUST NOT be used for different directions on a single "a=simulcast" line. Implementations that support both identification methods MAY include one "a=simulcast" line for each identification method for the same "m="-line. Multiple "a=simulcast" lines with the same identification method MUST NOT be used for a single "m="-line.

Attribute parameters are grouped by direction and consist of a listing of simulcast stream identifications to be used. The number of (non-alternative, see below) identifications in the list sets a limit to the number of supported simulcast streams in that direction. The order of the listed simulcast versions in the "send" direction suggests a proposed order of preference, in decreasing order: the stream listed first is the most preferred Section 3.1, and subsequent streams have progressively lower preference. The order of the listed simulcast streams in the "recv" direction expresses a preference which simulcast streams that are preferred, with the leftmost being most preferred. This can be of importance if the number of actually sent simulcast streams have to be reduced for some reason.

Formats that have explicit dependencies [RFC5583] [I-D.pthatcher-mmusic-rid] to other formats (even in the same media description) MAY be listed as different simulcast streams.

Alternative simulcast formats MAY be specified as part of the attribute parameters by expressing each simulcast stream as a comma-separated list of alternative format identifiers. In this case, there MUST NOT be any capability restriction in what alternative formats can be used across different simulcast streams, like requiring all simulcast streams to use the same codec format alternative. The order of the format alternatives within a simulcast stream is significant; the alternatives are listed from (left) most preferred to (right) least preferred. For the use of simulcast, this overrides the normal codec preference as expressed by format type ordering on the "m="-line, using regular SDP rules. This is to enable a separation of general codec preferences and simulcast stream configuration preferences.

A simulcast stream can use a codec defined such that the same RTP SSRC can change RTP payload type multiple times during a session, possibly even on a per-packet basis. A typical example can be a speech codec that makes use of Comfort Noise [RFC3389] and/or DTMF [RFC4733] formats. In those cases, such "related" formats MUST NOT be listed explicitly in the attribute parameters, since they are not strictly simulcast streams of the media source, but rather a specific way of generating the RTP stream of a single simulcast stream with varying RTP payload type. Instead, only a single simulcast stream identification MUST be used per simulcast stream or alternative simulcast format (if there are such) in the SDP. The used simulcast stream identification SHOULD be the codec format most relevant to the media description, if possible to identify, for example the audio codec rather than the DTMF. What codec format to choose in the case of switching between multiple equally "important" formats is left open, but it is assumed that in the presence of such strong relation it does not matter which is chosen.

If RTP stream pause/resume [I-D.ietf-avtext-rtp-stream-pause] is supported, the optional "paused=" parameter MAY be used in conjunction with "rid" simulcast stream identification to specify that a certain simulcast stream is initially paused already from start of the RTP session. In this case, support for RTP stream pause/resume MUST also be included under the same "m="-line listing "a=simulcast". Initially paused simulcast streams MUST NOT be used with "pt" identification. Initially paused simulcast streams are resumed as described by the RTP pause/resume specification.

An initially paused simulcast stream in "send" direction MUST be considered equivalent to an unsolicited locally paused stream, and be handled accordingly.

An initially paused simulcast stream in "recv" direction SHOULD cause the remote RTP sender to put the stream as unsolicited locally paused, unless there are other RTP stream receivers that do not mark the simulcast stream as initially paused. The reason to require an initially paused "recv" stream to be considered locally paused by the remote RTP sender, instead of making it equivalent to implicitly sending a pause request, is because the pausing RTP sender cannot know which SSRC owns the restriction when TMMBR/TMMBN are used for pause/resume signaling since the RTP receiver's SSRC in send direction is not known yet.

Use of the redundant audio data [RFC2198] format could be seen as a form of simulcast for loss protection purposes, but is not considered conflicting with the mechanisms described in this memo and MAY therefore be used as any other format. In this case the "red" format, rather than the carried formats, SHOULD be the one to list as a simulcast stream on the "a=simulcast" line.

6.1.1. Declarative Use

When used as a declarative media description, a=simulcast "recv" direction formats indicates the configured end point's required capability to recognize and receive a specified set of RTP streams as simulcast streams. In the same fashion, a=simulcast "send" direction requests the end point to send a specified set of RTP streams as simulcast streams.

If multiple simulcast formats are listed, it means that the configured end point MUST be prepared to receive any of the "recv" formats, and MAY send any of the "send" formats for that simulcast stream.

Editor’s note: The RID identification mechanism currently lacks a declarative use definition. As declarative use may also not follow unified plan with a single media source per ’"m="-line, it is uncertain if declarative can be defined for the mechanism in its current shape.

6.1.2. Offer/Answer Use

An offerer wanting to use simulcast SHALL include the "a=simulcast" attribute in the offer. An offerer that receives an answer without "a=simulcast" MUST NOT use simulcast towards the answerer. An offerer that receives an answer with "a=simulcast" not listing a direction or without any simulcast stream identifications in a specified direction MUST NOT use simulcast in that direction.

An answerer that does not understand the concept of simulcast will also not know the attribute and will remove it in the SDP answer, as defined in existing SDP Offer/Answer [RFC3264] procedures.

An answerer that does understand the attribute and that wants to support simulcast in an indicated direction SHALL reverse directionality of the unidirectional direction parameters; "send" becomes "recv" and vice versa, and include it in the answer. Note that, like all other use of SDP format tags ("pt:") for the send direction in Offer/Answer, format tags related to the simulcast stream identification send direction in an offer are placeholders that refer to information in the offer SDP, and the actual formats that will be used on the wire (including RTP Payload Format numbers) depends on information included in the SDP answer.

An offerer listing a set of receive simulcast streams and/or alternative formats in the offer MUST be prepared to receive RTP streams for any of those simulcast streams and/or alternative formats from the answerer.

An answerer that receives an offer with simulcast containing an "a=simulcast" attribute listing alternative formats for simulcast streams MAY keep all the alternatives in the answer, but it MAY also choose to remove any non-desirable alternatives per simulcast stream in the answer. The answerer MUST NOT add any alternatives that were not present in the offer.

An answerer that receives an offer with simulcast that lists a number of simulcast streams, MAY reduce the number of simulcast streams in the answer, but MUST NOT add simulcast streams.

An offerer that receives an answer where some simulcast formats are kept MUST be prepared to receive any of the kept send direction alternatives, and MAY send any of the kept receive direction alternatives from the answer. Similarly, the answerer MUST be prepared to receive any of the kept receive direction alternatives, and MAY send any of the kept send direction alternatives in the answer.

The offerer and answerer MUST NOT send more than a single alternative format at a time (based on RTP timestamps) per simulcast stream, but MAY change format on a per-RTP packet basis. This corresponds to the existing (non-simulcast) SDP offer/answer case when multiple formats are included on the "m="-line in the SDP answer.

An offerer that receives an answer where some of the simulcast streams are removed MAY release the corresponding resources (codec, transport, etc) in its receive direction and MUST NOT send any RTP streams corresponding to the removed simulcast streams.

Simulcast streams or formats using undefined simulcast stream identifications MUST NOT be used as valid simulcast streams by an RTP stream receiver.

An offerer that is capable of using both simulcast stream identification methods MAY include one "a=simulcast" line per identification method in the offer. Note that it is in general not expected that the "pt" identification method will provide feature parity with the "rid" method, and the different "a=simulcast" lines can therefore express different use of simulcast functionality. However, for some configurations the different identification methods can be equivalent.

An answerer receiving an offer listing both simulcast stream identification methods MUST choose only one and remove the other from the answer. An answerer not supporting a simulcast stream identification method in the offer MUST remove the non-supported "a=simulcast" line from the answer, possibly falling back to not using simulcast at all.

The media formats and corresponding characteristics of encoded streams used in a simulcast SHOULD be chosen such that they are different. If this difference is not required, RTP duplication [RFC7104] procedures SHOULD be considered instead of simulcast.

Note: The inclusion of "a=simulcast" or the use of simulcast does not change any of the interpretation or Offer/Answer procedures for other SDP attributes, like "a=fmtp" or "a=rid".

6.2. Relating Simulcast Streams

As long as there is only a single media source per SDP media description, simulcast RTP streams can be related on RTP level through the RTP payload type and (optionally) RID [I-D.pthatcher-mmusic-rid], as specified in the SDP "a=simulcast" attribute (Section 6.1) parameters. When using BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] with multiple SDP media descriptions to specify a single RTP session, there is an identification mechanism that allows relating RTP streams back to individual media descriptions, after which the above RTP payload type and RID relations can be used.

BUNDLE's MID is an RTCP source description (SDES) item. To ensure rapid initial reception, required to correctly process the RTP streams, it is also defined as an RTP header extension [RFC5285].

6.3. Signaling Examples

These examples describe a client to video conference service, using a centralized media topology with an RTP mixer.

```
+---+      +-----------+      +---+
| A |<---->|           |<---->| B |
+---+      |           |      +---+
           |   Mixer   |
+---+      |           |      +---+
| F |<---->|           |<---->| J |
+---+      +-----------+      +---+
```

Figure 2: Four-party Mixer-based Conference

6.3.1. Unified Plan Client

Alice is calling in to the mixer with a simulcast-enabled Unified Plan client capable of a single media source per media type. The client can send a simulcast of 2 video resolutions and frame rates: HD 1280x720p 30fps and thumbnail 320x180p 15fps. This is defined below using the "imageattr" [RFC6236]. Media formats (RTP payload types) are used as simulcast stream identification. Alice's Offer:

```
v=0
o=alice 2362969037 2362969040 IN IP4 192.0.2.156
s=Simulcast Enabled Unified Plan Client
t=0 0
c=IN IP4 192.0.2.156
m=audio 49200 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 49300 RTP/AVP 97 98
a=rtpmap:97 H264/90000
a=rtpmap:98 H264/90000
a=fmtp:97 profile-level-id=42c01f; max-fs=3600; max-mbps=108000
a=fmtp:98 profile-level-id=42c00b; max-fs=240; max-mbps=3600
a=imageattr:97 send [x=1280,y=720] recv [x=1280,y=720]
a=imageattr:98 send [x=320,y=180] recv [x=320,y=180]
a=simulcast: send pt=97;98 recv pt=97
```

Figure 3: Unified Plan Simulcast Offer

The only thing in the SDP that indicates simulcast capability is the line in the video media description containing the "simulcast" attribute. The included format parameters indicates that sent simulcast streams can differ in video resolution.

The Answer from the server indicates that it too is simulcast capable. Should it not have been simulcast capable, the "a=simulcast" line would not have been present and communication would have started with the media negotiated in the SDP.

```
v=0
o=server 823479283 1209384938 IN IP4 192.0.2.2
s=Answer to Simulcast Enabled Unified Plan Client
t=0 0
c=IN IP4 192.0.2.43
m=audio 49672 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 49674 RTP/AVP 97 98
a=rtpmap:97 H264/90000
a=rtpmap:98 H264/90000
a=fmtp:97 profile-level-id=42c01f; max-fs=3600; max-mbps=108000
a=fmtp:98 profile-level-id=42c00b; max-fs=240; max-mbps=3600
a=imageattr:97 send [x=1280,y=720] recv [x=1280,y=720]
a=imageattr:98 send [x=320,y=180] recv [x=320,y=180]
a=simulcast: recv pt=97;98 send pt=97
```

Figure 4: Unified Plan Simulcast Answer

Since the server is the simulcast media receiver, it reverses the direction of the "simulcast" attribute parameters.

6.3.2. Multi-Source Client

Fred is calling in to the same conference as in the example above with a two-camera, two-display system, thus capable of handling two separate media sources in each direction, where each media source is simulcast-enabled in the send direction. Fred's client is restricted to a single media source per media description.

The first two simulcast streams for the first media source use different codecs, H264-SVC [RFC6190] and H264 [RFC6184]. These two simulcast streams also have a temporal dependency. Two different video codecs, VP8 [I-D.ietf-payload-vp8] and H264, are offered as alternatives for the third simulcast stream for the first media source. RID is used as simulcast stream identification, reducing the number of media formats needed. Only the highest fidelity simulcast stream are sent from start, the lower fidelity streams being initially paused.

The second media source is offered with three different simulcast streams. All video streams of this second media source are loss protected by RTP retransmission [RFC4588]. RID is used as simulcast stream identification. Also here, all but the highest fidelity simulcast stream are initially paused.

Fred's client is also using BUNDLE to send all RTP streams from all media descriptions in the same RTP session on a single media transport. Although using many different simulcast streams in this example, use of RID as simulcast stream identification enables use of a low number of RTP payload types. Note that the use of both BUNDLE and RID recommends using the RTP header extension [RFC5285] for carrying these fields.

```
v=0
o=fred 238947129 823479223 IN IP4 192.0.2.125
s=Offer from Simulcast Enabled Multi-Source Client
t=0 0
c=IN IP4 192.0.2.125
a=group:BUNDLE foo bar zen

m=audio 49200 RTP/AVP 99
a=mid:foo
a=rtpmap:99 G722/8000

m=video 49600 RTP/AVPF 100 101 103
a=mid:bar
a=rtpmap:100 H264-SVC/90000
a=rtpmap:101 H264/90000
a=rtpmap:103 VP8/90000
a=fmtp:100 profile-level-id=42400d; max-fs=3600; max-mbps=108000; \
    mst-mode=NI-TC
a=fmtp:101 profile-level-id=42c00d; max-fs=3600; max-mbps=54000
a=fmtp:103 max-fs=900; max-fr=30
a=rid:1 send pt=100;max-width=1280;max-height=720;max-fr=60;depend=2
a=rid:2 send pt=101;max-width=1280;max-height=720;max-fr=30
a=rid:3 send pt=101;max-width=640;max-height=360
a=rid:4 send pt=103;max-width=640;max-height=360
a=depend:100 lay bar:101
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdrext:rid
a=rtcp-fb:* ccm pause nowait
a=simulcast: send rid=1;2;4,3 paused=2,3,4

m=video 49602 RTP/AVPF 96 104
a=mid:zen
a=rtpmap:96 VP8/90000
a=fmtp:96 max-fs=3600; max-fr=30
a=rtpmap:104 rtx/90000
a=fmtp:104 apt=96;rtx-time=200
a=rid:5 send pt=96;max-fs=921600;max-fr=30
a=rid:6 send pt=96;max-fs=614400;max-fr=15
a=rid:7 send pt=96;max-fs=230400;max-fr=30
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdrext:rid
a=rtcp-fb:* ccm pause nowait
a=simulcast: send rid=5;6;7 paused=6,7
```

Figure 5: Fred’s Multi-Source Simulcast Offer

Note: Empty lines in the SDP above are added only for readability and would not be present in an actual SDP.

7. Network Aspects

Simulcast is in this memo defined as the act of sending multiple alternative encoded streams of the same underlying media source. When transmitting multiple independent streams that originate from the same source, it could potentially be done in several different ways using RTP. A general discussion on considerations for use of the different RTP multiplexing alternatives can be found in Guidelines for Multiplexing in RTP [I-D.ietf-avtcore-multiplex-guidelines]. Discussion and clarification on how to handle multiple streams in an RTP session can be found in [I-D.ietf-avtcore-rtp-multi-stream].

The network aspects that are relevant for simulcast are:

Quality of Service: When using simulcast it might be of interest to prioritize a particular simulcast stream, rather than applying equal treatment to all streams. For example, lower bit-rate streams may be prioritized over higher bit-rate streams to minimize congestion or packet losses in the low bit-rate streams. Thus, there is a benefit to use a simulcast solution that supports QoS as good as possible.

NAT/FW Traversal: Using multiple RTP sessions incurs more cost for NAT/FW traversal unless they can re-use the same transport flow, which can be achieved by Multiplexing Negotiation Using SDP Port Numbers [I-D.ietf-mmusic-sdp-bundle-negotiation].

8. Limitations

The chosen approach has a few limitations that are described in this section. Some relate to the use of a single RTP session for all simulcast formats of a media source, while others relate to the two different simulcast stream identification methods.

8.1. Single RTP Session

The limitations in this section come from sending all simulcast streams related to a media source under the same SDP media description, which also means they are sent in the same RTP session.

It is not possible to use different simulcast streams on different transports, limiting the possibilities to apply different QoS to different simulcast streams. When using unicast, QoS mechanisms based on individual packet marking are feasible, since they do not require separation of simulcast streams into different RTP sessions to apply different QoS.

It is not possible to separate different simulcast streams into different multicast groups to allow a multicast receiver to pick the stream it wants, rather than receive all of them. In this case, the only reasonable implementation is to use different RTP sessions for each multicast group so that reporting and other RTCP functions operate as intended.

8.2. SDP Format Identification

The limitations in this section come from and thus apply only when using SDP format (RTP payload type) as simulcast stream identification method.

The available RTP payload type number space may not be sufficient when many different media formats and/or simulcast streams are used in the SDP. This can be particularly prominent when BUNDLE is used, and for any technology that adds to the number of required RTP payload types in a multiplicative way, such as for example adding RTP retransmission [RFC4588] and Forward Error Correction [RFC5109]. Flexible FEC Scheme [I-D.ietf-payload-flexible-fec-scheme] can be used for RTP retransmissions and would avoid the double consumption of the PT space that RTP Retransmission [RFC4588] causes.

Only existing SDP attributes and parameters can be used to define codec configuration for a simulcast format. Any codec that does not define a sufficient set of codec parameters in "a=fmtp", or can make use of other SDP attributes, may not be capable of expressing the desired simulcast format dimensions (Section 3.1) with necessary precision, or not at all. One example of this is the ability to separate simulcast formats by bandwidth for codecs lacking a codec-specific bandwidth parameter, since the SDP "b="-line covers all RTP payload types listed on an "m="-line.

A simulcast stream signaled as initially paused is not possible to resume by a remote peer, because it cannot know which target SSRC to use in the RESUME message [I-D.ietf-avtext-rtp-stream-pause].

8.3. RID Identification

The limitations in this section come from and thus apply only when using RID as simulcast stream identification method.

Use of the additional "a=rid"-line in SDP and the corresponding RID RTCP SDES item and RTP header extension requires some additional implementation complexity, and incurs some extra bandwidth cost to carry the RID RTCP SDES item and RTP header extension.

9. IANA Considerations

This document requests to register a new SDP attribute, simulcast.

Formal registrations to be written.

10. Security Considerations

The simulcast capability, configuration attributes and parameters are vulnerable to attacks in signaling.

A false inclusion of the "a=simulcast" attribute may result in simultaneous transmission of multiple RTP streams that would otherwise not be generated. The impact is limited by the media description joint bandwidth, shared by all simulcast streams irrespective of their number. There may however be a large number of unwanted RTP streams that will impact the share of bandwidth allocated for the originally wanted RTP stream.

A hostile removal of the "a=simulcast" attribute will result in simulcast not being used.

Neither of the above will likely have any major consequences and can be mitigated by signaling that is at least integrity and source authenticated to prevent an attacker to change it.

11. Contributors

Morgan Lindqvist and Fredrik Jansson, both from Ericsson, have contributed with important material to the first versions of this document. Robert Hansen and Cullen Jennings, from Cisco, and Peter Thatcher, from Google, contributed significantly to subsequent versions.

12. Acknowledgements

13. References

13.1. Normative References

[I-D.ietf-avtext-rtp-stream-pause]
Burman, B., Akram, A., Even, R., and M. Westerlund, "RTP Stream Pause and Resume", draft-ietf-avtext-rtp-stream-pause-10 (work in progress), September 2015.

[I-D.pthatcher-mmusic-rid]
Thatcher, P., Zanaty, M., Nandakumar, S., Burman, B., Roach, A., and B. Campen, "RTP Payload Format Constraints", draft-pthatcher-mmusic-rid-02 (work in progress), October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc4566>.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <http://www.rfc-editor.org/info/rfc5109>.

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

[RFC7104] Begen, A., Cai, Y., and H. Ou, "Duplication Grouping Semantics in the Session Description Protocol", RFC 7104, DOI 10.17487/RFC7104, January 2014, <http://www.rfc-editor.org/info/rfc7104>.

13.2. Informative References

[I-D.ietf-avtcore-multiplex-guidelines]
Westerlund, M., Perkins, C., and H. Alvestrand, "Guidelines for using the Multiplexing Features of RTP to Support Multiple Media Streams", draft-ietf-avtcore-multiplex-guidelines-03 (work in progress), October 2014.

[I-D.ietf-avtcore-rtp-multi-stream]
Lennox, J., Westerlund, M., Wu, W., and C. Perkins, "Sending Multiple Media Streams in a Single RTP Session", draft-ietf-avtcore-rtp-multi-stream-09 (work in progress), September 2015.

[I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-10 (work in progress), July 2015.

[I-D.ietf-avtext-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-ietf-avtext-rtp-grouping-taxonomy-08 (work in progress), July 2015.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-23 (work in progress), July 2015.

[I-D.ietf-payload-flexible-fec-scheme]
Singh, V., Begen, A., Zanaty, M., and G. Mandyam, "RTP Payload Format for Flexible Forward Error Correction (FEC)", draft-ietf-payload-flexible-fec-scheme-01 (work in progress), October 2015.

[I-D.ietf-payload-vp8]
Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", draft-ietf-payload-vp8-17 (work in progress), September 2015.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <http://www.rfc-editor.org/info/rfc2198>.

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

[RFC3389] Zopf, R., "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, DOI 10.17487/RFC3389, September 2002, <http://www.rfc-editor.org/info/rfc3389>.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <http://www.rfc-editor.org/info/rfc4588>.

[RFC4733] Schulzrinne, H. and T. Taylor, "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals", RFC 4733, DOI 10.17487/RFC4733, December 2006, <http://www.rfc-editor.org/info/rfc4733>.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, DOI 10.17487/RFC5117, January 2008, <http://www.rfc-editor.org/info/rfc5117>.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <http://www.rfc-editor.org/info/rfc5285>.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <http://www.rfc-editor.org/info/rfc5576>.

[RFC5583] Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", RFC 5583, DOI 10.17487/RFC5583, July 2009, <http://www.rfc-editor.org/info/rfc5583>.

[RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <http://www.rfc-editor.org/info/rfc6184>.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <http://www.rfc-editor.org/info/rfc6190>.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, DOI 10.17487/RFC6236, May 2011, <http://www.rfc-editor.org/info/rfc6236>.

Appendix A. Changes From Earlier Versions

NOTE TO RFC EDITOR: Please remove this section prior to publication.

A.1. Modifications Between WG Version -02 and -03

- o Removed text on multicast / broadcast from use cases, since it is not supported by the solution.

- o Removed explicit references to unified plan draft.

- Added possibility to initiate simulcast streams in paused mode.

- Enabled an offerer to offer multiple stream identification (pt or rid) methods and have the answerer choose which to use.

- Added a preference indication also in send direction offers.

- Added a section on limitations of the current proposal, including identification method specific limitations.

A.2. Modifications Between WG Version -01 and -02

- Relying on the new RID solution for codec constraints and configuration identification. This has resulted in changes in syntax to identify if pt or RID is used to describe the simulcast stream.

- Renamed simulcast version and simulcast version alternative to simulcast stream and simulcast format respectively, and improved definitions for them.

- Clarification that it is possible to switch between simulcast version alternatives, but that only a single one be used at any point in time.

- Changed the definition so that ordering of simulcast formats for a specific simulcast stream do have a preference order.

A.3. Modifications Between WG Version -00 and -01

- No changes. Only preventing expiry.

A.4. Modifications Between Individual Version -00 and WG Version -00

- Added this appendix.

Authors' Addresses

Bo Burman
Ericsson
Kistavagen 25
SE-164 80 Stockholm
Sweden

Email: bo.burman@ericsson.com

Magnus Westerlund
Ericsson
Farogatan 2
SE-164 80 Stockholm
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Suhas Nandakumar
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: snandaku@cisco.com

Mo Zanaty
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: mzanaty@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

P. Thatcher
Google
M. Zanaty
S. Nandakumar
Cisco Systems
B. Burman
Ericsson
A. Roach
B. Campen
Mozilla

October 19, 2015

RTP Payload Format Constraints
draft-pthatcher-mmusic-rid-02

Abstract

In this specification, we define a framework for identifying Source RTP Streams with the constraints on its payload format in the Session Description Protocol. This framework uses "rid" SDP attribute to: a) effectively identify the Source RTP Streams within a RTP Session, b) constrain their payload format parameters in a codec-agnostic way beyond what is provided with the regular Payload Types and c) enable unambiguous mapping between the Source RTP Streams to their media format specification in the SDP.

Note-1: The name 'rid' is not yet finalized. Please refer to Section 12 for more details on the naming.

Status of This Memo

Table of Contents

1. Introduction

Payload Type (PT) in RTP provides mapping between the format of the RTP payload and the media format description specified in the signaling. For applications that use SDP for signaling, the constructs rtpmap and/or fmtp describe the characteristics of the media that is carried in the RTP payload, mapped to a given PT.

Recent advances in standards such as RTCWEB and NETVC have given rise to rich multimedia applications requiring support for multiple RTP Streams with in a RTP session [I-D.ietf-mmusic-sdp-bundle-negotiation], [I-D.ietf-mmusic-sdp-simulcast] or having to support multiple codecs, for example. These demands have unearthed challenges inherent with:

- o The restricted RTP PT space in specifying the various payload configurations,
- o The codec-specific constructs for the payload formats in SDP,
- o Missing or underspecied payload format parameters,
- o Ambiguity in mapping between the individual Source RTP Streams and their equivalent format specification in the SDP.

This specification defines a new SDP framework for constraining Source RTP Streams (Section 2.1.10 [I-D.ietf-avtext-rtp-grouping-taxonomy]), called "Restriction Identifier (rid)", along with the SDP attributes to constrain their payload formats in a codec-agnostic way. The "rid" framework can be thought of as complementary extension to the way the media format parameters are specified in SDP today, via the "a=fmtp" attribute. This specification also proposes a new RTCP SDES item to carry the "rid" value, to provide correlation between the RTP Packets and their format specification in the SDP. This SDES item also uses the header extension mechanism [I-D.ietf-avtext-sdes-hdr-ext] to provide correlation at stream startup, or stream changes where RTCP isn't sufficient.

Note that the "rid" parameters only serve to further constrain the parameters that are established on a PT format. They do not relax any existing constraints.

As described in Section 7.2.1, this mechanism achieves backwards compatibility via the normal SDP processing rules, which require unknown a= parameters to be ignored. This means that implementations need to be prepared to handle successful offers and answers from other implementations that neither indicate nor honor the constraints requested by this mechanism.

Further, as described in Section 7 and its subsections, this mechanism achieves extensibility by: (a) having offerers include all supported constraints in their offer, abd (b) having answerers ignore a=rid lines that specify unknown constraints.

2. Key Words for Requirements

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]

3. Terminology

The terms Source RTP Stream, Endpoint, RTP Session, and RTP Stream are used as defined in [I-D.ietf-avtext-rtp-grouping-taxonomy].

[RFC4566] and [RFC3264] terminology is also used where appropriate.

4. Motivation

This section summarizes several motivations for proposing the "rid" framework.

1. RTP PT Space Exhaustion: [RFC3550] defines payload type (PT) that identifies the format of the RTP payload and determine its interpretation by the application. [RFC3550] assigns 7 bits for the PT in the RTP header. However, the assignment of static mapping of payload codes to payload formats and multiplexing of RTP with other protocols (such as RTCP) could result in limited number of payload type numbers available for the application usage. In scenarios where the number of possible RTP payload configurations exceed the available PT space within a RTP Session, there is need a way to represent the additional constraints on payload configurations and to effectively map a Source RTP Stream to its corresponding constraints.

1. Multi-source and Multi-stream Use Cases: Recently, there is a rising trend with real-time multimedia applications supporting multiple sources per endpoint with various temporal resolutions (Scalable Video Codec) and spatial resolutions (Simulcast) per source. These applications are being challenged by the limited

RTP PT space and/or by the underspecified SDP constructs for exercising granular control on configuring the individual Source RTP Streams.

5. SDP 'rid' Media Level Attribute

This section defines new SDP media-level attribute [RFC4566], "a=rid". Roughly speaking, this attribute takes the following form (see Section 10 for a formal definition).

```
a=rid:<rid-identifier> <direction> pt=<fmt-list>;<constraint>=<value>...
```

A given "a=rid" SDP media attribute specifies constraints defining an unique RTP payload configuration identified via the "rid-identifier". A set of codec-agnostic "rid-level" constraints are defined (Section 6) that describe the media format specification applicable to one or more Payload Types speicified by the "a=rid" line.

The 'rid' framework MAY be used in combination with the 'a=fmtp' SDP attribute for describing the media format parameters for a given RTP Payload Type. However in such scenarios, the 'rid-level' constraints (Section 6) further constrains the equivalent 'fmtp' attributes.

The 'direction' identifies the either 'send', 'recv' directionality of the Source RTP Stream.

A given SDP media description MAY have zero or more "a=rid" lines describing various possible RTP payload configurations. A given 'rid-identifier' MUST NOT be repeated in a given media description.

The 'rid' media attribute MAY be used for any RTP-based media transport. It is not defined for other transports.

Though the 'rid-level' attributes specified by the 'rid' property follow the syntax similar to session-level and media-level attributes, they are defined independently. All 'rid-level' attributes MUST be registered with IANA, using the registry defined in Section 13

Section 10 gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for the "rid" attribute.

The "a=rid" media attribute is not dependent on charset.

6. 'rid-level' constraints

This section defines the 'rid-level' constraints that can be used to constrain the RTP payload encoding format in a codec-agnostic way.

The following constraints are intended to apply to video codecs in a codec-independent fashion.

- max-width, for spatial resolution in pixels. In the case that stream orientation signaling is used to modify the intended display orientation, this attribute refers to the width of the stream when a rotation of zero degrees is encoded.

- max-height, for spatial resolution in pixels. In the case that stream orientation signaling is used to modify the intended display orientation, this attribute refers to the width of the stream when a rotation of zero degrees is encoded.

- max-fps, for frame rate in frames per second. For encoders that do not use a fixed framerate for encoding, this value should constrain the minimum amount of time between frames: the time between any two consecutive frames SHOULD NOT be less than 1/max-fps seconds.

- max-fs, for frame size in pixels per frame. This is the product of frame width and frame height, in pixels, for rectangular frames.

- max-br, for bit rate in bits per second. The restriction applies to the media payload only, and does not include overhead introduced by other layers (e.g., RTP, UDP, IP, or Ethernet). The exact means of keeping within this limit are left up to the implementation, and instantaneous excursions outside the limit are permissible. For any given one-second sliding window, however, the total number of bits in the payload portion of RTP SHOULD NOT exceed the value specified in "max-br."

- max-pps, for pixel rate in pixels per second. This value SHOULD be handled identically to max-fps, after performing the following conversion: max-fps = max-pps / (width * height). If the stream resolution changes, this value is recalculated. Due to this recalculation, excursions outside the specified maximum are possible during near resolution change boundaries.

All the constraints are optional and are subjected to negotiation based on the SDP Offer/Answer rules described in Section 7

This list is intended to be an initial set of constraints; future documents may define additional constraints; see Section 13.4. While this document doesn't define constraints for audio codecs, there is no reason such constraints should be precluded from definition and registration by other documents.

Section 10 provides formal Augmented Backus-Naur Form(ABNF) [RFC5234] grammar for each of the "rid-level" attributes defined in this section.

7. SDP Offer/Answer Procedures

This section describes the SDP Offer/Answer [RFC3264] procedures when using the 'rid' framework.

Note that 'rid's are only required to be unique within a media section ("m-line"); they do not necessarily need to be unique within an entire RTP session. In traditional usage, each media section is sent on its own unique 5-tuple, which provides an unambiguous scope. Similarly, when using BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], MID values associate RTP streams uniquely to a single media description.

7.1. Generating the Initial SDP Offer

For each media description in the offer, the offerer MAY choose to include one or more "a=rid" lines to specify a configuration profile for the given set of RTP Payload Types.

In order to construct a given "a=rid" line, the offerer must follow the below steps:

1. It MUST generate a 'rid-identifier' that is unique within a media description

2. It MUST set the direction for the 'rid-identifier' to one of 'send' or 'recv'

3. It MAY include a listing of SDP format tokens (usually corresponding to RTP payload types) to which the constraints expressed by the 'rid-level' attributes apply. Any Payload Types chosen MUST be a valid payload type for the media section (that is, it must be listed on the "m=" line).

4. The Offerer then chooses the 'rid-level' constraints (Section 6) to be applied for the rid, and adds them to the "a=rid" line. If it wishes the answer to have the ability to specify a constraint, but does not wish to set a value itself, it MUST include the name

of the constraint in the "a=rid" line, but without any indicated value.

Note: If an 'a=fmtp' attribute is also used to provide media-format-specific parameters, then the 'rid-level' attributes will further constrain the equivalent 'fmtp' parameters for the given Payload Type for those streams associated with the 'rid'.

If a given codec would require "a=fmtp" line when used without "a=rid" then the offer MUST include a valid corresponding "a=fmtp" line even when using RID.

7.2. Answerer processing the SDP Offer

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

7.2.1. 'rid' unaware Answerer

If the receiver doesn't support the 'rid' framework proposed in this specification, the entire "a=rid" line is ignored following the standard [RFC3264] Offer/Answer rules.

Section 7.1 requires the offer to include a valid "a=fmtp" line for any codecs that otherwise require it (in other words, the "a=rid" line cannot be used to replace "a=fmtp" configuration). As a result, ignoring the "a=rid" line is always guaranteed to result in a valid session description.

7.2.2. 'rid' aware Answerer

If the answerer supports 'rid' framework, the following steps are executed, in order, for each "a=rid" line in a given media description:

1. Extract the rid-identifier from the "a=rid" line and verify its uniqueness. In the case of a duplicate, the entire "a=rid" line, and all "a=rid" lines with rid-identifiers that duplicate this line, are rejected and MUST NOT be included in the SDP Answer.

2. If the "a=rid" line contains a "pt=" parameter, the list of payload types is verified against the list of valid payload types for the media section (that is, those listed on the "m=" line). If there is no match for the Payload Type listed in the "a=rid" line, then remove the "a=rid" line.

3. The answerer ensures that "rid-level" parameters listed are supported and syntactically well formed. In the case of a syntax error or an unsupported parameter, the "a=rid" line is removed.

4. If the 'depend' rid-level attribute is included, the answerer MUST make sure that the rid-identifiers listed unambiguously match the rid-identifiers in the SDP offer. Any lines that do not are removed.

5. if the "a=rid" line contains a "pt=" parameter, the answerer verifies that the attribute values provided in the "rid-level" attributes are consistent with the corrsponding codecs and their other parameters. See Section 9 for more detail. If the rid-level parameters are incompatible with the other codec properties, then the "a=rid" line is removed.

7.3. Generating the SDP Answer

Having performed the verification of the SDP offer as described, the answerer shall perform the following steps to generate the SDP answer.

For each "a=rid" line:

1. The answerer MAY choose to modify specific 'rid-level' attribute value in the answer SDP. In such a case, the modified value MUST be more constrained than the ones specified in the offer. The answer MUST NOT include any constraints that were not present in the offer.

2. The answerer MUST NOT modify the 'rid-identifier' present in the offer.

3. The answerer is allowed to remove one or more media formats from a given 'a=rid' line. If the answerer chooses to remove all the media format tokens from an "a=rid" line, the answerer MUST remove the entire "a=rid" line.

4. In cases where the answerer is unable to support the payload configuration specified in a given "a=rid" line in the offer, the answerer MUST remove the corresponding "a=rid" line. This includes situations in which the answerer does not understand one or more of the constraints in the "a=rid" line that has an associated value.

Note: in the case that the answerer uses different PT values to represent a codec than the offerer did, the "a=rid" values in the answer use the PT values that were sent in the offer.

7.4. Offering Processing of the SDP Answer

The offerer shall follow the steps similar to answerer's offer processing with the following exceptions

1. The offerer MUST ensure that the 'rid-identifiers' aren't changed between the offer and the answer. If so, the offerer MUST consider the corresponding 'a=rid' line as rejected.

2. If there exist changes in the 'rid-level' attribute values, the offerer MUST ensure that the modifications can be supported or else consider the "a=rid" line as rejected.

3. If the SDP answer contains any "rid-identifier" that doesn't match with the offer, the offerer MUST ignore the corresponding "a=rid" line.

4. If the "a=rid" line contains a "pt=" parameter, the offerer verifies that the list of payload types is a subset of those sent in the corresponding "a=rid" line in the offer.

5. If the "a=rid" line contains a "pt=" parameter, the offerer verifies that the attribute values provided in the "rid-level" attributes are consistent with the corrsponding codecs and their other parameters. See Section 9 for more detail. If the rid-level parameters are incompatible with the other codec properties, then the "a=rid" line is removed.

7.5. Modifying the Session

Offers and answers inside an existing session follow the rules for initial session negotiation. Such an offer MAY propose a change the number of RIDs in use. To avoid race conditions with media, any RIDs with proposed changes SHOULD use a new ID, rather than re-using one from the previous offer/answer exchange. RIDs without proposed changes SHOULD re-use the ID from the previous exchange.

8. Usage of 'rid' in RTP and RTCP

The RTP fixed header includes the payload type number and the SSRC values of the RTP stream. RTP defines how you de-multiplex streams within an RTP session, but in some use cases applications need further identifiers in order to effectively map the individual RTP Streams to their equivalent payload configurations in the SDP.

This specification defines a new RTCP SDES item [RFC3550], 'RID', which is used to carry rids within RTCP SDES packets. This makes it possible for a receiver to associate received RTP packets (identifying the Source RTP Stream) with a media description having the format constraint specified.

This specification also uses the RTP header extension for RTCP SDES items [I-D.ietf-avtext-sdes-hdr-ext] to allow carrying RID information in RTP packets to provide correlation at stream startup, or after stream changes where the use of RTCP may not be sufficiently responsive.

8.1. RTCP 'RID' SDES Extension

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      RID=TBD  |     length    | rid                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The rid payload is UTF-8 encoded and is not null-terminated.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

8.2. RTP 'rid' Header Extension

Because recipients of RTP packets will typically need to know which "a=rid" constraints they correspond to immediately upon receipt, this specification also defines a means of carrying RID identifiers in RTP extension headers, using the technique described in [I-D.ietf-avtext-sdes-hdr-ext].

As described in that document, the header extension element can be encoded using either the one-byte or two-byte header, and the identification-tag payload is UTF-8 encoded, as in SDP.

As the identification-tag is included in an RTP header extension, there should be some consideration about the packet expansion caused by the identification-tag. To avoid Maximum Transmission Unit (MTU) issues for the RTP packets, the header extension's size needs to be taken into account when the encoding media. Note that set of header extensions included in the packet needs to be padded to the next 32-bit boundary using zero bytes [RFC5285]

It is RECOMMENDED that the identification-tag is kept short. Due to the properties of the RTP header extension mechanism, when using the one-byte header, a tag that is 1-3 bytes will result in that a minimal number of 32-bit words are used for the RTP header extension, in case no other header extensions are included at the same time. In many cases, a one-byte tag will be sufficient; it is RECOMMENDED that implementations use the shortest identifier that fits their purposes.

9. Interaction with Other Techniques

Historically, a number of other approaches have been defined that allow constraining media streams via SDP parameters. These include:

- o Codec-specific configuration set via format parameters ("a=fmtp"); for example, the H.264 "max-fs" format parameter

- o Size restrictions imposed by image attribute attributes ("a=imgattr") [RFC6236]

When the mechanism described in this document is used in conjunction with these other restricting mechanisms, it is intended to impose additional restrictions beyond those communicated in other techniques.

In an offer, this means that a=rid lines, when combined with other restrictions on the media stream, are expected to result in a non-empty union. For example, if image attributes are used to indicate that a PT has a minimum width of 640, then specification of "max-width=320" in an "a=rid" line that is then applied to that PT is nonsensical. According to the rules of Section 7.2.2, this will result in the corresponding "a=rid" line being ignored by the recipient.

Similarly, an answer the a=rid lines, when combined with the other restrictions on the media stream, are also expected to result in a non-empty union. If the implementation generating an answer wishes to restrict a property of the stream below that which would be allowed by other parameters (e.g., those specified in "a=fmtp" or "a=imgattr"), its only recourse is to remove the "a=rid" line altogether, as described in Section 7.3. If it instead attempts to constrain the stream beyond what is allowed by other mechanisms, then the offerer will ignore the corresponding "a=rid" line, as described in Section 7.4.

10. Formal Grammar

This section gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for each of the new media and rid-level attributes defined in this document.

```
   rid-syntax        = "a=rid:" rid-identifier SP rid-dir
                       [ rid-pt-param-list / rid-param-list ]

   rid-identifier    = 1*(alpha-numeric / "-" / "_")

   rid-dir           = "send" / "recv"

   rid-pt-param-list = SP rid-fmt-list *(";" rid-param)

   rid-param-list    = SP rid-param *(";" rid-param)

   rid-fmt-list      = "pt=" fmt *( "," fmt )
                        ; fmt defined in {{RFC4566}}

   rid-param         = rid-width-param
                       / rid-height-param
                       / rid-fps-param
                       / rid-fs-param
                       / rid-br-param
                       / rid-pps-param
                       / rid-depend-param
                       / rid-param-other

   rid-width-param   = "max-width" [ "=" int-param-val ]

   rid-height-param  = "max-height" [ "=" int-param-val ]

   rid-fps-param     = "max-fps" [ "=" int-param-val ]

   rid-fs-param      = "max-fs" [ "=" int-param-val ]

   rid-br-param      = "max-br" [ "=" int-param-val ]

   rid-pps-param     = "max-pps" [ "=" int-param-val ]

   rid-depend-param  = "depend=" rid-list

   rid-param-other   = 1*(alpha-numeric / "-") [ "=" param-val ]

   rid-list          = rid-identifier *( "," rid-identifier )

   int-param-val     = 1*DIGIT

   param-val         = *( %x20-58 / %x60-7E )
                       ; Any printable character except semicolon
```

11. SDP Examples

Note: see [I-D.ietf-mmusic-sdp-simulcast] for examples of RID used in simulcast scenarios.

11.1. Many Bundled Streams using Many Codecs

In this scenario, the offerer supports the Opus, G.722, G.711 and DTMF audio codecs, and VP8, VP9, H.264 (CBP/CHP, mode 0/1), H.264-SVC (SCBP/SCHP) and H.265 (MP/M10P) for video. An 8-way video call (to a mixer) is supported (send 1 and receive 7 video streams) by offering 7 video media sections (1 sendrecv at max resolution and 6 recvonly at smaller resolutions), all bundled on the same port, using 3 different resolutions. The resolutions include:

- o 1 receive stream of 720p resolution is offered for the active speaker.
- o 2 receive streams of 360p resolution are offered for the prior 2 active speakers.
- o 4 receive streams of 180p resolution are offered for others in the call.

Expressing all these codecs and resolutions using 32 dynamic PTs (2 audio + 10x3 video) would exhaust the primary dynamic space (96-127). RIDs are used to avoid PT exhaustion and express the resolution constraints.

NOTE: The SDP given below skips few lines to keep the example short and focused, as indicated by either the "..." or the comments inserted.

Example 1

```
Offer:
...
m=audio 10000 RTP/SAVPF 96 9 8 0 123
a=rtpmap:96 OPUS/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:123 telephone-event/8000
a=mid:a1
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
a=rtpmap:98 VP8/90000
```

```
a=fmtp:98 max-fs=3600; max-fr=30
a=rtpmap:99 VP9/90000
a=fmtp:99 max-fs=3600; max-fr=30
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42401f; packetization-mode=0
a=rtpmap:101 H264/90000
a=fmtp:101 profile-level-id=42401f; packetization-mode=1
a=rtpmap:102 H264/90000
a=fmtp:102 profile-level-id=640c1f; packetization-mode=0
a=rtpmap:103 H264/90000
a=fmtp:103 profile-level-id=640c1f; packetization-mode=1
a=rtpmap:104 H264-SVC/90000
a=fmtp:104 profile-level-id=530c1f
a=rtpmap:105 H264-SVC/90000
a=fmtp:105 profile-level-id=560c1f
a=rtpmap:106 H265/90000
a=fmtp:106 profile-id=1; level-id=93
a=rtpmap:107 H265/90000
a=fmtp:107 profile-id=2; level-id=93
a=sendrecv
a=mid:v1 (max resolution)
a=rid:1 send max-width=1280;max-height=720;max-fps=30
a=rid:2 recv max-width=1280;max-height=720;max-fps=30
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
...same rtpmap/fmtp as above...
a=recvonly
a=mid:v2 (medium resolution)
a=rid:3 recv max-width=640;max-height=360;max-fps=15
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
...same rtpmap/fmtp as above...
a=recvonly
a=mid:v3 (medium resolution)
a=rid:3 recv max-width=640;max-height=360;max-fps=15
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
...same rtpmap/fmtp as above...
a=recvonly
a=mid:v4 (small resolution)
a=rid:4 recv max-width=320;max-height=180;max-fps=15
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
...same rtpmap/fmtp as above...
...same rid:4 as above for mid:v5,v6,v7 (small resolution)...
...
```

```
Answer:
...same as offer but swap send/recv...
```

11.2. Scalable Layers

Adding scalable layers to the above simulcast example gives the SFU further flexibility to selectively forward packets from a source that best match the bandwidth and capabilities of diverse receivers. Scalable encodings have dependencies between layers, unlike independent simulcast streams. RIDs can be used to express these dependencies using the "depend" parameter. In the example below, the highest resolution is offered to be sent as 2 scalable temporal layers (using MRST).

Example 3

```
Offer:
...
m=audio ...same as Example 1 ...
...
m=video ...same as Example 1 ...
...same rtpmap/fmtp as Example 1...
a=sendrecv
a=mid:v1 (max resolution)
a=rid:0 send max-width=1280;max-height=720;max-fps=15
a=rid:1 send max-width=1280;max-height=720;max-fps=30;depend=0
a=rid:2 recv max-width=1280;max-height=720;max-fps=30
a=rid:5 send max-width=640;max-height=360;max-fps=15
a=rid:6 send max-width=320;max-height=180;max-fps=15
a=simulcast: send rid=0;1;5;6 recv rid=2
...
...same m=video sections as Example1 for mid:v2-v7...
...

Answer:
...same as offer but swap send/recv...
```

12. Open Issues

12.1. Name of the identifier

The name ’rid’ is provisionally used and is open for further discussion.

Here are the few options that were considered while writing this draft

- o CID: Constraint ID, which is a rather precise description of what we are attempting to accomplish.

- o ESID: Encoded Stream ID, does not align well with taxonomy which defines Encoded Stream as before RTP packetization.

- o RSID or RID: RTP Stream ID, aligns better with taxonomy but very vague.

- o LID: Layer ID, aligns well for SVC with each layer in a separate stream, but not for other SVC layerings or independent simulcast which is awkward to view as layers.

- o EPT or XPT: EXtended Payload Type, conveys XPT.PT usage well, but may be confused with PT, for example people may mistakenly think they can use it in other places where PT would normally be used.

13. IANA Considerations

13.1. New RTP Header Extension URI

This document defines a new extension URI in the RTP Compact Header Extensions subregistry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

```
Extension URI: urn:ietf:params:rtp-hdrext:sdes:rid
Description:   RTP Stream Restriction Identifier
Contact:       <mmusic@ietf.org>
Reference:     RFCXXXX
```

13.2. New SDES item

RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

This document adds the MID SDES item to the IANA "RTCP SDES item types" registry as follows:

```
Value:          TBD
Abbrev.:        RID
Name:           Restriction Identification
Reference:      RFCXXXX
```

13.3. New SDP Media-Level attribute

This document defines "rid" as SDP media-level attribute. This attribute must be registered by IANA under "Session Description Protocol (SDP) Parameters" under "att-field (media level only)".

The "rid" attribute is used to identify characteristics of RTP stream with in a RTP Session. Its format is defined in Section 10.

13.4. Registry for RID-Level Parameters

This specification creates a new IANA registry named "att-field (rid level)" within the SDP parameters registry. The rid-level parameters MUST be registered with IANA and documented under the same rules as for SDP session-level and media-level attributes as specified in [RFC4566].

Parameters for "a=rid" lines that modify the nature of encoded media MUST be of the form that the result of applying the modification to the stream results in a stream that still complies with the other parameters that affect the media. In other words, parameters always have to restrict the definition to be a subset of what is otherwise allowable, and never expand it.

New parameter registrations are accepted according to the "Specification Required" policy of [RFC5226], provided that the specification includes the following information:

- contact name, email address, and telephone number
- parameter name (as it will appear in SDP)
- long-form parameter name in English
- whether the parameter value is subject to the charset attribute
- an explanation of the purpose of the parameter
- a specification of appropriate attribute values for this parameter
- an ABNF definition of the parameter

The initial set of rid-level parameter names, with definitions in Section 6 of this document, is given below:

Type	SDP Name	Reference
att-field	(rid level)	
	max-width	[RFCXXXX]
	max-height	[RFCXXXX]
	max-fps	[RFCXXXX]
	max-fs	[RFCXXXX]
	max-br	[RFCXXXX]
	max-pps	[RFCXXXX]
	depend	[RFCXXXX]

It is conceivable that a future document wants to define a RID-level parameter that contains string values. These extensions need to take care to conform to the ABNF defined for rid-param-other. In particular, this means that such extensions will need to define escaping mechanisms if they want to allow semicolons, unprintable characters, or byte values greater than 127 in the string.

OPEN ITEM: Do we need to do more than this regarding escaping?

14. Security Considerations

As with most SDP parameters, a failure to provide integrity protection over the a=rid attributes provides attackers a way to modify the session in potentially unwanted ways. This could result in an implementation sending greater amounts of data than a recipient wishes to receive. In general, however, since the "a=rid" attribute can only restrict a stream to be a subset of what is otherwise allowable, modification of the value cannot result in a stream that is of higher bandwidth than would be sent to an implementation that does not support this mechanism.

The actual identifiers used for RIDs are expected to be opaque. As such, they are not expected to contain information that would be sensitive, were it observed by third-parties.

15. Acknowledgements

Many thanks to review from Cullen Jennings, Magnus Westerlund, and Paul Kyzivat.

16. References

16.1. Normative References

[I-D.ietf-avtext-sdes-hdr-ext]
Westerlund, M., Burman, B., Even, R., and M. Zanaty, "RTP Header Extension for RTCP Source Description Items", draft-ietf-avtext-sdes-hdr-ext-02 (work in progress), July 2015.

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

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

[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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <http://www.rfc-editor.org/info/rfc5285>.

16.2. Informative References

[I-D.ietf-avtext-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-ietf-avtext-rtp-grouping-taxonomy-08 (work in progress), July 2015.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-23 (work in progress), July 2015.

[I-D.ietf-mmusic-sdp-simulcast]
Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in SDP and RTP Sessions", draft-ietf-mmusic-sdp-simulcast-02 (work in progress), October 2015.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, DOI 10.17487/RFC6236, May 2011, <http://www.rfc-editor.org/info/rfc6236>.

Authors’ Addresses

Peter Thatcher
Google

Email: pthatcher@google.com

Mo Zanaty
Cisco Systems

Email: mzanaty@cisco.com

Suhas Nandakumar
Cisco Systems

Email: snandaku@cisco.com

Bo Burman
Ericsson

Email: bo.burman@ericsson.com

Adam Roach
Mozilla

Email: adam@nostrum.com

Byron Campen
Mozilla

Email: bcampen@mozilla.com