Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 4, 2017

B. Burman
M. Westerlund
Ericsson
S. Nandakumar
M. Zanaty
Cisco

January 31, 2017

Using Simulcast in SDP and RTP Sessions
draft-ietf-mmusic-sdp-simulcast-07

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 describes how to accomplish simulcast in RTP and how to signal it in SDP. The described solution uses an RTP/RTCP identification method to identify RTP streams belonging to the same media source, and makes an extension to SDP to relate those RTP streams as being different simulcast formats of that media source. The SDP extension consists of a new media level SDP attribute that expresses capability to send and/or receive simulcast RTP streams.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Definitions 4
  2.1. Terminology 4
  2.2. Requirements Language 4
3. Use Cases 5
  3.1. Reaching a Diverse Set of Receivers 5
  3.2. Application Specific Media Source Handling 7
  3.3. Receiver Media Source Preferences 7
4. Requirements 7
5. Overview 9
6. Detailed Description 9
  6.1. Simulcast Attribute 10
  6.2. Simulcast Capability 11
  6.3. Offer/Answer Use 14
    6.3.1. Generating the Initial SDP Offer 14
    6.3.2. Creating the SDP Answer 14
    6.3.3. Offerer Processing the SDP Answer 15
    6.3.4. Modifying the Session 15
  6.4. Use with Declarative SDP 16
  6.5. Relating Simulcast Streams 16
  6.6. Signaling Examples 16
    6.6.1. Single-Source Client 17
    6.6.2. Multi-Source Client 18
7. RTP Aspects 21
  7.1. Outgoing from Endpoint with Media Source 21
  7.2. RTP Middlebox to Receiver 21
    7.2.1. Media-Switching Mixer 23
    7.2.2. Selective Forwarding Middlebox 24
  7.3. RTP Middlebox to RTP Middlebox 25
8. Network Aspects 26
  8.1. Bitrate Adaptation 26
9. Limitation 26
10. IANA Considerations 27
11. Security Considerations 28
12. Contributors 28
13. Acknowledgements 28

14. References 28
14.1. Normative References 28
14.2. Informative References 29
Appendix A. Changes From Earlier Versions 32
A.1. Modifications Between WG Version -05 and -06 32
A.2. Modifications Between WG Version -04 and -05 32
A.3. Modifications Between WG Version -03 and -04 33
A.4. Modifications Between WG Version -02 and -03 33
A.5. Modifications Between WG Version -01 and -02 33
A.6. Modifications Between WG Version -00 and -01 34
A.7. Modifications Between Individual Version -00 and WG Version -00 34
Authors' Addresses 34

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.

The intended scope of the defined mechanism is to support negotiation and usage of simulcast when using SDP offer/answer and media transport over RTP. The media transport topologies considered are point to point RTP sessions as well as centralized multi-party RTP sessions, where a media sender will provide the simulcasted streams to an RTP middlebox or endpoint, and middleboxes may further distribute the simulcast streams to other middleboxes or endpoints. Usage of multicast or broadcast transport is out of scope and left for future extension.

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

2. Definitions

2.1. Terminology

This document makes use of the terminology defined in RTP Taxonomy [RFC7656], and RTP Topologies [RFC7667]. The following terms are especially noted or here defined:

RTP Mixer: An RTP middle node, defined in [RFC7667] (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 [RFC7667].

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 [RFC7656]. 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 in offer/answer [RFC3264], any one of the negotiated alternative formats can be used in a single RTP stream at a given point in time, but not more than one (based on RTP timestamp). What format is used can change dynamically from one RTP packet to another.

Simulcast Stream Identifier (SCID): The identification value used to refer to an individual simulcast format, identical to the "rid-id" identification value for an RTP Payload Format Restriction [I-D.ietf-mmusic-rid] and the corresponding content of "RtpStreamId" RTCP SDES Item [I-D.ietf-avtext-rid].

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 target 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, typically degrades media Quality of Experience (QoE) such as end-to-end delay for the receiving participants, and requires RTP Mixer access to media content.

- 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, has very limited impact on media QoE, and does not require RTP Mixer (full) access to media content. 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 with best possible resolution 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:

REQ-1.1: It must be possible to identify a set of simulcasted RTP streams as originating from the same media source in SDP signaling.

REQ-1.2: An RTP endpoint must be capable of identifying the simulcast stream a received RTP stream is associated with, knowing the content of the SDP signalling.

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-5.4: A single simulcast stream can consist of multiple RTP streams, to support codecs where a dependent stream is dependent on a set of encoded and dependent streams, each potentially carried in their own RTP stream.

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 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]:

- An offer or answer can contain a number of simulcast streams, separate for send and receive directions.

- An offer or answer can contain multiple, alternative simulcast stream formats in the same fashion as multiple, alternative formats can be offered in a media description.

- A single media source per SDP media description is assumed, which is aligned with the concepts defined in [RFC7656] and will specifically work in a WebRTC context, both with and without BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] grouping.

- The codec configuration for a simulcast stream is expressed through use of separately specified RTP payload format restrictions [I-D.ietf-mmusic-rid] with an associated RTP-level identification mechanism [I-D.ietf-avtext-rid] to identify which RTP payload format restrictions an RTP stream adheres to. This complements and effectively extends simulcast stream identification and configuration possibilities that could be provided by using only SDP formats as identifier. Use of multiple RTP streams with the same (non-redundancy) media type in the context of a single media source, where those RTP streams are using different RtpStreamId, is a strong but not totally unambiguous indication of those RTP streams being part of a simulcast.

- It is possible to use source-specific signaling [RFC5576] with the proposed solution, but it is only in certain cases possible to learn from that signaling which SSRC will belong to a particular simulcast stream.

6. Detailed Description

This section further details the overview above (Section 5). First, formal syntax is provided (Section 6.1), followed by the rest of the SDP attribute definition in Section 6.2. Relating Simulcast Streams

(Section 6.5) provides the definition of the RTP/RTCP mechanisms used. The section is concluded with a number of examples.

6.1. Simulcast Attribute

This document defines a new SDP media-level "a=simulcast" attribute with the following ABNF [RFC5234] syntax:

```
sc-attr      = "a=simulcast:" sc-value
sc-value     = sc-str-list [SP sc-str-list]
sc-str-list  = sc-dir SP sc-alt-list *( ";" sc-alt-list )
sc-dir       = "send" / "recv"
sc-alt-list  = sc-id *( "," sc-id )
sc-id-paused = "~"
sc-id        = [sc-id-paused] rid-identifier
; SP defined in [RFC5234]
; rid-identifier defined in [I-D.ietf-mmusic-rid]
```

Figure 1: ABNF for Simulcast

The "a=simulcast" attribute has a parameter in the form of one or two simulcast stream descriptions, each consisting of a direction ("send" or "recv"), followed by a list of one or more simulcast streams. Each simulcast stream consists of one or more alternative simulcast formats. Each simulcast format is identified by a simulcast stream identifier (SCID). The SCID MUST have the form of an RTP stream identifier, as described by RTP Payload Format Restrictions [I-D.ietf-mmusic-rid].

In the list of simulcast streams, each simulcast stream is separated by a semicolon (";"). Each simulcast stream can in turn be offered in one or more alternative formats, represented by SCIDs, separated by a comma (","). Each SCID can also be specified as initially paused [RFC7728], indicated by prepending a "~" to the SCID. The reason to allow separate initial pause states for each SCID is that pause capability can be specified individually for each RTP payload type referenced by an SCID. Since pause capability specified via the "a=rtcp-fb" attribute and SCID specified by "a=rid" can refer to common payload types, it is unfeasible to pause streams with SCID where any of the related RTP payload type(s) do not have pause capability.

Examples:

```
a=simulcast:send 1,2,3;~4,~5 recv 6;~7,~8
a=simulcast:recv 1;4,5 send 6;7
```

Figure 2: Simulcast Examples

Above are two examples of different "a=simulcast" lines.

The first line is an example offer to send two simulcast streams and to receive two simulcast streams. The first simulcast stream in send direction can be sent in three different alternative formats (SCID 1, 2, 3), and the second simulcast stream in send direction can be sent in two different alternative formats (SCID 4, 5). Both of the second simulcast stream alternative formats in send direction are offered as initially paused. The first simulcast stream in receive direction has no alternative formats (SCID 6). The second simulcast stream in receive direction has two alternative formats (SCID 7, 8) that are both offered as initially paused.

The second line is an example answer to the first line, accepting to send and receive the two offered simulcast streams, however send and receive directions are specified in opposite order compared to the first line, which lets the answer keep the same order of simulcast streams in the SDP as in the offer, for convenience, even though directionality is reversed. This example answer has removed all offered alternative formats for the first simulcast stream (keeping only SCID 1), but kept alternative formats for the second simulcast stream in receive direction (4, 5). The answer thus accepts to send two simulcast streams, without alternatives. The answer does not accept initial pause of any simulcast streams, in either direction. More examples can be found in Section 6.6.

6.2. **Simulcast Capability**

Simulcast capability is expressed through a new media level SDP attribute, "a=simulcast" (Section 6.1). The meaning of the attribute on SDP session level is undefined, MUST NOT be used by implementations of this specification and MUST be ignored if received on session level. Extensions to this specification MAY define such session level usage. The meaning of including multiple "a=simulcast" lines in a single SDP media description is undefined, MUST NOT be used by implementations of this specification, and any additional "a=simulcast" lines beyond the first in a media description MUST be ignored if received.

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

Simulcast streams using undefined SCID MUST NOT be used as valid simulcast streams by an RTP stream receiver. The direction for an SCID MUST be aligned with the direction specified for the corresponding RTP stream identifier on the "a=rid" line.

The listed number of simulcast streams for a direction sets a limit to the number of supported simulcast streams in that direction. The order of the listed simulcast streams in the "send" direction suggests a proposed order of preference, in decreasing order: the SCID listed first is the most preferred and subsequent streams have progressively lower preference. The order of the listed SCID in the "recv" direction expresses 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.

SCID that have explicit dependencies [RFC5583] [I-D.ietf-mmusic-rid] to other SCID (even in the same media description) MAY be used.

Use of more than a single, alternative simulcast format for a simulcast stream MAY be specified as part of the attribute parameters by expressing the simulcast stream as a comma-separated list of alternative SCID. In this case, it is not possible to align what alternative SCID that are used across different simulcast streams, like requiring all simulcast streams to use SCID alternatives referring to the same codec format. The order of the SCID alternatives within a simulcast stream is significant; the SCID 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 defined as having their own SCID 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.

If RTP stream pause/resume [RFC7728] is supported, any SCID MAY be prefixed by a "~" character to indicate that the corresponding 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 where "a=simulcast" is included.

All RTP payload types related to such initially paused simulcast stream MUST be listed in the SDP as pause/resume capable as specified by [RFC7728], e.g. by using the "*" wildcard format for "a=rtcp-fb".

An initially paused simulcast stream in "send" direction MUST be considered equivalent to an unsolicited locally paused stream, and be handled accordingly. Initially paused simulcast streams are resumed as described by the RTP pause/resume specification. An RTP stream receiver that wishes to resume an unsolicited locally paused stream needs to know the SSRC of that stream. The SSRC of an initially paused simulcast stream can be obtained from an RTP stream sender RTCP Sender Report (SR) including both the desired SSRC as "SSRC of sender", and the SCID value in an RtpStreamId RTCP SDES item [I-D.ietf-avtext-rid].

Including an initially paused simulcast stream in "recv" direction in an SDP towards an RTP sender, 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 receiving SSRC owns the restriction when TMMBR/TMMBN are used for pause/resume signaling since the RTP receiver's SSRC in send direction is sometimes not yet known.

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.

The media formats and corresponding characteristics of simulcast streams SHOULD be chosen such that they are different, e.g. as different SDP formats with differing "a=rtpmap" and/or "a=fmtp" lines, or as differently defined RTP payload format restrictions. If this difference is not required, RTP duplication [RFC7104] procedures SHOULD be considered instead of simulcast. To avoid complications in implementations, a single SCID MUST NOT occur more than once per "a=simulcast" line. Note that this does not eliminate use of simulcast as an RTP duplication mechanism, since it is possible to define multiple different SCID that are effectively equivalent.

6.3. Offer/Answer Use

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".

6.3.1. Generating the Initial SDP Offer

An offerer wanting to use simulcast SHALL include the "a=simulcast" attribute in the offer. An offerer listing a set of receive simulcast streams and/or alternative formats as SCID in the offer MUST be prepared to receive RTP streams for any of those simulcast streams and/or alternative formats from the answerer.

6.3.2. Creating the SDP Answer

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. Similarly, an answerer that receives an offer with the "a=simulcast" attribute on session level SHALL remove it in the answer. An answerer that understands the attribute but receives multiple "a=simulcast" attributes in the same media description and that desires to use simulcast SHALL ignore and remove all but the first in the answer.

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.

An answerer that receives an offer with simulcast containing an "a=simulcast" attribute listing alternative SCID MAY keep all the alternative SCID in the answer, but it MAY also choose to remove any non-desirable alternative SCID in the answer. The answerer MUST NOT add any alternative SCID in send direction in the answer that were not present in the offer receive direction. The answerer MUST be prepared to receive any of the receive direction SCID alternatives, and MAY send any of the send direction alternatives that are kept in the answer.

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 answerer that receives an offer without RTP stream pause/resume capability MUST NOT mark any simulcast streams as initially paused in the answer.

An RTP stream pause/resume capable answerer that receives an offer with RTP stream pause/resume capability MAY mark any SCID that refer to pause/resume capable formats as initially paused in the answer.

An answerer that receives indication in an offer of an SCID being initially paused SHOULD mark that SCID as initially paused also in the answer, regardless of direction, unless it has good reason for the SCID not being initially paused. One such reason could, for example, be that the answerer would otherwise initially not receive any media of that type at all.

6.3.3. Offerer Processing the SDP Answer

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" without any SCID in a specified direction MUST NOT use simulcast in that direction.

An offerer that receives an answer where some SCID alternatives are kept MUST be prepared to receive any of the kept send direction SCID alternatives, and MAY send any of the kept receive direction SCID alternatives.

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

An offerer that offered some of its SCID as initially paused and that receives an answer that does not indicate RTP stream pause/resume capability, MUST NOT initially pause any simulcast streams.

An offerer with RTP stream pause/resume capability that receives an answer where some SCID are marked as initially paused, SHOULD initially pause those RTP streams regardless if they were marked as initially paused also in the offer, unless it has good reason for those RTP streams not being initially paused. One such reason could, for example, be that the answerer would otherwise initially not receive any media of that type at all.

6.3.4. Modifying the Session

Offers and answers inside an existing session follow the rules for initial session negotiation, with the additional restriction that any SCID marked as initially paused in such offer or answer MUST already be paused, thus a new offer/answer MUST NOT replace use of RTP stream pause/resume [RFC7728] in the session. Session modification

restrictions in section 6.5 of RTP payload format restrictions [I-D.ietf-mmusic-rid] also apply.

6.4. Use with Declarative SDP

This document does not define the use of "a=simulcast" in declarative SDP, partly motivated by use of the simulcast format identification [I-D.ietf-mmusic-rid] not being defined for use in declarative SDP. If concrete use cases for simulcast in declarative SDP are identified in the future, we expect that additional specifications will address such use.

6.5. Relating Simulcast Streams

Simulcast RTP streams MUST be related on RTP level through RtpStreamId [I-D.ietf-avtext-rid], as specified in the SDP "a=simulcast" attribute (Section 6.2) parameters. This is sufficient as long as there is only a single media source per SDP media description. When using BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], where multiple SDP media descriptions jointly specify a single RTP session, the SDES MID identification mechanism in BUNDLE allows relating RTP streams back to individual media descriptions, after which the above described RtpStreamId relations can be used. Use of the RTP header extension [RFC5285] for both MID and RtpStreamId identifications can be important to ensure rapid initial reception, required to correctly interpret and process the RTP streams. Implementers of this specification MUST support the RTCP source description (SDES) item method and SHOULD support RTP header extension method to signal RtpStreamId on RTP level.

RTP streams MUST only use a single alternative SCID at a time (based on RTP timestamps), but MAY change format (and SCID) 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, enabling per-RTP packet change of RTP payload type.

6.6. 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 3: Four-party Mixer-based Conference

6.6.1. Single-Source Client

Alice is calling in to the mixer with a simulcast-enabled 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]. In this example, only the "pt" "a=rid" parameter is used, effectively achieving a 1:1 mapping between RtpStreamId and media formats (RTP payload types), to describe simulcast stream formats. Alice's Offer:

```
v=0
o=alice 2362969037 2362969040 IN IP4 192.0.2.156
s=Simulcast Enabled 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=rid:1 pt=97 send
a=rid:2 pt=98 send
a=rid:3 pt=97 recv
a=simulcast:send 1;2 recv 3
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:RtpStreamId
```

Figure 4: Single-Source 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 "a=fmtp" and "a=imageattr" parameters indicates that sent simulcast streams can differ in video resolution.

The RTP header extension for RtpStreamId is offered to avoid issues with the initial binding between RTP streams (SSRCs) and the RtpStreamId identifying the simulcast stream and its format.

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. Also the usage of the RtpStreamId RTP header extension is accepted.

```
v=0
o=server 823479283 1209384938 IN IP4 192.0.2.2
s=Answer to Simulcast Enabled 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=rid:1 pt=97 recv
a=rid:2 pt=98 recv
a=rid:3 pt=97 send
a=simulcast:recv 1;2 send 3
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:RtpStreamId
```

Figure 5: Single-Source Simulcast Answer

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

6.6.2. 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 [RFC7741] and H264, are offered as alternatives for

the third simulcast stream for the first media source. Only the highest fidelity simulcast stream is 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]. 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, the use of RtpStreamId as simulcast stream identification enables use of a low number of RTP payload types. Note that the use of both BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] and "a=rid" [I-D.ietf-mmusic-rid] recommends using the RTP header extension [RFC5285] for carrying these RTP stream identification fields, which is consequently also included in the SDP. Note also that for "a=rid", the corresponding SDES attribute is named RtpStreamId [I-D.ietf-avtext-rid].

```
v=0
o=fred 238947129 823479223 IN IP6 2001:db8::c000:27d
s=Offer from Simulcast Enabled Multi-Source Client
t=0 0
c=IN IP6 2001:db8::c000:27d
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-fps=60;depend=2
a=rid:2 send pt=101;max-width=1280;max-height=720;max-fps=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:sdes:RtpStreamId
a=rtcp-fb:* ccm pause nowait
a=simulcast:send 1;2;~4,3

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:1 send pt=96;max-fs=921600;max-fps=30
a=rid:2 send pt=96;max-fs=614400;max-fps=15
a=rid:3 send pt=96;max-fs=230400;max-fps=30
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdrext:sdes:RtpStreamId
a=rtcp-fb:* ccm pause nowait
a=simulcast:send 1;~2;~3
```

Figure 6: 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. RTP Aspects

This section discusses what the different entities in a simulcast media path can expect to happen on RTP level. This is explored from source to sink by starting in an endpoint with a media source that is simulcasted to an RTP middlebox. That RTP middlebox sends media sources both to other RTP middleboxes (cascaded middleboxes), as well as selecting some simulcast format of the media source and sending it to receiving endpoints. Different types of RTP middleboxes and their usage of the different simulcast formats results in several different behaviors.

7.1. Outgoing from Endpoint with Media Source

The most straightforward simulcast case is the RTP streams being emitted from the endpoint that originates a media source. When simulcast has been negotiated in the sending direction, the endpoint can transmit up to the number of RTP streams needed for the negotiated simulcast streams for that media source. Each RTP stream (SSRC) is identified by associating (Section 6.5) it with an RtpStreamId SDES item, transmitted in RTCP and possibly also as an RTP header extension. In cases where multiple media sources have been negotiated for the same RTP session and thus BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation] is used, also the MID SDES item will be sent similarly to the RtpStreamId.

Each RTP stream may not be continuously transmitted due to any of the following reasons; temporarily paused using Pause/Resume [RFC7728], sender side application logic temporarily pausing it, or lack of network resources to transmit this simulcast stream. However, all simulcast streams that have been negotiated have active and maintained SSRC (at least in regular RTCP reports), even if no RTP packets are currently transmitted. The relation between an RTP Stream (SSRC) and a particular simulcast stream is not expected to change, except in exceptional situations such as SSRC collisions. At SSRC changes, the usage of MID and RtpStreamId should enable the receiver to correctly identify the RTP streams even after an SSRC change.

7.2. RTP Middlebox to Receiver

RTP streams in a multi-party RTP session can be used in multiple different ways, when the session utilizes simulcast at least on the media source to middlebox legs. This is to a large degree due to the different RTP middlebox behaviors, but also the needs of the

application. This text assumes that the RTP middlebox will select a media source and choose which simulcast stream for that media source to deliver to a specific receiver. In many cases, at most one simulcast stream per media source will be forwarded to a particular receiver at any instant in time, even if the selected simulcast stream may vary. For cases where this does not hold due to application needs, then the RTP stream aspects will fall under the middlebox to middlebox case Section 7.3.

The selection of which simulcast streams to forward towards the receiver, is application specific. However, in conferencing applications, active speaker selection is common. In case the number of media sources possible to forward, N, is less than the total amount of media sources available in an multi-media session, the current and previous speakers (up to N in total) are often the ones forwarded. To avoid the need for media specific processing to determine the current speaker(s) in the RTP middlebox, the endpoint providing a media source may include meta data, such as the RTP Header Extension for Client-to-Mixer Audio Level Indication [RFC6464].

The possibilities for stream switching are media type specific, but for media types with significant interframe dependencies in the encoding, like most video coding, the switching needs to be made at suitable switching points in the media stream that breaks or otherwise deals with the dependency structure. Even if switching points can be included periodically, it is common to use mechanisms like Full Intra Requests [RFC5104] to request switching points from the endpoint performing the encoding of the media source.

Inclusion of the RtpStreamId SDES item for an SSRC in the middlebox to receiver direction should only occur when use of RtpStreamId has been negotiated in that direction. It is worth noting that one can signal multiple RtpStreamIds when simulcast signalling indicates only a single simulcast stream, allowing one to use all of the RtpStreamIds as alternatives for that simulcast stream. One reason for including the RtpStreamId in the middlebox to receiver direction for an RTP stream is to let the receiver know which restrictions apply to the currently delivered RTP stream. In case the RtpStreamId is negotiated to be used, it is important to remember that the used identifiers will be specific to each signalling session. Even if the central entity can attempt to coordinate, it is likely that the RtpStreamIds need to be translated to the leg specific values. The below cases will have as base line that RtpStreamId is not used in the mixer to receiver direction.

7.2.1. Media-Switching Mixer

This section discusses the behavior in cases where the RTP middlebox behaves like the Media-Switching Mixer (Section 3.6.2) in RTP Topologies [RFC7667]. The fundamental aspect here is that the media sources delivered from the middlebox will be the mixer's conceptual or functional ones. For example, one media source may be the main speaker in high resolution video, while a number of other media sources are thumbnails of each participant.

The above results in that the RTP stream produced by the mixer is one that switches between a number of received incoming RTP streams for different media sources and in different simulcast versions. The mixer selects the media source to be sent as one of the RTP streams, and then selects among the available simulcast streams for the most appropriate one. The selection criteria include available bandwidth on the mixer to receiver path and restrictions based on the functional usage of the RTP stream delivered to the receiver. An example of the latter, is that it is unnecessary to forward a full HD video to a receiver if the display area is just a thumbnail. Thus, restrictions may exist to not allow some simulcast streams to be forwarded for some of the mixer's media sources.

This will result in a single RTP stream being used for a particular of the RTP mixer's media sources. This RTP stream is at any point in time a selection of one particular RTP stream arriving to the mixer, where the RTP header field values are rewritten to provide a consistent, single RTP stream. If the RTP mixer doesn't receive any incoming stream matched to this media source, the SSRC will not transmit, but be kept alive using RTCP. The SSRC and thus RTP stream for the mixer's media source is expected to be long term stable. It will only be changed by signalling or other disruptive events. Note that although the above talks about a single RTP stream, there can in some cases be multiple RTP streams carrying the selected simulcast stream for the originating media source, including repair or other auxiliary RTP streams.

The mixer may communicate the identity of the originating media source to the receiver by including the CSRC field with the originating media source's SSRC value. Note that due to the possibility that the RTP mixer switches between simulcast versions of the media source, the CSRC value may change, even if the media source is kept the same.

It is important to note that any MID SDES item from the originating media source needs to be removed and not be associated with the RTP stream's SSRC. This as there is nothing in the signalling between the mixer and the receiver that is structured around the originating

media sources, only the mixer's media sources. If they would be associated with the SSRC, the receiver would likely believe that there has been an SSRC collision, and that the RTP stream is spurious as it doesn't carry the identifiers used to relate it to the correct context. However, this is not true for CSRC values, as long as they are never used as SSRC. In these cases one could provide CNAME and MID as SDES items. A receiver could use this to determine which CSRC values that are associated with the same originating media source.

If RtpStreamIds are used in this scenario, it should be noted that the RtpStreamId on a particular SSRC will change based on the actual simulcast stream selected for switching. These RtpStreamId identifiers will be local to this leg's signalling context. In addition, the defined RtpStreamIds and their parameters need to cover all the media sources and simulcast streams that can be switched into this media source.

7.2.2. Selective Forwarding Middlebox

This section discusses the behavior in cases where the RTP middlebox behaves like the Selective Forwarding Middlebox (Section 3.7) in RTP Topologies [RFC7667]. Applications for this type of RTP middlebox results in that each originating media source will have a corresponding media source on the leg between the middlebox and the receiver. A SFM could go as far as exposing all the simulcast streams for an media source, however this section will focus on having a single simulcast stream that can contain any of the simulcast formats. This section will assume that the SFM projection mechanism works on media source level, and maps one of the media source's simulcast streams onto one RTP stream from the SFM to the receiver.

This usage will result in that the individual RTP stream(s) for one media source can switch between being active to paused, based on the subset of media sources the SFM wants to provide the receiver for the moment. With SFMs there exist no reasons to use CSRC to indicate the originating stream, as there is a one to one media source mapping. If the application requires knowing the simulcast version received to function well, then RtpStreamId should be negotiated on the SFM to receiver leg. Which simulcast stream that is being forwarded is not made explicit unless RtpStreamId is used on the leg.

Any MID SDES items being sent by the SFM to the receiver are only those agreed between the SFM and the receiver, and no MID values from the originating side of the SFM are to be forwarded.

A SFM could expose corresponding RTP streams for all the media sources and their simulcast streams, and then for any media source

that is to be provided forward one selected simulcast stream. However, this is not recommended as it would unnecessarily increase the number of RTP streams and require the receiver to timely detect switching between simulcast streams. The above usage requires the same SFM functionality for switching, while avoiding the uncertainties of timely detecting that a RTP stream ends. The benefit would be that the received simulcast stream would be implicitly provided by which RTP stream would be active for a media source. However, using RtpStreamId to make this explicit also exposes which alternative format is used. The conclusion is that using one RTP stream per simulcast stream is unnecessary. The issue with timely detecting end of streams, independent if they are stopped temporarily or long term, is that there is no explicit indication that the transmission has intentionally been stopped. The RTCP based Pause and Resume mechanism [RFC7728] includes a PAUSED indication that provides the last RTP sequence number transmitted prior to the pause. Due to usage, the timeliness of this solution depends on when delivery using RTCP can occur in relation to the transmission of the last RTP packet. If no explicit information is provided at all, then detection based on non increasing RTCP SR field values and timers need to be used to determine pause in RTP packet delivery. This results in that one can usually not determine when the last RTP packet arrives (if it arrives) that this will be the last. That it was the last is something that one learns later.

7.3. RTP Middlebox to RTP Middlebox

This relates to the transmission of simulcast streams between RTP middleboxes or other usages where one wants to enable the delivery of multiple simultaneous simulcast streams per media source, but the transmitting entity is not the originating endpoint. For a particular direction between middlebox A and B, this looks very similar to the originating to middlebox case on a media source basis. However, in this case there is usually multiple media sources, originating from multiple endpoints. This can create situations where limitations in the number of simultaneously received media streams can arise, for example due to limitation in network bandwidth. In this case, a subset of not only the simulcast streams, but also media sources can be selected. This results in that individual RTP streams can be become paused at any point and later being resumed based on various criteria.

The MIDs used between A and B are the ones agreed between these two identities in signalling. The RtpStreamId values will also be provided to ensure explicit information about which simulcast stream they are. The RTP stream to MID and RtpStreamId associations should here be long term stable.

8. 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 with good QoS support.

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].

8.1. Bitrate Adaptation

Use of multiple simulcast streams can require a significant amount of network resources. If the amount of available network resources varies during an RTP session such that it does not match what is negotiated in SDP, the bitrate used by the different simulcast streams may have to be reduced dynamically. What simulcast streams to prioritize when allocating available bitrate among the simulcast streams in such adaptation SHOULD be taken from the simulcast stream order on the "a=simulcast" line and ordering of alternative simulcast formats Section 6.2. Simulcast streams that have pause/resume capability and that would be given such low bitrate by the adaptation process that they are considered not really useful can be temporarily paused until the limiting condition clears.

9. Limitation

The chosen approach has a limitation that relates to the use of a single RTP session for all simulcast formats of a media source, which

comes from sending all simulcast streams related to a media source under the same SDP media description.

It is not possible to use different simulcast streams on different media 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 also 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. Such simulcast usage in multicast context is out of scope for the current document and would require additional specification.

10. IANA Considerations

This document requests to register a new media-level SDP attribute, "simulcast", in the "att-field (media level only)" registry within the SDP parameters registry, according to the procedures of [RFC4566] and [I-D.ietf-mmusic-sdp-mux-attributes].

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

Attribute name: simulcast

Long-form attribute name: Simulcast stream description

Charset dependent: No

Attribute value: See Section 6.1 of RFC XXXX.

Purpose: Signals simulcast capability for a set of RTP streams

MUX category: NORMAL

Note to RFC Editor: Please replace "RFC XXXX" with the assigned number of this RFC.

11. 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.

Security considerations related to the use of "a=rid" and the RtpStreamId SDES item is covered in [I-D.ietf-mmusic-rid] and [I-D.ietf-avtext-rid]. There are no additional security concerns related to their use in this specification.

12. 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, Peter Thatcher, from Google, and Adam Roach, from Mozilla, contributed significantly to subsequent versions.

13. Acknowledgements

The authors would like to thank Bernard Aboba, Thomas Belling, Roni Even, and Adam Roach for the feedback they provided during the development of this document.

14. References

14.1. Normative References

[I-D.ietf-avtext-rid]
Roach, A., Nandakumar, S., and P. Thatcher, "RTP Stream Identifier Source Description (SDES)", draft-ietf-avtext-rid-09 (work in progress), October 2016.

[I-D.ietf-mmusic-rid]
Thatcher, P., Zanaty, M., Nandakumar, S., Burman, B., Roach, A., and B. Campen, "RTP Payload Format Restrictions", draft-ietf-mmusic-rid-08 (work in progress), October 2016.

[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-36 (work in progress), October 2016.

[I-D.ietf-mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-16 (work in progress), December 2016.

[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>.

[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>.

[RFC7728] Burman, B., Akram, A., Even, R., and M. Westerlund, "RTP Stream Pause and Resume", RFC 7728, DOI 10.17487/RFC7728, February 2016, <http://www.rfc-editor.org/info/rfc7728>.

14.2. 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, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session", draft-ietf-avtcore-rtp-multi-stream-11 (work in progress), December 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>.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <http://www.rfc-editor.org/info/rfc5104>.

[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>.

[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>.

[RFC6464] Lennox, J., Ed., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, DOI 10.17487/RFC6464, December 2011, <http://www.rfc-editor.org/info/rfc6464>.

[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>.

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

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <http://www.rfc-editor.org/info/rfc7667>.

[RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <http://www.rfc-editor.org/info/rfc7741>.

Appendix A. Changes From Earlier Versions

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

A.1. Modifications Between WG Version -05 and -06

- Added section on RTP Aspects
- Added a requirement (5-4) on that capability exchange must be capable of handling multi RTP stream cases.
- Added extmap attribute also on first signalling example as it is a recommended to use mechanism.
- Clarified the definition of the simulcast attribute and how simulcast streams relates to simulcast formats and SCIDs.
- Updated References list and moved around some references between informative and normative categories.
- Editorial improvements and corrections.

A.2. Modifications Between WG Version -04 and -05

- Aligned with recent changes in draft-ietf-mmusic-rid and draft-ietf-avtext-rid.
- Modified the SDP offer/answer section to follow the generally accepted structure, also adding a brief text on modifying the session that is aligned with draft-ietf-mmusic-rid.
- Improved text around simulcast stream identification (as opposed to the simulcast stream itself) to consistently use the acronym SCID and defined that in the Terminology section.
- Changed references for RTP-level pause/resume and VP8 payload format that are now published as RFC.
- Improved IANA registration text.
- Removed unused reference to draft-ietf-payload-flexible-fec-scheme.
- Editorial improvements and corrections.

A.3. Modifications Between WG Version -03 and -04

- Changed to only use RID identification, as was consensus during IETF 94.

- ABNF improvements.

- Clarified offer-answer rules for initially paused streams.

- Changed references for RTP topologies and RTP taxonomy documents that are now published as RFC.

- Added reference to the new RID draft in AVTEXT.

- Re-structured section 6 to provide an easy reference by the updated IANA section.

- Added a sub-section 7.1 with a discussion of bitrate adaptation.

- Editorial improvements.

A.4. Modifications Between WG Version -02 and -03

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

- 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.5. 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.6. Modifications Between WG Version -00 and -01

- No changes. Only preventing expiry.

A.7. Modifications Between Individual Version -00 and WG Version -00

- Added this appendix.

Authors' Addresses

Bo Burman
Ericsson
Gronlandsgatan 31
SE-164 60 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