

STRAW Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 25, 2017

L. Miniero
Meetecho
S. Garcia Murillo
Medooze
V. Pascual
Oracle
December 22, 2016

Guidelines to support RTCP end-to-end in Back-to-Back User Agents
(B2BUAs)
draft-ietf-straw-b2bua-rtcp-17

Abstract

SIP Back-to-Back User Agents (B2BUAs) are often designed to also be on the media path, rather than just intercepting signalling. This means that B2BUAs often implement an RTP/RTCP stack as well, thus leading to separate multimedia sessions that the B2BUA correlates and bridges together. If not disciplined, though, this behaviour can severely impact the communication experience, especially when statistics and feedback information contained in RTCP messages get lost because of mismatches in the reported data.

This document defines the proper behaviour B2BUAs should follow when also acting on the signalling/media plane in order to preserve the end-to-end functionality of RTCP.

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 June 25, 2017.

Copyright Notice

Copyright (c) 2016 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	2
2. Terminology	4
3. Signalling/Media Plane B2BUAs	4
3.1. Media Relay	5
3.2. Media-aware Relay	6
3.3. Media Terminator	11
4. IANA Considerations	12
5. Security Considerations	12
6. IANA Considerations	13
7. Change Summary	13
8. Acknowledgements	16
9. References	16
9.1. Normative References	16
9.2. Informative References	18
Authors' Addresses	19

1. Introduction

Session Initiation Protocol [RFC3261] Back-to-Back User Agents (B2BUAs) are SIP entities that can act as a logical combination of both a User Agent Server (UAS) and a User Agent Client (UAC). As such, their behaviour is not always completely adherent to the standards, and can lead to unexpected situations. [RFC7092] presents a taxonomy of the most commonly deployed B2BUA implementations, describing how they differ in terms of the functionality and features they provide.

Such components often do not only act on the signalling plane, that is intercepting and possibly modifying SIP messages, but also on the media plane. This means that, in order to receive and manage all RTP and RTCP [RFC3550] packets in a session, these components also

manipulate the session descriptions [RFC4566] in the related offer/answer exchanges [RFC3264]. The reasons for such a behaviour can be different. The B2BUA may want, for instance, to provide transcoding functionality for participants with incompatible codecs, or it may need the traffic to be directly handled for different reasons. This can lead to several different topologies for RTP-based communication, as documented in [RFC7667].

Whatever the reason, such a behaviour does not come without a cost. In fact, whenever a media-aware component is placed on the path between two or more participants that want to communicate by means of RTP/RTCP, the end-to-end nature of such protocols is broken. While this may not be a problem for RTP packets, which can be quite easily relayed, it definitely can cause serious issue for RTCP messages, which carry important information and feedback on the communication quality the participants are experiencing. Consider, for instance, the simple scenario only involving two participants and a single RTP session depicted in Figure 1:



Figure 1: B2BUA modifying RTP headers

In this common scenario, a participant (Alice) is communicating with another participant (Bob) as a result of a signalling session managed by a B2BUA: this B2BUA is also on the media path between the two, and is acting as a media relay. This means that two separate RTP sessions are involved (one per side), each carrying two RTP streams (one per media direction). As part of this process, though, the B2BUA is also rewriting some of the RTP header information on the way. In this example, just the SSRC of the incoming RTP streams is changed, but more information may be modified as well (e.g., sequence numbers, timestamps, etc.). In particular, whenever Alice sends an RTP packet, she sets her SSRC (SSRC1) in the RTP header of her RTP source stream. The B2BUA rewrites the SSRC (SSRC3) before relaying the packet to Bob. At the same time, RTP packets sent by Bob (SSRC4) get their SSRC rewritten as well (SSRC2) before being relayed to Alice.

Assuming now that Alice needs to inform Bob she has lost several packets in the last few seconds, she will place the related received RTP stream SSRC she is aware of (SSRC2), together with her own

(SSRC1), in RTCP Reports and/or NACKs. Since the B2BUA is making use of different SSRCs for the RTP streams in the RTP session it established with each participant, blindly relaying Alice's incoming RTCP messages to Bob would cause issues. These RTCP messages would reference SSRCs Bob doesn't know about, which would result in precious feedback being dropped. In fact, Bob is only aware of SSRCs SSRC4 (the one his source RTP stream uses) and SSRC3 (the one he's receiving from the B2BUA in the received RTP stream), and knows nothing about SSRCs SSRC1 and SSRC2 in the messages he received instead. Considering the feedback being dropped because of this may contain precious information, e.g., related to packet loss, congestion, and other network issues or considerations, the inability to take them into account may lead to severe issues. For instance, Bob may flood Alice with more media packets she can handle, and/or not retransmit Alice the packets she missed and asked for. This may easily lead to a very bad communication experience, if not eventually to an unwanted termination of the communication itself.

This is just a trivial example that, together with additional scenarios, will be addressed in the following sections. Nevertheless, it is a valid example of how such a simple mishandling of precious information may lead to serious consequences. This is especially true if we picture more complex scenarios involving several participants at the same time, multiple RTP sessions (e.g., a video stream along audio) rather than a single one, redundancy RTP streams, SSRC multiplexing and so on. Considering how common B2BUA deployments are, it is very important for them to properly address RTCP messages, in order to be sure that their activities on the media plane do not break or interfere with anything relevant to the session.

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].

Besides, this document addresses, where relevant, the RTP-related terminology as disciplined in [RFC7656].

3. Signalling/Media Plane B2BUAs

As described in the introductory section, it's very common for B2BUA deployments to also act on the media plane, rather than just signalling alone. In particular, [RFC7092] describes three different categories of such B2BUAs: a B2BUA, in fact, may act as a simple media relay (1), effectively unaware of anything that is transported; it may be a media-aware relay (2), also inspecting and/or modifying

RTP and RTCP messages as they flow by; or it may be a full-fledged media termination entity (3), terminating and generating RTP and RTCP messages as needed.

[RFC3550] and [RFC7667] already mandate some specific behaviours in the presence of certain topologies. Anyway, due to their mixed nature B2BUAs sometimes can't or won't implement all relevant specifications. This means that it's not rare to encounter issues that may be avoided with a more disciplined behaviour in that regard, that is if the B2BUAs followed at least a set of guidelines to ensure no known problems occur. For this reason, the following subsections will describe the proper behaviour B2BUAs, whatever above category they fall in, should follow in order not to impact any end-to-end RTCP effectiveness.

3.1. Media Relay

A media relay, as identified in [RFC7092], simply forwards all RTP and RTCP messages it receives, without either inspecting or modifying them. Using the RTP Topologies terminology, this can be seen as a RTP Transport Translator. As such, B2BUA acting as media relays are not aware of what traffic they're handling. This means that both packet payloads and packet headers are opaque to them. Many Session Border Controllers (SBC) implement this kind of behaviour, e.g., when acting as a bridge between an inner and outer network.

Considering all headers and identifiers in both RTP and RTCP are left untouched, issues like the SSRC mismatch described in the previous section would not occur. Similar problems could still happen, though, for different reasons, as for instance if the session description prepared by the B2BUA, whether it has been modified or not, ends up providing incorrect information. This may happen, for example, if the SDP on either side contains 'ssrc' [RFC5576] attributes that don't match the actual SSRC being advertized on the media plane, or when the B2BUA advertized support for NACK because it implements it, while the original INVITE didn't. Such issues might occur, for instance, when the B2BUA acting as a media relay is generating a new session description when bridging an incoming call, rather than using the original session description. This may cause participants to find a mismatch between the SSRCs advertized in the SDP and the ones actually observed in RTP and RTCP messages, or to have them either ignore or generate RTCP feedback packets that were not explicitly advertized as supported.

In order to prevent such an issue, a media-relay B2BUA SHOULD forward all the SSRC- and RTCP-related SDP attributes when handling a multimedia session setup between participants: this includes attributes like 'ssrc' [RFC3261], 'rtcp-fb' [RFC4585], 'rtcp-xr-

attrib' [RFC3611] and others. However, certain SDP attributes may lead to call failures when forwarded by a media relay, as they have an implied assumption that the attribute describes the immediate peer. A clear example of this is the 'rtcp' [RFC3605] attribute, which describes the expected RTCP peer port. Other attributes might include the immediate peer's IP address, preferred transport, etc. In general, the guideline is to require rewriting of attributes that are implicitly describing the immediate peer. B2BUAs SHOULD forward all other SDP attributes in order to avoid breaking additional functionality endpoints may be relying on. If implementors have doubts about whether this guidance applies to a specific attribute, they should test to determine if call failures occur.

The cited 'rtcp' example is also relevant whenever RTP/RTCP multiplexing [RFC5761] support is being negotiated. If the B2BUA acting as a Media Relay is unaware of the specifics of the traffic it is handling, and as such may not have RTP/RTCP parsing capabilities, it SHOULD reject RTP/RTCP multiplexing by removing the 'rtcp-mux' SDP attribute. If instead the Media Relay is able to parse RTP/RTCP, and can verify that demultiplexing can be performed without any RTP Payload Type rewrites (i.e., no overlap between any RTP Payload Types and the RTCP Payload Type space has been detected), then the B2BUA SHOULD negotiate RTP/RTCP multiplexing support if advertized.

It is worth mentioning that, leaving RTCP messages untouched, a media relay may also leak information that, according to policies, may need to be hidden or masqueraded, e.g., domain names in CNAME items. Besides, these CNAME items may actually contain IP addresses: this means that, should a NAT be involved in the communication, this may actually result in CNAME collisions, which could indeed break the end-to-end RTCP behaviour. While [RFC7022] can prevent this from happening, there may be implementations that don't make use of it. As such, a B2BUA MAY rewrite CNAME items if any potential collision is detected, even in the Media Relay case. If a B2BUA does indeed decide to rewrite CNAME items, though, then it MUST generate new CNAMEs following [RFC7022]. The same SHOULD be done in case RTP extensions involving CNAMEs are involved (e.g., "urn:ietf:params:rtp-hdrex:sdes:cname", [RFC7941]). If that is not possible, e.g., because the Media Relay does not have RTP header editing capabilities or does not support these extensions, then the B2BUA MUST reject the negotiation of such extensions when negotiating the session.

3.2. Media-aware Relay

A Media-aware relay, unlike the the Media Relay addressed in the previous section, is aware of the media traffic it is handling. This means it inspects RTP and RTCP messages flowing by, and may even modify their headers. Using the RFC3550 terminology, this can be

seen as a RTP Translator. A B2BUA implementing this role, though, typically does not inspect the RTP payloads, which would be opaque to them: this means that the actual media would not be manipulated (e.g, transcoded).

This makes them quite different from the Media Relay previously discussed, especially in terms of the potential issues that may occur at the RTCP level. In fact, being able to modify the RTP and RTCP headers, such B2BUAs may end up modifying RTP related information like SSRC/CSRC, sequence numbers, timestamps and others in an RTP stream, before forwarding the modified packets to the other interested participants. This means that, if not properly disciplined, such a behaviour may easily lead to issues like the one described in the introductory section. For this reason, it is very important for a B2BUA modifying RTP-related information across two related RTP streams to also modify, in a coherent way, the same information in RTCP messages.

It is worthwhile to point out that such a B2BUA may not necessarily forward all the packets it receives, though. Selective Forwarding Units (SFU) [RFC7667], for instance, may be implemented to aggregate or drop incoming RTCP messages, while at the same time originating new ones on their own. It is important to clarify that a B2BUA SHOULD NOT randomly drop or forward RTCP feedback of the same type (e.g., a specific XR block type, or specific Feedback messages) within the context of the same session, as that may lead to confusing, if not broken, feedback to the recipients of the message due to gaps in the communication. As to the messages that are forwarded and/or aggregated, though, it's important to make sure the information is coherent.

Besides the behaviour already mandated for RTCP translators in Section 7.2 of [RFC3550], a media-aware B2BUA MUST handle incoming RTCP messages to forward following this guideline:

SR: [RFC3550]

If the B2BUA has changed the SSRC of the sender RTP stream a Sender Report refers to, it MUST update the SSRC in the SR packet header as well. If the B2BUA has changed the SSRCs of other RTP streams too, and any of these streams are addressed in any of the SR report blocks, it MUST update the related values in the SR report blocks as well. If the B2BUA has also changed the base RTP sequence number when forwarding RTP packets, then this change MUST be reflected in the 'extended highest sequence number received' field in the Report Blocks. In case the B2BUA is acting as a Selective Forwarding Units (SFU) [RFC7667], it needs to track in the outgoing SR the relevant number of packets sent and total amount of bytes sent to the receiver.

RR: [RFC3550]

The same guidelines given for SR apply for RR as well.

SDES: [RFC3550]

If the B2BUA has changed the SSRC of any RTP stream addressed in any of the chunks of an incoming SDES message, it MUST update the related SSRCs in all the chunks. The same considerations made with respect to CNAME collisions at the end of Section 3.1 apply here as well.

BYE: [RFC3550]

If the B2BUA has changed the SSRC of any RTP stream addressed in the SSRC/CSRC identifiers included in a BYE packet, it MUST update them in the message.

APP: [RFC3550]

If the B2BUA has changed the SSRC of any RTP stream addressed in the header of an APP packet, it MUST update the identifier in the message. Should the B2BUA be aware of any specific APP message format that contains additional information related to SSRCs, it SHOULD update them as well accordingly.

Extended Reports (XR): [RFC3611]

If the B2BUA has changed the SSRC of the RTP stream associated with the originator of an XR packet, it MUST update the SSRC in the XR message header. The same guidelines given for SR/RR, with respect to SSRC identifiers in report blocks, apply for all the Report Block types in the XR message as well. If the B2BUA has also changed the base RTP sequence number when forwarding RTP packets, then this change MUST be reflected in the 'begin_seq' and 'end_seq' fields that are available in most of the Report Block types that are part of the XR specification.

Receiver Summary Information (RSI): [RFC5760]

If the B2BUA has changed any SSRC of RTP streams addressed in a RSI packet, it MUST update the SSRC identifiers in the message. This includes the distribution source SSRC, which MUST be rewritten with the one the B2BUA uses to send RTP packets to each sender participant, the summarized SSRC and, when a Collision Sub-Report Block is available, the SSRCs in the related list.

Port Mapping (TOKEN): [RFC6284]

If the B2BUA has changed any SSRC of RTP streams addressed in a TOKEN packet, it MUST update the SSRC identifiers in the message. This includes the Packet Sender SSRC, which MUST be rewritten with the one the B2BUA uses to send RTP packets to each sender participant, and the Requesting Client SSRC when the message is a

response, which MUST be rewritten using the related sender participant(s) SSRC.

Feedback messages: [RFC4585]

All Feedback messages have a common packet format, which includes the SSRC identifier of the packet sender and the SSRC identifier of the media source the feedback is related to. Just as described for the previous messages, these SSRC identifiers MUST be updated in the message if the B2BUA has changed the SSRC of the RTP streams addressed there. It MUST NOT, though, change a media source SSRC that was originally set to zero, unless zero is actually the SSRC that was chosen by one of the involved endpoints, in which case the above mentioned rules as to SSRC rewriting apply. Considering that many feedback messages also include additional data as part of their specific Feedback Control Information (FCI), a media-aware B2BUA MUST take care of them accordingly, if it can parse and regenerate them, according to the following guidelines:

NACK: [RFC4585]

A media-aware B2BUA MUST properly rewrite the Packet ID (PID) of all addressed lost packets in the NACK FCI if it changed the RTP sequence numbers.

TMMBR/TMMBN/FIR/TSTR/TSTN/VBCM: [RFC5104]

A media-aware B2BUA MUST properly rewrite the additional SSRC identifier in the specific FCI, if it changed the related RTP SSRC of the media sender.

REMB: [I-D.alvestrand-rmcat-remb]

This draft describes an RTCP Payload-Specific feedback message that reports the receiver's available bandwidth to the sender. As of the time of this writing, REMB has been widely deployed, but has not been standardized. The REMB mechanism will not function correctly across a media-aware B2BUA that changes the SSRC of the media sender unless it also changes the SSRC values in the REMB packet.

Explicit Congestion Notification (ECN): [RFC6679]

The same guidelines given for SR/RR management apply, considering the presence of sequence numbers in the ECN Feedback Report format. For what concerns the management of RTCP XR ECN Summary Report messages, the same guidelines given for generic XR messages apply.

Apart from the generic guidelines related to Feedback messages, no additional modifications are needed for PLI, SLI and RPSI feedback messages.

Of course, the same considerations about the need for SDP and RTP/RTCP information to be coherent applies to media-aware B2BUAs. This means that, if a B2BUA changes any SSRC, it MUST update the related 'ssrc' attributes, if present, before sending it to the recipient. Besides, it MUST rewrite the 'rtcp' attribute if provided. At the same time, while a media-aware B2BUA is typically able to inspect/modify RTCP messages, it may not support all RTCP messages. This means that a B2BUA may choose to drop RTCP messages it can't parse. In that case, a media-aware B2BUA MUST advertize its RTCP level of support in the SDP in a coherent way, in order to prevent, for instance, a UAC to from sending NACK messages that would never reach the intended recipients. It's important to point out that, in case a compound RTCP packet was received and any RTCP message in it needs to be dropped, then the B2BUA SHOULD NOT drop the whole compound RTCP packet, but only the selected messages.

The same considerations on CNAMEs made when talking of Media Relays apply for Media-aware Relays as well. Specifically, if RTP extensions involving CNAMEs are involved (e.g., "urn:ietf:params:rtp-hdrext:sdes:cname", [RFC7941]) and negotiated because the B2BUA supports them, then the B2BUA MUST update the CNAME value in there as well, if it was changed. It is worth pointing out that, if the new CNAME is larger than the old one, this would result in a larger RTP packet than originally received. If the length of the updated packet exceeds the MTU of any of the networks the packet will traverse, this can result in the packet being dropped and lost by the recipient.

A different set of considerations is worthwhile for what concerns RTP/RTCP multiplexing [RFC5761] and Reduced-Size RTCP [RFC5506]. While the former allows for a better management of network resources by multiplexing RTP packets and RTCP messages over the same transport, the latter allows for a compression of RTCP messages, thus leading to less network traffic. For what concerns RTP/RTCP multiplexing, a B2BUA acting as a Media Relay may use it on either RTP session independently. This means that, for instance, a Media Relay B2BUA may use RTP/RTCP multiplexing on one side of the communication, and not use it on the other side, if the endpoint does not support it. This allows for a better management of network resources on the side that does support it. In case any of the parties in the communications supports it and the B2BUA does too, the related 'rtcp-mux' SDP attribute MUST be forwarded on the other side(s). If the B2BUA detects that any of the parties in the communication do not support the feature, it may decide to either disable it entirely or still advertize it for the RTP sessions with parties that do support it. In case the B2BUA decides to involve RTP/RTCP multiplexing, it MUST ensure that there are no conflicting RTP payload type numbers on either side. When there are, it MUST rewrite RTP payload type numbers to prevent conflicts in the session

where the RTP/RTCP multiplexing is applied. Should RTP payload types be rewritten, the related information in the SDP MUST be updated accordingly.

For what concerns Reduced-Size RTCP, instead, the considerations are a bit different. In fact, while a Media Relay B2BUA may choose to use it on the side that supports it and not on the side that doesn't, there are several reasons for discouraging such a behaviour. While Reduced-Size allows indeed for less network traffic related to RTCP messaging in general, this gain may lead a Reduced-Size RTCP implementation to also issue a higher rate of RTCP feedback messages. This would result in an increased RTCP traffic on the side that does not support Reduced-Size, and could as a consequence be actually counterproductive if the available bandwidth is different on the two sides. Negotiating a session with both sides would allow the B2BUA to discover which one supports Reduced-Size and which doesn't, and in case decide whether to allow the sides to independently use Reduced-Size or not. Should the B2BUA decide to disable the feature on all sides, which is suggested in case Reduced-Size is not supported by all parties involved, it MUST NOT advertize support for the Reduced-Size RTCP functionality on either side, by removing the 'rtcp-rsize' attribute from the SDP.

3.3. Media Terminator

A Media Terminator B2BUA, unlike simple relays and media-aware ones, is also able to terminate media itself. As such, it can inspect and/or modify RTP payloads as well. This means that such components, for instance, can act as media transcoders and/or originate specific RTP media. Using the RTP Topologies terminology, this can be seen as a RTP Media Translator. Such a topology can also be seen as a Back-to-back RTP sessions through a Middlebox, as described in Section 3.2.2 of [RFC7667]. Such a capability makes them quite different from the previously introduced B2BUA typologies. Since such a B2BUA would terminate RTP itself, it can take care of the related statistics and feedback functionality directly, with no need to simply relay any message between the participants in the multimedia session.

For this reason, no specific guideline is needed to ensure a proper end-to-end RTCP behaviour in such scenarios, mostly because most of the times there would be no end-to-end RTCP interaction among the involved participants in the first place. Nevertheless, should any RTCP message actually need to be forwarded to another participant in the multimedia session, the same guidelines provided for the media-aware B2BUA case apply.

For what concerns RTP/RTCP multiplexing support, the same considerations already given for the Media Relay management also

apply for a Media Terminator. Some different considerations might be given as to the Reduced-Size RTCP functionality, instead. In fact, in the Media Terminator case it is safe to use the feature independently on each side, as the B2BUA would terminate RTCP. In that case, the B2BUA SHOULD advertize and negotiate support for Reduced-Size if available, and MUST NOT otherwise.

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

The discussion made in the previous sections on the management of RTCP messages by a B2BUA worked under the assumption that the B2BUA has actually access to the RTP/RTCP information itself. This is indeed true if we assume that plain RTP and RTCP is being handled, but may not be once any security is enforced on RTP packets and RTCP messages by means of SRTP [RFC3711].

While typically not an issue in the Media Relay case, where RTP and RTCP packets are forwarded without any modification no matter whether security is involved or not, this could definitely have an impact on Media-aware Relays and Media Terminator B2BUAs. To make a simple example, if we envisage a SRTP/SRTCP session across a B2BUA, where the B2BUA itself has no access to the keys used to secure the session, there would be no way to manipulate SRTP headers without violating the hashing on the packet. At the same time, there would be no way to rewrite the RTCP information accordingly either.

For this reason, it is important to point out that the operations described in the previous sections are only possible if the B2BUA has a way to effectively manipulate the packets and messages flowing by. This means that, when media security is involved, only the Media-unaware Relay scenario can be properly addressed. Attempting to cover Media-aware Relay and Media Termination scenarios when involving secure sessions will inevitably lead to the B2BUA acting as a man-in-the-middle, and consequently its behaviour is unspecified and discouraged. More considerations on this are provided in [RFC7879].

It is also worth pointing out that there are scenarios where an improper management of RTCP messaging across a B2BUA may lead, willingly or not, to situations not unlike an attack. To make a simple example, an improper management of a REMB feedback message containing, e.g., information on the limited bandwidth availability for a user, may lead to missing or misleading information to its peer. This may cause the peer to increase the encoder bitrate, maybe

up to a point where a user with poor connectivity will inevitably be choked by an amount of data it cannot process. This scenario may thus result in what looks like a Denial of Service (DOS) attack towards the user.

6. IANA Considerations

This document has no IANA actions.

7. Change Summary

Note to RFC Editor: Please remove this whole section.

The following are the major changes between the 16 and the 17 versions of the draft:

- o Clarified the meaning of a sentence.

The following are the major changes between the 14 and the 15 versions of the draft:

- o Several changes addressing the IESG review (list follows).
- o Addressed 'rtcp-mux' in 3.1 as well, and not only 3.2.
- o Clarified that, if CNAMEs are rewritten, RTP extensions referencing them (e.g., [RFC7941]) should be updated too. Clarified that MTU issues can occur if the rewriting results in a larger RTP packet.
- o Clarified that when handling SR packets, the an SFU B2BUA must track packets/bytes sent.
- o Removed references to billing, lawful interception, etc. from the intro.
- o Moved some references (especially those affected by MUSTs in 3.2) to Normative.
- o Rewritten the "Such attributes SHOULD NOT be forwarded" section to clarify the context of the attributes that may lead to a failure if not taken care of.
- o Clarified that randomly dropping RTCP packets can lead to confusion on the recipient.
- o Updated text related to REMB.

- o Smaller fixes here and there.

The following are the major changes between the 13 and the 14 versions of the draft:

- o Removed first paragraph of Security Considerations which was unclear.
- o Added an IANA Considerations section to clarify there are no actions.

The following are the major changes between the 12 and the 13 versions of the draft:

- o Updated authors' affiliations and mail addresses.

The following are the major changes between the 11 and the 12 versions of the draft:

- o Addressed remaining points in Ben's second review.
- o Updated reference of STRAW's DTLS-SRTP draft to new [RFC7879].

The following are the major changes between the 10 and the 11 versions of the draft:

- o Addressed Ben's second review.

The following are the major changes between the 09 and the 10 versions of the draft:

- o Replaced references to obsoleted RFC 5117 with [RFC7667].
- o Made reference to [RFC7656] normative.
- o Clarified text across the whole document to address Ben's review.

The following are the major changes between the 08 and the 09 versions of the draft:

- o Updated references to documents which have become RFC in the meanwhile, [RFC7667] and [RFC7656].

The following are the major changes between the 06 and the 07 versions of the draft:

- o Clarified the suggested change by Colin Perkins on the management of CNAME items in SDES, and added reference to [RFC7022].

- o Addressed comment by Simon Perreault on CNAME collisions management.

The following are the major changes between the 05 and the 06 versions of the draft:

- o Addressed comment by Colin Perkins on the management of CNAME items in SDES.

The following are the major changes between the 04 and the 05 versions of the draft:

- o Clarified behaviour when SSRC is zero.
- o Fixed a couple of nits found by the Idnits tool.

The following are the major changes between the 03 and the 04 versions of the draft:

- o Addressed review by Magnus Westerlund.
- o Added guidelines for ECN RTCP messages.
- o Clarified that if an RTCP message is dropped because unsupported, only the unsupported packet is dropped and not the compound packet that contains it.
- o Added reference to Section 3.2.2 of [RFC7667] to Section 3.3.
- o Added considerations on RTP/RTCP multiplexing and Reduced-Size RTCP.

The following are the major changes between the 02 and the 03 versions of the draft:

- o Rephrased the Media Path Security section to take into account the MITM-related discussion in Honolulu.
- o Added some Security Considerations.

The following are the major changes between the 01 and the 02 versions of the draft:

- o Updated terminology to better adhere to [RFC7656].
- o Rephrased the Media Path Security section to take into account the MITM-related discussion in Toronto.

- o Clarified that NACK management might be trickier when SRTP is involved.

The following are the major changes between the 00 and the 01 versions of the draft:

- o Updated references and mapping per taxonomy RFC (7092).
- o Added a reference to RTP topologies, and tried a mapping as per-discussion in London.
- o Added more RTCP message types to the Media-Aware section.
- o Clarified that fixing the 'rtcp' SDP attribute is important.
- o Added a new section on the impact of media security.

8. Acknowledgements

The authors would like to thank Flavio Battimo and Pierluigi Palma for their invaluable feedback in the early stages of the document. The authors would also like to thank Colin Perkins, Bernard Aboba, Albrecht Schwarz, Hadriel Kaplan, Keith Drage, Jonathan Lennox, Stephen Farrell, Magnus Westerlund, Simon Perreault and Ben Campbell for their constructive comments, suggestions, and reviews that were critical to the formulation and refinement of this document.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<http://www.rfc-editor.org/info/rfc4566>>.

- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<http://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<http://www.rfc-editor.org/info/rfc7656>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <<http://www.rfc-editor.org/info/rfc3611>>.
- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, DOI 10.17487/RFC5760, February 2010, <<http://www.rfc-editor.org/info/rfc5760>>.
- [RFC6284] Begen, A., Wing, D., and T. Van Caenegem, "Port Mapping between Unicast and Multicast RTP Sessions", RFC 6284, DOI 10.17487/RFC6284, June 2011, <<http://www.rfc-editor.org/info/rfc6284>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<http://www.rfc-editor.org/info/rfc5104>>.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <<http://www.rfc-editor.org/info/rfc6679>>.

- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <<http://www.rfc-editor.org/info/rfc5761>>.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, DOI 10.17487/RFC5506, April 2009, <<http://www.rfc-editor.org/info/rfc5506>>.
- [RFC7022] Begen, A., Perkins, C., Wing, D., and E. Rescorla, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMES)", RFC 7022, DOI 10.17487/RFC7022, September 2013, <<http://www.rfc-editor.org/info/rfc7022>>.
- [RFC7941] Westerlund, M., Burman, B., Even, R., and M. Zanaty, "RTP Header Extension for the RTP Control Protocol (RTCP) Source Description Items", RFC 7941, DOI 10.17487/RFC7941, August 2016, <<http://www.rfc-editor.org/info/rfc7941>>.

9.2. Informative References

- [RFC7092] Kaplan, H. and V. Pascual, "A Taxonomy of Session Initiation Protocol (SIP) Back-to-Back User Agents", RFC 7092, DOI 10.17487/RFC7092, December 2013, <<http://www.rfc-editor.org/info/rfc7092>>.
- [RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <<http://www.rfc-editor.org/info/rfc7667>>.
- [I-D.alvestrand-rmcat-remb] Alvestrand, H., "RTCP message for Receiver Estimated Maximum Bitrate", draft-alvestrand-rmcat-remb-03 (work in progress), October 2013.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <<http://www.rfc-editor.org/info/rfc5576>>.
- [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <<http://www.rfc-editor.org/info/rfc3605>>.

- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<http://www.rfc-editor.org/info/rfc3711>>.
- [RFC7879] Ravindranath, R., Reddy, T., Salgueiro, G., Pascual, V., and P. Ravindran, "DTLS-SRTP Handling in SIP Back-to-Back User Agents", RFC 7879, DOI 10.17487/RFC7879, May 2016, <<http://www.rfc-editor.org/info/rfc7879>>.

Authors' Addresses

Lorenzo Miniero
Meetecho

Email: lorenzo@meetecho.com

Sergio Garcia Murillo
Medooze

Email: sergio.garcia.murillo@gmail.com

Victor Pascual
Oracle

Email: victor.pascual.avila@oracle.com