Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 6, 2012

M. Westerlund
B. Burman
Ericsson
February 3, 2012

Multi-Stream Media Conferencing
draft-westerlund-clue-multistream-conference-00

Abstract

This memo describes a multimedia multi-party conferencing architecture based on use of multiple Real-Time Transport Protocol (RTP) streams.

Table of Contents

1. Introduction

Multimedia multi-party conferencing is being improved through the use of multiple media streams per media type. At the same time, the number of different types of end-points that are capable of participating in a multimedia conference increases, and so does the number of access types that end-points may use to connect. This memo describes a number of use cases relevant to that scenario. The use cases aims to use as high media quality as possible, while making as efficient use of available resources as possible and accommodating a high degree of end-point and network diversity.

2. Requirements

The use cases in this memo should handle diversity in end-point capability such as media quality, processing power, and network interface. The perceived end-user media quality is in turn impacted by media stream bitrate and also number of streams per media type, as well as end-to-end delay, which should also be taken into account. Diversity in network capacity, network quality, and user preferences based on location, device, etc, should also be handled. These properties should be expected to change between conference sessions and even within the same session.

Different types of conference Topologies must be supported, including centralized, multicast, and point-to-point. It should also be possible to use cascaded, centralized conferences as well as combinations of unicast and multicast. The relevant RTP base topologies are described in [RFC5117].

Use cases including an RTP Mixer should avoid adding delay or reducing quality by forwarding streams as unmodified as possible, with reduced processing requirements in the RTP Mixer as added advantage.

The conference use cases should strive for continuous presence, presenting media from as many participants as reasonably possible. At the same time, it should strive to use as high quality media as possible from each participant. The bandwidth used for the conference should at the same time be as low as possible. Those three requirements are in conflict and there is no generally accepted, optimum trade-off.

The number of participants in the conference shall assumed to be unlimited, and it may thus not be possible to create a continuous presence experience with media from all participants being presented to all other participants, and only media from a limited sub-set of

participants may be presented to each individual participant. This is seen as the hardest conferencing use case and most other use cases should also be covered by that use case. How many other participants' media that are presented simultaneously on a certain end-point is allowed to vary and may depend on a number of preconditions.

3. Use Cases

The sub-sections below describe multi-stream conferencing use cases relevant to each RTP base topology. Each use case is constructed as to cover as many requirements as possible and must consequently include a number of different end-point types.

In the figures below, the involved end-points are for clarity drawn as only sending or only receiving, but may in a real scenario of course have the possibility to both send and receive.

The common figure legends for the end-point capability categories used in all use cases are:

D: Dual channel, high quality, for example conference room client. Streams in figures are denoted by '||' or '='. The term 'channel' is chosen as a generic term and may apply to any media. A channel is related to a single media source in the sender and a media sink in the receiver. For video media, the channel source is typically a camera and the sink a screen or window. For audio, the channel source is typically a microphone and the sink a loudspeaker. A dual channel sending end-point thus has two independent media sources that can be sent simultaneously to a receiving end-point. Nothing prevents those dual channels from using other qualities than 'high', but that is chosen as an example in this memo to simplify the description.

H: High quality, single channel, for example meeting room client. Streams in figures are denoted by '|' or '-'.

M: Medium quality, single channel, for example desktop client. Streams in figures are denoted by ':' or '..'.

L: Low quality, single channel, for example mobile client. Streams in figures are denoted by '''.

It is of course possible to divide end-points into more categories, but the chosen ones should make it possible to highlight most of the relevant topics.

Note also that the use cases are kept as separated and clean-cut as possible to simplify the description. Most use cases, especially the RTP Mixer ones (Section 3.2), could be combined into larger, more complex scenarios.

3.1. Point to Point

The point to point case is close to trivial, since it is assumed that capability exchange during setup will be able to negotiate the best quality between the two end-points. When the number of channels differ between sender and receiver, the channel aspects of Section 3.2.5 apply.

3.2. RTP Mixer

Each link between the RTP Mixer and an end-point is in principle a Point to Point connection (Section 3.1) as described above. One major difference from a Point to Point Connection is that the RTP Mixer should represent and act according to some combination of the wishes and needs of multiple end-points on the other side of the Mixer. That may include handling of conflicting or partly conflicting requirements, and the way to resolve those is not generally defined but will typically depend on RTP Mixer design, configuration, applied policies, or some combination.

3.2.1. Incompatible Codecs

This use case is the only feasible one when end-points have incompatible codecs, and transcoding is in that case always necessary and cannot be avoided. The incompatibility is not necessarily only due to different codec types, but may also be caused by limited codec capacity, or limitations in media stream transport between the end-points. This use case is trivial in the sense that the Mixer is assumed to always be capable of creating a dedicated media mix to each receiving end-point. It may be used as an overall strategy, or as part of other use cases.

A special case of transcoding is when the sender is configured to use scalable encoding and receivers do not support scalability. Transcoding of scalable streams to non-scalable streams is often a less complex operation than transcoding in general.

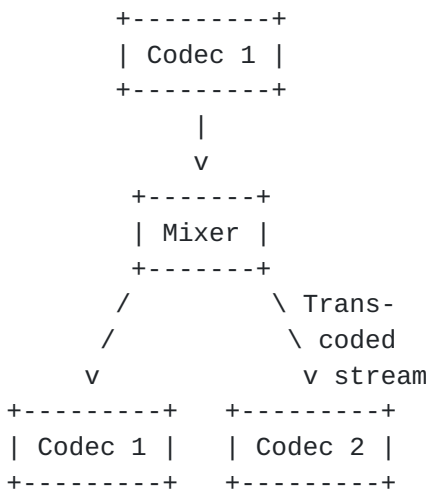

Figure 1: Transcoding Incompatible Codecs

3.2.2. Low Quality End-Point

A low quality end-point has per definition the lowest media quality in the conference, and as a sender it is assumed that all other end-points can receive and present the media without restrictions. Some of the more capable end-points will have to choose how to present the received media in the best way, but it can always be presented.

As a receiver, a low quality end-point is only capable of receiving streams from other low quality end-points (without transcoding).

It is conceivable that a receiver of a certain quality category (not only low quality) can receive higher quality streams and reduce the quality locally such that it is feasible for presentation, but that will in general waste both bandwidth and processing resources.

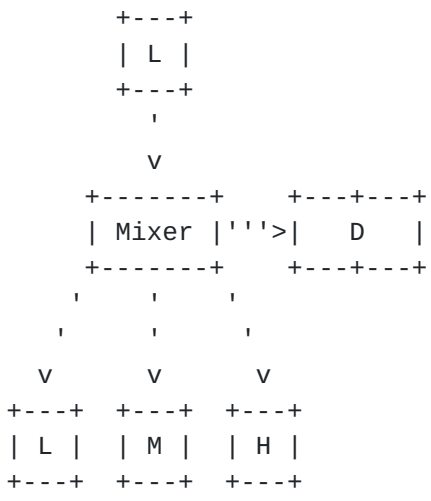

Figure 2: Low Quality Stream

3.2.3. Medium Quality End-Point

Similar to above, the medium quality sender media stream is assumed to be possible to receive without restrictions in all but the low quality end-point. For the medium quality media stream to reach also the low quality end-point, there are three options that are described in the sub-sections below.

When receiving, a medium quality end-point is capable of receiving other medium quality streams, as well as low quality streams (without transcoding).

3.2.3.1. Transcoding

Transcode the stream when sent towards low quality receivers, as described above (Section 3.2.1). This could sometimes be feasible quality-wise, especially if the quality difference between the medium quality and the low quality streams are large, making the reduced medium quality stream be relatively close quality-wise to un-encoded low quality media. End-to-end delay will however always suffer.

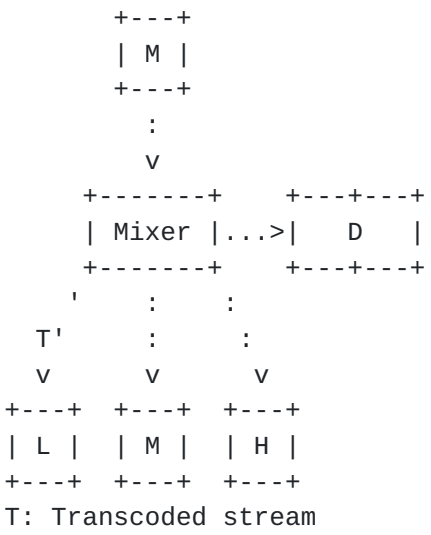

Figure 3: Medium Quality Stream Transcoding

3.2.3.2. Simulcast

Encode both a medium quality and a low quality stream from the same un-encoded source data and simulcast them. The RTP Mixer can, without having to transcode, forward the low quality stream towards the low quality end-points, and forward the medium quality stream towards all other end-points.

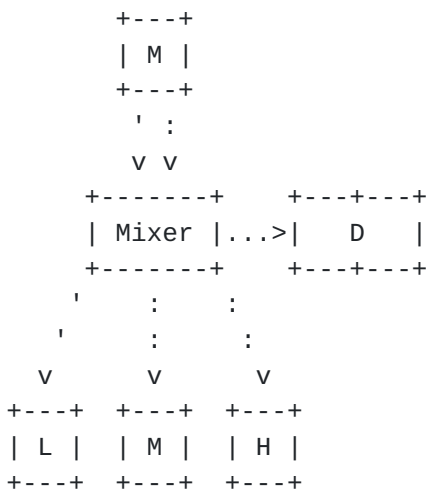

Figure 4: Medium Quality Stream Simulcast

3.2.3.3. Scalable Coding

As a variant of simulcast, if it is possible to use a scalable codec, create a scalable stream with one low quality sub-stream and one sub-stream that together with the low quality sub-stream can reconstruct a medium quality stream. Similar but not identical to the above, the RTP Mixer can, without having to transcode, forward the low quality sub-stream towards the low quality end-points, and forward both the low quality and the medium quality sub-streams (jointly describing a medium quality stream) to all other end-points.

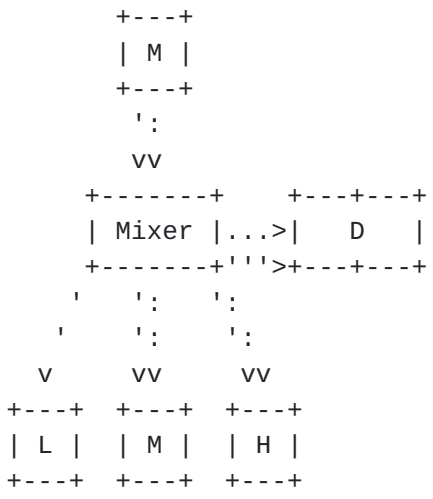

Figure 5: Medium Quality Scalable Stream

3.2.4. Single Channel High Quality End-Point

This use case is very similar to the medium quality end-point case (Section 3.2.3) above. The difference is that there are now two different (sample) categories of end-points that cannot receive the high quality stream instead of one category. Simulcast and scalable streams must thus be extended to three versions or three sub-streams, respectively.

As a receiver, all streams from other end-points can be received. The only exception is when multiple streams from a single end-point are used, such as from D.

3.2.4.1. Transcoding

Similar to Medium Quality (Section 3.2.3.1), just that the transcoding needs to produce two different qualities instead of one.

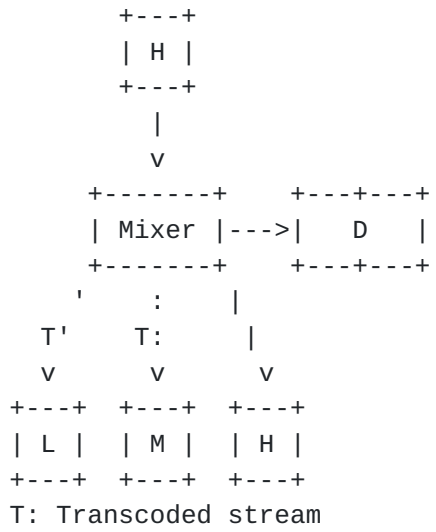

Figure 6: High Quality Stream Transcoding

3.2.4.2. Simulcast

Similar to Medium Quality (Section 3.2.3.2), just that three instead of two simulcast streams need to be sent.

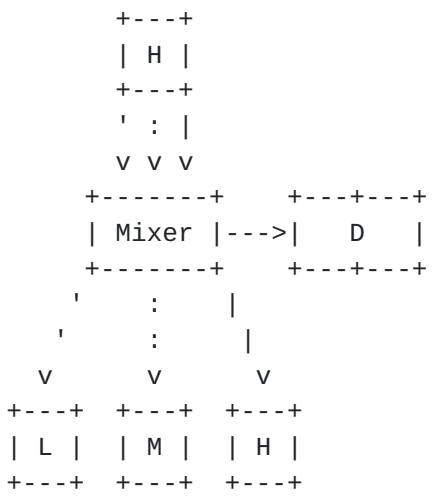

Figure 7: High Quality Stream Simulcast

3.2.4.3. Scalable Coding

Similar to Medium Quality (Section 3.2.3.3), just that three instead of two scalable layers are used.

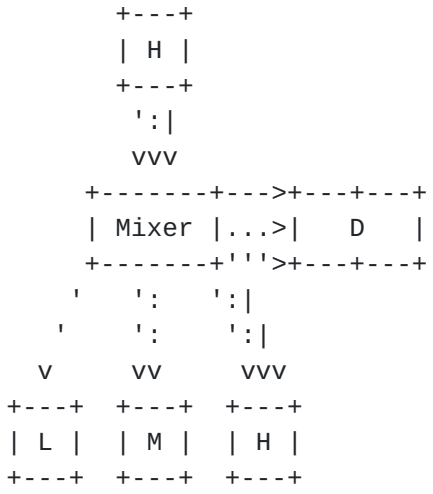

Figure 8: High Quality Scalable Stream

3.2.5. Dual Channel High Quality End-Point

Again, this use case is very similar to the one above (Section 3.2.4). The major difference is that this end-point is capable of sending and receiving dual, high quality streams where each stream has to be treated in a similar way to the previous section.

When using multiple inter-related media, such as video with

corresponding audio, those media streams need not only be synchronized time-wise, just as for single channel end-points, but their spatial relation need also be established. For example, a left camera with an attached microphone and a right camera with an attached microphone. In general it is likely always desirable to be able to relate streams from a multi-channel end-point in a defined way, representing related sub-parts of a larger scene, both intra-media and inter-media. Description and signaling of stream relations is a complex problem in itself, which is currently work in progress in CLUE WG [I-D.ietf-clue-framework] and will not be elaborated further in this memo.

Another major, additional, aspect to account for is that the RTP Mixer needs to choose how to map dual (or multiple) streams onto a single stream, when forwarding towards end-points that has fewer receive channels than the sender. This problem is similar to choosing a limited set of participants from a potentially unlimited set, which is described below (Section 3.2.6).

A dual-channel (or multi-channel) receiving end-point that is receiving fewer simultaneous channel streams from a sending end-point than the maximum possible this end-point can handle, will have to decide which one(s) of the available receive channels should be used for each received stream. This decision can also be made by the RTP Mixer, if it knows the concept of multi-channel clients, has information about how many simultaneous channels the individual receiver supports, and knows how those channels should be related.

There may of course be end-points that have capability for more than two simultaneous channels. It is also possible to envision end-points where the number of receive channels differ from the number of send channels.

3.2.6. Mixer Source and Sink Selection

When a Mixer cannot forward all available streams to each client, it has to choose a small set out of a potentially very large set of streams in the conference. Multiple strategies are possible for that choice. The Mixer may use different strategies towards different receivers, depending on for example their capabilities or preferences.

Another dimension of selection exist when the conference contains end-points with different number of channels (multiple media streams of the same media type). On the Mixer receive side, it may be necessary to select a few streams from a multi-stream media. On the Mixer send side, it may be necessary to select to which channel in a multi-stream capable receiver a certain stream should be sent.

The choice of source may be either algorithmic (pre-configured) or manual (user controlled from one or more end-points). Speech activity is a commonly used algorithmic measure to choose which participants' media streams to forward, but it is not the only conceivable measure. Which and how many streams to select can either be based on some algorithm included in the RTP Mixer (for example the N most active speakers), or it can be controlled by the conference owner, or even by the receiving users individually.

The figure below depicts the case where a receiving end-point explicitly selects streams through signaling (*) to the Mixer. Both the media senders and their streams are numbered for clarity, and the conceptual signaling message is contained in {...}.

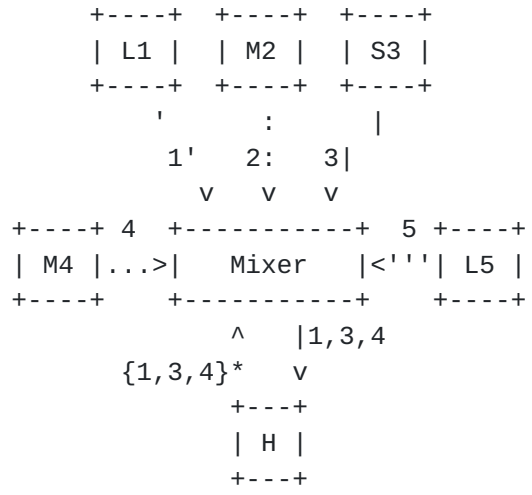

Figure 9: Receiver Stream Selection

To be able to make an informed choice on what streams to select, the user at the receiving end-point will need information about which conference participant correspond to which stream, and possibly also other meta-information about the streams and sending end-points.

3.2.7. Media Composition

When it is desirable that the RTP Mixer selects (Section 3.2.6) and forwards a larger number of simultaneous streams than what the receiving end-point can support, the Mixer has the option to make a composition of multiple streams onto fewer streams, possibly to only a single stream. Which and how many streams to compose is typically based on the selection, as described in the previous section above.

The composition operation is basically independent from selection. In general, Mixer composition requires transcoding. A Mixer composition use case example is depicted below. To simplify the

picture, only a single receiver is included. Also, both the media senders and their streams are numbered for clarity. In the below example, the Mixer has chosen to compose stream 1, 3 and 4 into the single stream sent to the receiving end-point.

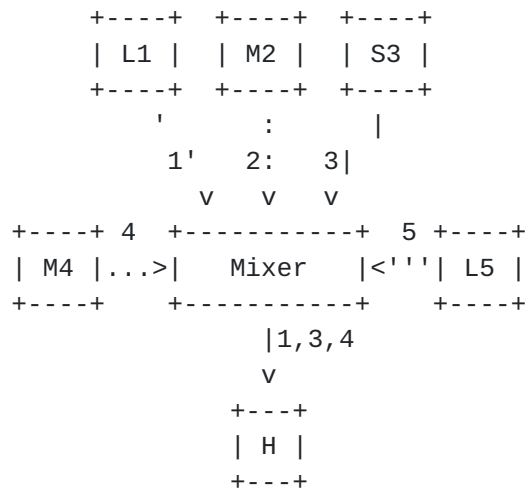

Figure 10: Mixer Composition

An alternative to make composition in the Mixer is to let the end-point do local composition by sending it multiple, un-composed, streams. This could avoid transcoding at the cost of sending multiple streams, which is depicted below, where stream 2, 3 and 5 are sent to the receiving client for local composition, as an example.

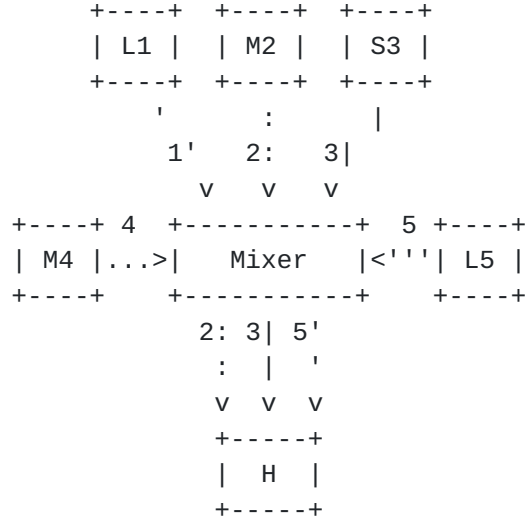

Figure 11: Local Composition

When the senders offer streams of multiple qualities, either the

mixer or the local composition can select and combine media of different qualities. Use of multiple qualities could help optimizing resource utilization for transport, decoding and rendering. In the figure below, one high quality (3), one medium quality (4) and two low qualities (1 and 2) are selected for local composition, as an example.

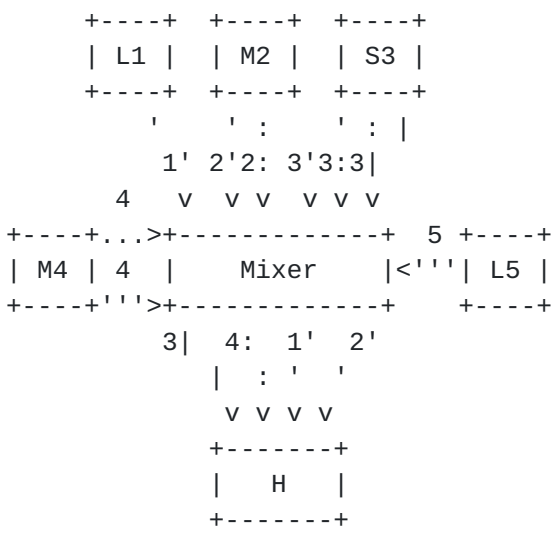

Figure 12: Multi Quality Local Composition

This local composition scenario can be further enhanced by the Mixer providing different quality streams to the receiver, based on the Mixer selection algorithm. One example could be to let the Mixer forward the stream from the most active speaker as a high quality stream, and forward the less active speakers as lower quality streams. To use this in the local composition, the receiving end-point must know the streams' different roles, which requires a stream role agreement between the Mixer and the receiving end-point. In the figure above, this can be achieved by tagging for example the leftmost stream from the mixer as having the "most active" role. The role agreement can be made through signaling between Mixer and receiving end-point.

When the receiving end-point supports reception and presentation of several channels (for example has several screens), it is possible to combine Mixer composition with local composition of multiple un-composed streams by sending one or more composed streams and one or more un-composed streams from the Mixer.

3.3. Multicast

In the multicast or multi-unicast case, each media stream from a single sender will reach multiple receivers unmodified. This can be

achieved by multicast addressing, or by multi-unicast and RTP Translators [RFC5117].

This use case is similar to when an RTP Mixer is neither performing composition (Section 3.2.7) nor source selection (Section 3.2.6), but is forwarding all streams (and qualities, if more than one) to all receivers. The entire load of stream and quality selection for presentation is put on the receiving end-point.

In the figure below, multi-unicast through the use of an RTP Translator is depicted since the figure becomes clearer than with a full mesh multicast. The figure illustrates non-scalable streams, but it is of course also possible to multicast scalable streams.

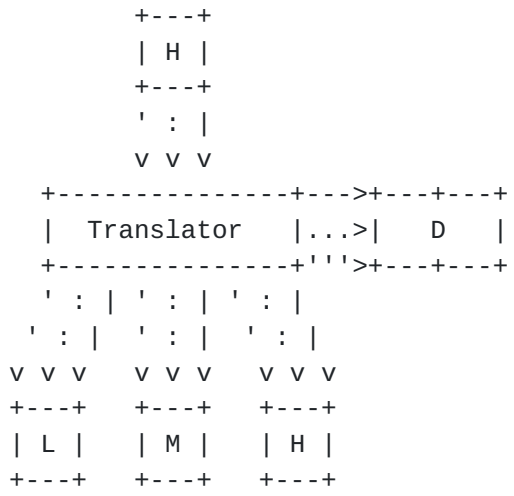

Figure 13: Multi-Unicast of Multiple Qualities

4. RTP Usage

This section discusses how RTP transport [RFC3550] could be used with the scenarios discussed in the previous section. It complements, extends and partially presents an alternative solution to what is described in [I-D.lennox-clue-rtp-usage].

4.1. Use of SSRC and CSRC

It is assumed that each RTP media stream in the use cases in the previous section is identified by an SSRC. There already exist methods to convey information about each media stream and sending end-point in a conference [RFC4575]. Those methods also provide means to correlate that information with the stream SSRC. This information could be sufficient for a receiving end-point to make informed media stream selection decisions (Section 3.2.6).

When the RTP Mixer associates a role to a stream (Section 3.2.7), for example "most active speaker" or "leftmost channel", it is possible to associate that additional property to an SSRC belonging to the Mixer, while also keeping the original SSRC in the RTP packet as CSRC. This way, it is possible to apply special treatment to received streams based on their SSRC without losing the ability to identify the original source, using existing RTP functionality.

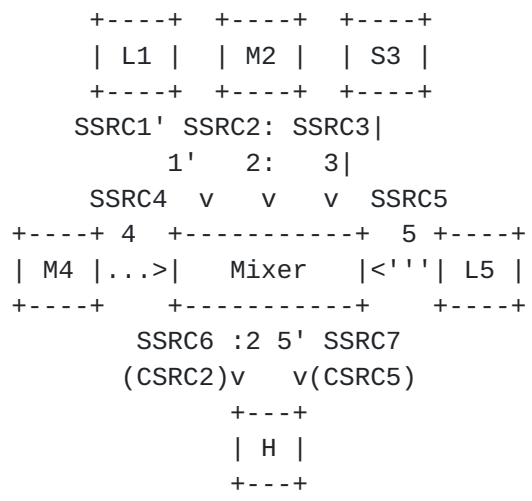

Figure 14: SSRC and CSRC from Mixer

It is also possible in a receiving end-point to let each role (and Mixer SSRC) map towards a specific media decoder, since that Mixer SSRC would rarely (if ever) change during a session other than due to SSRC conflicts, while the CSRC would typically change every time a new stream is selected for that specific role, for example "active speaker".

Note that when simulcast is used, different simulcast versions can typically use different SSRC. When scalable coding is used, different layers can sometimes be sent within a single SSRC using a single Payload Type and thus cannot be distinguished on RTP level. Identification of different layers will then have to be codec specific. Some scalable codecs can also send different layers on separate SSRC or using separate Payload Types.

4.2. Signaling Extensions

End-points and Mixers supporting multiple channels (Section 3.2.5) need to know how many simultaneous channels that can be accepted in a receiver and will be used from a sender. Assuming that each channel is sent as a single SSRC, there should be signaling that limits the number of SSRC in an RTP session [I-D.westerlund-avtcore-max-ssrc]. This maps well with the above suggested relation between SSRC and

media decoders, since the suggested limitation then also expresses the maximum simultaneously available decoding resources.

When representing a specific media source in several different qualities and when using simulcast to transport (Section 3.2.3.2) them rather than as scalable layers contained in a single stream, those separate streams need to be signaled as simulcast versions [I-D.westerlund-avtcore-rtp-simulcast], in order for the receiver to be able to apply correct selection logic (Section 3.2.6).

When a conference is configured to let individual users at receiving end-points choose which streams to receive (Section 3.2.6), responsive selection signaling between end-point and RTP Mixer [I-D.westerlund-dispatch-stream-selection] is needed to initiate the stream selection. This selection is applicable to the media streams included in a compositions also.

Note that when simulcast is used, different simulcast versions can typically use different SSRC. When scalable coding is used, different layers can sometimes be sent within a single SSRC using a single Payload Type and thus cannot use the parts of SDP signaling that relies on those identifiers. Identification of different layers will then have to be codec specific. Some scalable codecs can also send different layers on separate SSRC or using separate Payload Types.

4.3. Optimizations

When it is desirable to minimize the number of UDP ports used by an end-point, for example to reduce the resources for NAT and firewall traversal, it should be possible to send all media streams from all RTP sessions on a single UDP port [I-D.westerlund-avtcore-transport-multiplexing]. This should preferably be done without losing any important RTP functionality. Transport resource priority and Quality of Service handling are typically performed based on 5-tuple (source and destination addresses and ports, and protocol), which together with desired differentiation of media stream priority can require use of more than one UDP port (5-tuple).

In a conference use case with multiple sending end-points and where the receiving end-points do not make use of all available streams, there is a risk that some of the sent streams are not used by any receiver. The probability for this increases when end-points provide multiple streams of different qualities. The need for a certain stream can change very quickly, for example when the need is based on conditions of other streams such as speech activity. To save bandwidth and processing resources in the sending end-point, it would

thus be desirable for an RTP Mixer to be able to quickly turn off or pause individual streams [I-D.westerlund-avtext-rtp-stream-pause] that are no longer used in any media mix sent to receiving end-points, and even more importantly be able to quickly resume needed streams when they are needed again.

In use cases where multiple media streams (Section 3.2.5) are used in a single RTP session, when SDP is used as signaling protocol, and specifically when the number of streams depends on the SDP negotiation outcome (Section 4.2), the currently defined bandwidth signaling attribute is only capable of describing the maximum possible bandwidth usage for the most demanding alternative. It would be desirable to express bandwidth requirements in a more precise way [I-D.westerlund-mmusic-sdp-bw-attribute].

While any RTP stream relations such as for example spatial co-location of related audio and video streams should be possible to express in session signaling or other application signaling protocol, there may be times when it is desirable that RTP stream SSRC relations [I-D.westerlund-avtext-rtcp-sdes-srcname] such as simulcast alternatives or related FEC streams can be seen directly in the RTP or RTCP streams. This would allow for processing of media and related streams in middle boxes, without the need to have access to all higher layer signaling. Keeping protocol layer separation will enable some architectural freedom and may ease future extensions.

5. Security Considerations

Any security considerations relevant to this memo are described in the RFCs and drafts referenced in the RTP Usage section (Section 4).

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Acknowledgements

8. Informative References

[I-D.ietf-clue-framework]
Romanow, A., Duckworth, M., Pepperell, A., and B. Baldino,

"Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-02 (work in progress), January 2012.

[I-D.lennox-clue-rtp-usage]
Lennox, J., Romanow, A., and P. Witty, "Real-Time Transport Protocol (RTP) Usage for Telepresence Sessions", draft-lennox-clue-rtp-usage-01 (work in progress), October 2011.

[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-00 (work in progress), October 2011.

[I-D.westerlund-avtcore-rtp-simulcast]
Westerlund, M., Burman, B., Lindqvist, M., and F. Jansson, "Using Simulcast in RTP sessions", draft-westerlund-avtcore-rtp-simulcast-00 (work in progress), October 2011.

[I-D.westerlund-avtcore-transport-multiplexing]
Westerlund, M. and C. Perkins, "Multiple RTP Session on a Single Lower-Layer Transport", draft-westerlund-avtcore-transport-multiplexing-01 (work in progress), October 2011.

[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-00 (work in progress), October 2011.

[I-D.westerlund-avtext-rtp-stream-pause]
Akram, A., Burman, B., Grondal, D., and M. Westerlund, "RTP Media Stream Pause and Resume", draft-westerlund-avtext-rtp-stream-pause-00 (work in progress), October 2011.

[I-D.westerlund-dispatch-stream-selection]
Grondal, D., Burman, B., and M. Westerlund, "Media Stream Selection (MESS)", draft-westerlund-dispatch-stream-selection-00 (work in progress), October 2011.

[I-D.westerlund-mmusic-sdp-bw-attribute]
Frankkila, T., Westerlund, M., and B. Burman, "Extensible

Bandwidth Attribute for SDP", draft-westerlund-mmusic-sdp-bw-attribute-00 (work in progress), October 2011.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

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

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