

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

M. Westerlund
B. Burman
M. Lindqvist
F. Jansson
Ericsson
July 16, 2012

Using Simulcast in RTP sessions
draft-westerlund-avtcore-rtp-simulcast-01

Abstract

In some applications it may be necessary to send multiple media streams derived from the same media source. This is called Simulcast. This document discusses the best way of accomplishing this in RTP. It is concluded that a session based solution provides best support for simulcast, and a solution for that is defined. There are two necessary extensions. The first extension is how to group RTP sessions belonging to the same simulcast source using the grouping framework, and the second is how to identify which SSRCs that are the same media source by using a new RTCP SDES item SRCNAME.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

Copyright (c) 2012 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 (<http://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	4
2.	Definitions	5
2.1.	Terminology	5
2.2.	Requirements Language	5
3.	Simulcast and Applicability	5
3.1.	Simulcasting to RTP Mixer	5
3.1.1.	Simulcast Combined with Scalable Encoding	7
3.2.	Multicast Transported Simulcasted Media	7
3.2.1.	Diversity in Receiver Population	7
3.2.2.	Bit-rate Adaptation	8
3.3.	Simulcasting to a Consuming End-Point	9
3.4.	Same Encoding to Multiple Destinations	9
3.5.	Different Encoding to Independent Destinations	10
4.	Simulcast Alternatives	10
4.1.	Using the Payload Type	11
4.2.	Using Single RTP session	11
4.3.	Using Multiple RTP sessions	11
5.	Analysis	12
5.1.	RTP/RTCP Aspects	12
5.2.	Signalling Aspects	13
5.3.	Network Aspects	13
5.4.	Security Aspects	14
5.5.	Summary	14
6.	Signaling Support for Multiple RTP session based Simulcast	15
6.1.	Grouping Simulcast RTP Sessions	15
6.1.1.	Declarative Use	15
6.1.2.	Offer/Answer Use	16
6.2.	Media Stream Requirements	16
6.3.	Relating Alternative Encodings	16
6.4.	Multiple Stream handling	16
7.	Simulcast Signalling Examples	17
7.1.	Alice: Desktop Client	17
7.2.	Bob: Telepresence Room	19
7.3.	Fred: Dial-out to Legacy Client	23
7.4.	Joe: Dial-out to Desktop Client	26
8.	IANA Considerations	30
9.	Security Considerations	30
10.	Acknowledgements	30
11.	References	30
11.1.	Normative References	30
11.2.	Informative References	31
	Authors' Addresses	32

1. Introduction

Simulcast is the act of simultaneously sending multiple different versions of the same media content, e.g. the same video source encoded with different video encoders. This can be done in several ways and for different purposes. This document focuses on the case where one wants to provide multiple streams with different encodings over RTP [[RFC3550](#)] towards an intermediary so that the intermediary can select which encoding to forward to other participants in the session, and more specifically how the grouping of the streams is defined.

The different encodings of a media content considered in this document can differ in:

Bit-rate: The difference is the amount of bits spent to encode the media thus giving different quality.

Codec: Different media codecs are used to ensure that different receivers that do not have a common set of decoders can decode at least one of the versions. This can include codec configuration options that are not compatible, like video encoder profiles, or the capability of receiving the transport packetization.

Sampling: Different sampling of media, in spatial as well as in temporal domain, may be used to suit different rendering capabilities or needs at the receiving endpoints, as well as a method to achieve different bit-rates. 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. Obviously, a difference in sampling may result in difference in bit-rate.

There are different reasons for an application to provide a single media source in different encodings. As soon as an application has the need to send multiple encodings, there is a potential need for simulcast. This need can arise even when using media codecs that have scalability features built in. The purpose of this document is to find the most suitable solution for the non-trivial variants of simulcast and in order to do this, different ways of multiplexing the different encodings are discussed. Following the presentation of the alternatives, an analysis is performed on how different aspects like RTP mechanisms, signaling possibilities, and network features are affected by the alternatives. This is a specific application of the aspects discussed in RTP Multiplexing Architecture [[I-D.westerlund-avtcore-multiplex-architecture](#)]. The discussion results in a conclusion, a solution, and a proposal for the

standardization work required to support simulcast.

[2.](#) Definitions

[2.1.](#) Terminology

The following terms and abbreviations are used in this document:

Encoding: A particular encoding is the choice of the media encoder (codec) that has been used to compress the media and the fidelity of that encoding through the choice of sampling, bit-rate and other codec configuration parameters.

Different encodings: An encoding is different when some parameter that characterize the encoding of a particular media source is changed. Such changes can be one or more of the following parameters; codec, codec configuration, bit-rate, sampling.

Simulcast versions: Media streams used for simulcast that use different encodings and thus constitute different versions of the same media source.

[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.](#) Simulcast and Applicability

This section discusses different usage scenarios for the term simulcast and clarifies which of those this document focuses on. It also reviews why simulcast and scalable codecs can be a useful combination.

[3.1.](#) Simulcasting to RTP Mixer

This scenario relates to a multi-party session where one or more central nodes are used to facilitate the media transport between the session participants. Thus, this targets the RTP Mixer Topology defined in [[RFC5117](#)] ([Section 3.4](#): Topo-Mixer). This scenario is targeted for further discussion in this document.

Simulcasting different media encodings of video that differ both in resolution and in bit-rate is highly applicable to video conferencing scenarios. For example, an RTP mixer selects the video of the most

active speaker and sends that participant's video stream as a high resolution stream to the other participants, and in addition also sends a number of low resolution video streams of the other participants, enabling the receiving user to both display the current speaker in high quality and monitor the other participants in lower quality/resolution/size. As the participants should not receive the stream showing themselves, the set of streams will be unique to all participants.

A number of alternatives exist to provide both high and low resolutions from an RTP Mixer:

Simulcast: The clients send one stream for the low resolution and another for the high resolution.

Scalable Video Coding: The clients are using a video encoder that can provide one stream that is both providing the high resolution and also enables the mixer to extract a low resolution representation from that single stream.

Transcoding in the Mixer: The clients send a high resolution stream to the RTP Mixer which performs a transcoding to a lower resolution stream.

The Transcoding alternative requires that the RTP mixer has sufficient amount of transcoding resources to produce the number of low resolution streams required. In worst case, all participants' streams may need to be transcoded. If the resources are not available, a different solution is needed. There will also normally be a quality loss and an increase in latency associated with the transcoding operation.

Scalable video encoding requires a more complex encoder compared to non-scalable encoding. Also, if the resolution difference between the streams is large, a scalable codec may in fact be only marginally more bandwidth efficient than the simulcast case where the different resolutions are sent as separate streams from the clients to the mixer. At the same time, with scalable video encoding, the transmission of all but the lowest resolution will consume more bandwidth from the mixer to the other participants than with a non-scalable encoding.

Simulcasting has the benefit that it is conceptually simple. It enables the use of any media codec that the participants agree on, allowing the RTP mixer to be codec-agnostic. With the currently available video encoders, simulcasting may be less bit-rate efficient in the path from the sending client to the mixer but more efficient in the mixer to receiver path compared to Scalable Video Coding.

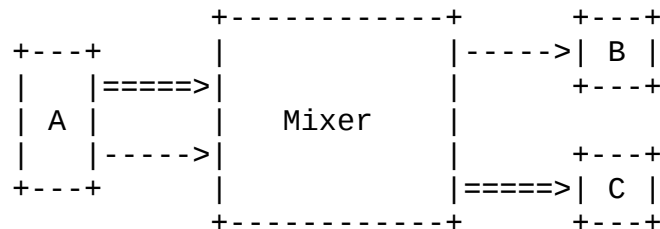


Figure 1: RTP Mixer selecting from simulcast versions

The sender A provides the mixer with both a high resolution version "====>" and a low resolution version "---->". The mixer selects who in it's receiver population should get a particular version.

3.1.1. Simulcast Combined with Scalable Encoding

As explained in the previous section, a scalable codec is not always more bandwidth efficient than simulcast, especially in the path from the mixer to the receiver.

There are however cases where a combination of simulcast and scalable encoding can be beneficial. By using simulcast in cases where the scalable codec is less efficient, one can optimize the efficiency of the complete system. A good example of this usage would be where the video is encoded using SVC transported in RTP [[RFC6190](#)], where each simulcast stream has a different resolution, and each SVC media stream uses temporal scalability and signal to noise ratio (SNR) scalability within that single media stream. If only resolution and temporal variations are needed, this can be implemented using the non-scalable part of H.264, as each simulcast version provides the different resolution, and each media stream within a simulcast encoding has temporal scalability through the use of non-reference frames.

3.2. Multicast Transported Simulcasted Media

When using multicast, particularly Source-Specific Multicast (SSM) [[RFC3569](#)] to distribute RTP/RTCP packets to a large receiver population one faces some issues. There are at least two different issues where simulcast can potentially be useful.

3.2.1. Diversity in Receiver Population

If there is any diversity in the receivers regarding e.g. capability, codec support or code base, there are potentially restrictions in what streams can be delivered to the receivers. If using the lowest common denominator over a diverse receiver population isn't acceptable, simulcast can be one possible solution. By offering

different stream alternatives, it is possible to let the receivers choose the simulcast version that matches their capabilities. By using explicit signalling for simulcast, it is not necessary for the stream distributor to handle multiple receiver configurations individually for a multi-media session, nor to ensure that each receiver gets an encoding that matches their capabilities.

The simulcast version granularity the receivers can select will be on multicast group level. Thus, this use case puts a strict requirement on supporting RTP session multiplexing. The reason being that having a single RTP session straddle several multicast groups makes any reporting on the received sources very difficult to interpret. Using one RTP session per simulcast version instead provides consistency.

3.2.2. Bit-rate Adaptation

If the network paths from the media sender to the receivers can support different bit-rates, there is a need to support media streams encoded to different bit-rates. If these path differences are of a more static nature, for example depending primarily on the underlying link layers, using simulcast has an advantage over scalable encoding. The reason is that the efficiency of scalable coding will never be better than encoding to a single target rate. When the receiver can determine current network interface connectivity, it can choose simulcast version with certainty. That choice will also be correct until the event of another network interface becoming the active one. This assumes that the multicast transmission uses dedicated resources and will thus not be congested due to other network traffic. To support this behavior, the signalling must support indication of which media streams that are alternatives to each other, and it is also necessary to be able to determine aggregate bit-rate for the selected multicast group(s) compared to available network properties.

Simulcast is possible to use also in more dynamic situations where each receiver continuously gathers reception statistics to detect path congestion and based on that may change which version to receive. The main issue with such usage is how to achieve a switch from one version to another with minimal playback interruption and also avoiding to put extra load on the network during the actual switch. Here, scalable encoding in general have better characteristics since scalability layers are typically synchronized.

When comparing simulcast and scalable encoding, the trade-offs are different and the down-sides occur at different places. Simulcast will have a higher bit-rate load at a media sender and that will also be the case for any network path shared between receivers of multiple simulcast versions. However, for parts of the network path where there is only a single simulcast version, the achievable quality at a

given bit-rate will be slightly higher for simulcast. It will also be more difficult to seamlessly switch between simulcast versions than between different scalable encodings, as simulcast actually switches from one media stream version to another instead of adding or removing some enhancement layers.

3.3. Simulcasting to a Consuming End-Point

This scenario is based on an RTP Transport Translator ([Section 3.3: Topo-Trn-Translator](#)) [[RFC5117](#)]. The transport translator functions as a relay and transmits all streams received from one participant to all other participants. For example, when simulcasting a low resolution and a high resolution video stream, the RTP Translator would send all the streams to all clients. This clearly increases the bit-rate transmitted on the paths to the clients compared to the mixer case in the previous section. The only simulcast benefit for the receiving client over a single stream scenario would be reduced decoding complexity for the low resolution streams. A single stream scenario which only transmits the high resolution stream would allow the receiver to decode it and scale it down to the desired resolution.

The usage of transport translator and simulcast becomes efficient if each receiving client is allowed to control or configure the relay with respect to which version it wants to receive. However, such usage of RTP has some potential issues with RTCP. One example is when a receiver has indicated to the transport translator that it does not want to receive a particular stream, but at the same time it is receiving and reporting on other streams from the same sender. In this case, the sender will receive no RTCP messages about the non-forwarded stream and therefore get the impression that the stream somehow is lost. Thus some consideration and mechanism are needed to support such a use case in order not to break RTCP reception reporting.

This scenario is considered in the continuation of the document but with less emphasis than on the RTP mixer case.

3.4. Same Encoding to Multiple Destinations

One interpretation of simulcast is when one encoding is sent to multiple receivers. This is well supported in RTP by simply copying all outgoing RTP and RTCP traffic to several transport destinations, if the intention is to create a common RTP session. As long as all participants do the same, a full mesh is constructed and everyone in the multi party session have a similar view of the joint RTP session. This is analog to an Any Source Multicast (ASM) session but without the traffic optimization as multiple copies of the same content is

likely to have to pass over the same link.

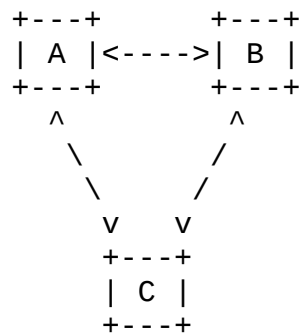


Figure 2: Full Mesh / Multi-unicast

As this type of simulcast is analog to ASM usage and RTP has good support for ASM sessions, no further consideration for this scenario is made in this document.

3.5. Different Encoding to Independent Destinations

Another alternative interpretation of simulcast is multiple destinations, where each destination gets a specifically tailored version, but where the destinations are independent. A typical example for this would be a streaming server distributing the same live session to a number of receivers, adapting the quality and resolution of the multi-media session to each receiver's capability and available bit-rate. This case can be solved in RTP by having independent RTP sessions between the sender and the receivers. Thus this case is not considered further.

4. Simulcast Alternatives

Simulcast is defined in this document as the act of sending multiple alternative encodings 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. The below sub-sections describe potential ways of achieving stream multiplexing and identification of which streams are alternative encodings of the same source. In the following descriptions it is also included how this interacts with multiple sources (SSRCs) in the same RTP session for other reasons than simulcast. Multiple SSRCs may occur for various reasons such as multiple participants in multipoint topologies such as multicast, transport relays or full mesh transport simulcasting, multiple source devices, such as multiple cameras or microphones at one end-point, or other RTP mechanisms such as RTP Retransmission [[RFC4588](#)].

4.1. Using the Payload Type

This alternative uses only the RTP payload type to identify the different simulcast streams. Thus all simulcast streams would be sent in the same RTP session using only a single SSRC per actual media source. However, as discussed in Guidelines for using the Multiplexing Features of RTP

[[I-D.westerlund-avtcore-multiplex-architecture](#)], using Payload Type Multiplexing does not work and is hereby dismissed as potential solution.

4.2. Using Single RTP session

This idea is based on using a unique SSRC for each alternative encoding of an actual media source within a single RTP session. The identification of how streams are considered to be alternative needs an additional mechanism, for example using SSRC grouping [[RFC5576](#)] and a new SDES item such as SRCNAME proposed in [[I-D.westerlund-avtext-rtcp-sdes-srcname](#)] with a semantics that indicate them as alternatives of a particular media source. When there are multiple actual media sources in a session, each media source will have to use a number of SSRCS to represent the different alternatives it produces. For example, if all actual media sources are similar and produce the same number of simulcast versions, there will be $n*m$ SSRCS in use in the RTP session, where n is the number of actual media sources and m the number of simulcast versions they can produce. Each SSRC can use any of the configured payload types for this RTP session. All session level attributes and parameters that are not source specific will apply and must function with all the alternative encodings intended to be used.

4.3. Using Multiple RTP sessions

Using multiple RTP sessions means that each different simulcast version of an actual media source is transmitted in a separate RTP session, using whatever session identifier to distinguish the different versions. This solution needs explicit session grouping [[RFC5888](#)] with a semantics that indicate them as alternatives. It is also important to identify the SSRCS in the different sessions that are alternative encodings of the same media source. This could be accomplished using the same SSRC across the sessions, but that is not robust against SSRC collisions and could potentially force cascading SSRC changes between sessions. A better choice would be to use the same value for the a new SDES item proposed in [[I-D.westerlund-avtext-rtcp-sdes-srcname](#)]. Each RTP session will have its own set of configured RTP payload types available for use with any SSRC in that session. In addition, all other attributes for sessions or sources can be used as normal to indicate the

configuration of that particular alternative.

5. Analysis

This section provides an analysis of simulcast as a specific case of the aspects discussed in Guidelines for using the Multiplexing Features of RTP [[I-D.westerlund-avtcore-multiplex-architecture](#)] to determine what is the most suitable solution. The below section discusses the relevant points for simulcast and contrasts using only SSRCs with using both RTP sessions and SSRC.

5.1. RTP/RTCP Aspects

The RTP/RTCP aspects of relevance are:

RTP Specification: From a base RTP specification point of view, there is no real difference between a single RTP session or using multiple RTP sessions.

Multiple SSRC Legacy Considerations: Dealing with legacy handling of multiple SSRCs in one RTP session for simulcast is a minor issue as end-points supporting simulcast will implement the necessary support. They should also determine if there is necessary support based on signalling. However, for cases where usage of simulcast is combined with legacy in the same scenario, multiple RTP sessions will have an advantage as the number of SSRCs in each session does not increase due to simulcast, only the number of sessions.

Cross Session RTCP Requests: In the case of simulcast, the findings in the architecture document stands and might be relevant when switching between simulcast versions to configure current code control state.

Binding Related Sources: Simulcast will require a clear binding between the SSRCs carrying the different simulcast versions. This issue will be independent of using one or multiple RTP sessions.

Transport Translators: Transport translators and simulcast is not the best match. This as the core of the functionality desired in simulcast is usually to be able to switch between alternatives, which is not really possible with transport translators as they do not manipulate the media streams. However, if one uses multiple RTP sessions, a session participant can control the simulcast version it receives in a very coarse grained fashion by joining the right RTP session. However, it is not capable of switching individual sources within the sessions.

Regarding RTP/RTCP aspects, multiple RTP sessions based solution can handle legacy better, while an single RTP session solution has some advantage if there is need for synchronized requests across multiple stream versions, but there are no major differences.

5.2. Signalling Aspects

The signalling aspects is one of the major issues for simulcast. In the currently used signalling system based on SDP [[RFC4566](#)] and Offer/Answer [[RFC3264](#)], the properties of media streams are negotiated on RTP session level. This is discussed in [Section 7.3.1](#) of the Guidelines for using the Multiplexing Features of RTP [[I-D.westerlund-avtcore-multiplex-architecture](#)].

As simulcast is all about being able to signal and negotiate what the different simulcast versions should be, it becomes important that the signalling supports such usage. A SSRC only solution does not prevent such signalling to be developed, but SSRC centric signalling is currently almost non-existent. If Session and SSRC based solution is used instead, it is already possible to signal and negotiate the version properties on a session level. Negotiated media properties will apply to all media sources sent in the same RTP session, which is likely not an issue in most cases. For example, using a common simulcast version definition across all media sources at one end-point will allow an RTP mixer choose both which media sources and which simulcast versions of them to forward towards the other end-points.

From a signalling perspective, the only rapid way forward is multiple RTP sessions based solution.

5.3. Network Aspects

The network aspects that have any relevance for simulcast are:

Quality of Service: When using simulcast it might be of interest to prioritize a particular simulcast version, rather than applying equal treatment of all versions. For example, lower bit-rate versions may be prioritized over higher bit-rate versions to minimize congestion or packet losses in the low bit-rate versions. Thus, there is a benefit to use a simulcast solution that supports QoS as good as possible. By using RTP sessions over different transport flows, a simulcast version can be prioritized by flow based QoS mechanisms. If the application would like to prioritize a particular media source in one simulcast version then the two proposals are equal.

NAT/FW Traversal: Using multiple RTP sessions will incur more cost for NAT/FW traversal unless the solution for multiplexing multiple RTP sessions on a single lower layer transport [[I-D.westerlund-avtcore-transport-multiplexing](#)] is used, in which cases they are basically equal. That is both from NAT/FW traversal perspective and QoS possibilities. If flow based QoS with any differentiation is desirable, the cost for additional transport flows is likely necessary.

Multicast: To enable simulcast to be combined with multicast, it will be required to use multiple RTP sessions. Multicast groups need be separate for the different versions to allow a multicast receiver to pick the version 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.

Using multiple RTP Sessions are clearly the better choice when taking network aspects into account. Multiple RTP Sessions are required to support any multicast usage. In addition, it can provide support for differentiated flow based QoS. The extra NAT/FW traversal costs can be mitigated completely by multiplexing all RTP sessions over a single transport.

[5.4.](#) Security Aspects

The discussed security aspects has the following applicability or considerations when it comes to simulcast:

Security Context Scope: Both issues may be applicable to simulcast usage. If differentiation enforcement is based on encryption and keying then multiple RTP session based simulcast has a slight benefit.

Key-Management: There is no significant difference in the solution except that multiple RTP sessions may require keying more contexts. Having more contexts is also what brings additional freedom to make differentiation.

There is a small difference in security aspects where multiple RTP sessions provides more freedom, but also a higher cost in the amount of contexts needing to be keyed.

[5.5.](#) Summary

Defining multiple RTP sessions based simulcast appears to be the best choice. It supports the most use cases including the multicast based one, it has better support for flow based QoS, and the NAT/FW costs

can be mitigated. When it comes to signalling, multiple RTP sessions based simulcast appears to require a modest set of extensions to work, while a single RTP session seems to require large amounts of extensions to enable sets of SSRC to negotiate different parameters that differentiate the simulcast versions. Multiple RTP sessions also provide greater flexibility when it comes to key-management choices for the applications.

A single RTP session solution, as a complement to the multiple RTP sessions, is not considered due to the large amount of extensions required for signalling. The needed extensions to support single RTP session simulcast may be defined in the future.

6. Signaling Support for Multiple RTP session based Simulcast

To enable the usage of multiple RTP sessions based simulcast, some minimal additional signaling support is required. That support is discussed in this section. First of all, there is a need for a mechanism to identify the RTP sessions carrying simulcast versions from the same media source. Secondly, a receiver needs to be able to identify the SSRCs in the different sessions belonging to the same media source. Beyond the necessary signaling support for simulcast, some very useful optimizations regarding transmission of media streams are described that will also help RTP mixers to select which stream alternatives to deliver to a specific client, or request a client to encode in a particular way.

6.1. Grouping Simulcast RTP Sessions

The proposal is to define a new grouping semantics for the session groupings framework [[RFC5888](#)]. There is a need to separate the semantics of intent to send simulcast streams from the capability to recognize and receive simulcast streams. For that reason two new simulcast grouping semantics are defined, "SimulCast Receive" (SCR) and "SimulCast Send" (SCS). They both act as an indicator that session level simulcast is desired and provide one set of RTP sessions that carries simulcast versions of media sources. There may be multiple sets of RTP Sessions that carries simulcast versions.

6.1.1. Declarative Use

When used as a declarative media description, SCR indicates the configured end-point's required capability to recognize and receive a specified set of RTP streams as simulcast streams. In the same fashion, SCS requests the end-point to send a specified set of RTP streams as simulcast streams. SCR and SCS MAY be used independently and at the same time and they need not specify the same or even the

same number of RTP sessions in the group.

[6.1.2.](#) Offer/Answer Use

When used in an offer, SCS indicates the SDP providing agent's intent of sending simulcast and the particular set of RTP sessions, and SCR indicates the agent's capability of receiving simulcast streams within the configured set of RTP Sessions. SCS and SCR MAY be used independently and at the same time and they need not specify the same or even the same number of RTP sessions in the group. The answerer MUST change SCS to SCR and SCR to SCS in the answer, given that it has and wants to use the corresponding (reverse) capability. An answerer not supporting the SCS or SCR direction, or not supporting SCS or SCR grouping semantics at all, will remove that grouping attribute altogether, according to the grouping framework [[RFC5888](#)]. An offerer that receives an answer indicating lack of simulcast support in one or both directions, where SCR and/or SCS grouping are removed, MUST NOT use simulcast in the non-supported direction(s).

[6.2.](#) Media Stream Requirements

When doing simulcast, the media streams that are alternatives need certain considerations to ensure that switching between alternative streams are as issue-free as possible. The following considerations are needed:

Same Clock Base: To enable correct alignment of media packets on the source time-line, all alternative streams (SSRCs) MUST use the same underlying clock to relate their RTP timestamp values with the network time protocol (NTP) formatted sender time in the RTCP Sender Reports.

[6.3.](#) Relating Alternative Encodings

To ensure that simulcast streams can be related correctly, the usage of the SDES SRCNAME [[I-D.westerlund-avtext-rtcp-sdes-srcname](#)] with the same value across simulcast versions is belonging to the same media source is REQUIRED.

[6.4.](#) Multiple Stream handling

The grouping semantics SCR and SCS SHOULD be combined with the SDP attributes "a=max-send-ssrc" and "a=max-recv-ssrc" [[I-D.westerlund-avtcore-max-ssrc](#)] to indicate the number of simultaneous streams of each encoding that may be sent or that can be handled in the receive direction.

7. Simulcast Signalling Examples

This example is for a case of client to video conference service using a centralized media topology with an RTP mixer. Alice and Bob calls into a conference server for a conference call with audio and video sent to the RTP mixer, these clients being capable to send a few video simulcast versions. The conference server also dials out to Fred, which is a legacy client resulting in fallback behavior. When dialing out to Joe, more functionality is enabled as Joe is a client similar to Alice.

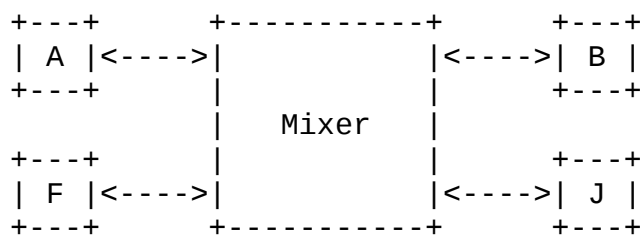


Figure 3: Four-party Mixer-based Conference

Example of Media plane for RTP mixer based multi-party conference with 4 participants.

7.1. Alice: Desktop Client

Alice is calling in to the mixer with an audiovisual single stream desktop client, only adding capability to send simulcast and announce SRCNAME, compared to a legacy client. The offer from Alice looks like

```
v=0
o=alice 2362969037 2362969040 IN IP4 192.0.2.156
s=Simulcast enabled Desktop Client
t=0 0
c=IN IP4 192.0.2.156
b=AS:825
a=group:SCS 2 3
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:145
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=ssrc:521923924 cname:alice@foo.example.com
a=ssrc:521923924 srcname:a
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:520
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=ssrc:192392452 cname:alice@foo.example.com
a=ssrc:192392452 srcname:v
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:160
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=ssrc:239245219 cname:alice@foo.example.com
a=ssrc:239245219 srcname:v
a=mid:3
a=sendonly
```

Figure 4: Alice Offer for a Simulcast Conference

As can be seen from the SDP, Alice has a simulcast-enabled client and offers two different simulcast versions sent from her single camera, indicated by the SCS grouping tag and the two media IDs (2 and 3). The first video version with media ID 2 prefers 360p resolution (signaled via imageattr) and the second video version with media ID 3 prefers 180p resolution. The first video media line also acts as the single receive video (making media line sendrecv), while the second video media line is only related to simulcast transmission and is thus offered sendonly. The two simulcast encoding streams and its related audio stream are bound together using SRCNAME SDP item with the identifier "v", a single level is required in this case. We also

declare the end-point CNAME as all sources belong to the same synchronization context.

7.2. Bob: Telepresence Room

Bob is calling in to the mixer with a telepresence client that has capability for both sending multi-stream, receiving and local rendering of those multiple streams, as well as sending simulcast versions to the mixer. More specifically, in this example the client has three cameras, each being sent in three different simulcast versions. In the receive direction, up to two main screens can show video from a (multi-stream) conference participant being active speaker, and still more screen estate can be used to show videos from up to 16 other conference listeners. Each camera has a corresponding (stereo) microphone that can also be negotiated down to mono by removing the stereo payload type from the answer. The capability to send and receive multiple SSRC in the same RTP session is explicitly announced through use of RTP multi-stream signalling [[I-D.westerlund-avtcore-max-ssrc](#)].

```
v=0
o=bob 129384719 9834727 IN IP4 192.0.2.35
s=Simulcast Enabled Multi Stream Telepresence Client
t=0 0
c=IN IP4 192.0.2.35
b=AS:6035
a=group:SCS 2 3 4
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:435
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=max-send-ssrc:* 3
a=max-recv-ssrc:* 3
a=ssrc:724847850 cname:bob@foo.example.com
a=ssrc:724847850 srcname:a1
a=ssrc:2847529901 cname:bob@foo.example.com
a=ssrc:2847529901 srcname:a2
a=ssrc:57289389 cname:bob@foo.example.com
a=ssrc:57289389 srcname:a3
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:4500
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:* send [x=1280,y=720] recv [x=1280,y=720]
[x=640,y=360] [x=320,y=180]
a=max-send-ssrc:96 3
```

```
a=max-recv-ssrc:96 2
a=ssrc:75384768 cname:bob@foo.example.com
a=ssrc:75384768 srcname:v1
a=ssrc:2934825991 cname:bob@foo.example.com
a=ssrc:2934825991 srcname:v2
a=ssrc:3582594238 cname:bob@foo.example.com
a=ssrc:3582594238 srcname:v3
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:1560
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=640,y=360]
a=max-send-ssrc:96 3
a=ssrc:1371234978 cname:bob@foo.example.com
a=ssrc:1371234978 srcname:v1
a=ssrc:897234694 cname:bob@foo.example.com
a=ssrc:897234694 srcname:v2
a=ssrc:239263879 cname:bob@foo.example.com
a=ssrc:239263879 srcname:v3
a=mid:3
a=sendonly
m=video 49500 RTP/AVP 96
b=AS:420
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=max-send-ssrc:96 3
a=ssrc:485723998 cname:bob@foo.example.com
a=ssrc:485723998 srcname:v1
a=ssrc:2345798212 cname:bob@foo.example.com
a=ssrc:2345798212 srcname:v2
a=ssrc:1295729848 cname:bob@foo.example.com
a=ssrc:1295729848 srcname:v3
a=mid:4
a=sendonly
m=video 49600 RTP/AVP 96 97 98
b=AS:2600
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:96 recv [x=1280,y=720]
a=rtpmap:97 H264/900000
a=fmtp:97 profile-level-id=42c01e
a=imageattr:97 recv [x=640,y=360]
a=rtpmap:98 H264/900000
a=fmtp:98 profile-level-id=42c00d
a=imageattr:98 recv [x=320,y=180]
```

```
a=max-recv-ssrc:96 1
a=max-recv-ssrc:97 4
a=max-recv-ssrc:98 16
a=max-recv-ssrc:* 16
a=mid:5
a=recvonly
a=content:alt
```

Figure 5: Bob Offer for a Multi-stream and Simulcast Telepresence Conference

Bob has a three-camera, three-screen, simulcast-enabled client with even higher performance than Alice's and can additionally support 720p video, as well as multiple receive streams of various resolutions. The client implementor has thus decided to offer three simulcast streams for each camera, indicated by the SCS grouping tag and the three media IDs (2, 3, and 4) in the SDP.

The first video media line with media ID 2 indicates the ability to send video from three simultaneous video sources (cameras) through the max-send-ssrc attribute with value 3. This media line is also marked as the main video by using the content attribute from [\[RFC4796\]](#). Also the receive direction has declared ability to handle multiple video sources, and in this example it is 2. The interpretation of content:main for those two streams in the receive direction is that the client expects and can present (in prime position) at most two main (active speaker) video streams from another multi-camera client.

The second and third video media lines with media ID 3 and 4 are the sendonly simulcast streams. Through the grouping, they can implicitly be interpreted as also being content:main for the send direction, but is not marked as such since multiple media blocks with content:main could be confusing for a legacy client.

The fourth video media line with media ID 5 is recvonly and is marked with content:alt. That media line should, as was intended for that content attribute value, receive alternative content to the main speaker, such as "audience". In a multi-party conference, that could for example be the next-to-most-active and/or non-active speakers. The SDP describes that those streams can be presented in a set of different resolutions, indicated through the different payload types. The maximum number of streams per payload type is indicated through the max-recv-ssrc attribute. In this example, at most one stream can have payload type 96, preferably 720p, as indicated by the related imageattr line. Similarly, at most 4 streams can have payload type 97, preferably using 360p resolution, and at most 16 streams can have payload type 98, preferably of 180p resolution. In any case, there

must never be more than 16 simultaneous streams of any payload type, but combinations of payload types may occur, such as for example two streams using payload type 97 and 8 streams using payload type 98.

The answer from a simulcast-enabled RTP mixer to this last SDP could look like:

```
v=0
o=server 238947290 239573929 IN IP4 192.0.2.2
s=Multi stream and Simulcast Telepresence Bob Answer
c=IN IP4 192.0.2.43
b=AS:7065
a=group:SCR 2 3 4
m=audio 49200 RTP/AVP 96
b=AS:435
a=rtpmap:96 G719/48000/2
a=max-send-ssrc:96 3
a=max-recv-ssrc:96 3
a=ssrc:4111848278 cname:server@conf1.example.com
a=ssrc:4111848278 srcname:r1
a=ssrc:835978294 cname:server@conf1.example.com
a=ssrc:835978294 srcname:r2
a=ssrc:2938491278 cname:server@conf1.example.com
a=ssrc:2938491278 srcname:r3
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:4650
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:* send [x=1280,y=720] [x=640,y=360] [x=320,y=180]
             recv [x=1280,y=720]
a=max-recv-ssrc:96 3
a=max-send-ssrc:96 2
a=ssrc:2938746293 cname:server@conf1.example.com
a=ssrc:2938746293 srcname:t1
a=ssrc:1207102398 cname:server@conf1.example.com
a=ssrc:1207102398 srcname:t2
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:1560
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* recv [x=640,y=360]
a=max-recv-ssrc:96 3
a=mid:3
a=recvonly
m=video 49500 RTP/AVP 96
```

```
b=AS:420
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:96 3
a=mid:4
a=recvonly
m=video 49600 RTP/AVP 96 97 98
b=AS:2600
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:96 send [x=1280,y=720]
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42c01e
a=imageattr:97 send [x=640,y=360]
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42c00d
a=imageattr:98 send [x=320,y=180]
a=max-send-ssrc:96 1
a=max-send-ssrc:97 4
a=max-send-ssrc:98 8
a=max-send-ssrc:* 8
a=ssrc:2981523948 cname:server@conf1.example.com
a=ssrc:2938237 cname:server@conf1.example.com
a=ssrc:1230495879 cname:server@conf1.example.com
a=ssrc:74835983 cname:server@conf1.example.com
a=ssrc:3928594835 cname:server@conf1.example.com
a=ssrc:948753 cname:server@conf1.example.com
a=ssrc:1293456934 cname:server@conf1.example.com
a=ssrc:4134923746 cname:server@conf1.example.com
a=mid:5
a=sendonly
a=content:alt
```

Figure 6: Server Answer for Bob Multi-stream and Simulcast Telepresence Conference

In this SDP answer, the grouping tag is changed to SCR, confirming that the sent simulcast streams will be received. The directionality of the streams themselves as well as the directionality of multi-stream and bandwidth attributes are changed. The number of allowed streams in the content:alt video session has been reduced from 16 to 8 in the answer.

7.3. Fred: Dial-out to Legacy Client

Fred has a simple legacy client that know nothing of the new signaling means discussed in this document. In this example, the

multi-stream and simulcast aware RTP mixer is calling out to Fred. Even though it is never actually sent, this would be Fred's offer SDP, should he have called in. It is included here to improve the reader's understanding of Fred's response to the conference SDP.

```
v=0
o=fred 82342187 237429834 IN IP4 192.0.2.213
s=Legacy Client
t=0 0
c=IN IP4 192.0.2.213
m=audio 50132 RTP/AVP 9 8
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
m=video 50134 RTP/AVP 96 97
b=AS:405
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
a=rtpmap:97 H263-2000/90000
a=fmtp:97 profile=0;level=30
```

Figure 7: Legacy Client Hypothetical Offer

Fred would offer a single mono audio and a single video, each with a couple of different codec alternatives.

The same conference server as in the previous example is calling out to Fred, offering the full set of multi-stream and simulcast features based on what the server itself can support.

```
v=0
o=server 323439283 2384192332 IN IP4 192.0.2.2
s=Multi stream and Simulcast Dial-out Offer
c=IN IP4 192.0.2.43
b=AS:7065
a=group:SCR 2 3 4
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:435
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=max-send-ssrc:* 4
a=max-recv-ssrc:* 3
a=ssrc:3293472833 cname:server@conf1.example.com
a=ssrc:3293472833 srcname:q9
a=ssrc:1734728348 cname:server@conf1.example.com
a=ssrc:1734728348 srcname:Gr
a=ssrc:1054453769 cname:server@conf1.example.com
```



```
a=ssrc:1054453769 srcname:S0
a=ssrc:3923447729 cname:server@conf1.example.com
a=ssrc:3923447729 srcname:AJ
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:4650
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:* send [x=1280,y=720] [x=640,y=360] [x=320,y=180]
             recv [x=1280,y=720]
a=max-recv-ssrc:96 3
a=max-send-ssrc:96 3
a=ssrc:78456398 cname:server@conf1.example.com
a=ssrc:78456398 srcname:bj
a=ssrc:3284726348 cname:server@conf1.example.com
a=ssrc:3284726348 srcname:ON
a=ssrc:2394871293 cname:server@conf1.example.com
a=ssrc:2394871293 srcname:ya
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:1560
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* recv [x=640,y=360]
a=max-recv-ssrc:96 3
a=mid:3
a=recvonly
m=video 49500 RTP/AVP 96
b=AS:420
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:96 3
a=mid:4
a=recvonly
m=video 49600 RTP/AVP 96 97 98
b=AS:2600
a=rtpmap:96 H264/900000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:96 send [x=1280,y=720]
a=rtpmap:97 H264/900000
a=fmtp:97 profile-level-id=42c01e
a=imageattr:97 send [x=640,y=360]
a=rtpmap:98 H264/900000
a=fmtp:98 profile-level-id=42c00d
a=imageattr:98 send [x=320,y=180]
a=max-send-ssrc:96 1
```

```
a=max-send-ssrc:97 4
a=max-send-ssrc:98 8
a=max-send-ssrc:* 8
a=ssrc:2342872394 cname:server@conf1.example.com
a=ssrc:1283741823 cname:server@conf1.example.com
a=ssrc:3294823947 cname:server@conf1.example.com
a=ssrc:1020408838 cname:server@conf1.example.com
a=ssrc:1999343791 cname:server@conf1.example.com
a=ssrc:2934192349 cname:server@conf1.example.com
a=ssrc:2234347728 cname:server@conf1.example.com
a=ssrc:3224283479 cname:server@conf1.example.com
a=mid:5
a=sendonly
a=content:alt
```

Figure 8: Server Dial-out Offer with Multi-stream and Simulcast

The answer from Fred to this offer would look like:

```
v=0
o=fred 9842793823 239482793 IN IP4 192.0.2.213
s=Legacy Client Answer to Server Dial-out
t=0 0
c=IN IP4 192.0.2.213
m=audio 50132 RTP/AVP 9
b=AS:80
a=rtpmap:9 G722/8000
m=video 50134 RTP/AVP 96
b=AS:405
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
m=video 0 RTP/AVP 96
m=video 0 RTP/AVP 96
m=video 0 RTP/AVP 96
```

Figure 9: Legacy Client Answer to Server Dial-out

as can be seen from the hypothetical offer, Fred does not understand any of the multistream or simulcast attributes, and does also not understand the grouping framework. Thus, all those lines are removed from the answer SDP and any surplus video media blocks except for the first are rejected. The media bandwidth are adjusted down to what Fred actually accepts to receive.

7.4. Joe: Dial-out to Desktop Client

This example is almost identical to the one above, with the difference that the answering end-point has some limited simulcast

and multi-stream capability. As above, this is the offer SDP that Joe would have used, should he have called in.

```
v=0
o=joe 82342187 237429834 IN IP4 192.0.2.117
s=Simulcast and Multistream enabled Desktop Client
t=0 0
c=IN IP4 192.0.2.117
b=AS:985
a=group:SCS 2 3
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:145
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=ssrc:1223883729 cname:joe@foo.example.com
a=ssrc:1223883729 srcname:jV
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:520
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:96 send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=ssrc:3842394823 cname:joe@foo.example.com
a=ssrc:3842394823 srcname:BD
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:160
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=ssrc:1214232284 cname:joe@foo.example.com
a=ssrc:1214232284 srcname:BD
a=mid:3
a=sendonly
m=video 49300 RTP/AVP 96
b=AS:320
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:* 2
a=mid:4
a=recvonly
a=content:alt
```

Figure 10: Desktop Client Hypothetical Offer

Joe would send two versions of simulcast, 360p and 180p, from a single camera and can receive three sources of multi-stream, one 360p and two 180p streams.

Again, the same conference server is calling out to Joe and the offer SDP from the server would be almost identical to the one in the previous example. It is therefore not included here. The response from Joe would look like:

```
v=0
o=joe 239482639 4702341992 IN IP4 192.0.2.117
s=Answer from Desktop Client to Server Dial-out
t=0 0
c=IN IP4 192.0.2.117
b=AS:985
a=group:SCS 2 3
m=audio 49200 RTP/AVP 96
b=AS:145
a=rtpmap:96 G719/48000/2
a=ssrc:1223883729 cname:joe@foo.example.com
a=ssrc:1223883729 srcname:iJ
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:520
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:96 send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=ssrc:3842394823 cname:joe@foo.example.com
a=ssrc:3842394823 srcname:YD
a=mid:2
a=content:main
m=video 0 RTP/AVP 96
a=mid:3
m=video 49400 RTP/AVP 96
b=AS:160
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=ssrc:1214232284 cname:joe@foo.example.com
a=ssrc:1214232284 srcname:YD
a=mid:4
a=sendonly
m=video 49300 RTP/AVP 96
b=AS:320
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:* 2
a=mid:5
a=recvonly
a=content:alt
```

Figure 11: Desktop Client Answer to Server Dial-out

Since the RTP mixer supports all of the features that Joe does and more, the SDP does not differ much from what it should have been in an offer. It can be noted that as stated in [[RFC5888](#)], all media

lines need mid attributes, even the rejected ones, which is why mid:3 is present even though the mid quality simulcast version offered by the mixer is rejected by Joe.

8. IANA Considerations

This document requests that two new SDP grouping semantics, SCS and SCR, are registered.

Formal registrations to be written.

9. Security Considerations

The Simulcast grouping semantics are vulnerable to attacks in the signalling.

A false grouping of non-simulcast streams as simulcast would risk that some streams are incorrectly ignored by receivers that know simulcast and that are uninterested in the assumed simulcast streams.

A hostile removal of simulcast grouping will prevent streams from being interpreted as simulcast, which obviously prevents use of the simulcast functionality. It will also risk that intended simulcast streams are instead presented as separate, independent streams to a receiver.

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.

10. Acknowledgements

11. References

11.1. Normative References

[I-D.westerlund-avtcore-max-ssrc]
Westerlund, M., Burman, B., and F. Jansson, "Multiple Synchronization sources (SSRC) in RTP Session Signaling", [draft-westerlund-avtcore-max-ssrc-02](#) (work in progress), July 2012.

[I-D.westerlund-avtext-rtcp-sdes-srcname]
Westerlund, M., Burman, B., and P. Sandgren, "RTCP SDES

Item SRCNAME to Label Individual Sources",
[draft-westerlund-avtext-rtcp-sdes-srcname-01](#) (work in progress), July 2012.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", [RFC 5576](#), June 2009.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", [RFC 5888](#), June 2010.

[11.2.](#) Informative References

- [I-D.westerlund-avtcore-multiplex-architecture]
Westerlund, M., Burman, B., Perkins, C., and H. Alvestrand, "Guidelines for using the Multiplexing Features of RTP",
[draft-westerlund-avtcore-multiplex-architecture-02](#) (work in progress), July 2012.
- [I-D.westerlund-avtcore-transport-multiplexing]
Westerlund, M. and C. Perkins, "Multiple RTP Sessions on a Single Lower-Layer Transport",
[draft-westerlund-avtcore-transport-multiplexing-03](#) (work in progress), July 2012.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [RFC3569] Bhattacharyya, S., "An Overview of Source-Specific Multicast (SSM)", [RFC 3569](#), July 2003.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", [RFC 4588](#), July 2006.
- [RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description

Protocol (SDP) Content Attribute", [RFC 4796](#),
February 2007.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", [RFC 5117](#),
January 2008.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis,
"RTP Payload Format for Scalable Video Coding", [RFC 6190](#),
May 2011.

Authors' Addresses

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

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

Bo Burman
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 13 11
Email: bo.burman@ericsson.com

Morgan Lindqvist
Ericsson
Farogatan 6
Kista, SE-164 80
Sweden

Phone: +46 10 719 00 00
Fax:
Email: morgan.lindqvist@ericsson.com
URI:

Fredrik Jansson
Ericsson
Farogatan 6
Kista, SE-164 80
Sweden

Phone: +46 10 719 00 00
Fax:
Email: fredrik.k.jansson@ericsson.com
URI: