Abstract

A SIP Back-To-Back User Agent (B2BUA) refers to the concatenation of a SIP User Agent Client (UAC) and a SIP User Agent Server (UAS). A transparent B2BUA is a particular type of B2BUA that forwards SIP messages in a SIP proxy-like way, and that also benefits from some
features of a User Agent (UA) element. This document recommends best current practices for the implementation of such a transparent B2BUA. Developing transparent B2BUAs that meet this set of requirements will greatly increase the likelihood that SIP applications will function properly.

Table of Contents

1. Introduction ........................................... 3
2. Terminology ............................................ 4
3. Features of a transparent B2BUA .......................... 4
   4.1. Forwarding SIP Messages .............................. 5
   4.1.1. Motivation ....................................... 5
   4.1.2. Examples ......................................... 5
   4.1.3. Recommendations .................................. 5
   4.2. Forwarding SIP Messages Referencing External Dialog .... 6
   4.2.1. Motivation ....................................... 6
   4.2.2. Examples ......................................... 7
   4.2.3. Recommendations .................................. 7
   4.3. Upstream and Downstream Forking ....................... 9
   4.3.1. Motivation ....................................... 9
   4.3.2. Examples ......................................... 9
   4.3.3. Recommendations .................................. 9
   4.4. B2BUA Forking ....................................... 9
   4.4.1. Motivation ....................................... 9
   4.4.2. Example ......................................... 10
   4.4.3. Recommendations .................................. 10
   4.5. Sending a BYE Request ................................ 10
   4.5.1. Motivation ....................................... 10
   4.5.2. Examples ......................................... 10
   4.5.3. Recommendations .................................. 10
   4.6. Third Party Call Control ............................ 10
   4.6.1. SIP and TLS ..................................... 10
5. Examples .................................................. 11
   5.1. Transparent B2BUA ................................... 11
   5.2. B2BUA and Conferencing, mapping ....................... 18
   5.3. B2BUA and Conferencing, no mapping ................... 22
6. Acknowledgements ......................................... 27
7. References ............................................... 27
Appendix A. Summary of the actions done by the B2BUA .......... 28
Authors’ Addresses ......................................... 30
Intellectual Property and Copyright Statements ................ 31
1. Introduction

SIP intermediaries often need to perform additional tasks that go beyond the scope of routing. Some examples of such tasks are topology hiding and network termination of dialogs, which are often implemented in so-called application servers and session border controllers.

Generally, these tasks can not be implemented with a SIP proxy element, as defined in RFC3261 [2], because the responsibility of a SIP proxy is basically limited to routing messages only. To circumvent this, the industry has adopted two different approaches:

1. The first one is the use of an "extended proxy": its behavior follows the SIP proxy behavior of Section 16 of [2] (i.e. the Call-Id and unknown headers are preserved when routing messages), except that it allows itself to perform additional features (e.g. can send a BYE message, can forward SIP message with a modified body, ...).

2. The second one is the use of a "transparent B2BUA": by nature, its behavior allows to implement UA features (e.g. can send BYE message, can generate a SIP message with any SIP body, ...). In addition, it also strives to route messages as a SIP proxy, even if many details need to be considered (e.g. when routing message, the Call-ID is modified, unknown headers are likely not to be preserved ...).

The difference between the two approaches is very weak. Most of the features of a "transparent B2BUA" are possible features for an "extended proxy".

Of course, if possible, a SIP proxy element should be used instead of these two approaches. Indeed, it is the single intermediary that is documented within SIP IETF specifications.

This document only discusses the "transparent B2BUA" approach because it leads to a huge number of end-to-end issues, if not implemented carefully. Indeed, RFC3261 [2] only mentions that a B2BUA is a concatenation of a UAC and UAS. This apparent flexibility is also a weakness: without more accurate details, the behavior of a SIP B2BUA is not predictable. When used as a SIP intermediary between two users, a B2BUA can thus potentially prevent them from using many SIP features.

Section 3 describes some cases that explain the proliferation of SIP B2BUA elements instead of a SIP Proxy element.

Ideally, these features should be implemented differently with end-to-end mechanisms, so that regular proxies would suffice.
However, the situation is that the industry has widely adopted such
SIP B2BUAs. This specification thus proposes some best current
practices in order to mitigate end-to-end SIP issues. They are
documented in Section 4.

The scope of this document is limited to a specific type of B2BUA:
those that basically behave as a proxy element plus the additional
features of Section 3. Therefore B2BUAs acting as SIP conferencing
elements or SIP relay elements are out-of-scope of this document.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED",
"SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT
RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as
described in BCP 14, [1] and indicate requirement levels for
compliant implementations.

3. Features of a transparent B2BUA

By its nature a B2BUA inserts itself into the call flow and modifies
the Call-ID. Indeed, the B2BUA is made of two User Agents: its first
UA terminates all SIP messages coming from the calling UA, and its
second UA originates all SIP messages towards the called UA.

A transparent B2BUA element is one type of B2BUA that routes SIP
messages as a proxy would do. However, being a B2BUA enables the
transparent B2BUA to:
- Hide user-identity, for example in the From and To header fields.
- Hide topology information, for example in the Call-ID, Via, Route,
  Record-Route, Contact header fields.
- Modify the SIP body, for example the media IP and port addresses
  in the SDP.
- Perform 3rd Party-Call Control, for example when REFER is not
  supported by one remote party.
- Relay an out-of-dialog request towards multiple destinations at
  the same time.
- Send a BYE request towards one, or even both parties, for example
  in prepaid applications.

Note that the document on Session Border Controllers [3] already
discusses the first four features in more detail.

([OPEN-POINT: The last 3 features are not transparent in the sense
that the intermediary does not only relay a SIP message, but the
intermediary also generates new messages on its own, which will]}
seriously increase the complexity of this work. Should we keep them?


This section gives some recommendations in order that a B2BUA be as transparent as possible within a SIP network.

4.1. Forwarding SIP Messages

4.1.1. Motivation

In order to minimize the impact on the SIP messages exchanged between two users, the B2BUA should forward all SIP messages. When forwarding a SIP message, the B2BUA should take care to preserve as many headers as possible, as well as the body.

4.1.2. Examples

If a SIP INVITE message sent by Alice indicates some supported extensions, it is important that the B2BUA forward these extensions in the SIP INVITE message sent to Bob. Otherwise, the two users will never be able to use these SIP extensions. Section 5.1 shows such an example.

If a SIP 200 OK of INVITE with an SDP offer is sent by Bob, it is necessary that the B2BUA forward the SIP 200 to Alice before generating the ACK request towards Bob. Otherwise, Bob’s UA will never receive the SDP answer.

4.1.3. Recommendations

When the UAS of the B2BUA receives a SIP request from an upstream proxy or UA, it must check whether it supports the extensions required that the UAC advertises in the Allow, Required, Supported, and any other additional headers. In practice, many extensions can be accepted transparently based on a local policy, however there definitely are some extensions that do require some implementation work within the B2BUA itself. For example, if a Require header field contains the option tag 100rel, the B2BUA needs to support the PRACK message.

If the B2BUA decides to relay the received request, its associated UAC generates a new downstream SIP request with its new Via, Max-Forwards, Call-ID, CSeq, and Contact header fields, as described in RFC3261. Route header fields of the upstream request MAY be copied in the downstream request, except the topmost Route header if it is
under the responsibility of the B2BUA. Additional Route header fields MAY also be added to the downstream request. Record-Route header fields of the upstream request are not copied in the new downstream request, as Record-Route is only meaningful for the upstream dialog. The UAC SHOULD copy other header fields and body from the upstream request into this downstream request before sending it.

[[TODO: discuss about OPTIONS and REGISTER messages.]]

[[TODO: add a discussion of Path (RFC3327) in the REGISTER section]]

Some SIP messages may contain information related to other SIP dialogs. In this case, it is important to also apply the recommendations of Section 4.2

[[TODO: more work is needed for the Contact header: duration of its life within the B2BUA server, GRUUs, event packages containing contacts...]]

When the UAC side of the B2BUA receives the downstream SIP response of a forwarded request, its associated UAS creates an upstream response (except for 100 responses). The creation of the Via, Max-Forwards, Call-ID, CSeq, Record-Route and Contact header fields follows the rules of [2]. The Record-Route header fields of the downstream response are not copied in the new upstream response, as Record-Route is only meaningful for the downstream dialog. The UAS SHOULD copy other header fields and body from the downstream response into this upstream response before sending it.

[[OPEN-ISSUE: which level of details is needed? Is it enough with the current description, or do we need to describe a better relationship with the Section 8 "General User Agent Behavior" of RFC3261 [2], or do we need a section similar to Section 16 "Proxy Behavior" of RFC3261 [2]? What about race conditions as done in [4]?]]

A summary of the transparency on these headers is also described in Appendix A.

4.2. Forwarding SIP Messages Referencing External Dialog

4.2.1. Motivation

As seen in the previous section, if a B2BUA is involved in a dialog on one side, there is another associated dialog on its other side. This can be considered as a first level of mapping, made of two dialogs.
There is also a second level of mapping to take into account. This happens when a SIP message contains a reference to another B2BUA dialog. There are two cases:

- The Request-URI of an out-of-dialog request contains a B2BUA’s contact from another dialog.
- An header field or a body of a message references another B2BUA dialog.

* The current list of such header fields is:
  + In-Reply-To (Call-ID) [2]
  + Replaces (Call-ID, to-tag, from-tag) [5]
  + Join (Call-ID, to-tag, from-tag) [6]
  + Target-Dialog (Call-ID, local-tag, remote-tag) [7]

* The current list of such bodies is:
  + Event Package for INVITE-Initiated Dialog[8]
  + Event Package for Conference State[9]
  + Event Package for Key Press Stimulus[10]

In both cases, it is again important that the B2BUA supports these extensions, and be able to update this dialog information.

4.2.2. Examples

Section 5.2 shows a transfer of a Point-to-Point Session into a Conference Call Control. In this example, the B2BUA is acting on the behalf of Alice and is systematically in the path of all SIP messages for Alice. In this case, the B2BUA is able to update the Replaces and the Target-Dialog header field.

Section 5.3 is the same example as the previous one except that the B2BUA is not always in the path of Alice’s messages. In this case, Carol’s UA receives a Replaces header with an unknown Call-ID. This is eventually fixed thanks to the B2BUA involved later in the call flow that updates the Replaces header field of the last INVITE to Bob.

4.2.3. Recommendations

4.2.3.1. In-Reply-To header field

The In-Reply-To header field [2] does not require that the referenced dialog still exist. This creates a specific problem for B2BUA functionality as the use of the In-Reply-To header would require that each B2BUA maintains a record of previous dialogs which have occurred so that the appropriate mapping of the Call-ID in the In-Reply-To header can occur when a new dialog using this header is received. The issue is how long must the Call-ID mapping be maintained to achieve this purpose, hours, days, weeks...?
Two alternatives exist for the handling of the In-Reply-To header:
- If the In-Reply-To does not match an existing dialog the B2B can strip the In-Reply-To header.
- The B2BUA can reject the INVITE containing the In-Reply-To header which does not match an existing dialog.

It is recommended not to use the In-Reply-To header field in networks in which B2BUAs are deployed. If the B2BUA receives a request containing an In-Reply-To header field value that does not match an existing dialog, it is recommended that the B2BUA strip the In-Reply-To header and pass on the request.

[[OPEN-POINT: From a terminal point of view, it is quite difficult for a UA to be aware of whether or not there is a B2BUA in the signalling path due to fact that it a man-in-the-middle device.]]

4.2.3.2. B2BUA Handling of Messages with Reference Headers

When a B2BUA receives a request creating a dialog and relays this request, it should record the mapping information between the dialog created on one side and the dialog created on the other side. This information is made of the two dialogs, which include the their different pieces (Call-ID, local tag, remote tag, local CSeq, remote CSeq, Route-set, local contact, remote target, and secure flag). This mapping information must be recorded until the end of the dialog.

When the B2BUA later receives an out-of-dialog request with a Request-URI targeting a local contact of the B2BUA, the B2BUA should insert the remote target of the associated dialog within the Request-URI of the outgoing request. This is done thanks to the recorded mapping information. (c.f. example flows labeled F13 and F14 of Section 5.3.)

When the B2BUA receives an out-of-dialog REFER request with a Refer-To header field containing a URI related to the B2BUA, the B2BUA should not modify the Refer-To URI, as this will be done in the resulting referred request. (c.f. example Section 5.3.)

When a B2BUA receives a message with a header field or a body referencing a dialog related to the B2BUA, it should update the referenced dialog (Call-ID and tags) thanks to the recorded mapping information.

The same basic handling is used for all three reference headers with care having to be taken with Target-Dialog as it is defined as Call-ID, local-tag, remote-tag. This makes Target-Dialog different from Replaces and Join (Call-ID, from-tag, to-tag) as the values used
in local and remote tags may be either the to or from tags from the original dialog. This depends upon the direction of the new request relative to the original request of the referenced dialog.

4.3. Upstream and Downstream Forking

4.3.1. Motivation

The B2BUA must cope with other SIP elements that may use SIP forking. Thus, a B2BUA must properly forward forked messages.

4.3.2. Examples

If an upstream proxy forks a SIP INVITE message, and if some of these forked messages arrive at a server hosting a B2BUA, the server must be ready to receive requests that only differ in the Request-URI and the Via header field, and forward them all downstream.

Similarly, if a downstream proxy forks, the B2BUA must be able to receive proxied responses that differ by their to-tag and forward them all upstream with a different to-tag. Otherwise, if two SDP offer/answer happen on the downstream side of the B2BUA, this may result in a single SDP offer with two SDP answers on the upstream side of the B2BUA.

4.3.3. Recommendations

If the B2BUA receives forked SIP requests, it must forward all forked requests downstream.

If the B2BUA receives forked SIP responses (i.e. responses with a different to-tag for a previously forwarded request), it must forward upstream responses with a different to-tag.

4.4. B2BUA Forking

[[Note: "B2BUA Forking" may be misleading as forking is only defined for SIP proxies. Can’t we find a better name?]]

4.4.1. Motivation

There are some cases where the B2BUA wants to perform a parallel search, which is a feature similar to the forking feature of a SIP proxy.
4.4.2. Example

Upon receiving an INVITE from Alice to Bob, the B2BUA forwards the INVITE in parallel towards two destinations: not only to Bob, but also to a Media Server that generates an early announcement.

4.4.3. Recommendations

Upon receiving a SIP request that can fork on its UAS side, a B2BUA MAY choose to forward the request in parallel to two destinations by creating multiple UACs. In case downstream responses with a different to-tag come back to the B2BUA, it must also forward upstream responses with a different to-tag.

4.5. Sending a BYE Request

4.5.1. Motivation

The intermediary needs to terminate a session.

4.5.2. Examples

Upon detecting a loss of connectivity with Bob, the B2BUA sends a SIP BYE message towards Alice to properly terminate the session.

When implementing a prepaid service, the B2BUA needs to be able to send a SIP BYE message to Alice, and also another one to Bob.

4.5.3. Recommendations

When sending a BYE on behalf of a user, the B2BUA must not try to forward associated responses.

[[OPEN-POINT: there is an issue if the BYE is challenged with a 407 response.]]

4.6. Third Party Call Control

[[OPEN-POINT: Do we want to discuss this feature being implemented by a B2BUA? The proposal is to include it.]]

4.6.1. SIP and TLS

[[TODO: How will credentials be established? Can a call be sips on one side of a B2B and sip on the other? ]]
5. Examples

For the sake of clarity, the values chosen for the following SIP header fields and parameters are simplified:
- branch parameter of Via
- tag parameter of From
- tag parameter of To
- sequence number of CSeq
- Call-ID

5.1. Transparent B2BUA
Alice      Alice’s Tr. B2BUA     Bob’s Proxy         Bob

<table>
<thead>
<tr>
<th>F1 INVITE</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F2 100</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>F3 INVITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F4 100</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>F5 INVITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F6 100</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>F7 180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F8 180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F9 180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F10 200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F11 200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F12 200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F13 ACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F14 ACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F15 ACK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTP flows</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F16 BYE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F17 BYE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F18 BYE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F19 200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F20 200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F21 200</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In this example, there is a B2BUA in the path between Alice and Bob. There is a first SIP dialog between Alice and the B2BUA. There is a second SIP dialog between the B2BUA and Bob.

Besides the minimum required header fields, INVITE labeled F1 also transports the Allow, Supported, and P-Visited-Network-ID header fields. Because the B2BUA supports them, they are all transparently copied in INVITE labeled F3.

[[OPEN-POINT: Is SDP "o=" field changed in F3 and F8?]]

F1 INVITE Alice -> B2BUA

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
Max-Forwards: 70
Route: <sip:b2bua.example.org;lr>
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>
Call-ID: 11111111
CSeq: 11 INVITE
Contact: <sip:alice@pc1.example.org>
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE
Supported: timer
P-Visited-Network-ID: "Visited network number 1"
Content-Type: application/sdp
Content-Length: 144

v=0
o=alice 2890844526 2890844526 IN IP4 pc1.example.org
s=-
c=IN IP4 pc1.example.org
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

F2 100 Trying B2BUA -> Alice

SIP/2.0 100 Trying
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>
Call-ID: 11111111
CSeq: 11 INVITE
Content-Length: 0
F3 INVITE B2BUA -> Bob

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>
Call-ID: 22222222
CSeq: 22 INVITE
Contact: <sip:alice@b2bua.example.org>
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE
Supported: timer
P-Visited-Network-ID: "Visited network number 1"
Content-Type: application/sdp
Content-Length: 144

v=0
o=alice 2890844526 2890844526 IN IP4 pc1.example.org
s=-
c=IN IP4 pc1.example.org
t=0 0
m=audio 49202 RTP/AVP 0
a=rtpmap:0 PCMU/8000

F4 100 Proxy -> B2BUA

SIP/2.0 100 Trying
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>
Call-ID: 22222222
CSeq: 22 INVITE
Content-Length: 0

F8 180 Bob -> B2BUA

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>;tag=222y
Call-ID: 22222222
Contact: <sip:bob@pc2.example.org>
Record-Route: <sip:example.org;lr>
Require: timer
Supported: timer
CSeq: 22 INVITE
F9 180 B2BUA -> Alice

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>;tag=111y
Call-ID: 11111111
Contact: <sip:bob@pc1.com>
Require: timer
Supported: timer
CSeq: 11 INVITE
Content-Length: 0

F11 200 Proxy -> B2BUA

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc2.example.org:5060;branch=z9hG4bK2222
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>;tag=222y
Call-ID: 22222222
CSeq: 22 INVITE
Contact: <sip:bob@pc2.example.org>
Record-Route: <sip:example.org;lr>
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE
Require: timer
Supported: timer
Session-Expires: 4000;refresher=uac
Content-Type: application/sdp
Content-Length: 142

v=0
o=bob 1390844527 1390844527 IN IP4 pc2.example.org
s=-
c=IN IP4 pc2.example.org
t=0 0
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000

F12 200 B2BUA -> Alice

SIP/2.0 200 OK
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK1111
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>;tag=111y
Call-ID: 11111111
CSeq: 11 INVITE
Contact: <sip:bob@b2bua.example.org>
Allow: INVITE, ACK, OPTIONS, CANCEL, BYE
Require: timer
Supported: timer
Session-Expires: 4000;refresher=uac
Content-Type: application/sdp
Content-Length: 142

v=0
o=bob 1390844527 1390844527 IN IP4 pc2.example.org
s=-
c=IN IP4 pc2.example.org
t=0 0
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000

F13 ACK Alice -> B2BUA

ACK sip:bob@b2bua.example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1112
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>;tag=111y
Call-ID: 11111111
CSeq: 1 ACK
Content-Length: 0

F14 ACK B2BUA -> Proxy

ACK sip:bob@pc2.example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2223
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>;tag=222y
Call-ID: 22222222
Route: <sip:example.org;lr>
CSeq: 22 ACK
Content-Length: 0

F17 BYE Proxy -> B2BUA

BYE sip:alice@b2bua.example.org SIP/2.0
Via: SIP/2.0/UDP pc2.example.org:5060;branch=z9hG4bK2224
Max-Forwards: 70
From: Bob <sip:bob@example.org>;tag=222y
To: Alice <sip:alice@example.org>;tag=222x
Call-ID: 22222222
CSeq: 222 BYE
Content-Length: 0

F18 BYE B2BUA -> Alice

BYE sip:alice@pc1.example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK1114
Max-Forwards: 70
From: Bob <sip:bob@example.org>;tag=111y
To: Alice <sip:alice@example.org>;tag=111x
Call-ID: 11111111
CSeq: 111 BYE
Content-Length: 0

F19 200 Alice -> B2BUA

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1114
From: Bob <sip:bob@example.org>;tag=111y
To: Alice <sip:alice@example.org>;tag=111x
Call-ID: 11111111
CSeq: 111 BYE
Content-Length: 0

F20 200 B2BUA -> Proxy

SIP/2.0 200 OK
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK1114
From: Bob <sip:bob@example.org>;tag=222y
To: Alice <sip:alice@example.org>;tag=222x
Call-ID: 22222222
CSeq: 222 BYE
Content-Length: 0
5.2. B2BUA and Conferencing, mapping

<table>
<thead>
<tr>
<th>Alice</th>
<th>Alice’s Tr</th>
<th>B2BUA</th>
<th>Bob</th>
<th>Carol</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 INVITE (d1)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F2 INVITE (d2)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F3 200</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F4 200</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F5 ACK</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F6 ACK</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F7 INVITE (d3)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F8 INVITE (d4)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F9 200</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F10 200</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F11 ACK</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F12 ACK</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F13 REFER (d5)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Refer-To: ...?Replaces:d1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target-D:d3</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F14 REFER (d6)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Refer-To: ...?Replaces:d2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target-D:d4</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>F15 INVITE (d7)</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Replaces=d2</td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>

This call flow is similar to the transfer of a Point-to-Point Session into a Conference Call Control as described in [11] (Section 5.10),
where 3 dialogs exist without any B2BUA.

This example introduces a B2BUA in the path of SIP messages between Alice, Bob and Carol, which induces three additional dialogs, namely d2, d4, and d6. It is important to note that if the B2BUA is not able to replace the reference to dialog d2 by a reference on dialog d3 in the Replaces header field, F7 INVITE will fail, as dialog d2 will be unknown to Carol.

In the flow below the AOR has been used as the Request-URI when establishing the communication between Alice and Bob and between Alice and Carol, F1, F2, F7 & F8. When the REFER request, F13 & F14, is sent however the Contact from the communication with Carol is used as the Request-URI as the use of the Target-Dialog header requires that the REFER request is directed to the terminal where the referenced dialog exists. This is similar to sending a subsequent request on an existing dialog. As there is a Replaces header parameter included in the Refer-To header of the REFER the Contact URI from the communication with Bob is used as the Refer-To URI. This is to ensure that the subsequent INVITE, F15, generated as a result of the REFER is directed to the UE on which the dialog to be replaced is terminated.

For the sake of brevity, the proxy-registrar of Bob and Alice is not described.

F1 INVITE Alice -> Alice’s B2BUA

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>
Call-ID: 11111111
CSeq: 11 INVITE
Contact: <sip:alice@pc1.example.org>
Route: <sip:b2bua.example.org;lr>
Content-Type: application/sdp
Content-Length: ...

F2 INVITE Alice’s B2BUA -> Bob

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>
Call-ID: 22222222
CSeq: 22 INVITE
Contact: <sip:alice@b2bua.example.org>
Content-Type: application/sdp
Content-Length: ...

F3 200 OK Bob -> Alice’s B2BUA

SIP/2.0 200 OK
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
From: Alice <sip:alice@pc1.example.org>;tag=222x
To: Bob <sip:bob@example.org>;tag=222y
Call-ID: 22222222
CSeq: 22 INVITE
Contact: <sip:bob@pc2.example.org>
Content-Type: application/sdp
Content-Length: ...

F4 200 OK Alice’s B2BUA -> Alice

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
From: Alice <sip:alice@pc1.example.org>;tag=111x
To: Bob <sip:bob@example.org>;tag=111y
Call-ID: 11111111
CSeq: 11 INVITE
Contact: <sip:bob@b2bua.example.org>
Content-Type: application/sdp
Content-Length: ...

F7 INVITE Alice -> Alice’s B2BUA

INVITE sip:carol@example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK3333
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=333x
To: Carol <sip:carol@example.org>
Call-ID: 33333333
CSeq: 11 INVITE
Contact: <sip:alice@pc1.example.org>
Route: <sip:b2bua.example.org;lr>
Content-Type: application/sdp
Content-Length: ...
F8 INVITE Alice’s B2BUA -> Carol

INVITE sip:carol@example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK4444
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=444x
To: Carol <sip:carol@example.org>
Call-ID: 44444444
CSeq: 44 INVITE
Contact: <sip:alice@b2bua.example.org>
Content-Type: application/sdp
Content-Length: ...

F9 200 OK Carol -> Alice’s B2BUA

SIP/2.0 200 OK
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK4444
From: Alice <sip:alice@example.org>;tag=444x
To: Carol <sip:carol@example.org>tag=444y
Call-ID: 44444444
CSeq: 44 INVITE
Contact: <sip:carol@pc3.example.org>
Content-Type: application/sdp
Content-Length: ...

F10 200 OK Alice’s B2BUA -> Alice

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK3333
From: Alice <sip:alice@example.org>;tag=333x
To: Carol <sip:carol@example.org>;tag=333y
Call-ID: 33333333
CSeq: 11 INVITE
Contact: <sip:carol@b2bua.example.org>
Content-Type: application/sdp
Content-Length: ...

F13 REFER Alice -> Alice’s B2BUA

REFER sip:carol@b2bua.example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK5555
Max-Forwards: 70
From: Alice <alice@example.org>;tag=555x
To: Carol <sip:carol@b2bua.example.org>
Call-ID: 55555555
CSeq: 2 REFER
Refer-To: <sip:bob@b2bua.example.org?Replaces=11111111
;to-tag=111y;from-tag=111x>
Target-Dialog: 33333333;local-tag=333x;remote-tag=333y
Contact: <sip:alice@pc1.example.org>
Content-Length: 0

F14 REFER Alice’s B2BUA -> Carol

REFER sip:carol@pc3.example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK6666
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=666x
To: Carol <sip:carol@pc3.example.org>
Call-ID: 66666666
CSeq: 2 REFER
Refer-To: <sip:bob@pc2.example.org?Replaces=22222222
;to-tag=222y;from-tag=222x>
Target-Dialog=44444444;local-tag=444x;remote-tag=444y>
Contact: <sip:alice@b2bua.example.org>
Content-Length: 0

F15 INVITE Carol -> Bob

INVITE sip:bob@pc2.example.org SIP/2.0
Via: SIP/2.0/UDP pc3.example.org:5060;branch=z9hG4bK7777
Max-Forwards: 70
From: Carol <sip:carol@example.org>;tag=777x
To: Bob <sip:bob@pc2.example.org>
Call-ID: 77777777
CSeq: 77 INVITE
Contact: <sip:alice@b2bua.example.org>
Replaces=22222222;to-tag=222y;from-tag=222x
Content-Type: application/sdp
Content-Length: ...

5.3. B2BUA and Conferencing, no mapping
This call flow is similar to the transfer of a Point-to-Point Session into a Conference Call Control as described in [11] (Section 5.10), where 3 dialogs exist without any B2BUA.

This example shows a B2BUA that does not systematically receive all SIP messages on behalf of Alice. Indeed, INVITE labeled F7 is not received by the B2BUA. In this case, the Refer-To and Replaces header fields received by Carol will contain information referencing a Contact URI and a dialog that are only known by Alice and its B2BUA.

Nevertheless, the B2BUA later in the flow will receive the F14 INVITE
and thus has the opportunity to update the F15 INVITE message.

A similar situation may also happen in case Alice’s UA load-balance SIP messages towards multiple B2BUAs that do not share the contextual information related to the mapping of Alice’s UA.

In the flow below the AOR has been used as the Request-URI when establishing the communication between Alice and Bob and between Alice and Carol, F1, F2, & F7. When the REFER request, F13, is sent however the Contact from the communication with Carol is used as the Request-URI as the use of the Target-Dialog header requires that the REFER request is directed to the terminal where the referenced dialog exists. This is similar to sending a subsequent request on an existing dialog. As there is a Replaces header parameter included in the Refer-To header of the REFER the Contact URI from the communication with Bob is used as the Refer-To URI. This is to ensure that the subsequent INVITE, F14 & F15, generated as a result of the REFER is directed to the UE on which the dialog to be replaced is terminated.

F1 INVITE Alice -> Alice’s B2BUA

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>
Call-ID: 11111111
CSeq: 11 INVITE
Contact: <sip:alice@pc1.example.org>
Route: <sip:b2bua.example.org;lr>
Content-Type: application/sdp
Content-Length: ...

F2 INVITE Alice’s B2BUA -> Bob

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>
Call-ID: 22222222
CSeq: 22 INVITE
Contact: <sip:alice@b2bua.example.org>
Content-Type: application/sdp
Content-Length: ...

F3 200 OK Bob -> Alice’s B2BUA

SIP/2.0 200 OK
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK2222
From: Alice <sip:alice@example.org>;tag=222x
To: Bob <sip:bob@example.org>;tag=222y
Call-ID: 22222222
CSeq: 22 INVITE
Contact: <sip:bob@pc2.example.org>
Content-Type: application/sdp
Content-Length: ...

F4 200 OK Alice’s B2BUA -> Alice

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK1111
From: Alice <sip:alice@example.org>;tag=111x
To: Bob <sip:bob@example.org>;tag=111y
Call-ID: 11111111
CSeq: 11 INVITE
Contact: <sip:bob@b2bua.example.org>
Content-Type: application/sdp
Content-Length: ...

F7 INVITE Alice -> Alice’s B2BUA

INVITE sip:carol@example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK3333
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=333x
To: Carol <sip:carol@example.org>
Call-ID: 33333333
CSeq: 11 INVITE
Contact: <sip:alice@pc1.example.org>
Content-Type: application/sdp
Content-Length: ...

F8 200 OK Bob -> Alice’s B2BUA

SIP/2.0 200 OK
Via: SIP/2.0/UDP b2bua.example.org:5060;branch=z9hG4bK4444
From: Alice <sip:alice@example.org>;tag=333x
To: Carol <sip:carol@example.org>;tag=333y
Call-ID: 33333333
CSeq: 11 INVITE
Contact: <sip:carol@pc3.example.org>
Content-Type: application/sdp
Content-Length: ...

F13 REFER Alice -> Carol

REFER sip:carol@pc3.example.org SIP/2.0
Via: SIP/2.0/UDP pc1.example.org:5060;branch=z9hG4bK5555
Max-Forwards: 70
From: Alice <sip:alice@example.org>;tag=555x
To: Carol <sip:carol@pc3.example.org>
Call-ID: 55555555
CSeq: 2 REFER
Refer-To: <sip:bob@b2bua.example.org?Replaces=11111111
 ;to-tag=111y;from-tag=111x>
Target-Dialog: 33333333;local-tag=333x;remote-tag=333y
Contact: <sip:alice@pc1.example.org>
Content-Length: 0

F14 INVITE Carol -> Alice’s B2BUA

INVITE sip:bob@b2bua.example.org SIP/2.0
Via: SIP/2.0/UDP pc3.example.org:5060;branch=z9hG4bK6666
Max-Forwards: 70
From: Carol <sip:carol@example.org>;tag=666x
To: Bob <sip:bob@b2bua.example.org>
Call-ID: 66666666
CSeq: 33 INVITE
Contact: <sip:carol@pc3.example.org>
Replaces=11111111;to-tag=111y;from-tag=111x>
Content-Type: application/sdp
Content-Length: ...

F15 INVITE Alice’s B2BUA -> Bob

INVITE sip:bob@example.org SIP/2.0
Via: SIP/2.0/UDP pc3.example.org:5060;branch=z9hG4bK7777
Max-Forwards: 70
From: Carol <sip:carol@example.org>;tag=777x
To: Bob <sip:bob@example.org>
6. Acknowledgements

The author wishes to thank Fan Hu, Jean Claude Le Rouzic, Paul Kyzivat, Thomas Leseney, and Youssef Chadli for their helpful discussions on this topic.

7. References


Appendix A. Summary of the actions done by the B2BUA

This appendix makes a summary of the different actions that are performed by the B2BUA upon receipt of an out-of-dialog request and upon receipt of the associated response.

Note that the B2BUA acts as a UAS on one side and as a UAC on the other, as defined in RFC3261 [2], but that additional details are added in order to copy as many information from the incoming request to the outgoing request in order to minimize the impact of the B2BUA.
<table>
<thead>
<tr>
<th>Header field</th>
<th>where</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow</td>
<td>R,r</td>
<td>If the B2BUA supports the allowed method, copy it else remove the method</td>
</tr>
<tr>
<td>Call-Id</td>
<td>R</td>
<td>Create a new Call-Id for outgoing request</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>Map Call-Id to corresponding incoming request</td>
</tr>
<tr>
<td>Contact</td>
<td>R</td>
<td>Create a new local contact for outgoing request[1]</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>Create a new local contact for incoming request[1]</td>
</tr>
<tr>
<td>Content-Type</td>
<td>R,r</td>
<td>If the body contains references to a mapped dialog then map dialog referenced within the body [4]</td>
</tr>
<tr>
<td>CSeq</td>
<td>R,r</td>
<td>May be copied</td>
</tr>
<tr>
<td>From</td>
<td>R</td>
<td>Should copy name-addr, may change name-addr if privacy is required, generate a new tag [2].</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>Copy name-addr, and map tag</td>
</tr>
<tr>
<td>In-Reply-To</td>
<td>R</td>
<td>If possible Map its Call-Id, else remove it</td>
</tr>
<tr>
<td>Join</td>
<td>R</td>
<td>Map Call-id and tags [6]</td>
</tr>
<tr>
<td>Max-Forwards</td>
<td>R</td>
<td>Reset Max-Forwards when creating requests [3]</td>
</tr>
<tr>
<td>To</td>
<td>R</td>
<td>Copy name-addr [2]</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>Copy name-addr, generate new tag.</td>
</tr>
<tr>
<td>Record-Route</td>
<td>R</td>
<td>Remove headers</td>
</tr>
<tr>
<td>Route</td>
<td>r</td>
<td>May copy [see 4.1.3]</td>
</tr>
<tr>
<td>RequestURI</td>
<td>R</td>
<td>hostname and uri-parameters may require action [5]</td>
</tr>
<tr>
<td>Replaces</td>
<td>R</td>
<td>Map Call-Id and tags. [6]</td>
</tr>
<tr>
<td>Required</td>
<td>R</td>
<td>If the B2BUA support the extension, copy it else remove the extension</td>
</tr>
<tr>
<td>Server</td>
<td>r</td>
<td>Replace the value with the B2BUA’s value</td>
</tr>
<tr>
<td>Supported</td>
<td>R,r</td>
<td>If the B2BUA supports the extension, copy it else remove the extension</td>
</tr>
<tr>
<td>Target-Dialog</td>
<td>R</td>
<td>Map Call-Id and tags [6]</td>
</tr>
<tr>
<td>User-Agent</td>
<td>R</td>
<td>Replace the value with the B2BUA’s value</td>
</tr>
<tr>
<td>Via</td>
<td>R</td>
<td>Create a new Via header field for outgoing request</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>Use Via from incoming request as per RFC3261</td>
</tr>
<tr>
<td>others</td>
<td>R,r</td>
<td>Should by default copy them</td>
</tr>
</tbody>
</table>

Table 1: Header modifications required by for out-of-dialog messages

R=request received by the B2BUA
r=response received by the B2BUA
[1] Contact header URI parameters may require mapping.
[2] URI parameters may require mapping. (?)
[3] If the request loops back to the B2BUA with the same target there is the potential for undetected loops. TODO: Can this be averted?
[5] References to the B2BUA hostname are mapped to the appropriate contact hostname. GRUU parameters require mapping. Others?
[6] Presence of these headers may require that the Request URI is also mapped to the Contact used in the referenced dialog.

Authors’ Addresses

Xavier Marjou
France Telecom
Rue Pierre Marzin
Lannion  22307
France

Email: xavier.marjou@orange-ftgroup.com

Ian Elz
Ericsson
New Century Park
Coventry  CV3 1JG
United Kingdom

Email: ian.elz@ericsson.com

Peter Musgrave
CounterPath
350 Legget Drive
Ottawa  K2K 2W7
Canada

Email: PMusgrave@newheights.com