Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2013

M. Westerlund
B. Burman
P. Sandgren
Ericsson

October 22, 2012

RTCP SDES Item SRCNAME to Label Individual Sources
draft-westerlund-avtext-rtcp-sdes-srcname-02

Abstract

This document defines a new SDES item called SRCNAME which uniquely identifies a single media source, like a camera or a microphone. That way anyone receiving the SDES information from a set of interlinked RTP sessions can determine which SSRCs are logically related to the same source. It can equally be used to label SSRC multiplexed related streams, such as FEC or Retransmission streams related to the original source stream in the same session. In addition the new SDES item is also defined for usage with the SDP source specific media attribute ("a=ssrc"), enabling an end-point to declare and learn the source bindings through signalling ahead of receiving RTP/RTCP packets.

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 April 25, 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

RTP [RFC3550] has always been a protocol that supports multiple participants, each sending their own media streams in RTP sessions. Previously, many implementations have aimed only at point to point voice over IP with a single source in each end-point. Even client implementations aimed at video conferences have often been built with the assumption around central mixers that only deliver a single media stream per media type. However, more advanced client implementations may transmit multiple streams in the same RTP session and there may be tight relations between different streams and their SSRCs. For example, a client with several cameras that uses simulcast to send streams with different encodings of the video from each camera have the need of conveying the relation of the streams to the receiver. A similar example is a client with several cameras that uses SVC multi-session transmission [RFC6190] and also here the receiver needs to know which streams relate to which video source. Other examples of tight RTP relations are a retransmission stream and its original stream, and cases of forward error correction (FEC), where a client needs to associate a number of source streams with, in general, a different number of repair streams.

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. Problem Description

In a scenario where an endpoint needs to send several RTP media streams, in a single RTP session or spread across several RTP sessions, and where two or more of those streams are somehow related, that relation information is today not always possible to convey in a timely manner to entities (endpoints and middle nodes) that need it.

An RTP Mixer [RFC5117], on the other hand, must have all the SDP information available and can provide it to any number of participants, since there must be a mapping from the original sources to the Mixer’s own streams, which are in turn distributed to all other participants. That is also true for a source projecting mixer, since there is a projection algorithm that must be made to work. It is even likely that the Mixer is allowed to provide the stream relation and impose that onto all of the clients, rather than trying to map a wide variety of different relations onto what it provides.

A single relation between two or more streams means that each stream has a certain "role" in that specific relation. A "role" is related to a specific reason to group a set of streams. The number of different grouping tags defined in various RFC for use with the SDP group attribute [RFC5888], as well as the media decoding dependency attribute [RFC5583] can be used as an indication of the different roles that may need to be described.

Those stream relational roles are typically application-specific, can sometimes be complex, and a single stream can even take on several roles. The major difference between roles is that they commonly do not share the same hierarchy root node and sometimes also middle nodes differ between roles. All roles however use the same hierarchy leaves, being the RTP media streams, but different roles may want to name leaves differently. It should be possible to express such relation structure and allow a single stream to hold several roles. It is believed to be sufficient if a single stream role can be described as being part of a relation hierarchy.

4. Motivation

This section contains a brief description of existing techniques that conceivably could be used to provide information on RTP stream relations, and a motivation why those are not always sufficient. In addition, there are defined milestones for RTP stream duplication [I-D.ietf-avtext-rtp-duplication] in IETF AVTEXT and stream duplication grouping [I-D.ietf-mmusic-duplication-grouping] in MMUSIC WG that makes normative references to this document.

4.1. RTP SSRC

To rely on using the same RTP Synchronization SouRCe (SSRC) for all streams related to a particular media source is many times not possible when the related streams are part of the same RTP session, since the SSRC itself is the identifier to tell the streams apart. This method is not robust against SSRC collision and potentially forces cascading SSRC changes between sessions. It does also not provide any information in how the streams are related.

4.2. RTCP SDES CNAME

CNAME is not sufficient to express the necessary type of relation, although that is commonly inferred from end-points that have only one media stream per media type. The primary use of CNAME in multi-source usages is instead to indicate which end-point and what synchronization context a particular media stream relates to, and that usually means that all streams sent from a client have the same

CNAME.

4.3. SDP

A common solution is to use SDP [RFC4566] attributes to convey the relation between streams. Session-multiplexed streams can be associated with an attribute that groups different SDP m-lines [RFC5888], and SSRC-multiplexed streams can be grouped at the media level for each SDP m-line [RFC5576]. For example, Forward Error Correction Grouping Semantics in the Session Description Protocol [RFC5956] uses that media level grouping with the "FEC-FR" tag to group FEC associations when the different streams from a source are SSRC-multiplexed in the same RTP session.

Using SDP attributes may work fine in the case when the receivers of the streams also get an SDP describing the bindings of all the streams, but that is not always the case. One such example is a highly dynamic conference session where a large amount of clients are communicating with each other via an RTP Translator [RFC5117]. The RTP Translator forwards all RTP and RTCP traffic from a client to all other clients and the clients can be prepared to receive any number of streams of certain specified media. When a new client joins the session, the other clients may not be notified via explicit signalling before starting to receive media streams from this new client. Such notification could for example be made through a SIP Update with a new SDP containing an explicit list of the new streams, but there are also other possibilities. The clients will instead detect the new client’s streams directly via RTP and RTCP. Similar situations typically arise in multicast scenarios. In those cases, there is no way for a client or middle node to identify if and how certain streams are related to each other, since that information was only included in the SDP, if at all.

4.4. Implicit Methods

RTP Retransmission Payload Format [RFC4588] describes a solution for finding the association between original streams and retransmission streams when SSRC-multiplexing is used. The association can be resolved when the receiver receives a retransmission packet matching a retransmission request sent earlier. However, the RFC states that this mechanism might fail if there are two outstanding requests for the same packet sequence number in two different original streams of a session. Therefore, to avoid ambiguity in unicast a receiver MUST NOT have two outstanding requests for the same packet sequence number in two different original streams before the association is resolved. For multicast, however, this ambiguity cannot be avoided and SSRC-multiplexing of original and retransmission streams is therefore prohibited in multicast. By defining a solution for one to one mapping between an original stream and any supporting streams, this issue can be avoided in the future.

> Note: This document does not update RFC 4588 to use the proposed solution, but it may be done in the future.

5. Proposed Solution

To enable an RTP session participant to determine the close relation of different streams without the above mentioned problems, a new method for identifying such sources is needed. This identification is called Source Name, or SRCNAME and is a unique identifier identifying a single media source, like a camera, a microphone, a particular media mix, or conceptual stream.

5.1. SRCNAME Contents

The basic idea is that streams with matching SRCNAME are related, similar to the idea with RTCP SDES CNAME.

It is assumed that related streams will share the same synchronization context, meaning that the SRCNAME is scoped by CNAME and need not duplicate any CNAME information.

The SRCNAME format includes "." (%x2E) as a hierarchy separator, allowing a stream to relate to another stream at a certain hierarchy level. Each hierarchy level is then a node in a hierarchy tree. For example, assume a video stream being provided in two different resolutions, named "lowres" and "hires", each being protected by a Forward Error Correction stream, with another additive FEC stream covering both resolutions. The low resolution video media stream could have a SRCNAME being "program1.video.lowres.media", and its FEC stream "program1.video.lowres.fec". By this, and although it is not a stream in itself, it is possible to use "program1.video.lowres" to refer to the set of related streams (in this case media and FEC) belonging to "lowres". If needed, it is still possible to refer to the individual, physical, streams by using one more level of the hierarchy (".media" and ".fec"). The SRCNAME for the additive FEC stream, covering both resolutions and their per-stream FEC, could be "program1.video.fec". Building on the same example, an high fidelity audio stream belonging to the above video could use an SRCNAME of "program1.audio.hifi".

Note that the hierarchy structure can be chosen entirely by the media sender, but it is anyway possible to decide stream relations, at what level the streams relate, and which other streams that are included in the relation at that level by matching SRCNAME hierarchically left-to-right between "." hierarchy separators. The specific type of relation is not encoded into SRCNAME in any mandated way, but need to be stringently described by other means, for example SDP, and is out of scope for this specification. SRCNAME needs only express that streams are related, not exactly how the related streams should be processed together.

Note that SRCNAME need not be particularly human-readable as long as each node in the hierarchy has a tag that is unique for that CNAME context, which makes it possible to limit the SRCNAME size.

5.2. SRCNAME in SDES

RTP [RFC3550] defines the Source Description RTCP Packet (SDES), which contains one or more chunks, each of which is composed of SDES items describing the SSRC identified in that chunk. None of the present SDES items is, however, suitable for uniquely identifying a media source.

Therefore, we propose to define a new SDES item called the SRCNAME, which uses a unique label to identify a single media source, like a camera or a microphone. The source may also be a particular media mix or conceptual stream, such as the "most active speaker" output by a RTP mixer performing stream switching. That way, anyone receiving the SDES information from a set of interlinked RTP sessions or multiple SSRCs in the same session can determine which SSRCs are the same source. Connecting streams with SRCNAME can be done irrespective of which multiplexing type is used and it solves the problems with the current solutions described above.

5.3. SRCNAME in SDP

It is, however, possible that a receiver will receive the RTP streams before receiving SDES packets with all SRCNAME items and that would mean that the receiver cannot make the connections between SSRCs and SRCNAMEs when starting to receive the media. "Source-Specific Media Attributes in the Session Description Protocol (SDP)" [RFC5576] defines a way of declaring different attributes for SSRCs in each session in SDP, and if a new source attribute is added to this framework, it would be suitable for conveying the connections between SSRCs and SRCNAMEs before the media communication starts. Thus, in addition to the new SDES item we also define a new SDP source-specific media attribute called "srcname", which enables an end-point to declare and learn the source bindings ahead of receiving RTP/RTCP packets. Of course, this new SDP source attribute will not be useful for the case described above when clients did not get updates with new client’s stream bindings, but it will be useful in most other cases.

5.4. SRCNAME in RTP Header Extension

There is a risk that neither RTCP SDES nor SDP attributes are timely enough in cases where RTP streams are received before the SDES has arrived, in which case an RTP header extension [RFC5285] could be negotiated for use, containing a combination of CNAME and SRCNAME information. This type of rapid information synchronization through RTP header extension is similar to what is described in [RFC6051]. The RTP header extension need not be present in every RTP packet, for example only in the beginning of the stream, at key points, or periodically, according to the application’s needs and as chosen by the media sender.

6. SRCNAME Format

The SRCNAME MUST fulfill the requirements Section 6.5 in RTP [RFC3550] puts on SDES item values in general. These requirements is that it is a UTF-8 [RFC3629] string that have a maximum length of 255 bytes.

In addition, there are format restrictions to accommodate the relation hierarchy and multiple roles, as described by the following ABNF [RFC5234]:

```
srcname-node =    1*(%x01-09 / %x0B-0C / %x0E-2D / %x2F-FF)
    ; Same as RFC 4566 "byte-string"
    ; except for the hierarchy separator

srcname-content = srcname-node *(%x2E srcname-node)
```

Figure 1: SRCNAME Format ABNF

It is RECOMMENDED to use per communication session unique random identifiers, applying srcname-node restrictions, as srcname-node. The length of such srcname-node identifiers MAY be limited down to a single character, especially when the resulting SRCNAME has several nodes.

7. SDES Item SRCNAME

Source Descriptions are a method that should work with all RTP topologies (assuming that any intermediary node is supporting this item) and existing RTP extensions. We propose to define a new SDES item called SRCNAME. That way, anyone receiving the SDES information from a set of interlinked RTP sessions or SSRCs in a single session can determine which SSRCs are related to the same source, and at what hierarchy level.

This SRCNAME’s relation to CNAME is the following. CNAME represents an end-point and a synchronization context. If the different sources identified by SRCNAMEs should be played out synchronized when receiving them in a multi-stream context, then the sources need to be in the same synchronization context. Thus in all cases, all SSRCs with the same SRCNAME will have the same CNAME. A given CNAME may contain multiple sets of sources using different SRCNAMEs.

The SDES SRCNAME item follows the same format as the other SDES items defined in RTP [RFC3550]:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| SRCNAME=TBA1  |     length    | source name                ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: SDES SRCNAME Format

The source name field MUST follow the above srcname-content definition. Multiple SDES SRCNAME describing different relation roles MAY be included.

When using the SRCNAME SDES item, it is equally important as CNAME. Thus SRCNAME is RECOMMENDED to be included in all full compound RTCP packets being sent. It MAY also be included in non-compound packets in cases where the implementation believes that there might be new receivers needing the information.

8. SRCNAME in SDP

"Source-Specific Media Attributes in the Session Description Protocol (SDP)" [RFC5576] defines a way of declaring attributes for SSRC in each session in SDP. With a new SDES item, it is possible to use this framework to define how SRCNAME can also be provided in the SDP for each SSRC in each RTP session, thus enabling an end-point to declare and learn the source bindings ahead of receiving RTP/RTCP packets.

Hence, we propose a new SDP source attribute called srcname with the following structure:

```
a=ssrc:<ssrc-id> srcname:<srcname>
```

The srcname value MUST be identical to the SRCNAME value the media sender will send in the SDES SRCNAME item in the SDES RTCP packets. Multiple srcname attributes MAY be used to describe multiple relation roles.

FormalABNF syntax [RFC5234] for the "srcname" attribute:

```
srcname-attr = "srcname:" srcname

srcname = srcname-content

attribute =/ srcname-attr
   ; The definition of "attribute" is in RFC 4566.
```

Figure 3: SRCNAME Attribute ABNF

When used in SDP, srcname-content MUST use ISO 10646 in UTF-8 encoding, and MUST be independent of any "a=charset".

9. SRCNAME as RTP Header Extension

When SRCNAME information is carried as RTP header extension [RFC5285], the header extension MUST contain both CNAME and SRCNAME information, since SRCNAME is scoped by CNAME. Separate header extension identities are defined for SRCNAME and CNAME. This is motivated by the fact that a single RTP stream can have several SRCNAME, but only a single CNAME.

The RTP header extensions for CNAME and SRCNAME MAY use either one of the one-byte or two-byte header formats, depending on the CNAME and SRCNAME value size. The one-byte header SHOULD be used when the value contains at most 16 bytes. Note that the RTP header extension specification does not allow to mix one-byte and two-byte headers for the same stream, so if the value size of either SRCNAME or CNAME requires the two-byte header, the other MUST also use that header format.

The header extension payload for SRCNAME contains the srcname-content, as defined in Section 6. The header extension payload for CNAME contains the CNAME value as defined in [RFC3550]. Figures Figure 4 and Figure 5 show samples of the structure of the header extension payload for the two header formats.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID   |  len  | CNAME or SRCNAME value ...                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ID       |      len      |  CNAME or SRCNAME value ...   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5

The URN identifiers to use with "a=extmap" SDP signaling for SRCNAME and CNAME, respectively, MUST be

```
urn:ietf:params:rtp-hdrext:srcname
urn:ietf:params:rtp-hdrext:cname
```

10. Examples

This section shows SDP examples of declaring the SRCNAME in SDP.

10.1. Simulcast

In this use case the end-point is a client with a single audio source and two video sources, and it uses simulcast for sending different encodings of the same video source. This example is based on Using Simulcast in RTP sessions [I-D.westerlund-avtcore-rtp-simulcast]. The following SDP describes this.

```
v=0
o=alice 3203093520 3203093520 IN IP4 foo.example.com
s=Simulcast enabled client
t=0 0
c=IN IP4 foo.example.com
m=audio 49200 RTP/AVP 96
a=rtpmap:96 G719/48000/2
a=ssrc:521923924 cname:alice@foo.example.com
a=ssrc:521923924 srcname:a1
a=mid:1
m=video 49300 RTP/AVP 96
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:192392452 cname:alice@foo.example.com
a=ssrc:192392452 srcname:v1
a=ssrc:834753488 cname:alice@foo.example.com
a=ssrc:834753488 srcname:v2
a=mid:2
a=content:main
m=video 49400 RTP/AVP 97
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42c00d
a=imageattr:97 send [x=320,y=180]
a=ssrc:239245219 cname:alice@foo.example.com
a=ssrc:239245219 srcname:v1
a=ssrc:734623563 cname:alice@foo.example.com
a=ssrc:734623563 srcname:v2
a=mid:3
a=sendonly
```

The audio session is proposing to use one stereo stream of G.719 and the video sessions are proposing to send two different encodings of each video source, one with the resolution 640x360 and one with 320x180. The end-point also declares the SSRCs it intends to use with bindings to CNAME and SRCNAME, enabling the receiver of the SDP to bind together the video streams that originate from the same video camera. For example, the two streams having an SRCNAME of "v1" originate from the same video camera and belong together.

The use of the srcname attribute in the SDP is optional and the information can be retrieved from RTCP reporting, but it will then not be possible to correctly relate the video sources until the first RTCP report is received.

10.2. SVC with multi-session transmission

Here an example is shown of a client that uses SVC with multi-session transmission as described in RTP Payload Format for Scalable Video Coding [RFC6190]. RTP Payload Format for Scalable Video Coding [RFC6190] only describes examples for a client with one video source and the decoder dependencies of the different sessions are grouped using the Session grouping DDP attribute as defined in Signaling Media Decoding Dependency in the Session Description Protocol (SDP) [RFC5583] and implicitly CNAME.

However, if a client has two video sources and wishes to use multi-session transmission and send streams from both sources in each session, an additional grouping mechanism is needed to group the different streams in the different sessions. SRCNAME is suitable for this and here we show an example where the DDP attribute groups the different sessions and the SRCNAME is used to relate the different SSRCs in each RTP session to one of the two video sources.

```
   v=0
   o=bob 8473948250 8473948250 IN IP4 foo.example.com
   s=SVC MST client
   t=0 0
   c=IN IP4 foo.example.com
   a=group:DDP L1 L2 L3
   m=audio 49500 RTP/AVP 96
   a=rtpmap:96 G719/48000/2
   a=ssrc:293848928 cname:bob@foo.example.com
   a=mid:A1
   m=video 20000 RTP/AVP 96
   a=rtpmap:96 H264/90000
   a=fmtp:96 profile-level-id=4de00a; packetization-mode=1;
    mst-mode=NI-TC; sprop-parameter-sets={sps0},{pps0};
   a=ssrc:743947584 cname:bob@foo.example.com
   a=ssrc:743947584 srcname:V1.L1
   a=ssrc:283894947 cname:bob@foo.example.com
   a=ssrc:283894947 srcname:V2.L1
   a=mid:L1
   m=video 20002 RTP/AVP 97
   a=rtpmap:97 H264-SVC/90000
   a=fmtp:97 profile-level-id=53000c; packetization-mode=1;
    mst-mode=NI-T; sprop-parameter-sets={sps1},{pps1};
   a=ssrc:492784823 cname:bob@foo.example.com
   a=ssrc:492784823 srcname:V1.L2
   a=ssrc:892362397 cname:bob@foo.example.com
   a=ssrc:892362397 srcname:V2.L2
   a=mid:L2
   a=depend:97 lay L1:96
   m=video 20004 RTP/AVP 98
   a=rtpmap:98 H264-SVC/90000
   a=fmtp:98 profile-level-id=53001F; packetization-mode=1;
    mst-mode=NI-T; sprop-parameter-sets={sps2},{pps2};
   a=ssrc:184562894 cname:bob@foo.example.com
   a=ssrc:184562894 srcname:V1.L3
   a=ssrc:305605682 cname:bob@foo.example.com
   a=ssrc:305605682 srcname:V2.L3
   a=mid:L3
   a=depend:98 lay L1:96 L2:97
```

Thus, the client declares that it will send two video streams in each RTP session and the receiver is then able to relate the streams in the different sessions by using the SRCNAME binding, with matching (first parts of the) SRCNAME value. Without the SRCNAME binding it would not be possible for the receiver to know which streams belong to the same source. Note that the audio stream does not have an explicit srcname attribute in this example, but only relate to the video streams through the same CNAME. Note that the last part of the SRCNAMEs in the example, ".L1", ".L2" and ".L3" are not necessary but allowed and will not impact the ability to tell that the streams belong together, since related streams have the first part in common.

10.3. Retransmission

This use case shows how SRCNAME can be used to connect retransmission streams to the original streams in the case of SSRC multiplexed RTP retransmission [RFC4588]. This is included to exemplify how RTP retransmission could be updated to provide explicit bindings between the source and the repair stream, but just an example and not a specification.

```
v=0
o=carol 3462534872 3462534872 IN IP4 foo.example.com
s=SSRC-multiplexed retransmission client
t=0 0
c=IN IP4 foo.example.com
m=audio 49800 RTP/AVP 96
a=rtpmap:96 G719/48000/2
a=ssrc:8372496978 cname:carol@foo.example.com
a=mid:1
m=video 49300 RTP/AVP 96 97
a=rtpmap:96 H264/90000
a=rtcp-fb:96 nack
a=fmtp:96 profile-level-id=42c01e
a=rtpmap:97 rtx/90000
a=fmtp:97 apt=96;rtx-time=200
a=ssrc:192392452 cname:carol@foo.example.com
a=ssrc:192392452 srcname:v1.o
a=ssrc:834753488 cname:carol@foo.example.com
a=ssrc:834753488 srcname:v1.r
a=ssrc:682394013 cname:carol@foo.example.com
a=ssrc:682394013 srcname:v2.o
a=ssrc:284576129 cname:carol@foo.example.com
a=ssrc:284576129 srcname:v2.r
a=mid:2
```

The client proposes to send two original video streams in the video session and a retransmission stream for each one of them. The retransmission streams are associated with the respective original stream by using matching SRCNAME and a receiver would then know which original stream a certain retransmission stream is associated with. This solves the ambiguity problem when SSRC-multiplexing is used for retransmission and it enables SSRC-multiplexing of original and retransmission streams to be used also in multicast sessions. Note that ".o" and ".r" parts of SRCNAME are not needed, but may improve understanding of the example and will not affect the ability to match related streams.

10.4. Forward Error Correction

Forward Error Correction Grouping Semantics in the Session Description Protocol [RFC5956] defines two SDP attributes for grouping the associated source and FEC-based repair streams. One can be used for grouping different RTP sessions and the other can be used for grouping SSRCs in the same RTP session, i.e. when session-respective SSRC-multiplexing is used. However, it may be advantageous to SSRC-multiplex the source streams in one RTP session and the repair streams in another since that gives a receiver the possibility to reject the repair session in case it does not support the proposed FEC. In this case, the above mentioned grouping attributes cannot be used to associate the repair streams with the respective source stream since grouping of SSRCs cannot be made across RTP sessions. The following example shows how SRCNAME can be used for that.

```
v=0
o=dave 7352395826 7352395826 IN IP4 foo.example.com
s=FEC client
t=0 0
c=IN IP4 foo.example.com
a=group:FEC-FR 2 3
m=audio 49300 RTP/AVP 96
a=rtpmap:96 G719/48000/2
a=ssrc:237847298 cname:dave@foo.example.com
a=mid:1
m=video 49200 RTP/AVP 100
a=rtpmap:100 MP2T/90000
a=ssrc:847612849 cname:dave@foo.example.com
a=ssrc:847612849 srcname:v1.o
a=ssrc:558237845 cname:dave@foo.example.com
a=ssrc:558237845 srcname:v2.o
a=exthdr:1 urn:ietf:params:rtp-hdrext:cname
a=exthdr:4 urn:ietf:params:rtp-hdrext:srcname
a=mid:2
m=application 49300 RTP/AVP 101
a=rtpmap:101 1d-interleaved-parityfec/90000
a=fmtp:101 L=5; D=10; repair-window=200000
a=ssrc:389572053 cname:dave@foo.example.com
a=ssrc:389572053 srcname:v1.r
a=ssrc:185729479 cname:dave@foo.example.com
a=ssrc:185729479 srcname:v2.r
a=exthdr:2 urn:ietf:params:rtp-hdrext:cname
a=exthdr:5 urn:ietf:params:rtp-hdrext:srcname
```

```
a=mid:3
```

In this example the client proposes to send two video streams in one session and two repair streams in the other session. The repair streams are associated with the respective video stream by using a matching SRCNAME. When receiving either this SDP, the SDES SRCNAME packets, or the SRCNAME/CNAME RTP header extensions (which are also offered), a receiver can make the connection between the source streams and the repair streams. Even a client not receiving the SDP will be able to do the association, by SRCNAME in either SDES or RTP header extension, if it has established one RTP session for receiving source streams and another for receiving repair streams. Note that ".o" and ".r" parts of SRCNAME are not needed, but may improve understanding of the example and will not affect the ability to match related streams (since they match on the highest hierarchical level).

11. Usage with the Offer/Answer Model

The SDP offer/answer procedures for a=ssrc are specified in Source-Specific Media Attributes in the Session Description Protocol (SDP) [RFC5576]. The SDP offer/answer procedures for a=exthdr are specified in A General Mechanism for RTP Header Extensions [RFC5285].

12. Backward Compatibility

Clients not supporting SRCNAME will not have the possibility to bind different streams to a specific media source, since they will not understand the SRCNAME SDES item or the RTP header extension. However, sending SRCNAME SDES items to a client not supporting it should not impose any problems since all clients should be prepared that new SDES items may be specified according to RTP [RFC3550].

According to the definition of SDP attributes in SDP: Session Description Protocol [RFC4566], if an attribute is received that is not understood, it MUST be ignored by the receiver. So a receiver not supporting the ssrc attribute will simply ignore it.

Source-Specific Media Attributes in the Session Description Protocol (SDP) [RFC5576] defines rules of how new source attributes should be registered, which means that a receiver supporting RFC 5576 should be prepared that new source attributes may be defined. This means that a user supporting some of the source attributes should not have any problems when the user receives an SDP with unknown source attributes.

RTP header extension will only be used when successfully negotiated in SDP, which requires support in both sender and receiver.

13. IANA Considerations

Following the guidelines in SDP [RFC4566], in The Session Description Protocol (SDP) Grouping Framework [RFC5888], and in RTP [RFC3550], the IANA is requested to register:

1. A new SDES item named SRCNAME, as defined in Section 7. This item needs to be assigned an identifier TBA1.

2. A new SDP source attribute named srcname, as defined in Section 8.

3. New RTP header extension URN identifiers for SRCNAME and CNAME, as defined in Section 9.

14. Security Considerations

The SDES or header extension SRCNAMEs being close to opaque identifiers could potentially carry additional meanings or function as overt channel. If the SRCNAME would be permanent between sessions, they have the potential for compromising the users' privacy as they can be tracked between sessions. See Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs) [RFC6222] for more discussion.

A third party modification of the srcname labels either in the RTCP SDES items, in the SDP a=ssrc attribute, or in the RTP header extension can cause service disruption. By modifying labels the wrong streams could be associated, with potentially serious effects including media disruptions. If streams that are to be associated aren't associated, then another type of failures occur. To prevent modification, insertion or deletion of the srcname labels, the carrying channel needs to be protected by integrity protection and source authentication. For RTCP and RTP header extension, various solutions exist, such as SRTP [RFC3711], DTLS [RFC6347], or IPsec [RFC4301]. For protecting the SDP, the signalling channel needs to provide protection. For SIP S/MIME [RFC3261] are the ideal, and hop by hopTLS [RFC5246] provides at least some protection, although not perfect. For SDPs retrieved using RTSP DESCRIBE [RFC2326], TLS would be the RECOMMENDED solution.

15. References

15.1. Normative References

[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.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

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

[RFC6222] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.

15.2. Informative References

[I-D.ietf-avtext-rtp-duplication] Begen, A. and C. Perkins, "Duplicating RTP Streams", draft-ietf-avtext-rtp-duplication-00 (work in progress), July 2012.

[I-D.ietf-mmusic-duplication-grouping] Begen, A., Cai, Y., and H. Ou, "Duplication Grouping Semantics in the Session Description Protocol", draft-ietf-mmusic-duplication-grouping-00 (work in progress), October 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 (work in progress), October 2011.

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, July 2008.

[RFC5583] Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", RFC 5583, July 2009.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

[RFC5956] Begen, A., "Forward Error Correction Grouping Semantics in the Session Description Protocol", RFC 5956, September 2010.

[RFC6051] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", RFC 6051, November 2010.

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

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

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

Patrik Sandgren
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 717 97 41
Email: patrik.sandgren@ericsson.com