CLUE WG
Internet-Draft
Intended status: Standards Track
Expires: March 16, 2013

R. Even
Huawei Technologies
J. Lennox
Vidyo

September 12, 2012

Mapping RTP streams to CLUE media captures
draft-even-clue-rtp-mapping-04.txt

Abstract

This document describes mechanisms and recommended practice for mapping RTP media streams defined in SDP to CLUE media captures.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Telepresence systems can send and receive multiple media streams. The CLUE framework [I-D.ietf-clue-framework] defines media captures as a source of Media, such as from one or more Capture Devices. A Media Capture (MC) may be the source of one or more Media streams. A Media Capture may also be constructed from other Media streams. A middle box can express Media Captures that it constructs from Media streams it receives.

SIP offer answer [RFC3264] uses SDP [RFC4566] to describe the RTP[RFC3550] media streams. Each RTP stream has a payload type number and SSRC. The content of the RTP stream is created by the encoder in the endpoint. This may be an original content from a camera or a content created by an intermediary device like an MCU.

This document makes recommendations, for this telepresence architecture, about how RTP and RTCP streams should be encoded and transmitted, and how their relation to CLUE Media Captures should be communicated. The proposed solution supports multiple RTP topologies

2. Terminology

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 RFC2119[RFC2119] and indicate requirement levels for compliant RTP implementations.

3. RTP topologies for CLUE

The typical RTP topologies used by Telepresence systems specify different behaviors for RTP and RTCP distribution. The relevant topologies include point-to-point, as well as media mixers, media-switching mixers, and source-projection mixers.

In the point-to-point topology, one peer communicates directly with a single peer over unicast. There can be one or more RTP sessions, and each RTP session can carry multiple RTP streams identified by their SSRC. All SSRCs will be recognized by the peers based on the information in the RTCP SDES report that will include the CNAME and SSRC of the sent RTP streams. There are different point to point use cases as specified in CLUE use case [I-D.ietf-clue-telepresence-use-cases]. There may be a difference between the symmetric and asymmetric use cases. While in the symmetric use case the typical mapping will be from a Media capture device to a render device (e.g. camera to monitor) in the asymmetric

case the render device may receive different capture information (RTP stream from a different camera) if it has fewer rendering devices (monitors). In some cases, a CLUE session which, at a high-level, is point-to-point may nonetheless have RTP which is best described by one of the mixer topologies below. For example, a CLUE endpoint can produce composited or switched captures for use by a receiving system with fewer displays than the sender has cameras.

In the Media Mixer topology, the peers communicate only with the mixer. The mixer provides mixed or composited media streams, using its own SSRC for the sent streams. There are two cases here. In the first case the mixer may have separate RTP sessions with each peer (similar to the point to point topology) terminating the RTCP sessions on the mixer; this is known as Topo-RTCP-Terminating MCU in [RFC5117]. In the second case, the mixer can use a conference-wide RTP session similar to RFC 5117's Topo-mixer or Topo-Video-switching. The major difference is that for the second case, the mixer uses conference-wide RTP sessions, and distributes the RTCP reports to all the RTP session participants, enabling them to learn all the CNAMEs and SSRCs of the participants and know the contributing source or sources (CSRCs) of the original streams from the RTP header. In the first case, the Mixer terminates the RTCP and the participants cannot know all the available sources based on the RTCP information. The conference roster information including conference participants, endpoints, media and media-id (SSRC) can be available using the conference event package [RFC4575] element.

In the Media-Switching Mixer topology, the peer to mixer communication is unicast with mixer RTCP feedback. It is conceptually similar to a compositing mixer as described in the previous paragraph, except that rather than compositing or mixing multiple sources, the mixer provides one or more conceptual sources selecting one source at a time from the original sources. The Mixer creates a conference-wide RTP session by sharing remote SSRC values as CSRCs to all conference participants.

In the Source-Projection Mixer topology, the peer to mixer communication is unicast with RTCP mixer feedback. Every potential sender in the conference has a source which is "projected" by the mixer into every other session in the conference; thus, every original source is maintained with an independent RTP identity to every receiver, maintaining separate decoding state and its original RTCP SDES information. However, RTCP is terminated at the mixer, which might also perform reliability, repair, rate adaptation, or transcoding on the stream. Senders' SSRCs may be renumbered by the mixer. The sender may turn the projected sources on and off at any time, depending on which sources it thinks are most relevant for the receiver; this is the primary reason why this topology must act as an

RTP mixer rather than as a translator, as otherwise these disabled sources would appear to have enormous packet loss. Source switching is accomplished through this process of enabling and disabling projected sources, with the higher-level semantic assignment of reason for the RTP streams assigned externally.

The above topologies demonstrate two major RTP/RTCP behaviors:

1. The mixer may either use the source SSRC when forwarding RTP packets, or use its own created SSRC. Still the mixer will distribute all RTCP information to all participants creating conference-wide RTP session/s. This allows the participants to learn the available RTP sources in each RTP session. The original source information will be the SSRC or in the CSRC depending on the topology. The point to point case behaves like this.

2. The mixer terminates the RTCP from the source, creating separate RTP sessions with the peers. In this case the participants will not receive the source SSRC in the CSRC. Since this is usually a mixer topology, the source information is available from the SIP conference event package [RFC4575]. Subscribing to the conference event package allows each participant to know the SSRCs of all sources in the conference.

4. Mapping CLUE Media Captures to RTP streams

The different topologies described in Section 3 support different SSRC distribution models and RTP stream multiplexing points.

Most video conferencing systems today can separate multiple RTP sources by placing them into separate RTP sessions using, the SDP description. For example, main and slides video sources are separated into separate RTP sessions based on the content attribute [RFC4796]. This solution works straightforwardly if the multiplexing point is at the UDP transport level, where each RTP stream uses a separate RTP session. This will also be true for mapping the RTP streams to Media Captures if each media capture uses a separate RTP session, and the consumer can identify it based on the receiving RTP port. In this case, SDP only needs to label the RTP session with an identifier that identifies the media capture in the CLUE description. In this case, it does not change the mapping even if the RTP session is switched using same or different SSRC. (The multiplexing is not at the SSRC level).

Even though Session multiplexing is supported by CLUE, for scaling reasons, CLUE recommends using SSRC multiplexing in a single or

multiple sessions. So we need to look at how to map RTP streams to Media Captures when SSRC multiplexing is used.

When looking at SSRC multiplexing we can see that in various topologies, the SSRC behavior may be different:

1. The SSRCs are static (assigned by the MCU/Mixer), and there is an SSRC for each media capture encoding defined in the CLUE protocol. Source information may be conveyed using CSRC, or, in the case of topo-RTCP-Terminating MCU, is not conveyed.

2. The SSRCs are dynamic, representing the original source and are relayed by the Mixer/MCU to the participants.

In the above two cases the MCU/Mixer creates its own advertisement, with a virtual room capture scene.

Another case we can envision is that the MCU / Mixer relays all the capture scenes from all advertisements to all consumers. This means that the advertisement will include multiple capture scenes, each representing a separate TP room with its own coordinate system. A general tools for distributing roster information is by using an event package, for example by extending the conference event package.

4.1. Review of current directions in MMUSIC, AVText and AVTcore

Editor's note: This section provides an overview of the RFCs and drafts that can be used a base for a mapping solution. This section is for information only, and if the WG thinks that it is the right direction, the authors will bring the required work to the relevant WGs.

The solution needs to also support the simulcast case where more than one RTP session may be advertised for a Media Capture.

When looking at the available tools based on current work in MMUSIC, AVTcore and AVText for supporting SSRC multiplexing the following documents are considered to be relevant.

SDP Source attribute [RFC5576] mechanisms to describe specific attributes of RTP sources based on their SSRC.

Negotiation of generic image attributes in SDP [RFC6236] provides the means to negotiate the image size. The image attribute can be used to offer different image parameters like size but in order to offer multiple RTP streams with different resolutions it does it using separate RTP session for each image option.

[I-D.westerlund-avtcore-max-ssrc] proposes a signaling solution for how to use multiple SSRCs within one RTP session.

A proposed solution to support simulcast is defined in [I-D.westerlund-avtcore-rtp-simulcast]. Simulcast is an application usage where multiple media streams derived from the same media source may be sent simultaneously. The document discusses the best way of accomplishing this in RTP using a session-based solution. The document describes a solution where each stream from the unicast stream will use a separate RTP session. Section 4.2 of the document looks at using a single RTP session using RFC5576 [RFC5576] and the proposed source name attribute specified in [I-D.westerlund-avtext-rtcp-sdes-srcname]. Another way for a single seesion support may be by using a different payload type numbers but section 4.1 of [I-D.westerlund-avtcore-rtp-simulcast] discourages such usage.

[I-D.westerlund-avtext-rtcp-sdes-srcname] provides an extension that may be send in SDP, as an RTCP SDES information or as an RTP header extension that uniquely identifies a single media source. It defines an hierarchical order of the SRCNAME parameter that can be used to for example to describe multiple resolution from the same source (see section 5.1 of [I-D.westerlund-avtcore-rtp-simulcast]). Still all the examples are using RTP session multiplexing.

Other documents reviewed by the authors but are currently not used in a proposed solution include:

[I-D.lennox-mmusic-sdp-source-selection] specifies how participants in a multimedia session can request a specific source from a remote party.

[I-D.westerlund-avtext-codec-operation-point](expired) extends the codec control messages by specifying messages that let participants communicate a set of codec configuration parameters.

Using the above documents it is possible to negotiate the max number of received and sent RTP streams inside an RTP session (m-line or bundled m-line). This allows also offering allowed combinations of codec configurations using different payload type numbers

Examples: max-recv-ssrc:{96:2 & 97:3) where 96 and 96 are different payload type numbers. Or max-send-ssrc{*:4}.

In the next sections, the document will propose mechanisms to map the RTP streams to media captures addressing the simulcast case.

4.2. Static Mapping

Static mapping is widely used in current MCU implementations. It is also common for a point to point symmetric use case when both endpoints have the same capabilities. For capture encodings with static SSRCs, it is most straightforward to indicate this mapping outside the media stream, in the CLUE or SDP signaling. An SDP source attribute [RFC5576] could be defined to associate CLUE capture IDs with SSRCs in SDP. Each SSRC will have a captureID value that will be specified also in the CLUE media capture as an attribute. The provider advertisement could, if it wished, use the same SSRC for media capture encodings that are mutually exclusive. (This would be natural, for example, if two advertised captures are implemented as different configurations of the same physical camera, zoomed in or out.). Section 6 provide an example of an SDP offer and CLUE advertisement.

For the simulcast case the major issue is to support the multiplexing of streams form the same source with different image attribute like image size. The description of the different resolutions is based on RFC6236 [RFC6236] imageattrib. Each RTP stream will have a different SSRC and in order to map an SSRC to a specific resolution the proposal is to use the srcname attribute [I-D.westerlund-avtext-rtcp-sdes-srcname] in SDP and define a srcname-node imgattrX where X is a number that reflects the order in the imageattr attribute so that imgattr1 is the first and imgattr2 is the second and so on.

For example in a=imageattr:98 send [x=1280,y=720] [x=640, y=360]

a=ssrc:11111 srcname:v1.imgattr1 is 720p.

This leads to another proposal for mapping by using the scrname as the RTP stream identifier for simulcast and non-simulcast use cases. See example in section 6

4.3. Dynamic mapping

Dynamic mapping using RTP header extension is described in draft-lennox-clue-rtp-usage [I-D.lennox-clue-rtp-usage] section 10.2. The value in the RTP header extension can be the parameter we chose to use in the static case (CaptureID or srcname).

When looking at the dynamic mode in the simulcast case it looks like in the MCU case, the MCU will create a common mode. For example if there are three participants in the conference and A can send highres and lowres simultenously while B can only send highres, the MCU will offer just the highres since he cannot provide the lowres from B

without transcoding. This becomes more complicated when the offered resolutions do not match. The issue above assumes that the MCU will need to offer an SDP that will fit the negotiated mode even if we negotiate the resolution in the CLUE protocol and not using the imageattr.

Note: in the dynamic case there is a need to verify how it will work if there is not all RTP streams of the same media type are multiplexed in a single RTP session.

4.4. Recommendations

The recommendation is that endpoints MUST support both the static declaration of capture encoding SSRCs, and the RTP header extension method of sharing capture IDs, with the extension in every media packet. For low bandwidth situations, this may be considered excessive overhead; in which case endpoints MAY support the combined approach from [I-D.lennox-clue-rtp-usage]. The SDP offer MAY specify the SSRC mapping to media capture. In the case of static mapping topologies there will be no need to use the header extensions in the media, since the SSRC for the RTP stream will remain the same during the call unless a collision is detected and handled according to RFC5576 [RFC5576]. If the used topology uses dynamic mapping then the RTP header extension will be used to indicate the RTP stream switch for the media capture. In this case the SDP description may be used to negotiate the initial SSRC but this will be left for the implementation. Note that if the SSRC is defined explicitly in the SDP the SSRC collision should be handled as in RFC5576.

5. Application to CLUE Media Requirements

[I-D.lennox-clue-rtp-usage] offers a number of requirements that are believed to be necessary for a CLUE RTP mapping. The solutions described in this document are believed to meet that requirement, though some of them are only possible for some of the topologies. (Since the requirements are generally of the form "it must be possible for a sender to do something", this is adequate; a sender which wishes to perform that action needs to choose a topology which allows the behavior it wants.

In this section we address only those requirements where the topologies or the association mechanisms treat the requirements differently.

Media-4: It must be possible for an original source to move among switched captures (i.e. at one time be sent for one switched capture, and at a later time be sent for another one).

This applies naturally for static sources with a Switched Mixer. For dynamic sources with a Source-Projecting Mixer, this just requires the capture tag in the header extension element to be updated appropriately.

Media-6: Whenever a given source is transmitted for a switched capture, it must be immediately possible for a receiver to determine the switched capture it corresponds to, and thus that any previous source is no longer being mapped to that switched capture.

For a Switched Mixer, this applies naturally. For a Source-Projecting mixer, this is done based on the header extension.

Media-7: It must be possible for a receiver to identify the original source that is currently being mapped to a switched capture, and correlate it with out-of-band (non-Clue) information such as rosters.

For a Switched Mixer, this is done based on the CSRC, if the mixer is providing CSRCs; if for a Source-Projecting Mixer, this is done based on the SSRC.

Media-8: It must be possible for a source to move among switched captures without requiring a refresh of decoder state (e.g., for video, a fresh I-frame), when this is unnecessary. However, it must also be possible for a receiver to indicate when a refresh of decoder state is in fact necessary.

This can be done by a Source-Projecting Mixer, but not by a Switching Mixer. The last requirement can be accomplished through an FIR message [RFC5104], though potentially a faster mechanism (not requiring a round-trip time from the receiver) would be preferable.

Media-9: If a given source is being sent on the same transport flow to satisfy more than one capture (e.g. if it corresponds to more than one switched capture at once, or to a static capture as well as a switched capture), it should be possible for a sender to send only one copy of the source.

For a Source-Projecting Mixer, this can be accomplished by sending multiple dynamic capture IDs for the same source; this can also be done for an environment with a hybrid of mixer topologies and static and dynamic captures, described below in Section 6. It is not possible for static captures from a Switched Mixer.

Media-12: If multiple sources from a single synchronization context are being sent simultaneously, it must be possible for a receiver to associate and synchronize them properly, even for sources that are mapped to switched captures.

For a Mixed or Switched Mixer topology, receivers will see only a single synchronization context (CNAME), corresponding to the mixer. For a Source-Projecting Mixer, separate projecting sources keep separate synchronization contexts based on their original CNAMEs, thus allowing independent synchronization of sources from independent rooms without needing global synchronization. In hybrid cases, however (e.g. if audio is mixed), all sources which need to be synchronized with the mixed audio must get the same CNAME (and thus a mixer-provided timebase) as the mixed audio.

6. Examples

It is possible for a CLUE device to send multiple instances of the topologies in Section 3 simultaneously. For example, an MCU which uses a traditional audio bridge with switched video would be a Mixer topology for audio, but a Switched Mixer or a Source-Projecting Mixer for video. In the latter case, the audio could be sent as a static source, whereas the video could be dynamic.

More notably, it is possible for an endpoint to send the same sources both for static and dynamic captures. Consider the example in Section 11.1 of [I-D.ietf-clue-framework], where an endpoint can provide both three cameras (VC0, VC1, and VC2) for left, center, and right views, as well as a switched view (VC3) of the loudest panel.

It is possible for a consumer to request both the (VC0 - VC2) set and VC3. It is worth noting that the content of VC3 is, at all times, exactly the content of one of VC0, VC1, or VC2. Thus, if the sender uses the Source-Selection Mixer topology for VC3, the consumer that receives these three sources would not need to send any additional media traffic over just sending (VC0 - VC2).

In this case, the advertiser could describe VC0, VC1, and VC2 in its initial advertisement or SDP with static SSRCs, whereas VC3 would need to be dynamic. The role of VC3 would move among VC0, VC1, or VC2, indicated by the RTP header extension on those streams' RTP packets.

6.1. Static mapping

Using the video capture example from the framework for a three camera system with four monitors where one is for the presentation stream [I-D.ietf-clue-framework] document:

o VC0- (the camera-left camera stream, purpose=main, switched:no

- VC1- (the center camera stream, purpose=main, switched:no

- VC2- (the camera-right camera stream), purpose=main, switched:no

- VC3- (the loudest panel stream), purpose=main, switched:yes

- VC4- (the loudest panel stream with PiPs), purpose=main, composed=true; switched:yes

- VC5- (the zoomed out view of all people in the room), purpose=main, composed=no; switched:no

- VC6- (presentation stream), purpose=presentation, switched:no

Where the physical simultaneity information is:

{VC0, VC1, VC2, VC3, VC4, VC6}

{VC0, VC2, VC5, VC6}

In this case the provider can send up to six simultaneous streams and receive four one for each monitor. This is the maximum case but it can be further limited by the capture scene entries which may propose sending only three camera streams and one presentation, still since the consumer can select any media captures that can be sent simultaneously the offer will specify 6 streams where VC5 and VC1 are using the same resource and are mutually exclusive.

In the Advertisement there may be two capture scenes:

The first capture scene may have four entries:

{VC0, VC1, VC2}

{VC3}

{VC4}

{VC5}

The second capture scene will have the following single entry.

{VC6}

We assume that an intermediary will need to look at CLUE if want to have better decision on handling specific RTP streams for example based on them being part of the same capture scene so the SDP will not group streams by capture scene.

The SIP offer may be

```
m=video 49200 RTP/AVP 99

a=extmap:1 urn:ietf:params:rtp-hdrex:clue-capture-id / for support
of dynamic mapping

a=rtpmap:99 H264/90000

a=max-send-ssrc:{*:6}

a=max-recv-ssrc:{*:4}

a=ssrc:11111 CaptureID:1

a=ssrc:22222 CaptureID:2

a=ssrc:33333 CaptureID:3

a=ssrc:44444 CaptureID:4

a=ssrc:55555 CaptureID:5

a=ssrc:66666 CaptureID:6
```

In the above example the provider can send up to five main streams and one presentation stream.

We define a new Media Capture ID attribute CaptureID which will have the mapping of the related RTP stream

Note that VC1 and VC5 have the same SSRC since they are using the same resource.

- VC0- (the camera-left camera stream, purpose=main, switched:no, CaptureID =1

- VC1- (the center camera stream, purpose=main, switched:no, CaptureID =2

- VC2- (the camera-right camera stream), purpose=main, switched:no, CaptureID =3

- VC3- (the loudest panel stream), purpose=main, switched:yes, CaptureID =4

- VC4- (the loudest panel stream with PiPs), purpose=main, composed=true; switched:yes, CaptureID =5

- VC5- (the zoomed out view of all people in the room), purpose=main, composed=no; switched:no, CaptureID =2

- VC6- (presentation stream), purpose=presentation, switched:no, CaptureID =6

Note: We can allocate an SSRC for each MC which will not require the indirection of using a CaptureId. This will require if a switch to dynamic is done to provide information about which SSRC is being replaced by the new one.

6.2. Simulcast Static Mapping

The next example adds the support for simulcast offering to send low and high resolution of the same media capture and is based on avtcore-rtp-simulcast, RFC5576 [RFC5576] and RFC6236 [RFC6236].

The offer example is from a telepresence endpoint to an MCU offering three different pairs of RTP streams providing high and low res each and one RTP stream at high res.

The example is using SSRC multiplexing; this is different from the [I-D.westerlund-avtcore-rtp-simulcast] example that provide a solution where the low and high resolutions version uses a different RTP session. RFC6236 [RFC6236] does not describe how to indicate that the offer can send more than one image size. This is addressed in [I-D.westerlund-avtcore-rtp-simulcast] but the draft only addresses the case of using multiple RTP sessions. There is a discussion in section 4.2 of [I-D.westerlund-avtcore-rtp-simulcast] about using a single RTP session but no protocol solution. The proposal is to use the SDP srcname as described in section X.

```
m=video 49200 RTP/AVPF 98, 99

a=extmap:1 urn:ietf:params:rtp-hdrex:clue-capture-id

a=rtpmap:98 H264/90000

a=fmtp:98 profile-level-id=42c01f

a=imageattr:98 send [x=1280,y=720] [x=640, y=360]

a=max-send-ssrc:{*:7}

a=max-recv-ssrc:{*:4}

a=ssrc:11111 CaptureID:1
```

```
   a=ssrc:11111 cname:alice@foo.com

   a=ssrc:11111 srcname:v1.imgattr1

   a=ssrc:11115 CaptureID:2

   a=ssrc:11115 cname:alice@foo.com

   a=ssrc:11115 srcname:v1.imgattr2

   a=ssrc:22222 CaptureID:3

   a=ssrc:22222 cname:alice@foo.com

   a=ssrc:22222 srcname:v2.imgattr1

   a=ssrc:22225 CaptureID:4

   a=ssrc:22225 cname:alice@foo.com

   a=ssrc:22225 srcname:v2.imgattr2

   a=ssrc:33333 CaptureID:5

   a=ssrc:33333 cname:alice@foo.com

   a=ssrc:33333 srcname:v3.imgattr1

   a=ssrc:33335 CaptureID:6

   a=ssrc:33335 cname:alice@foo.com

   a=ssrc:33335 srcname:v3.imgattr2

   a=ssrc:44444 CaptureID:7

   a=ssrc:44444 cname:alice@foo.com

   a=ssrc:44444 srcname:v4.imgattr1
```

[I-D.westerlund-avtext-rtcp-sdes-srcname] suggest to use the same ssrcname also as an RTCP SDES message and also as an RTP header extension. It make sense to use the srcname as the mapping identifier.

The offer from the first example will be

```
m=video 49200 RTP/AVP 99

a=extmap:1 urn:ietf:params:rtp-hdrex:srcname

a=rtpmap:99 H264/90000

a=max-send-ssrc:{*:6}

a=max-recv-ssrc:{*:4}

a=ssrc:11111 cname:alice@foo.com

a=ssrc:11111 srcname:v1

a=ssrc:22222 cname:alice@foo.com

a=ssrc:22222 srcname:v2

a=ssrc:33333 cname:alice@foo.com

a=ssrc:33333 srcname:v3

a=ssrc:44444 cname:alice@foo.com

a=ssrc:44444 srcname:v4

a=ssrc:55555 cname:alice@foo.com

a=ssrc:55555 srcname:v5

a=ssrc:66666 cname:alice@foo.com

a=ssrc:66666 srcname:v6
```

6.3. Dynamic Mapping

For topologies that use dynamic mapping there is no need to provide the SSRCs in the offer (they may not be available if the offers from the sources will not include them when connecting to the mixer or remote endpoint) In this case the captureID (srcname) will be specified first in the advertisement.

The SIP offer may be

```
m=video 49200 RTP/AVP 99

a=extmap:1 urn:ietf:params:rtp-hdrex:clue-capture-id
```

```
      a=rtpmap:99 H264/90000

      a=max-send-ssrc:{*:4}

      a=max-recv-ssrc:{*:4}
```

This will work for ssrc multiplex. It is not clear how it will work when RTP streams of the same media are not multiplexed in a single RTP session. How to know which encoding will be in which of the different RTP sessions.

7. Acknowledgements

place holder

8. IANA Considerations

TBD

9. Security Considerations

TBD.

10. References

10.1. Normative References

[I-D.ietf-clue-framework]
Romanow, A., Duckworth, M., Pepperell, A., and B. Baldino, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-06 (work in progress), July 2012.

[I-D.lennox-clue-rtp-usage]
Lennox, J., Witty, P., and A. Romanow, "Real-Time Transport Protocol (RTP) Usage for Telepresence Sessions", draft-lennox-clue-rtp-usage-04 (work in progress), June 2012.

[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.

10.2. Informative References

[I-D.ietf-clue-telepresence-use-cases]
Romanow, A., Botzko, S., Duckworth, M., Even, R., and I. Communications, "Use Cases for Telepresence Multi-streams", draft-ietf-clue-telepresence-use-cases-04 (work in progress), August 2012.

[I-D.lennox-mmusic-sdp-source-selection]
Lennox, J. and H. Schulzrinne, "Mechanisms for Media Source Selection in the Session Description Protocol (SDP)", draft-lennox-mmusic-sdp-source-selection-04 (work in progress), March 2012.

[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-01 (work in progress), July 2012.

[I-D.westerlund-avtext-codec-operation-point]
Westerlund, M., Burman, B., and L. Hamm, "Codec Operation Point RTCP Extension", draft-westerlund-avtext-codec-operation-point-00 (work in progress), March 2012.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[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.

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

State", RFC 4575, August 2006.

[RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", RFC 4796, February 2007.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

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

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, May 2011.

Authors' Addresses

Roni Even
Huawei Technologies
Tel Aviv,
Israel

Email: roni.even@mail01.huawei.com

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com