Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 6, 2016

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

February 3, 2016

Using Simulcast in SDP and RTP Sessions
draft-ietf-mmusic-sdp-simulcast-04

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.

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.

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]. In addition, the following terms are used:

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]. 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.

- 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:

> Editor's note: Consider adding an explicit requirement that the solution supports use of simulcast even when using multiple codecs and multiple redundant RTP streams per defined codec (or something similar), since this is really an existing requirement and should also fully motivate the use of RID as identification mechanism.

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 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 stream formats 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 [RFC7656] 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 is expressed through use of a separately specified RTP-level identification mechanism [I-D.ietf-mmusic-rid][I-D.roach-avtext-rid], which complements and effectively extends the available simulcast stream identification and configuration possibilities that could be provided by using only 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). 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 6.3) provides the definition of the RTP/RTCP mechanisms used. The section is concluded with a number of examples.

6.1. Simulcast Attribute

Name: simulcast

Value: sc-value

Usage Level: media

Charset Dependent: no

Multiplex Category: NORMAL

Syntax [RFC5234]:

```
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 / token
; SP defined in [RFC5234]
; token defined in [RFC4566]
; 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 in that list is separated by a semicolon (";"). Each simulcast stream can in turn be offered in one or more alternative formats, where each simulcast stream alternative is separated by a comma (","). The simulcast stream alternative MUST be described in the form of a RID, as described by [I-D.ietf-mmusic-rid]. Each simulcast stream can be initially paused [I-D.ietf-avtext-rtp-stream-pause], indicated by prepending a "~" to the simulcast stream. In case there are simulcast stream alternatives, pause can be specified individually for each alternative. The reason to allow separate initial pause states for each simulcast stream alternative is that pause capability can be specified individually for each RTP payload type referenced by a RID, which makes it infeasible to pause RID where any of the related RTP payload type(s) do not have pause capability.

Examples:

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

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 as three different alternatives (1, 2, 3), and the second simulcast stream in send direction can be sent as two different alternatives (4, 5). All second stream send alternatives are offered as initially paused. The first simulcast stream in receive direction has no alternatives (only 1). The second simulcast stream in receive direction has two alternatives (2, 5) 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, even though directionality is reversed. This example answer has removed all offered alternatives for the first simulcast stream (keeping only 1), but kept alternative formats for the second simulcast stream in receive direction (4, 5). The answer 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 6.4.

6.2. Simulcast Capability

Simulcast capability is expressed as a new media level SDP attribute, "a=simulcast" (Section 6.1), with multiplex category [I-D.ietf-mmusic-sdp-mux-attributes] NORMAL.

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 meaning of including multiple "a=simulcast" lines in a single SDP media description is undefined and MUST NOT be used. 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.

The different simulcast streams MUST be identified through the RTP-level "RID" identification mechanism [I-D.ietf-mmusic-rid].

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.ietf-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, it is not possible to align what alternative formats that are used between different simulcast streams, like requiring all simulcast streams to use alternatives with the same codec format. 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.

If RTP stream pause/resume [I-D.ietf-avtext-rtp-stream-pause] is supported, any simulcast stream identification 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 listing "a=simulcast". If the simulcast stream is specified as a list of alternative formats, the indication is prepended to the first format of the list and applies to whatever alternative that is eventually chosen. All RTP payload types related to such initially paused simulcast stream MUST be listed in the SDP as pause/resume capable as specified by [I-D.ietf-avtext-rtp-stream-pause].

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 stream RID identification as an RID RTCP SDES item.

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 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.

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

6.2.1. Declarative Use

When used as a declarative media description, "a=simulcast" line "recv" direction formats indicate 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" line "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: It may not be beneficial for declarative use to be limited to a single media source per "m=" line, as elaborated further in Section 8.

6.2.2. 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" 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.

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

An answerer that receives an offer with RTP stream pause/resume capability MAY mark any simulcast streams as initially paused in the answer.

An answerer that receives indication in an offer of a simulcast stream being initially paused , SHOULD mark that simulcast stream as initially paused also in the answer, regardless of direction, unless it has good reason for the stream not being initially paused.

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

An offerer with RTP stream pause/resume capability that receives an answer where some simulcast streams are marked as initially paused, SHOULD initially pause them regardless if they were marked as initially paused also in the offer, unless it has good reason for those streams not being initially paused.

> 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. Relating Simulcast Streams

Simulcast RTP streams MUST be related on RTP level through RID [I-D.roach-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 RID relations can be used. Use of the RTP header extension [RFC5285] for both MID and RID identifications can be important to ensure rapid initial reception, required to correctly interpret and process the RTP streams. Implementers of this specification MUST support RTCP source description (SDES) item and SHOULD support RTP header extension method to signal RID on RTP level.

6.4. 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

6.4.1. 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" RID parameter is used, effectively achieving a 1:1 mapping between RID 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
a=rid:2 pt=98
a=simulcast:send 1;2 recv 1
```

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 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 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
a=rid:2 pt=98
a=simulcast:recv 1;2 send 1
```

Figure 5: Single-Source Simulcast Answer

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

6.4.2. 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. 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]. 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 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, which is consequently also included in the SDP.

```
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:sdes:rid
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: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:sdes:rid
a=rtcp-fb:* ccm pause nowait
a=simulcast:send 5;~6;~7
```

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. 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].

7.1. 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. 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.

8. Limitations

The chosen approach has a few limitations that are described in this section. The only one currently described relates to the use of a single RTP session for all simulcast formats of a media source.

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 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 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.

9. IANA Considerations

This document requests to register a new SDP attribute, simulcast, as defined in Section 6.1.

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.

Security considerations related to the use of RID is covered in [I-D.ietf-mmusic-rid] and [I-D.roach-avtext-rid]. There are no additional security concerns related to its use in this specification.

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, Peter Thatcher, from Google, and Adam Roach, from Mozilla, 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.ietf-mmusic-rid]
Thatcher, P., Zanaty, M., Nandakumar, S., Burman, B., Roach, A., and B. Campen, "RTP Payload Format Constraints", draft-ietf-mmusic-rid-01 (work in progress), February 2016.

[I-D.ietf-mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-12 (work in progress), January 2016.

[I-D.roach-avtext-rid]
Roach, A., Nandakumar, S., and P. Thatcher, "RTP Payload Format Constraints", draft-roach-avtext-rid-01 (work in progress), February 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>.

[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, 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.

[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-25 (work in progress), January 2016.

[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>.

[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>.

[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>.

Appendix A. Changes From Earlier Versions

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

A.1. 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.2. 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.3. 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.4. Modifications Between WG Version -00 and -01

- No changes. Only preventing expiry.

A.5. 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