

SIP WG
Internet-Draft
Intended status: Standards Track
Expires: September 5, 2007

R. Mahy
Plantronics
C. Jennings
Cisco Systems
March 4, 2007

**Remote Call Control in the Session Initiation Protocol (SIP) using the
REFER method and the session-oriented dialog package
draft-mahy-sip-remote-cc-05**

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/lid-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 September 5, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes how to use the SIP REFER method and the dialog package to manipulate conversations, dialogs, and sessions on remote User Agents. Specifically it extends the REFER mechanism to allow the specification of a response that a UA should send in a dialog. This functionality is most useful for collections of loosely coupled User Agents that wish to present a coordinated user

experience. It does not require a Third-Party Call Control controller to be involved in any of the manipulated dialogs.

Table of Contents

1.	Conventions	3
2.	Introduction	3
3.	Examples of Remote Call Control Operations	4
4.	User Agent Behavior	8
4.1.	Organizing requests within dialogs	8
4.2.	Addressing the relevant parties	10
4.3.	Selecting an existing dialog context for the triggered request	11
4.4.	Placing call on/off hold	11
4.5.	Remote Call Control Primitives	12
5.	Formal Syntax	12
6.	Security Considerations	13
6.1.	Authorizing remote call control requests	13
7.	IANA Considerations	13
8.	Acknowledgments	13
9.	References	13
9.1.	Normative References	13
9.2.	Informational References	14
	Authors' Addresses	15
	Intellectual Property and Copyright Statements	16

1. Conventions

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 [RFC-2119](#) [2].

To simplify discussions related to the REFER method and its extensions, three new terms will be used:

REFER-Issuer: the UA issuing the REFER request. Sometimes this document will also use the term "controller".

REFER-Recipient: the UA receiving the REFER request

REFER-Target: the UA designated in the Refer-To URI

2. Introduction

The SIP [1] core protocol describes how User Agents originate and terminate sessions. Remote call control is the manipulation of conversations and session-oriented dialogs by a UA that is not directly involved in any of the relevant conversations, dialogs, or sessions. This manipulation generally involves sending REFER [4] requests to a UA which is directly involved, using information obtained via the dialog package [5]. (Although many are familiar with REFER only as used to implement call transfer [11], the authors of the REFER method never intended this limitation. In fact the REFER method was created when the SIP working group realized that a generic request to ask another UA to do something on your behalf was much more powerful than just doing transfers.)

For convenience we can describe the SIP entity that sends requests to manipulate remote sessions "the controller", but this is just a logical role. A UA that acts as a controller for one request can terminate and originate its own sessions, and even receive remote call control requests as other requests.

Remote call control is especially useful for collections of loosely coupled User Agents which would like to present a coordinated user experience. Among other things, this allows User Agents which handle orthogonal media types but which would like to be present in a single conversation to add and remove each other from the conversation as needed. This is especially appropriate when coordinating conversations among organizers, general purpose computers, and special purpose communications appliances like telephones, Internet televisions, in-room video systems, electronic whiteboards, and gaming devices.

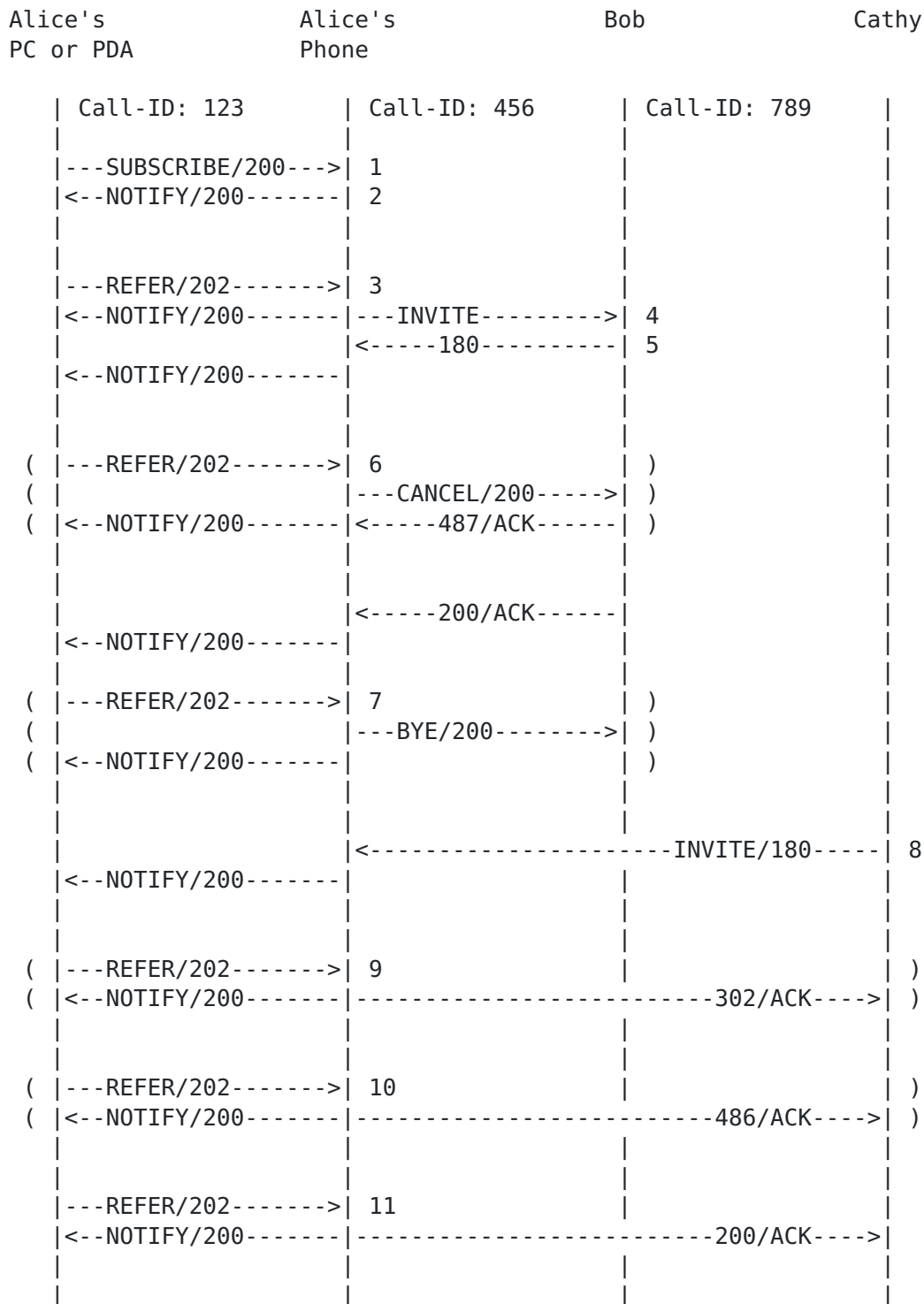
For example using remote call control, an Instant Messaging client could initiate a multiplayer gaming session and an audio session to a

chat conversation. Likewise a telephone could add an electronic whiteboard session to a voice conversation. Finally, a computer or organizer could cause a nearby phone to dial from numbers or URIs in a document, email, or address book; allow users to answer or deflect incoming calls without removing hands from the computer keyboard; place calls on hold; and join other sessions on the phone or otherwise.

3. Examples of Remote Call Control Operations

This entire section provide non-normative examples of functionality where a computer or PDA manipulates a telephone. The behavior for remote call control with other types of devices is similar, but describing similar manipulations for other media or device types would naturally use a different set of vocabulary.

In the requests labeled with 1 and 2, Alice's PC or PDA sets up a subscription to the dialog package from Alice's phone (messages shown in a later section). All of the subsequent NOTIFY messages are notifications about changes in the dialog state at Alice's phone. In message 3, Alice's PC or PDA asks her phone to "call Bob" (message 4), which eventually results in an early dialog (5) with one of Bob's Contacts. [Note Well: Parts of the flow marked in parentheses (including messages 6, 7, 9, and 10) show alternative outcomes in the call flow.]



Messages 3, 4, and 5 follow. The norefersub option-tag on each REFER suppresses the implicit subscription which would normally follow the

REFER (the notifications in the call flow diagram are for the dialog package subscription in messages 1 and 2).

Via and Max-Forward headers and session descriptions are omitted for brevity and clarity. In some cases, display names are added for simplify the task of the reader following the examples. Note that URIs in SIP cannot wrap lines. Due to RFC formatting conventions, this draft splits URIs across lines where the URI would exceed 72 characters. A backslash character marks where this line folding has taken place. Finally, some of the URIs shown here are not escaped properly to aid in readability. In message 9 the @ in the Refer-To URI should be escaped.

Message 3:

```
REFER sip:reg2@192.0.2.3 SIP/2.0
To: "Alice's phone" <sip:reg2@192.0.2.3>;tag=def
From: "Alice's PC or PDA" <sip:alice1@192.0.2.2>;tag=abc
Call-ID: 123
CSeq: 2 REFER
Require: remotecc
Supported: norefersub
Refer-To: "Bob" <sip:bob@example.net>
```

Message 4:

```
INVITE sip:bob@example.net SIP/2.0
To: "Bob" <sip:bob@example.net>
From: "Alice" <sip:alice@example.com>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Alice's Phone" <sip:reg2@192.0.2.3>
Content-Type: application/sdp
Content-Length: xxx
```

Message 5:

```
SIP/2.0 180 Ringing
To: "Bob" <sip:bob@example.net>;tag=uvw
From: "Alice's phone" <sip:reg2@192.0.2.3>;tag=xyz
Call-ID: 456
CSeq: 1 INVITE
Contact: "Bob's Contact" <sip:line1@192.0.2.5>
Content-Type: application/sdp
Content-Length: xxx
```

The rest of the REFER messages in this example are identical to the REFER in Message 3 except for the Refer-To header, Target-Dialog [\[6\]](#)

header, CSeq header, and Via branch id (not shown). Message fragment 6 and 7 show the Refer-To headers which Alice's PC or PDA would send to cause Alice's phone to terminate the session which Message 4 attempted to originate. The parameters in the Target-Dialog header are used to explicitly match a specific dialog. They are more fully described in a later section.

Headers for Message 6:

```
Refer-To: "Bob's Contact" <sip:line1@192.0.2.5;method=CANCEL>  
Target-Dialog: 456;remote-tag=;local-tag=xyz
```

Header for Message 7:

```
Refer-To: "Bob's Contact" <sip:line1@192.0.2.5;method=BYE>  
Target-Dialog: 456;remote-tag=uvw;local-tag=xyz
```

Message 8 is an invitation received by Alice's phone from Cathy. Refer-To headers for Messages 9, 10, and 11 are shown below which would cause Alice's phone to redirect, reject, or accept the invitation. They use the response URI parameter defined in [7]. Note that some specialized User Agents might not be capable of accepting an invitation autonomously. For example, a SIP user agent which connect to an analog telephone cannot physically force the phone to go offhook.

Message 8:

```
INVITE sip:reg2@192.0.2.3 SIP/2.0
To: "Alice" <sip:alice@example.com>
From: "Cathy" <sip:cathy@example.net>;tag=ijk
Call-ID: 789
CSeq: 1 INVITE
Contact: "Cathy's Contact" <sip:cathy-pc.example.net>
Content-Type: application/sdp
Content-Length: xxx
```

Header for 9:

```
Refer-To: <sip:cathy-pc.example.net;method=INVITE;response=302 \
?Contact=sip:doug@example.com>
Target-Dialog: 789;remote-tag=ijk;local-tag=lmn
```

Header for 10:

```
Refer-To: <sip:cathy-pc.example.net;method=INVITE;response=486>
Target-Dialog: 789;remote-tag=ijk;local-tag=lmn
```

Header for 11:

```
Refer-To: <sip:cathy-pc.example.net;method=INVITE;response=200>
Target-Dialog: 789;remote-tag=ijk;local-tag=lmn
```

4. User Agent Behavior

4.1. Organizing requests within dialogs

REFER messages used for call transfer typically arrive within an existing dialog which was created with the INVITE method. In general, REFER messages can be sent within an existing dialog, or they can start a new dialog (the dialog used by the implicit subscription they create). In many use cases of remote call control, receiving notifications about the status of a REFER request are superfluous, as the Refer-Issuer typically maintains a long duration subscription to the dialog package. This situation is complicated by the possible presence of the norefersub option-tag, defined in section 7 of [7]. When the norefersub option tag is present, a REFER request which would have created a new subscription and dialog becomes a standalone transaction instead. Each such standalone REFER transaction MUST use a new (unique) Call-Id header field value. The following three use cases are suggested:

1. In the most common usage, the controller maintains a long duration subscription to the dialog package, and sends REFER requests within that dialog. Each REFER is sent within the context of the dialog created for the subscription to the dialog package, and could include the norefersub option-tag in a Supported header field value.
2. Occasionally the dialog package is only supported via a dialog state agent separate from the Refer-Receiver, in which case the controller maintains a long duration subscription to the dialog package to a dialog state agent, and the controller sends these individual REFER requests as standalone requests each with a different (unique) Call-ID header field value, which could also include the norefersub option-tag in a Supported header field value.
3. In some cases, the controller does not typically maintain a dialog package subscription for the Refer-Receiver. This might be the case for a "webdialer" or other application which associates with other UAs on an adhoc and intermitent basis. An initial REFER request is sent to start a new dialog, which is followed by notifications for the refer event type (the norefersub option-tag SHOULD NOT be used in this case). These notifications could contain message/sipfrag or application/dialog-info+xml notification bodies as described in Section 4 of [7].

Message 1:

```
SUBSCRIBE sip:reg2@192.0.2.3 SIP/2.0
To: "Alice's phone" <sip:reg2@192.0.2.3>
From: "Alice's PC or PDA" <sip:alice1@192.0.2.2>;tag=abc
Call-ID: 123
CSeq: 1 SUBSCRIBE
Event: dialog
Contact: <sip:alice1@192.0.2.2>
```

Message 2:

```
NOTIFY sip:reg2@192.0.2.3 SIP/2.0
To: "Alice's PC or PDA" <sip:alice1@192.0.2.2>;tag=abc
From: "Alice's phone" <sip:reg2@192.0.2.3>;tag=def
Call-ID: 123
CSeq: 1 NOTIFY
Event: dialog
Contact: <sip:reg2@192.0.2.3>
Subscription-State: active;expires=3600
Content-Type: application/dialog-info+xml
Content-Length: xxx
```


4.2. Addressing the relevant parties

REFER requests contain a number of URIs which need to address the appropriate parties. A list of the relevant fields include the Request-URI, To header URI, From header URI, Contact header URI, Refer-To header URI, and the Referred-By header URI. This section attempts to clarify what needs to be placed in each field.

In most cases, remote call control seeks to manipulate dialogs or sessions on a specific UA. For this reason, the Request URI of the REFER request SHOULD be a valid Globally Routeable User-agent URI (GRUU) [8] for a single UA (a Contact URI). Contact URIs for a UA can be discovered by subscribing to the registration package for the relevant AORs.

In the rare exceptions when the controller does not care which specific UA it manipulates, an AOR MAY be used instead. When an AOR is used, the REFER request can include appropriate caller-preferences to encourage selection of an appropriate Contact. The norefersub option-tag MUST NOT be used when the REFER Request-URI is an AOR, as the REFER Request could fork and cause very odd behavior. While, the controller can discourage a proxy from forking remote call control request by using the Request-Disposition: no-fork header field, insuring that no proxy forks requires the use of the callerpref option-tag in a Proxy-Require header field value. Because any proxy in the chain of this request which did not support caller preferences would cause the request to fail, use of Proxy-Require is NOT RECOMMENDED.

For remote call control requests to operate as expected, the Refer-Issuer needs to be confident that the Refer-Receiver supports the extensions and conventions described here. Otherwise, the triggered request might have completely different semantics from the request which was indicated in the Refer-To header. (Most implementations ignore unknown URI and header parameters). For example a REFER intended to cause the Refer-Receiver to send a 486 Busy Here response for an existing dialog, might instead trigger a new INVITE to the sender of the original INVITE. Implementations which send remote call control requests MUST include the remotecc option-tag in a Require header field value in each REFER request. (Note that support for this option-tag also implies support for the response URI parameter in a Refer-To header.)

The To header field in the REFER request should contain the same URI as in the Request-URI, and the From identifies the AOR of the controller. The Refer-To is set to whatever URI would normally appear in the triggered request if the request were initiated autonomously by the Refer-Receiver. A REFER triggering a standalone

request or dialog starting request, could send to either an AOR or a Contact address, but typically to an AOR. A REFER request triggering a request which is in a dialog MUST always place a Contact URI in the Refer-To header.

4.3. Selecting an existing dialog context for the triggered request

Many uses of remote call control require that the Refer-Receiver generate a new request or response in the context of an existing dialog. For example, the controller might want the Refer-Receiver to send a BYE, CANCEL, or response to an INVITE in the context of a dialog created with INVITE. For subscriptions, the controller might want the Refer-Receiver to unsubscribe (send a SUBSCRIBE with an Expires header field of 0).

To select the appropriate dialog from which to source the request, the Target-Dialog header specified in [6] MUST be used.

4.4. Placing call on/off hold

A common usage of remote call control is to request the Refer-Receiver to place an existing call on or off hold. While the concept of hold used in telephony does not map perfectly to SIP, the normal usage of placing a call on hold is to change all sendrecv media streams to inactive, and all sendonly media streams to inactive. This could be done by escaping an SDP offer into the Refer-To URI with the "body=" parameter. However, this document proposes using instead the +sip.rendering feature tag [5] and a new URI parameter "featuretag" used to convey a SIP media feature tag [9] associated with this URI.

Note that this mechanism for conveying feature tags is different from the mechanism defined in [10]. The feature tag conveyed in that mechanism is the feature tag of the Refer-Target, not the Refer-Receiver.

For example, to place Bob on hold, the following Refer-To header field could be used:

```
Refer-To: <sip:bob@192.0.2.5;method=INVITE \
;featuretag=+sip.rendering%3Dno>
Target-Dialog: 456;remote-tag=;local-tag=xyz
```

To place Bob off hold, the following Refer-To header field could be used:


```
Refer-To: <sip:bob@192.0.2.5;method=INVITE \
;featuretag+=sip.rendering%3Dyes>
Target-Dialog: 456;remote-tag=;local-tag=xyz
```

4.5. Remote Call Control Primitives

The following remote call control primitives are enabled by the conventions described in this document:

- Initiate a new session
- Terminate a session
- Reject an alerting session
- Deflect an alerting session
- Answer an alerting session
- Single step transfer of session
- Completing a transfer between sessions

The following remote call control primitives are not enabled by any functionality described in this document. A complete solution to remote call control will require that these primitives are addressed as well:

- Move a session to the foreground
- Move a session to the background
- Merge sessions

The concept of foreground and background requires additional explanation. In an audio context, typically only session is in the "foreground" while all other sessions are "on hold" in the background. In a video or text context, the concept of foreground and background refers to "focus" in the user interface. There is currently no proposal for how to request sessions move to the foreground or background. A few options worth considering are a) a pair of new methods, or b) a PUBLISH request sent with a new event type and body.

It is possible to Single step transfer or Complete a transfer between a target and a SIP conferencing [[12](#)] User Agent. However, the target User Agent may be capable of local mixing, or it may be configured to use a specific conference server. As with the Join header defined in [RFC3911](#) [[13](#)], it is desirable to ask the User Agent to use whatever conferencing behavior it would if it was locally invoked.

5. Formal Syntax

The following syntax specification extends the SIP URI ABNF in

[RFC3261](#) to have a response URI tag using the augmented Backus-Naur Form (BNF) as described in [RFC 4234](#) [3]. Note that the "/"= notation used in this ABNF indicates an extension of the production on the left-hand side.

```
uri-parameter      /= response-param / featuretag-param
```

```
response-param     = "response=" 3DIGIT
```

```
featuretag-param   = "featuretag=" feature-param
```

Where feature-param is defined in [9].

6. Security Considerations

The functionality described in this document allows an authorized party to manipulate SIP sessions and dialogs in arbitrary ways. Any user agent that accepts these types of requests needs to be very careful in who it authorizes to send these types of requests.

6.1. Authorizing remote call control requests

It is RECOMMENDED that for any refer that includes a response, if a user agent registers with the address-of-record X, that this user agent authorize subscriptions that come from any entity that can authenticate itself as X. This can be done by using Digest Authentication.

7. IANA Considerations

Need to register the SIP URI parameter specified in [Section 5](#) and the 'remotecc' option-tag.

8. Acknowledgments

Many thanks to Sean Olson, Orit Levin, Robert Sparks, and Alan Johnston.

9. References

9.1. Normative References

- [1] 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.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
 - [3] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), October 2005.
 - [4] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", [RFC 3515](#), April 2003.
 - [5] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", [RFC 4235](#), November 2005.
 - [6] Rosenberg, J., "Request Authorization through Dialog Identification in the Session Initiation Protocol (SIP)", [RFC 4538](#), June 2006.
 - [7] Levin, O., "Suppression of Session Initiation Protocol (SIP) REFER Method Implicit Subscription", [RFC 4488](#), May 2006.
 - [8] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", [draft-ietf-sip-gruu-11](#) (work in progress), October 2006.
 - [9] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [RFC 3840](#), August 2004.
 - [10] Levin, O. and A. Johnston, "Conveying Feature Tags with the Session Initiation Protocol (SIP) REFER Method", [RFC 4508](#), May 2006.

[9.2.](#) Informational References

- [11] Sparks, R., "Session Initiation Protocol Call Control - Transfer", [draft-ietf-sipping-cc-transfer-07](#) (work in progress), October 2006.
- [12] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", [RFC 4353](#), February 2006.
- [13] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header", [RFC 3911](#), October 2004.

Authors' Addresses

Rohan Mahy
Plantronics
345 Encinal Street
Santa Cruz, CA
USA

Email: rohan@ekabal.com

Cullen Jennings
Cisco Systems
170 West Tasman Drive
Mailstop SJC-21/2
San Jose, CA 95134
USA

Phone: +1 408 902-3341
Email: fluffy@cisco.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).

