STRAW Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 18, 2015

L. Miniero
Meetecho
S. Garcia Murillo
Medooze
V. Pascual
Quobis

April 16, 2015

Guidelines to support RTCP end-to-end in Back-to-Back User Agents (B2BUAs)
draft-ietf-straw-b2bua-rtcp-06

Abstract

SIP Back-to-Back User Agents (B2BUAs) are often envisaged to also be on the media path, rather than just intercepting signalling. This means that B2BUAs often implement an RTP/RTCP stack as well, whether to act as media transcoders or to just passthrough the media themselves, 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 packets 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 October 18, 2015.

Table of Contents

1. Introduction 2
2. Terminology 4
3. Signalling/Media Plane B2BUAs 5
3.1. Media Relay 5
3.2. Media-aware Relay 6
3.3. Media Terminator 10
4. Media Path Security 11
5. IANA Considerations 12
6. Security Considerations 12
7. Change Summary 12
8. Acknowledgements 14
9. References 14
9.1. Normative References 14
9.2. Informative References 14
Authors' Addresses 16

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 completelely adherent to the standards, and can lead to unexpected situations the IETF is trying to address. [RFC7092] presents a taxonomy of the most 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, when on the signalling path between two or more participants willing to communicate, such components also

manipulate the session description [RFC4566] in order to have all RTP and RTCP [RFC3550] pass through it as well within the context of an SDP offer/answer [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 like billing, lawful interception, session recording and so on. This can lead to several different topologies for RTP-based communication, as documented in [RFC5117]. These topologies are currently being updated to address new commonly encountered scenarios as well [I-D.ietf-avtcore-rtp-topologies-update].

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, and their effectiveness may be affected as a consequence. While this may not be a problem for RTP packets, which from a protocol point of view just contain opaque media packets and as such can be quite easily relayed, it definitely can cause serious issue for RTCP packets, which carry important information and feedback on the communication quality the participants are experiencing. In fact, RTCP packets make use of specific ways to address the media they are referring to. Consider, for instance, the simple scenario only involving two participants and a single RTP session depicted in Figure 1:

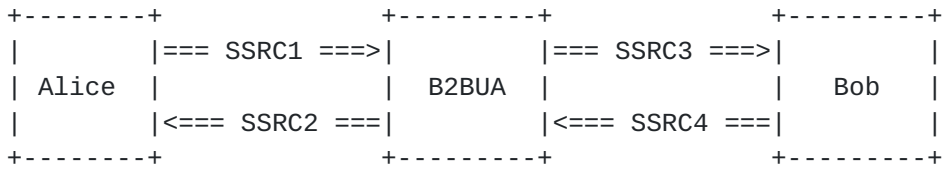

Figure 1: B2BUA modifying RTP headers

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, it is also rewriting some of the RTP header information on the way, for instance because that's how its RTP relaying stack works: in this example, just the SSRC of the incoming RTP audio streams is changed, but more information may be changed as well (e.g., sequence numbers, timestamps, etc.). In particular, whenever Alice sends an audio RTP packet, she sets her SSRC (SSRC1) to 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 audio packets in the last few seconds, maybe because of a network congestion, she would of course place the related received RTP stream SSRC she is aware of (SSRC2), together with her own (SSRC1), in RTCP Reports and/or NACKS to do so, hoping for a retransmission [RFC4588] or for Bob to slow down. Since the B2BUA is making use of different SSRCs for the RTP streams in the RTP session it established with each participant, a blind relaying of the RTCP packets to Bob would in this case result, from Bob's perspective, in unknown SSRCs being addressed, thus resulting in the precious information 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 RTCP packets he would receive instead. As a consequence of the feedback being dropped, unaware of the issue Bob may continue to flood Alice with even more media packets and/or not retransmit Alice the packets she missed, which 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 trivial mishandling of precious information may lead to serious consequences, especially considering that more complex scenarios may involve 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 such feedback, in order to be sure that their activities on the media plane do not break anything they're not supposed to.

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 [I-D.ietf-avtext-rtp-grouping-taxonomy].

3. Signalling/Media Plane B2BUAs

As anticipated 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, according to the level of activities performed on the media plane: 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 packets as they flow by; or it may be a full-fledged media termination entity, terminating and generating RTP and RTCP packets as needed.

While [RFC3550] and [RFC5117] already mandate some specific behaviours when specific topologies are deployed, not all deployments strictly adhere to the specifications and as such it's not rare to encounter issues that may be avoided with a more disciplined behaviour in that regard. For this reason, the following subsections will describe the proper behaviour B2BUAs, whatever above category they fall in, should follow in order to avoid, or at least minimize, any impact on end-to-end RTCP effectiveness.

3.1. Media Relay

A media relay as identified in [RFC7092] basically just forwards, from an application level point of view, all RTP and RTP packets 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, meaning that not only the packet payloads are opaque to them, but headers as well. 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 occur, though, should the session description end up providing incorrect information about the media flowing (e.g., if the SDP on either side contain 'ssrc' [RFC5576] attributes that don't match the actual SSRC being advertized on the media plane) or about the supported RTCP mechanisms (e.g., in case the B2BUA advertized support for NACK because it implements it, but the original INVITE didn't). Such an issue might occur, for instance, in case the B2BUA acting as a media relay is generating a new session description when bridging an incoming call, rather than taking into account the original session description in the first place. This may cause the participants to find a mismatch between the SSRCs advertized in SDP and the ones actually observed in

RTP and RTCP packets (which may indeed change during a multimedia session anyway, but having them synced during setup would help nonetheless), or having 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 interested participants: this includes attributes like 'ssrc' [RFC3261], 'rtcp-fb' [RFC4585], 'rtcp-xr-attrib' [RFC3611] and others. It SHOULD NOT, though, blindly forward all SDP attributes, as some of them (e.g., candidates, fingerprints, crypto, etc.) may lead to call failures for different reasons out of scope to this document. One notable example is the 'rtcp' [RFC3605] attribute that UAC may make use of to explicitly state the port they're willing to use for RTCP: considering the B2BUA would relay RTCP packets, the port as seen by the other UAC involved in the communication would differ from the one negotiated originally, and as such it MUST be rewritten accordingly.

Besides, it is worth mentioning that, leaving RTCP packets untouched, a media relay may also let through information that, according to policies, may be best left hidden or masqueraded, e.g., domain names in CNAME items. Nevertheless, that information cannot break the end-to-end RTCP behaviour.

3.2. Media-aware Relay

A Media-aware relay, unlike the the Media Relay addressed in the previous section, is actually aware of the media traffic it is handling. As such, it is able to inspect RTP and RTCP packets flowing by, and may even be able to modify the headers in any of them before forwarding them. Using the RFC3550 terminology, this can be seen as a RTP Translator. A B2BUA implementing this role would typically not, though, inspect the RTP payloads as well, 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 (and hence CSRC lists, that must of course be updated accordingly), sequence numbers, timestamps and the like in an RTP stream, before forwarding the modified packets to the other interested participants in the multimedia sessions on the RTP streams they're using to receive the media. This means that, if not properly disciplined, such a behaviour may easily lead to issues like the one described in the introductory section. As such, it is very important

for a B2BUA modifying RTP-related information across two related RTP streams to also modify the same information in RTCP packets as well, and in a coherent way, so that not to confuse any of the participants involved in a communication.

It is worthwile to point out that such a B2BUA would not necessarily forward all the packets it is receiving, though: Selective Forwarding Units (SFU) [I-D.ietf-avtcore-rtp-topologies-update], for instance, could aggregate or drop incoming RTCP messages, while at the same time originating new ones on their own. For 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 also handle incoming RTCP messages to forward following this guideline:

SR: [RFC3550]
: If the B2BUA has changed any SSRC in any RTP streams relation, it MUST update the SSRC-related information in the incoming SR packet before forwarding it. This includes the sender SSRC, which MUST be rewritten with the one the B2BUA uses in the RTP stream used to receive RTP packets from each participant, and the SSRC information in all the blocks, which MUST be rewritten using the related sender participant(s) SSRC. If the B2BUA has also changed the base RTP sequence number when forwarding RTP packets, then this change needs to be properly addressed in the 'extended highest sequence number received' field in the Report Blocks.

RR: [RFC3550]
: The same guidelines given for SR apply for RR as well.

SDES: [RFC3550]
: If the B2BUA has changed any SSRC in any direction, it MUST update the SSRC-related information in all the chunks in the incoming SDES packet before forwarding it. In case the SSRC in any of the chunks is changed, the related CNAME item SHOULD be updated as well in order to avoid potential CNAME collisions, especially if the RFC 3550 CNAME algorithm is used.

BYE: [RFC3550]
: If the B2BUA has changed any SSRC in any direction, it MUST update the SSRC in the BYE message.

APP: [RFC3550]
: If the B2BUA has changed any SSRC in any direction, it MUST update the SSRC in the APP 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.

Extended Reports (XR): [RFC3611]

If the B2BUA has changed any SSRC in any direction, it MUST update the SSRC-related information in the incoming XR message header before forwarding it. This includes the source SSRC, which MUST be rewritten with the one the B2BUA uses to send RTP packets to each sender participant, and the SSRC information in all the block types that include it, which MUST be rewritten using the related sender participant(s) SSRC. If the B2BUA has also changed the base RTP sequence number when forwarding RTP packets, then this change needs to be properly addressed 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 in any direction, it MUST update the SSRC-related information in the incoming RSI message header before forwarding it. 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, in case a Collision Sub-Report Block is available, the SSRCs in the related list.

Port Mapping (TOKEN): [RFC6284]

If the B2BUA has changed any SSRC in any direction, it MUST update the SSRC-related information in the incoming TOKEN message before forwarding it. 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 in case 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 of the packet sender and the one of the media source the feedack is related to. Just as described for the previous messages, these SSRC identifiers MUST be updated if the B2BUA has changed any SSRC in any direction. 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. Besides, 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]
Besides the common packet format management for feedback messages, a media-aware B2BUA MUST also properly rewrite the Packet ID (PID) of all addressed lost packets in the NACK FCI if it changed the RTP sequence numbers before forwarding a packet.

TMMBR/TMMBN/FIR/TSTR/TSTN/VBCM: [RFC5104]
Besides the common packet format management for feedback messages, a media-aware B2BUA MUST also properly rewrite the additional SSRC identifier all those messages envisage as part of their specific FCI if it changed the related RTP SSRC of the media sender.

REMB: [I-D.alvestrand-rmcat-remb]
Besides the common packet format management for feedback messages, a media-aware B2BUA MUST also properly rewrite the additional SSRC identifier(s) REMB packets envisage as part of their specific FCI if it changed the related RTP SSRC of the media sender.

Explicit Congestion Notification (ECN): [RFC6679]
Besides the common packet format management for feedback messages, the same guidelines given for SR/RR management apply as well, 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 instead.

Of course, the same considerations about the need for SDP and RTP/RTCP information to be coherent also applies to media-aware B2BUAs. This means that, if a B2BUA is going to change any SSRC, it SHOULD update the related 'ssrc' attributes if they were present in the original description before sending it to the recipient, just as it MUST rewrite the 'rtcp' attribute if provided. At the same time, the ability for a media-aware B2BUA to inspect/modify RTCP packets may also mean such a B2BUA may choose to drop RTCP packets it can't parse: in that case, a media-aware B2BUA MUST also advertize its RTCP level of support in the SDP in a coherent way, in order to prevent, for instance, a UAC to make use of NACK messages that would never reach the intended recipients. It's important to point out that, in case any RTCP packet needs to be dropped, then only the offending RTCP packet needs to be dropped, and not the whole compound RTCP packet it may belong to.

A different set of considerations, instead, 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 can 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 it's not supported. 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 does 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 both sides, and in case there are, it MUST rewrite RTP payload type numbers to ensure no conflict in the domain 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 other aspects to take into account before doing so. 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 bandwidth is different on each side. That said, the B2BUA can choose whether or not to advertize support for Reduced-Size RTCP on either side by means of the 'rtcp-rsize' SDP attribute. Should a B2BUA decide to allow the sides to independently use or not Reduced-Size, then the B2BUA MUST advertize support for the feature on the sides that support it, and MUST NOT advertize it on the sides that don't, by removing the related attribute from the SDP before forwarding it. Should the B2BUA decide to disable the feature on all sides, instead, it MUST NOT advertize support for the Reduced-Size RTCP functionality on any 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, that is taking care of RTP payloads as well and not only headers. 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 [I-D.ietf-avtcore-rtp-topologies-update]. Such a capability makes them quite different from the previously introduced B2BUA typologies, as this means they are going to terminate RTCP as well: in fact, since the media is terminated by themselves, the related statistics and feedback functionality can be taken care directly by the B2BUA, and does not need to be relayed to the other 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 at all, as the B2BUA would terminate them all and take care of them accordingly. Nevertheless, should any RTCP packet 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 basically apply for a Media Terminator as well. 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 leg. In that case, the same considerations about advertizing the support, or lack of support, of the feature on either side as described for the Media Relay case apply here as well.

4. Media Path Security

The discussion made in the previous sections on the management of RTCP messages by a B2BUA has so far mostly 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 this may not be true once any security is enforced on RTP packets and RTCP messages by means of SRTP [RFC3711], whether the keying is done using Secure Descriptions [RFC4568] or DTLS-SRTP [RFC5764].

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 think of 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, as most of the packet (especially when RTCP compound packets are involved) would be encrypted.

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, in case media security is involved, only the Media-unaware Relay scenario can be properly addressed. Attempting to cover Media-aware Relay and Media Terminarion scenarios when involving secure sessions will inevitably lead to the B2BUA acting as a man-in-the-middle, and as such its behaviour is unspecified and discouraged.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

This document, being a summary and vest common practice overview that covers different standards, does not introduce any additional consideration to those described by the aforementioned standard documents themselves.

It is worth pointing out, though, 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 information to its peer, who may end up increasing the encoder bitrate up to a point where the user with poor connectivity will inevitably be choked by an amount of data it cannot process. This scenario may as such result in what looks like a Denial of Service (DOS) attack towards the user.

7. Change Summary

Note to RFC Editor: Please remove this whole section.

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

- 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:

- Clarified behaviour when SSRC is zero.

- 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:

- Addressed review by Magnus Westerlund.

- Added guidelines for ECN RTCP messages.

- Clarified that if an RTCP packet is dropped because unsupported, only the unsupported packet is dropped and not the compound packet that contains it.

- Added reference to Section 3.2.2 of [I-D.ietf-avtcore-rtp-topologies-update] to Section 3.3.

- 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:

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

- Added some Security Considerations.

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

- Updated terminology to better adhere to [I-D.ietf-avtext-rtp-grouping-taxonomy].

- Rephrased the Media Path Security section to take into account the MITM-related discussion in Toronto.

- 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:

- Updated references and mapping per taxonomy RFC (7092).

- Added a reference to RTP topologies, and tried a mapping as per-discussion in London.
- Added more RTCP packet types to the Media-Aware section.
- Clarified that fixing the 'rtcp' SDP attribute is important.
- 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 and Magnus Westerlund 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, March 1997.

[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.

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

[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.

9.2. Informative References

[RFC7092] Kaplan, H. and V. Pascual, "A Taxonomy of Session Initiation Protocol (SIP) Back-to-Back User Agents", RFC 7092, December 2013.

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

[I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-07 (work in progress), April 2015.

[I-D.ietf-avtext-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., and G. Salgueiro, "A Taxonomy of Grouping Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-ietf-avtext-rtp-grouping-taxonomy-06 (work in progress), March 2015.

[I-D.alvestrand-rmcat-remb]
Alvestrand, H., "RTCP message for Receiver Estimated Maximum Bitrate", draft-alvestrand-rmcat-remb-03 (work in progress), October 2013.

[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, July 2006.

[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.

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

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, October 2003.

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.

[RFC6284] Begen, A., Wing, D., and T. Van Caenegem, "Port Mapping between Unicast and Multicast RTP Sessions", RFC 6284, June 2011.

[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.

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

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.

[RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

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

Authors' Addresses

Lorenzo Miniero
Meetecho

Email: lorenzo@meetecho.com

Sergio Garcia Murillo
Medooze

Email: sergio.garcia.murillo@gmail.com

Victor Pascual
Quobis

Email: victor.pascual@quobis.com