

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 30, 2007

M. Procter
Citel.
February 26, 2007

**An approach to Call Park/Retrieve using SIP
draft-procter-bliss-call-park-extension-00**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 30, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Call Park and Call Retrieve are useful telephony services that are normally found on traditional PBXs. Implementing these services using the Session Initiation Protocol (SIP) in the way described in the SIP Service Examples draft [1] is straightforward, but suffers from a useability problem when implemented using SIP User Agents resembling traditional business telephones. This draft discusses a simple extension to cater for this style of endpoint.

1. Overview

When parking and retrieving a call, it is clearly important to be able to identify a parked call to allow subsequent retrieval. The approach described in [\[1\]](#) uses the SIP dialog ID between the parked endpoint and the park server itself. This dialog ID is unique, allocated by both the parked user and the Park Server, and also long. Mechanisms for transferring this identifier between the parking party and the retrieving party are outside the scope of [\[1\]](#), but given the nature of the dialog ID, transferring this information electronically is likely to be the only practical mechanism.

Traditional PBX users have become accustomed to parking a call against a short number (typically 3 or 4 digits), and then using this identifier to communicate to the retrieving party which call to retrieve. This information may be passed verbally, or by means of small paper notes. Whilst collisions may occur, they are generally avoided satisfactorily by administrative policies.

This draft attempts to reconcile these two models by allowing the parking party to specify a short tag to attach to the parked call (the 'orbit'). The retrieving party can then use the same tag to locate the relevant information to retrieve the parked call.

2. Call Park

This message flow of parking a call is identical to that illustrated in [\[1\]](#). The difference that this draft introduces is in the REFER message to the Park Server. The details of the REFER message changes are discussed below.

Alice	Bob	Park Server	Carol
	INVITE F1		
	----->		
	180 Ringing F2		
	<-----		
	200 OK F3		
	<-----		
	ACK F4		
	----->		
	RTP Media		
	<=====>		
	Bob Parks Call		
	REFER Refer-To: A F5		
	----->		
	202 F6		
	<-----		
	NOTIFY F7		
	<-----		
	200 F8		
	----->		
	INVITE F9 Replaces: B		
	<-----		
	200 OK F10		
	----->		
	ACK F11		
	<-----		
	RTP Music		
	<=====>		
	BYE F12		
	----->		
	NOTIFY F14		
	200 OK F13		
	<-----		
	200 OK F15		
	----->		

The URI <sips:park@server.example.com;orbit=1234> is used instead of directing the request to the URI <sips:park@server.example.com>. The addition of the orbit parameter effectively tags the parked call with a short memorable code entered by the user.

F5 REFER Bob -> Park Server

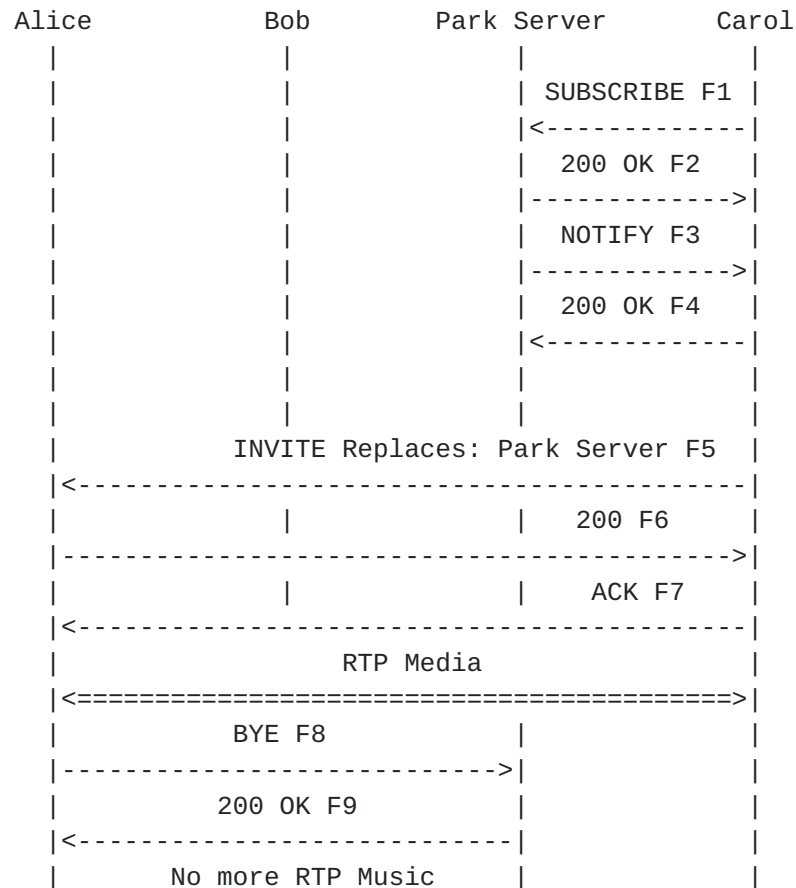
```
REFER sips:park@server.example.com;orbit=1234 SIP/2.0
Via: SIP/2.0/TLS client.biloxi.example.com:5061
    ;branch=z9hG4bKnashds9
Max-Forwards: 70
From: Bob <sips:bob@biloxi.example.com>;tag=02134
To: Park Server <sips:park@server.example.com;orbit=1234>
Call-ID: 4802029847@biloxi.example.com
CSeq: 1 REFER
<allOneLine>
Refer-To: <sips:alice@client.atlanta.example.com?Replaces=
12345601%40atlanta.example.com%3Bfrom-tag%3D314159
%3Bto-tag%3D1234567>
</allOneLine>
Referred-By: <sips:bob@biloxi.example.com>
Contact: <sips:bob@client.biloxi.example.com>
Content-Length: 0
```

[3.](#) Call Retrieve

In order to retrieve the call using the approach described in [\[1\]](#), we need to obtain the dialog identifiers, given only the orbit. In fact, [\[1\]](#) points us to the solution in this extract:

Note that if the Park Server did not return the dialog identifiers (Call-ID, To and From tags) in the NOTIFY, Carol could send a SUBSCRIBE to retrieve this information.

By subscribing to the dialog event package [\[2\]](#) at the same URI used for parking the call, i.e. <sips:park@server.example.com;orbit=1234>, all the information that is required for the call to be picked up by C is delivered in the corresponding NOTIFY.



F1 SUBSCRIBE Carol -> Park Server

```

SUBSCRIBE sips:park@server.example.com;orbit=1234 SIP/2.0
Via: SIP/2.0/TLS chicago.example.com:5061;branch=z9hG4bK92bz
Max-Forwards: 70
From: Carol <sips:carol@chicago.example.com>;tag=8672349
To: <sips:park@server.example.com;orbit=1234>
Call-ID: xt4653gs2ham@chicago.example.com
CSeq: 1 SUBSCRIBE
Contact: <sips:carol@client.chicago.example.com>
Event: dialog
Subscription-State: active;expires=0
Accept: application/dialog-info+xml
Content-Length: 0

```


F2 200 OK Park Server -> Carol

SIP/2.0 200 OK

Via: SIP/2.0/TLS chicago.example.com:5061;branch=z9hG4bK92bz
;received=192.0.2.114

Max-Forwards: 70

From: Carol <sips:carol@chicago.example.com>;tag=8672349

To: <sips:park@server.example.com;orbit=1234>;tag=1234567

Call-ID: xt4653gs2ham@chicago.example.com

CSeq: 1 SUBSCRIBE

Content-Length: 0

F3 NOTIFY Park Server -> Carol

NOTIFY sips:carol@client.chicago.example.com SIP/2.0

Via: SIP/2.0/TLS chicago.example.com:5061;branch=z9hG4bK93ca

Max-Forwards: 70

To: Carol <sips:carol@chicago.example.com>;tag=8672349

From: <sips:park@server.example.com;orbit=1234>;tag=1234567

Call-ID: xt4653gs2ham@chicago.example.com

CSeq: 2 NOTIFY

Contact: <sips:park@server.example.com;orbit=1234>

Event: dialog

Subscription-State: terminated

Content-Type: application/dialog-info+xml

Content-Length: ...

```
<?xml version="1.0"?>
```

```
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
  version="0" state="full"
```

```
  entity="sips:park@park.server.example.com;orbit=1234">
```

```
<dialog id="94992014524" call-id="12345600@atlanta.example.com"
```

```
  local-tag="3145678" remote-tag="1234567" direction="recipient"
```

```
  remote-uri="alice@atlanta.example.com"
```

```
  remote-target="alice@client.atlanta.example.com">
```

```
<state>confirmed</state>
```

```
</dialog>
```

```
</dialog-info>
```


F4 200 OK Carol -> Park Server

SIP/2.0 200 OK

Via: SIP/2.0/TLS chicago.example.com:5061;branch=z9hG4bK93ca

To: Carol <sips:carol@chicago.example.com>;tag=8672349

From: <sips:park@server.example.com;orbit=1234>;tag=1234567

Call-ID: xt4653gs2ham@chicago.example.com

CSeq: 2 NOTIFY

Contact: <sips:carol@client.chicago.example.com>

Content-Length: 0

The remainder of the frames are the same as the corresponding frames from [1], since the required dialog ID has been obtained through the SUBSCRIBE / NOTIFY cycle from the Park Server.

4. A failed attempt to park a call

If an attempt is made to park a call against an orbit that is already in use, then the park attempt may fail.

Alice	Bob	Park Server	Carol
INVITE F1			
----->			
180 Ringing F2			
<-----			
200 OK F3			
<-----			
ACK F4			
----->			
RTP Media			
<=====>			
Bob Parks Call			
	REFER Refer-To: A F5		
	----->		
	486 Busy Here		
	<-----		

Under these circumstances, Bob may choose to attempt to park the call again, but using a different orbit number.

5. Enforcing a policy to avoid orbit collisions

Sometimes an orbit number assignment policy needs to be implemented. This may be to ensure that all orbit numbers are a particular length, or have a form that means that they can be dialled directly (given

suitable extensions to an Application Server). It may also be implemented to eliminate the problem of trying to park more than one call on the same orbit.

To enforce a policy, we ensure that the orbit number is not allocated by the UA (entered by the user, or by configuration etc.) but is instead allocated by the Park Server, and relayed to the UA. The natural location for this information is for the Park Server to add the orbit parameter to the Contact header that it returns in the response to the REFER message.

Alice	Bob	Park Server	Carol
	INVITE F1		
	----->		
	180 Ringing F2		
	<-----		
	200 OK F3		
	<-----		
	ACK F4		
	----->		
	RTP Media		
	<=====		
	Bob Parks Call		
	REFER Refer-To: A F5		
	----->		
	202 Accepted F6		
	<-----		

F5 REFER Bob -> Park Server

```

REFER sips:park@server.example.com SIP/2.0
Via: SIP/2.0/TLS client.biloxi.example.com:5061
    ;branch=z9hG4bKnashdsB
Max-Forwards: 70
From: Bob <sips:bob@biloxi.example.com>;tag=22134
To: Park Server <sips:park@server.example.com>
Call-ID: 4802029847@biloxi.example.com
CSeq: 1 REFER
<allOneLine>
  Refer-To: <sips:alice@client.atlanta.example.com?Replaces=
    12345601%40atlanta.example.com%3Bfrom-tag%3D314159
    %3Bto-tag%3D1234567>
</allOneLine>
Referred-By: <sips:bob@biloxi.example.com>
Contact: <sips:bob@client.biloxi.example.com>
Content-Length: 0

```


F6 202 Accepted Park Server -> Bob

SIP/2.0 202 Accepted

Via: SIP/2.0/TLS client.biloxi.example.com:5061

;branch=z9hG4bKnashdsB

;received=192.0.2.105

From: Bob <sips:bob@biloxi.example.com>;tag=22134

To: Park Server <sips:park@server.example.com>;tag=56324

Call-ID: 4802029848@biloxi.example.com

CSeq: 1 REFER

Contact: <sips:park@server.example.com;orbit=7001>

Content-Length: 0

This approach is analogous to the Conference Factory described in [3], as it permits a single configurable value (the URI of the Park Server) to be used by multiple UAs to provide unique parking orbits.

6. Security Considerations

None.

7. References

- [1] Johnston, A., "Session Initiation Protocol Service Examples", [draft-ietf-sipping-service-examples-12](#) (work in progress), January 2007.
- [2] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", [RFC 4235](#), November 2005.
- [3] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", [BCP 119](#), [RFC 4579](#), August 2006.

Author's Address

Michael Procter
Citel.
Wheatcroft Business Park
Landmere Lane, Edwalton
Nottingham NG12 4DG
UK

Email: michael.procter@citel.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

