
Stream: Internet Engineering Task Force (IETF)
RFC: [8853](#)
Category: Standards Track
Published: May 2020
ISSN: 2070-1721
Authors: B. Burman M. Westerlund S. Nandakumar M. Zanaty
Ericsson Ericsson Cisco Cisco

RFC 8853

Using Simulcast in Session Description Protocol (SDP) and RTP Sessions

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 the Session Description Protocol (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 indicate that those RTP streams are 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

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8853>.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions

with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction
2. Definitions
 - 2.1. Terminology
 - 2.2. Requirements Language
3. Use Cases
 - 3.1. Reaching a Diverse Set of Receivers
 - 3.2. Application-Specific Media Source Handling
 - 3.3. Receiver Media-Source Preferences
4. Overview
5. Detailed Description
 - 5.1. Simulcast Attribute
 - 5.2. Simulcast Capability
 - 5.3. Offer/Answer Use
 - 5.3.1. Generating the Initial SDP Offer
 - 5.3.2. Creating the SDP Answer
 - 5.3.3. Offerer Processing the SDP Answer
 - 5.3.4. Modifying the Session
 - 5.4. Use with Declarative SDP
 - 5.5. Relating Simulcast Streams
 - 5.6. Signaling Examples
 - 5.6.1. Single-Source Client
 - 5.6.2. Multisource Client
 - 5.6.3. Simulcast and Redundancy
6. RTP Aspects
 - 6.1. Outgoing from Endpoint with Media Source

- 6.2. [RTP Middlebox to Receiver](#)
 - 6.2.1. [Media-Switching Mixer](#)
 - 6.2.2. [Selective Forwarding Middlebox](#)
- 6.3. [RTP Middlebox to RTP Middlebox](#)
- 7. [Network Aspects](#)
 - 7.1. [Bitrate Adaptation](#)
- 8. [Limitation](#)
- 9. [IANA Considerations](#)
- 10. [Security Considerations](#)
- 11. [References](#)
 - 11.1. [Normative References](#)
 - 11.2. [Informative References](#)
- [Appendix A. Requirements](#)
- [Acknowledgements](#)
- [Contributors](#)
- [Authors' Addresses](#)

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 multiparty 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. Simulcast could be used point to point between middleboxes as part of a distributed multiparty scenario. Usage of multicast or broadcast transport is out of scope and left for future extensions.

This document describes a few scenarios that motivate the use of 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 "[A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol \(RTP\) Sources](#)" [RFC7656] and "[RTP Topologies](#)" [RFC7667]. The following terms are especially noted or here defined:

RTP mixer: An RTP middlebox, in the wide sense of the term, encompassing Sections 3.6 to 3.9 of [RFC7667].

RTP session: An association among a group of participants communicating with RTP, as defined in [RFC3550] and amended by [RFC7656].

RTP stream: A stream of RTP packets containing media data, as defined in [RFC7656].

RTP switch: A common short term for the terms "switching RTP mixer", "source projecting middlebox", and "video switching Multipoint Control Unit (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 nonsimulcast 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.

2.2. Requirements Language

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

3. Use Cases

The use cases of simulcast described in this document relate to a multiparty 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 principal approaches for an RTP mixer to provide this adapted view of the communication session to each receiving participant:

- Transcoding (decoding and re-encoding) received RTP streams with characteristics adapted to each receiving participant. This often includes 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 creating end-to-end delay for the receiving participants, and requires the RTP mixer to have access to media content.
- Switching a subset of all received RTP streams or substreams 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 the RTP mixer to have (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 for the resource situation of a receiving participant. It is also a disadvantage that sending multiple RTP streams consumes more network resources from the sending participant to the RTP mixer.

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 RTP payload format MIME type) and can include codec configuration. A couple of codec resources that differ only in codec configuration will be "different" if they are somehow not "compatible", such as 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 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.

Bitrate: This relates to the number of bits sent per second to transmit the media source as an RTP stream, which typically also affects the 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 trade-off 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, 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.

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](#)) in a receiving client. Many conferencing systems do not send the active speaker's media back to the sender itself, which means there is some other participant's media that instead is forwarded to 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 state preferences on the characteristics of the RTP stream they like to 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. Overview

This memo defines [SDP \[RFC4566\]](#) signaling that covers the above described simulcast use cases and functionalities. A number of requirements for such signaling are elaborated in [Appendix A](#).

The Restriction Identifier (RID) mechanism, as defined in [\[RFC8851\]](#), enables an SDP offerer or answerer to specify a number of different RTP stream restrictions for a rid-id by using the "a=rid" line. Examples of such restrictions are maximum bitrate, maximum spatial video resolution (width and height), maximum video frame rate, etc. Each rid-id may also be restricted to use only a subset of the RTP payload types in the associated SDP media description. Those RTP payload types can have their own configurations and parameters affecting what can be sent or received, using the "a=fmtp" line as well as other SDP attributes.

A new SDP media-level attribute, "a=simulcast", is defined. The attribute describes, independently for "send" and "receive" directions, the number of simulcast RTP streams as well as potential alternative formats for each simulcast RTP stream. Each simulcast RTP stream, including alternatives, is identified using the RID identifier (rid-id), defined in [\[RFC8851\]](#).

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

If this line is included in an SDP offer, the "send" part indicates the offerer's capability and proposal to send two simulcast RTP streams. Each simulcast stream is described by one or more RTP stream identifiers (rid-ids), and each group of rid-ids for a simulcast stream is separated by a semicolon (";"). When a simulcast stream has multiple rid-ids that are separated by a comma (","), they describe alternative representations for that particular simulcast RTP stream. Thus, the "send" part shown above is interpreted as an intention to send two simulcast RTP streams. The first simulcast RTP stream is identified and restricted according to rid-id 1. The second simulcast RTP stream can be sent as two alternatives, identified and restricted according to rid-ids 2 and 3. The "recv" part of the line shown here indicates that the offerer desires to receive a single RTP stream (no simulcast) according to rid-id 4.

A more complete example SDP-offer media description is provided in [Figure 1](#).

```
m=video 49300 RTP/AVP 97 98 99
a=rtpmap:97 H264/90000
a=rtpmap:98 H264/90000
a=rtpmap:99 VP8/90000
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=fmtp:99 max-fs=240; max-fr=30
a=rid:1 send pt=97;max-width=1280;max-height=720
a=rid:2 send pt=98;max-width=320;max-height=180
a=rid:3 send pt=99;max-width=320;max-height=180
a=rid:4 recv pt=97
a=simulcast:send 1;2,3 recv 4
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
```

Figure 1: Example Simulcast Media Description in Offer

The SDP media description in [Figure 1](#) can be interpreted at a high level to say that the offerer is capable of sending two simulcast RTP streams: one H.264 encoded stream in up to 720p resolution, and one additional stream encoded as either H.264 or VP8 with a maximum resolution of 320x180 pixels. The offerer can receive one H.264 stream with maximum 720p resolution.

The receiver of this SDP offer can generate an SDP answer that indicates what it accepts. It uses the "a=simulcast" attribute to indicate simulcast capability and specify what simulcast RTP streams and alternatives to receive and/or send. An example of such an answering "a=simulcast" attribute, corresponding to the above offer, is:

```
a=simulcast:recv 1;2 send 4
```

With this SDP answer, the answerer indicates in the "recv" part that it wants to receive the two simulcast RTP streams. It has removed an alternative that it doesn't support (rid-id 3). The "send" part confirms to the offerer that it will receive one stream for this media source according to rid-id 4. The corresponding, more complete example SDP answer media description could look like [Figure 2](#).

```
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=rid:1 recv pt=97;max-width=1280;max-height=720
a=rid:2 recv pt=98;max-width=320;max-height=180
a=rid:4 send pt=97
a=simulcast:recv 1;2 send 4
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
```

Figure 2: Example Simulcast Media Description in Answer

It is assumed that a single SDP media description is used to describe a single media source. This is aligned with the concepts defined in [RFC7656] and will work in a WebRTC context, both with and without BUNDLE grouping of media descriptions [RFC8843].

To summarize, the "a=simulcast" line describes "send"- and "receive"-direction simulcast streams separately. Each direction can in turn describe one or more simulcast streams, separated by semicolons. The identifiers describing simulcast streams on the "a=simulcast" line are rid-ids, as defined by "a=rid" lines in [RFC8851]. Each simulcast stream can be offered as a list of alternative rid-ids, with each alternative separated by a comma as shown in the example offer in Figure 1. A detailed specification can be found in Section 5, and more detailed examples are outlined in Section 5.6.

5. Detailed Description

This section provides further details to the overview in Section 4. First, formal syntax is provided (Section 5.1), followed by the rest of the SDP attribute definition in Section 5.2. "Relating Simulcast Streams" (Section 5.5) provides the definition of the RTP/RTCP mechanisms used. The section concludes with a number of examples.

5.1. Simulcast Attribute

This document defines a new SDP media-level "a=simulcast" attribute, with value according to the syntax in Figure 3, which uses ABNF [RFC5234] and its update, "Case-Sensitive String Support in ABNF" [RFC7405]:

```
sc-value      = ( sc-send [SP sc-recv] ) / ( sc-recv [SP sc-send] )
sc-send      = %s"send" SP sc-str-list
sc-recv      = %s"recv" SP sc-str-list
sc-str-list  = sc-alt-list *( ";" sc-alt-list )
sc-alt-list  = sc-id *( "," sc-id )
sc-id-paused = "~"
sc-id        = [sc-id-paused] rid-id
; SP defined in [RFC5234]
; rid-id defined in [RFC8851]
```

Figure 3: ABNF for Simulcast Value

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 (rid-id). The rid-id **MUST** have the form of an RTP stream identifier, as described by "RTP Payload Format Restrictions" [RFC8851].

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 rid-ids, separated by commas (","). Each rid-id can also be specified as initially **paused** [RFC7728],

indicated by prepending a "~" to the rid-id. The reason to allow separate initial pause states for each rid-id is that pause capability can be specified individually for each RTP payload type referenced by a rid-id. Since pause capability specified via the "a=rtcp-fb" attribute applies only to specified payload types, and a rid-id specified by "a=rid" can refer to multiple different payload types, it is unfeasible to pause streams with rid-id where any of the related RTP payload type(s) do not have pause capability.

5.2. Simulcast Capability

Simulcast capability is expressed through a new media-level SDP attribute, "a=simulcast" (Section 5.1). The use of this attribute at the session level is undefined. Implementations of this specification **MUST NOT** use it at the session level and **MUST** ignore it if received at the session level. Extensions to this specification may define such session-level usage. Each SDP media description **MUST** contain at most one "a=simulcast" line.

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

Simulcast streams using undefined rid-ids **MUST NOT** be used as valid simulcast streams by an RTP stream receiver. The direction for a rid-id **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 rid-id listed first is the most preferred, and subsequent streams have progressively lower preference. The order of the listed rid-ids in the "recv" direction expresses which simulcast streams are preferred, with the leftmost being most preferred. This can be of importance if the number of actually sent simulcast streams has to be reduced for some reason.

rid-ids that have explicit [dependencies](#) [RFC5583] [RFC8851] to other rid-ids (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 rid-ids. The order of the rid-id alternatives within a simulcast stream is significant; the rid-id 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. However, the choice of which alternative to use per simulcast stream is independent, and there is currently no mechanism for the offerer to force the answerer to choose the same alternative for multiple simulcast streams.

A simulcast stream can use a codec defined such that the same RTP synchronization source (SSRC) can change RTP payload type multiple times during a session, possibly even on a per-packet basis. A typical example is a speech codec that makes use of formats for [Comfort Noise \[RFC3389\]](#) and/or [dual-tone multifrequency \(DTMF\) \[RFC4733\]](#).

If [RTP stream pause/resume \[RFC7728\]](#) is supported, any rid-id **MAY** be prefixed by a "~" character to indicate that the corresponding simulcast stream is paused already from the 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 an 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 the "send" direction for the endpoint sending the SDP **MUST** be considered equivalent to an unsolicited locally paused stream and 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) or Receiver Report (RR) that includes both the desired SSRC as initial SSRC in the source description (SDES) chunk, optionally a [MID SDDES item \[RFC8843\]](#) (if used and if rid-ids are not unique across "m=" lines), and the rid-id value in an [RtpStreamId RTCP SDDES item \[RFC8852\]](#).

If the endpoint sending the SDP includes a "recv"-direction simulcast stream that is initially paused, then the remote RTP sender receiving the SDP **SHOULD** put its RTP stream in an unsolicited locally paused state. The simulcast stream sender does not put the stream in the locally paused state if there are other RTP stream receivers in the session that do not mark the simulcast stream as initially paused. However, in centralized conferencing, the RTP sender usually does not see the SDP signaling from RTP receivers and cannot make this determination. The reason for requiring that an initially paused "recv" stream be considered locally paused by the remote RTP sender instead of making it equivalent to implicitly sending a pause request is that the pausing RTP sender cannot know which receiving SSRC owns the restriction when Temporary Maximum Media Stream Bit Rate Request (TMMBR) and Temporary Maximum Media Stream Bit Rate Notification (TMMBN) are used for pause/resume signaling ([Section 5.6 of \[RFC7728\]](#)); this is because the RTP receiver's SSRC in the "send" direction is sometimes not yet known.

Use of the redundant audio data format [\[RFC2198\]](#) could be seen as a form of simulcast for loss-protection purposes, but it 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, it is **RECOMMENDED** to use RTP duplication procedures [\[RFC7104\]](#) instead of simulcast.

To avoid complications in implementations, a single rid-id **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 rid-ids that are effectively equivalent.

5.3. 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, such as "a=fmtp" or "a=rid".

5.3.1. Generating the Initial SDP Offer

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

5.3.2. 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 procedures [RFC3264]. Since SDP session-level simulcast is undefined in this memo, an answerer that receives an offer with the "a=simulcast" attribute on the SDP 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 **SHALL** disable use of simulcast by removing all "a=simulcast" lines for that media description in the answer.

An answerer that does understand the attribute and 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 rid-ids **MAY** keep all the alternative rid-ids in the answer, but it **MAY** also choose to remove any nondesirable alternative rid-ids in the answer. The answerer **MUST NOT** add any alternative rid-ids in the "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 rid-id alternatives and **MAY** send any of the "send"-direction alternatives that are part of 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 it **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 answerer capable of pause/resume that receives an offer with RTP stream pause/resume capability **MAY** mark any rid-ids that refer to pause/resume capable formats as initially paused in the answer.

An answerer that receives indication in an offer of a rid-id being initially paused **SHOULD** mark that rid-id as initially paused also in the answer, regardless of direction, unless it has good reason for the rid-id not being initially paused. One reason to remove an initial pause in the answer compared to the offer could be, for example, that all "receive"-direction simulcast streams for a media source the answerer accepts in the answer would otherwise be paused.

5.3.3. 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 rid-id in a specified direction **MUST NOT** use simulcast in that direction.

An offerer that receives an answer where some rid-id alternatives are kept **MUST** be prepared to receive any of the kept "send"-direction rid-id alternatives and **MAY** send any of the kept "receive"-direction rid-id alternatives.

An offerer that receives an answer where some of the rid-ids 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 rid-ids.

An offerer that offered some of its rid-ids as initially paused and 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 rid-ids are marked as initially paused **SHOULD** initially pause those RTP streams, even 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 be, for example, that the answerer would otherwise initially not receive any media of that type at all.

5.3.4. Modifying the Session

Offers inside an existing session follow the same rules as for initial SDP offer, with these additions:

1. rid-ids marked as initially paused in the offerer's "send" direction **SHALL** reflect the offerer's opinion of the current pause state at the time of creating the offer. This is purely informational, and RTP stream pause/resume signaling [RFC7728] in the ongoing session **SHALL** take precedence in case of any conflict or ambiguity.
2. rid-ids marked as initially paused in the offerer's "receive" direction **SHALL** (as in an initial offer) reflect the offerer's desired rid-id pause state. Except for the case where the offerer already paused the corresponding RTP stream through RTP stream pause/resume [RFC7728] signaling, this is identical to the conditions at an initial offer.

Creation of SDP answers and processing of SDP answers inside an existing session follow the same rules as described above for initial SDP offer/answer.

Session modification restrictions in Section 6.5 of "RTP Payload Format Restrictions" [RFC8851] also apply.

5.4. Use with Declarative SDP

This document does not define the use of "a=simulcast" in declarative SDP, partly because use of the [simulcast format identification](#) [RFC8851] is not defined for use in declarative SDP. If concrete use cases for simulcast in declarative SDP are identified in the future, the authors of this memo expect that additional specifications will address such use.

5.5. Relating Simulcast Streams

Simulcast RTP streams **MUST** be related on the RTP level through [RtpStreamId](#) [RFC8852], as specified in the SDP "a=simulcast" attribute (Section 5.2) parameters. This is sufficient as long as there is only a single media source per SDP media description. When using [BUNDLE](#) [RFC8843], where multiple SDP media descriptions jointly specify a single RTP session, the SDES MID (Media Identification) mechanism in BUNDLE allows relating RTP streams back to individual media descriptions, after which the RtpStreamId relations described above can be used. Use of the RTP header extension for the [RTCP source description items](#) [RFC7941] 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 the RTP level.

NOTE: For the case where it is clear from SDP that the RTP PT uniquely maps to a corresponding RtpStreamId, an RTP receiver can use RTP PT to relate simulcast streams. This can sometimes enable decoding even in advance of receiving RtpStreamId information in RTCP SDES and/or RTP header extensions.

RTP streams **MUST** only use a single alternative rid-id at a time (based on RTP timestamps) but **MAY** change format (and rid-id) on a per-RTP packet basis. This corresponds to the existing (nonsimulcast) 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.

5.6. Signaling Examples

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

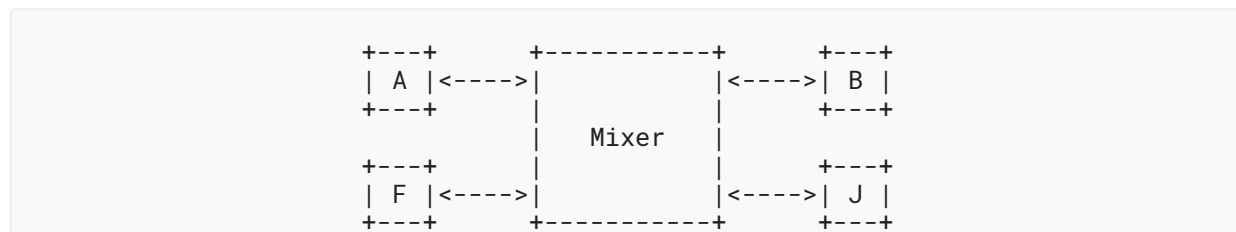


Figure 4: Four-Party Mixer-Based Conference

5.6.1. 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 to describe simulcast stream formats, effectively achieving a 1:1 mapping between RtpStreamId and media formats (RTP payload types). Alice's Offer:

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

Figure 5: 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 indicate 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
c=IN IP4 192.0.2.43
t=0 0
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 recv pt=97
a=rid:2 recv pt=98
a=rid:3 send pt=97
a=simulcast:recv 1;2 send 3
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
```

Figure 6: Single-Source Simulcast Answer

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

5.6.2. Multisource 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 \[RFC4588\]](#). In addition, all but the highest-fidelity simulcast stream are initially paused. Note that the lower resolution is more prioritized than the medium-resolution simulcast stream.

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 when using both [BUNDLE \[RFC8843\]](#) and "a=rid" [\[RFC8851\]](#), it is recommended to use the RTP header extension for the [RTCP source descriptions](#)

items [RFC7941] for carrying these RTP stream-identification fields, which is consequently also included in the SDP. Note also that for "a=rid", the corresponding RtpStreamId SDP attribute RTP header extension is named `rtp-stream-id` [RFC8852].

```
v=0
o=fred 238947129 823479223 IN IP6 2001:db8::c000:27d
s=Offer from Simulcast-Enabled Multi-Source Client
c=IN IP6 2001:db8::c000:27d
t=0 0
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=216000; \
  mst-mode=NI-TC
a=fmtp:101 profile-level-id=42c00d;max-fs=3600;max-mbps=108000
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-hdext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
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 max-fs=921600;max-fps=30
a=rid:2 send max-fs=614400;max-fps=15
a=rid:3 send max-fs=230400;max-fps=30
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id
a=rtcp-fb:* ccm pause nowait
a=simulcast:send 1;~3;~2
```

Figure 7: Fred's Multisource Simulcast Offer

5.6.3. Simulcast and Redundancy

The example in this section looks at applying simulcast with audio and video redundancy formats. The audio media description uses codec and bitrate restrictions, combined with the [RTP payload for redundant audio data \[RFC2198\]](#) for enhanced packet-loss resilience. The video media description applies both resolution and bitrate restrictions, combined with Forward Error Correction (FEC) in the form of [flexible FEC \[RFC8627\]](#) and [RTP retransmission \[RFC4588\]](#).

The audio source is offered to be sent as two simulcast streams. The first simulcast stream is encoded with Opus, restricted to 64 kbps (rid-id=1), and the second simulcast stream (rid-id=2) is encoded with either G.711, or G.711 combined with linear predictive coding (LPC) for redundancy and explicit comfort noise (CN). Both simulcast streams include telephone-event capability. In this example, stand-alone LPC is not offered as a possible payload type for the second simulcast stream's RID, which could be motivated by, for example, not providing sufficient quality.

The video source is offered to be sent as two simulcast streams, both with two alternative simulcast formats. Redundancy and repair are offered in the form of both flexible FEC and RTP retransmission. The flexible FEC is not bound to any particular RTP streams and is therefore able to be used across all RTP streams that are being sent as part of this media description.

```

o=fred 238947129 823479223 IN IP6 2001:db8::c000:27d
s=Offer from Simulcast-Enabled Client using Redundancy
c=IN IP6 2001:db8::c000:27d
t=0 0
a=group:BUNDLE foo bar
m=audio 49200 RTP/AVP 97 98 99 100 101 102
a=mid:foo
a=rtpmap:97 G711/8000
a=rtpmap:98 LPC/8000
a=rtpmap:99 OPUS/48000/1
a=rtpmap:100 RED/8000/1
a=rtpmap:101 CN/8000
a=rtpmap:102 telephone-event/8000
a=fmtp:99 useinbandfec=1;usedtx=0
a=fmtp:100 97/98
a=fmtp:102 0-15
a=ptime:20
a=maxptime:40
a=rid:1 send pt=99,102;max-br=64000
a=rid:2 send pt=100,97,101,102
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
a=simulcast:send 1;2
m=video 49600 RTP/AVPF 103 104 105 106 107
a=mid:bar
a=rtpmap:103 H264/90000
a=rtpmap:104 VP8/90000
a=rtpmap:105 rtx/90000
a=rtpmap:106 rtx/90000
a=rtpmap:107 flexfec/90000
a=fmtp:103 profile-level-id=42c00d;max-fs=3600;max-mbps=108000
a=fmtp:104 max-fs=3600; max-fr=30
a=fmtp:105 apt=103;rtx-time=200
a=fmtp:106 apt=104;rtx-time=200
a=fmtp:107 repair-window=100000
a=rid:1 send pt=103;max-width=1280;max-height=720;max-fps=30
a=rid:2 send pt=104;max-width=1280;max-height=720;max-fps=30
a=rid:3 send pt=103;max-width=640;max-height=360;max-br=300000
a=rid:4 send pt=104;max-width=640;max-height=360;max-br=300000
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
a=extmap:2 urn:ietf:params:rtp-hdext:sdes:rtp-stream-id
a=extmap:3 urn:ietf:params:rtp-hdext:sdes:repaired-rtp-stream-id
a=rtcp-fb:* ccm pause nowait
a=simulcast:send 1,2;3,4

```

Figure 8: Simulcast and Redundancy Example

6. RTP Aspects

This section discusses what the different entities in a simulcast media path can expect to happen on the 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 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.

6.1. 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 it ([Section 5.5](#)) 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 \[RFC8843\]](#) is used, the MID SDES item will also be sent, similarly to the `RtpStreamId`.

Each RTP stream might 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 SSRCs (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.

6.2. RTP Middlebox to Receiver

RTP streams in a multiparty 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, the RTP stream aspects will fall under the middlebox-to-middlebox case ([Section 6.3](#)).

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 number of media sources available in a multimedia 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 metadata, 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 signaling 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 signaling 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 assume that `RtpStreamId` is not used in the mixer to receiver direction.

6.2.1. Media-Switching Mixer

This section discusses the behavior in cases where the RTP middlebox behaves like the media-switching mixer in RTP topologies ([Section 3.6.2 of \[RFC7667\]](#)). 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 the RTP stream produced by the mixer being 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. As an example of the latter, 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 each of the RTP mixer's media sources. At any point in time, this RTP stream is 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 signaling 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 redundancy or other auxiliary RTP streams.

The mixer may communicate the identity of the originating media source to the receiver by including the Contributing Source (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. That is, there is nothing in the signaling between the mixer and the receiver that is structured around the originating media sources, only the mixer's media sources. If they were associated with the SSRC, the receiver would likely believe that there has been an SSRC collision and the RTP stream is spurious, because 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 an 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 the scenario described by this section, 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 signaling context. In addition, the defined RtpStreamIds and their parameters need to cover all the media sources and simulcast streams received by the RTP mixer that can be switched into this media source, sent by the RTP mixer.

6.2.2. Selective Forwarding Middlebox

This section discusses the behavior in cases where the RTP middlebox behaves like the Selective Forwarding Middlebox in RTP topologies ([Section 3.7](#) of [\[RFC7667\]](#)). Applications for this type of RTP middlebox result in each originating media source having a corresponding media source on the leg between the middlebox and the receiver. A Selective Forwarding Middlebox (SFM) could go as far as exposing all the simulcast streams for a 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 the 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 the individual RTP stream(s) for one media source being able to switch between being active and 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.

An 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 of whether 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 nonincreasing RTCP SR field values and timers need to be used to determine pause in RTP packet delivery. As a result, when the last RTP packet arrives (if it arrives), one usually cannot determine that this will be the last. That it was the last is something that one learns later.

6.3. 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 middleboxes A and B, this looks very similar to the originating-to-middlebox case on a media-source basis. However, in this case, there are 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. As a result, individual RTP streams can become paused at any point and later be resumed based on various criteria.

The MIDs used between A and B are the ones agreed between these two identities in signaling. 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.

7. Network Aspects

Simulcast is in this memo defined as the act of sending multiple alternative encoded streams of the same underlying media source. Transmitting multiple independent streams that originate from the same source 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 using the Multiplexing Features of RTP to Support Multiple Media Streams](#)" [[MULTIPLEX](#)]. Discussion and clarification on how to handle multiple streams in an RTP session can be found in [[RFC8108](#)].

The network aspects that are relevant for simulcast are:

Quality of Service (QoS): 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-bitrate streams may be prioritized over higher-bitrate streams to minimize congestion or packet losses in the low-bitrate streams. Thus, there is a benefit to using a simulcast solution with good QoS support.

NAT/FW Traversal (Network Address Translator / Firewall Traversal): Using multiple RTP sessions incurs more cost for NAT/FW traversal unless they can reuse the same transport flow, which can be achieved by [multiplexing negotiation using SDP port numbers \[RFC8843\]](#).

7.1. Bitrate Adaptation

Use of multiple simulcast streams can require a significant amount of network resources. The aggregate bandwidth for all simulcast streams for a media source (and thus SDP media description) is bounded by any SDP "b=" line applicable to that media source. It is assumed that a suitable congestion-control mechanism is used by the application to ensure that it doesn't cause persistent congestion. 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. When a simulcasting media source uses a single media transport for all of the simulcast streams, it is likely that a joint congestion control across all simulcast streams is used for that media source. 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 5.2](#)). 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.

8. 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, which limits the possibilities for applying 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 a multicast context is out of scope for the current document and would require additional specification.

9. IANA Considerations

This document registers a new media-level SDP attribute, "simulcast", in the "att-field (media level only)" registry within the "Session Description Protocol (SDP) Parameters" registry, according to the procedures of [\[RFC4566\]](#) and [\[RFC8859\]](#).

Contact name, email: The IESG (iesg@ietf.org)

Attribute name: simulcast

Long-form attribute name: Simulcast stream description

Charset dependent: No

Attribute value: sc-value; see [Section 5.1](#) of RFC 8853.

Purpose: Signals simulcast capability for a set of RTP streams

Mux category: NORMAL

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. However, there may 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.

Integrity protection and source authentication of all SDP signaling, including simulcast attributes, can mitigate the risks of such attacks that attempt to alter signaling.

Security considerations related to the use of "a=rid" and the RtpStreamId SDES item are covered in [\[RFC8851\]](#) and [\[RFC8852\]](#). There are no additional security concerns related to their use in this specification.

11. References

11.1. Normative References

[\[RFC2119\]](#)

- Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <<https://www.rfc-editor.org/info/rfc7405>>.
- [RFC7728] Burman, B., Akram, A., Even, R., and M. Westerlund, "RTP Stream Pause and Resume", RFC 7728, DOI 10.17487/RFC7728, February 2016, <<https://www.rfc-editor.org/info/rfc7728>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8843] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", RFC 8843, DOI 10.17487/RFC8843, April 2020, <<https://www.rfc-editor.org/info/rfc8843>>.
- [RFC8851] Roach, A.B., Ed., "RTP Payload Format Restrictions", DOI 10.17487/RFC8851, RFC 8851, April 2020, <<https://www.rfc-editor.org/info/rfc8851>>.
- [RFC8852] Roach, A.B., Nandakumar, S., and P. Thatcher, "RTP Stream Identifier Source Description (SDS)", DOI 10.17487/RFC8852, RFC 8852, April 2020, <<https://www.rfc-editor.org/info/rfc8852>>.
- [RFC8859] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", RFC 8859, DOI 10.17487/RFC8859, April 2020, <<https://www.rfc-editor.org/info/rfc8859>>.

11.2. Informative References

[MULTIPLEX]

- Westerlund, M., Burman, B., Perkins, C., Alvestrand, H., and R. Even, "Guidelines for using the Multiplexing Features of RTP to Support Multiple Media Streams", Work in Progress, Internet-Draft, draft-ietf-avtcore-multiplex-guidelines-12, 16 June 2020, <<https://tools.ietf.org/html/draft-ietf-avtcore-multiplex-guidelines-12>>.
- [RFC2198]** Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J.C., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <<https://www.rfc-editor.org/info/rfc2198>>.
- [RFC3389]** Zopf, R., "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, DOI 10.17487/RFC3389, September 2002, <<https://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, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc5109>>.
- [RFC5583]** Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", RFC 5583, DOI 10.17487/RFC5583, July 2009, <<https://www.rfc-editor.org/info/rfc5583>>.
- [RFC6184]** Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <<https://www.rfc-editor.org/info/rfc6184>>.
- [RFC6190]** Wenger, S., Wang, Y.-K., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <<https://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, <<https://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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc7656>>.
- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<https://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, <<https://www.rfc-editor.org/info/rfc7741>>.
- [RFC7941] Westerlund, M., Burman, B., Even, R., and M. Zanaty, "RTP Header Extension for the RTP Control Protocol (RTCP) Source Description Items", RFC 7941, DOI 10.17487/RFC7941, August 2016, <<https://www.rfc-editor.org/info/rfc7941>>.
- [RFC8108] Lennox, J., Westerlund, M., Wu, Q., and C. Perkins, "Sending Multiple RTP Streams in a Single RTP Session", RFC 8108, DOI 10.17487/RFC8108, March 2017, <<https://www.rfc-editor.org/info/rfc8108>>.
- [RFC8627] Zanaty, M., Singh, V., Begen, A., and G. Mandyam, "RTP Payload Format for Flexible Forward Error Correction (FEC)", RFC 8627, DOI 10.17487/RFC8627, July 2019, <<https://www.rfc-editor.org/info/rfc8627>>.

Appendix A. Requirements

The following requirements are met by the defined solution to support the [use cases](#) (Section 3):

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 that a received RTP stream is associated with, knowing the content of the SDP signaling.

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](#) [RFC8843] SDP media descriptions.

REQ-3: Capability negotiation. The following must be possible:

REQ-3.1: The sender can express capability of sending simulcast.

- REQ-3.2: The receiver can express capability of receiving simulcast.
- REQ-3.3: The sender can express the maximum number of simulcast streams that can be provided.
- REQ-3.4: The receiver can express the maximum number of simulcast streams that can be received.
- REQ-3.5: The sender can detail the characteristics of the simulcast streams that can be provided.
- REQ-3.6: The 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 DTME.
 - 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 nonsimulcast legacy clients using a single media source per media type.
 - REQ-6.2: WebRTC environment with a single media source per SDP media description.

Acknowledgements

The authors would like to thank Bernard Aboba, Thomas Belling, Roni Even, Adam Roach, Iñaki Baz Castillo, Paul Kyzivat, and Arun Arunachalam for the feedback they provided during the development of this document.

Contributors

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

Authors' Addresses

Bo Burman

Ericsson
Gronlandsgatan 31
SE-164 60 Stockholm
Sweden
Email: bo.burman@ericsson.com

Magnus Westerlund

Ericsson
Torshamnsgatan 23
SE-164 83 Stockholm
Sweden
Email: magnus.westerlund@ericsson.com

Suhas Nandakumar

Cisco
170 West Tasman Drive
San Jose, CA 95134
United States of America
Email: snandaku@cisco.com

Mo Zanaty

Cisco
170 West Tasman Drive
San Jose, CA 95134
United States of America
Email: mzanaty@cisco.com