Guidelines to support RTCP end-to-end in Back-to-Back User Agents (B2BUAs)
draft-miniero-straw-b2bua-rtcp-00

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 pass through the media themselves, thus leading to separate media legs 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 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 April 23, 2014.

Copyright Notice
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 the IETF is trying to address. [I-D.ietf-straw-b2bua-taxonomy] 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 parties 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. The reasons for such a behaviour can be different: the B2BUA may want, for instance, to provide transcoding functionality for peers 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.

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 peers 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 peers are experiencing. In fact, RTCP packets make use of specific ways to address the media they are referring to. Consider, for instance, the scenario depicted in Figure 1:

```
+--------+    +--------+    +--------+
|        |=== SSRC1 ===|        |=== SSRC3 ===|        |
| Alice  |    B2BUA    |        |   Bob    |
|        |<=== SSRC2 ===|        |<=== SSRC4 ===|
+--------+    +--------+    +--------+
```

Figure 1: B2BUA modifying RTP headers

In this common scenario, a party (Alice) is communicating with a peer (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. It is also, though, 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 audio SSRC 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 adds her SSRC (SSRC1) to the RTP header; 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 peer audio SSRC she is aware of (SSRC2), together with her own (SSRC1), in RTCP Reports and/or NACKS to do so, hoping for a retransmission or for Bob to slow down. Since the B2BUA is making use of different SSRCs for the RTP communication with the party and the peer, a blind relaying of the RTCP packets to Bob would in this case result, from his 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 he’s originating) and SSRC3 (the one he’s receiving from the B2BUA), and knows nothing about SSRCs SSRC1 and SSRC2 in the RTCP packets he receives. As a consequence of the feedback being dropped, unaware of the issue Bob may continue to flood the party with even more media packets and/or not send Alice the packets she misses, 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 simple 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. 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].

3. 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, [I-D.ietf-straw-b2bua-taxonomy] 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.

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 [I-D.ietf-straw-b2bua-taxonomy] basically just forwards, from an application level point of view, all RTP and RTCP packets it receives, without either inspecting or modifying them. 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’ attributes that don’t match the actual SSRC being advertised on the media plane) or about the supported RTCP mechanisms (e.g., in case the B2BUA advertised 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 peers to find a mismatch between the SSRCs advertised in SDP and the ones actually observed in RTP and RTCP packets, or having them either ignore or generate RTCP feedback packets that were not explicitly advertised 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 session setup between interested parties: this includes attributes like ‘ssrc’ [RFC3261] and ‘rtcp-fb’ [RFC4585]. 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.

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 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. A B2BUA implementing this role would not, though, inspect the RTP payloads as well, which would be opaque to them.

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, sequence numbers, timestamps and the like before forwarding packets from one peer to another. 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 to also modify the same information in RTCP packets as well, and in a coherent way, so that not to confuse any of the peers involved in a communication. 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 the SSRC of any direction, it MUST update the SSRC-related information in the incoming SR packet before forwarding it. This includes the sender SSRC, which MUST be replaced by the one the B2BUA uses to send RTP packets to the sender peer, and the SSRC information in all the blocks, which MUST be replaced using the related sender peer(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 the SSRC of any direction, it MUST update the SSRC-related information in all the chunks in the incoming SDES packet before forwarding it.

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

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

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
feedback is related to. Just as described for the previous messages, these SSRC identifiers MUST be updated if the B2BUA has changed the SSRC of any direction. It MUST NOT, though, change a media source SSRC that was originally set to zero. 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 replace 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 replace 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 replace 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.

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. 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 SHOULD also advertise 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.
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. Such a capability makes them quite different from the previously introduced B2BUA typologies, as this means they most likely 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 logical peer in the multimedia communication.

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 peers at all, as the B2BUA would terminate them all and take care of them accordingly. Nevertheless, should any RTCP packet actually need to be delivered to the actual peer, the same guidelines provided for the media-aware B2BUA case apply.

4. IANA Considerations

This document makes no request of IANA.

5. Security Considerations

TBD. Not any additional consideration to what the standards already give? Probably this section will need a few words about how NOT following the guidelines can lead to security issues: e.g., not properly translating REMB messages can cause an increasing flow of media packets, that may be seen as attacks to devices that can’t handle the amount of data.

6. Acknowledgements

TBD.

7. References

7.1. Normative References


7.2. Informative References

[I-D.ietf-straw-b2bua-taxonomy]

[I-D.alvestrand-rmcat-remb]


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