

SIPREC
Internet-Draft
Intended status: Informational
Expires: April 23, 2011

A. Hutton
Siemens Enterprise Communications
L. Portman
Nice Systems
R. Jain
IPC Systems
K. Rehor
Cisco Systems, Inc.
October 20, 2010

**An Architecture for Media Recording using the Session Initiation
Protocol
draft-ietf-siprec-architecture-01**

Abstract

Session recording is a critical requirement in many communications environments such as call centers and financial trading. In some of these environments, all calls must be recorded for regulatory, compliance, and consumer protection reasons. Recording of a session is typically performed by sending a copy of a media stream to a recording device. This document describes architectures for deploying session recording solutions in an environment which is based on the Session Initiation Protocol (SIP).

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, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Terminology	3
2.	Definitions	3
3.	Introduction	4
4.	Session Recording Architecture	5
4.1.	Location of the Session Recording Client	5
4.1.1.	B2BUA acts as a Session Recording Client	5
4.1.2.	Endpoint acts as Session Recording Client	6
4.1.3.	A SIP Proxy cannot be a Session Recording Client	7
4.1.4.	Interaction with MEDIACTRL	7
4.1.5.	Interaction with Conference Focus	8
4.2.	Establishing the Recording Session	9
4.2.1.	Session Recording Client Initiated Recording	9
4.2.2.	Session Recording Server Initiated Recording	10
4.2.3.	Pause/Resume Recording Session	10
4.2.4.	Media Stream Mixing	10
4.2.5.	Media Transcoding	11
4.3.	Media Recording Metadata	11
4.3.1.	Contents of recording metadata	11
4.3.2.	Mechanisms for delivery of metadata to Session Recording Server	11
4.4.	Notifications to the Recorded User Agents	12
4.5.	Preventing the recording of a SIP session	12
5.	IANA considerations	13
6.	Security considerations	13
7.	References	13
7.1.	Normative References	13
7.2.	Informative References	14
	Authors' Addresses	15

1. 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](#)] and indicate requirement levels for compliant mechanisms.

2. Definitions

Session Recording Server (SRS): A Session Recording Server (SRS) is a SIP User Agent (UA) that acts as the sink of the recorded media. An SRS is a logical function that typically archives media for extended durations of time and provides interfaces for search and retrieval of the archived media. An SRS is typically implemented as a multi-port device that is capable of receiving media from several sources simultaneously. An SRS is typically also the sink of the recorded session metadata. Note that the term "Server" does not imply the SRS is the server side of a signaling protocol - the SRS may be the initiator of recording requests, for example.

Session Recording Client (SRC): A Session Recording Client (SRC) is a SIP User Agent (UA) that acts as the source of the recorded media, sending it to the Session Recording Server. A Session Recording Client is a logical function. Its capabilities may be implemented across one or more physical devices. In practice, a Session Recording Client could be a personal device (such as a SIP phone), a SIP Media Gateway (MG), a Session Border Controller (SBC), Media Server, or an Application Server. The Session Recording Client is also the source of the recorded session metadata.

Communication Session (CS): A SIP session created between two or more UA's for the purpose of communication which may be recorded.

Recording Session (RS): The session created between an Session Recording Client and Session Recording Server for the purpose of recording a Communication Session.

Recording aware UA: A SIP User Agent that can at a minimum understand notifications indicating that a Communication Session in which it is involved is being recorded. It may also be able to express preferences relating to whether a Communication session can or should be recorded.

Media Recording Metadata (MRM): The metadata describing the communication session that is required by the Session Recording Server. This will include for example the identity of users that participate in the Communication Session and dialog state. Typically

this metadata is archived with the replicated media at the Session Recording Server. The media recording metadata is delivered in real-time to the Session Recording Server.

Replicated Media: A copy of the media associated with the Communication Session created by the Session Recording Client and sent to the Session Recording Server. It may contain all the media associated with the communication session (E.g. Audio and Video) or just a subset (E.g. Audio).

3. Introduction

Session recording is a critical requirement in many communications environments such as call centers and financial trading. In some of these environments, all calls must be recorded for regulatory, compliance, and consumer protection reasons. Recording of a session is typically performed by sending a copy of a media stream to a recording device. This document describes architectures for deploying session recording solutions in an environment which is based on the Session Initiation Protocol (SIP) the requirements for which are described in [[I-D.ietf-siprec-req](#)].

This document focuses on how sessions are established between a Session Recording Client (SRC) and the Session Recording Server (SRS) for the purpose of conveying the Replicated Media and Media Recording Metadata (e.g. Identity of parties involved) relating to the Communication Session.

Once the Replicated Media and Media Recording Metadata have been received by the Session Recording Server they will typically be archived for retrieval at a later time. The procedures relating to the archiving and retrieval of this information is outside the scope of this document.

This document only considers active recording, where the Session Recording Client purposefully streams media to a Session Recording Server. Passive recording, where a recording device detects media directly from the network (E.g. using port mirroring techniques), is outside the scope of this document. In addition, lawful intercept is outside the scope of this document which takes account of the IETF policy on wiretapping [[RFC2804](#)].

The Recording Session that is established between the Session Recording Client and the Session Recording Server uses the normal procedures for establishing INVITE initiated dialogs as specified in [[RFC3261](#)] and uses SDP for describing the media to be used during the session as specified in [[RFC4566](#)]. However it is intended that some

extensions to SIP (E.g. Headers, Option Tags, Etc.) will be defined to support the requirements for media recording. The Replicated Media is required to be sent in real-time to the Session Recording Server and is not buffered by the Session Recording Client to allow for real-time analysis of the media by the Session Recording Server.

4. Session Recording Architecture

4.1. Location of the Session Recording Client

This section contains some example session recording architectures showing how the Session Recording Client is a logical function that can be located in or split between various physical components.

4.1.1. B2BUA acts as a Session Recording Client

A SIP B2BUA which has access to the media that is to be recorded may act as a Session Recording Client. The B2BUA may already be aware that a session needs to be recorded before the initial establishment of the communication session or the decision to record the session may occur after the session has been established.

If the B2BUA/SRC makes the decision to initiate the Recording Session (RS) then it will initiate the establishment of a SIP Session by sending an INVITE to the Session Recording Server.

If the Session Recording Server makes the decision to initiate the recording session then it will initiate the establishment of a SIP Session by sending an INVITE to the B2BUA/Session Recording Client.

The RS INVITE will need to contain information which identifies the session as being established for the purposes of recording and prevents the session from being accidentally rerouted to a UA which is not a SRS.

The B2BUA/SRC is responsible for notifying the UA's involved in the communication session that the session is being recorded.

The B2BUA/SRC is responsible for honoring any indication from recording aware UA's or through some configured policies that the communication session must not be recorded.

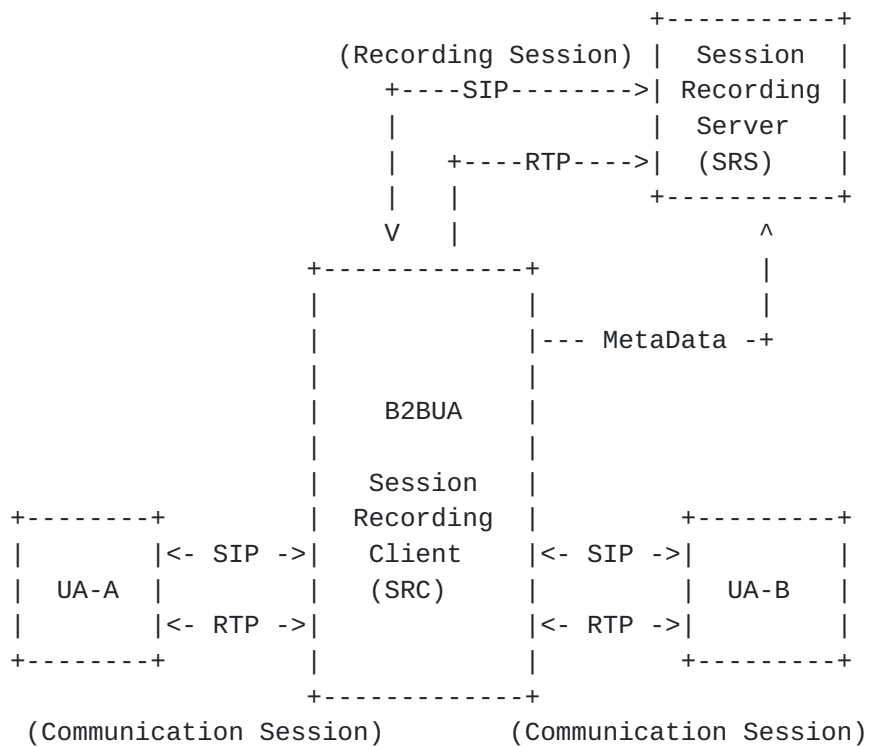


Figure 1: B2BUA Acts as the Session Recording Client.

4.1.2. Endpoint acts as Session Recording Client

A SIP Endpoint / User Agent may act as a Session Recording Client in which case the endpoint sends the Replicated Media to the Session Recording Server

If the endpoint makes the decision to initiate the Recording Session then it will initiate the establishment of a SIP Session by sending an INVITE to the Session Recording Server.

If the Session Recording Server makes the decision to initiate the Recording Session then it will initiate the establishment of a SIP Session by sending an INVITE to the endpoint.

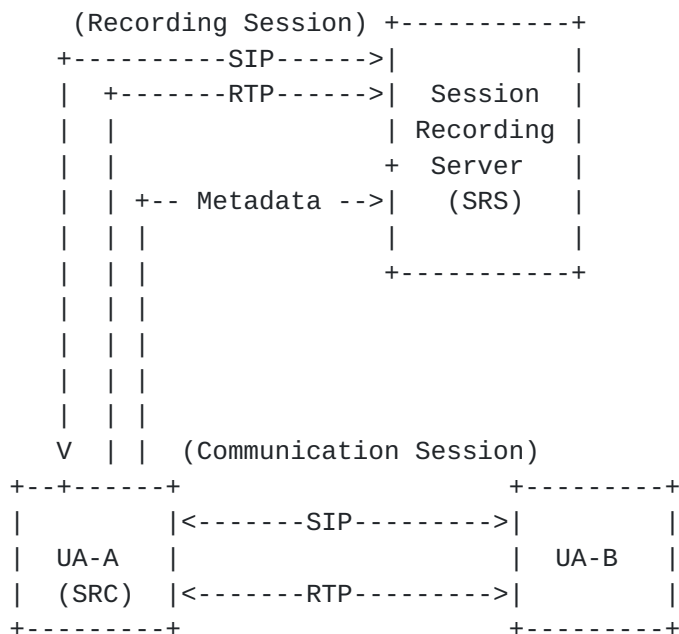


Figure 2: SIP Endpoint acts as the Session Recording Client

4.1.3. A SIP Proxy cannot be a Session Recording Client

A SIP Proxy is unable to act as an SRC because it does not have access to the media and therefore has no way of enabling the delivery of the replicated media to the SRS

4.1.4. Interaction with MEDIACTRL

The mediactrl architecture [[RFC5567](#)] describes an architecture in which an application server (AS) controls a Media Server (MS) which may be used for purposes such as conferencing and recording media streams. In this architecture the AS typically uses SIP Third Party Call Control (3PCC) to instruct the SIP UA's to direct the media to the Media Server.

The Session Recording Client and Session Recording Server described in this document may act as an application server as described in [[RFC5567](#)] and therefore may when further decomposed be made up of an application server which uses a mediactrl interface to control a media server for the purpose of recording the media streams however this interface is considered outside the scope of this document.

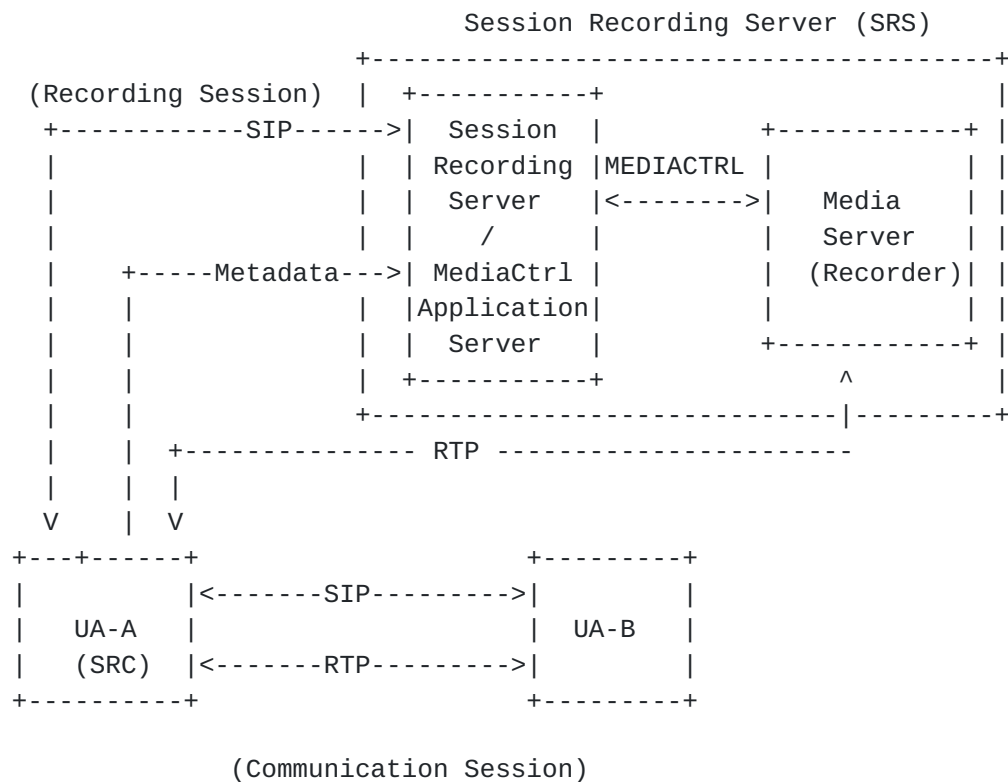


Figure 3: Example of Session Recording Server using MEDIACtrl

4.1.5. Interaction with Conference Focus

In the case of a centralised conference a combination of the conference focus and mixer [RFC4353] may act as a SRC and therefore provide the SRS with the replicated media and associated media recording metadata. In this arrangement the SRC is able to provide media and metadata relating to each of the participants, including, for example, any side conversations where the media passes through the mixer

The conference focus may also act as a Recording Aware UA in the case when one of the participants acts as a SRC.

In an alternative arrangement a SIP endpoint which is as a conference participant can act as an SRC. The SRC will in this case have access to the media and metadata relating to that particular participant and may be able to obtain additional metadata from the conference focus. The SRC may for example use the conference event package as described in [RFC4575] or [I-D.ietf-xcon-event-package] to obtain information about other participants which it provides to the SRS within the media recording metadata.

The SRC may be involved in the conference from the very beginning or may join at some later point of time.

4.2. Establishing the Recording Session

The Session Recording Client or the Session Recording Server may initiate the Recording Session.

It should be noted that the Recording Session is a completely independent from the Communication Session that is being recorded at both the SIP dialog level and at the session level. For example if media encryption is used for the Communication Session the Session Recording Client must decrypt any media received on the Communication Session and, if required, re-encrypt the media using a separate SRTP key for the Recording Session before sending the media to the Session Recording Server.

Concerning media negotiation, regular SIP/SDP capabilities should be used, and existing transcoding capabilities should not be precluded

4.2.1. Session Recording Client Initiated Recording

When the Session Recording Client initiates the Recording Session for the purpose of conveying media to the Session Recording Server it performs the following actions.

- o The SRC is provisioned with a URI for the SRS, which is resolved through normal [[RFC3263](#)] procedures.
- o Initiates the dialog by sending an INVITE request to the Session Recording Server. The dialog is established according to the normal procedures for establishing an INVITE initiated dialog as specified in [[RFC3261](#)].
- o Include in the INVITE an indication that the session is established for the purpose of recording the associated media. Possible mechanisms for this include using the Require header and/or a new media feature tag similar to the use of "isfocus" as defined in [[RFC3840](#)]
- o Indicate support for the recording metadata event package.
- o If the Replicated Media is to be started immediately then the Session Recording Client will include an SDP attribute of "a=sendonly" for each media line or "a=inactive" if it is not ready to transmit the media.

- o The Recording Session may replicate all media associated with the Communication Session or only a subset.
- o Replicate the media streams that are to be recorded and transmit the media to the Session Recording Server.

4.2.2. Session Recording Server Initiated Recording

When the Session Recording Server initiates the media recording session with the Session Recording Client it performs the following actions.

- o Send an INVITE request to the Session Recording Client
- o Include in the INVITE an indication that the session is established for the purpose of recording the associated media. Possible mechanisms for this include using the Require header or a media feature tag similar to the use of "isfocus" as defined in [[RFC3840](#)]
- o Identify the session that is to be recorded - Possibly using the Join header [[RFC3911](#)]
- o If the Recording Session is to be started immediately then the Session Recording Client will include an SDP attribute of "a=recvonly" for each media line or "a=inactive" if it is not ready to receive the media

If the Session Recording Server does not have prior knowledge of what media streams are available to be recorded it can make use of an offerless INVITE which allows the Session Recording Client to make the initial SDP offer.

4.2.3. Pause/Resume Recording Session

The Session Recording Server or the Session Recording Client may pause the recording by changing the SDP direction attribute to "inactive" and resume the recording by changing the direction back to "sendonly" or "recvonly"

4.2.4. Media Stream Mixing

In a basic session involving only audio there are typically two audio/RTP streams between the two UA's involved transporting media in each direction. When recording this media the two streams may be mixed at the SRC before being transmitted to the SRS or it may be a requirement of the recording server that the media streams are not mixed and are sent to the SRS as two separate streams. The case when

media is mixed at the SRC is simple as only a single media stream is required to be sent to the SRS. However in the case when the media streams are not mixed then the SDP offer sent to the SRS must describe two separate media streams.

4.2.5. Media Transcoding

The communication session (CS) and the recording session (RS) are negotiated separately using a standard SDP offer/answer exchange which may result in the SRC having to perform media transcoding between the two sessions. If the SRC is not capable of performing media transcoding it may limit the media formats in the offer to the SRS depending on what media is negotiated on the CS or may limit what it includes in the offer on the CS if it has prior knowledge of the media formats supported by the SRS. However typically the SRS will be the more capable device which can provide a wide range of media format options to the SRC and may also be able to make use of a media transcoder as detailed in [[RFC5369](#)].

4.3. Media Recording Metadata

4.3.1. Contents of recording metadata

The content of the recording metadata will be defined in a separate specification and therefore the following list is just a guide to the type of information that may be conveyed by the Session Recording Client to the Session Recording Server in the recording metadata.

- o Dialog identifiers for the Communication Session
- o Identities of SIP UA's taking part in the Communication Session
- o Dialog state of the Communication Session
- o Session state relating to the Communication Session(i.e. sendonly, inactive, sendrecv).
- o Etc.

4.3.2. Mechanisms for delivery of metadata to Session Recording Server

The SRS obtains session recording metadata from the SRC. There are several potential sources of information received from the SRC that can be used by the SRS to obtain recording metadata. These mechanisms include the following:

- o SIP Headers: Metadata may be obtained from SIP Headers received within the Recording Session
- o SDP: Metadata may be obtained from within SDP [[RFC4566](#)]
- o Event Package: The metadata may be conveyed from the SRC to SRS by defining an event-package and using it with the SIP events notification mechanism [[RFC3265](#)]. This allows the metadata to be communicated within its own SUBSCRIBE initiated dialog as opposed to being within the INVITE-initiated dialog of the Recording Session. This allows for sending metadata where the CS and RS lifecycles are different.
- o INFO Package: The metadata may be conveyed from SRC to SRS by defining an INFO package [[I-D.ietf-sipcore-info-events](#)]. Since INFO is a mid-dialog request, this mechanism requires the metadata to be communicated within the SIP INVITE RS dialog.
- o Using Non-SIP Mechanism: The recording metadata may be conveyed using some other non-SIP means (e.g. HTTP, WS-*).

OPEN ISSUE: SIPREC may need to choose a mechanism for metadata delivery for standardisation.

[4.4.](#) Notifications to the Recorded User Agents

Typically a user that is involved in a session that is to be recorded is notified by an announcement at the beginning of the session or may receive some warning tones within the media. However the standardization of media recording protocols when using SIP enable an indication that the call is being recorded to be included in the signaling associated with that communication session

It is the Session Recording Client that must provide a notification to all users for which it is replicated received media for the purpose of recording including the local user if the Session Recording Client is a SIP endpoint.

If the Session Recording Client is aware that the remote UA is a recording aware UA then it MUST notify the UA using a SIP based mechanism. If the Session Recording Client is not aware that the peer UA is a recording aware UA then it MUST use an in-band tone or announcement to notify the remote UA.

[4.5.](#) Preventing the recording of a SIP session

A Recording Aware UA may during the initial session establishment or during an established session provide one of the following

indications.

- o No Recording Allowed
- o Recording Required

5. IANA considerations

None.

6. Security considerations

The Recording Session is fundamentally a standard SIP dialog and media session and therefore make use of existing SIP security mechanisms for securing the Recording Session and Media Recording Metadata. SRTP is used for securing the Replicated Media.

The intended use of this architecture is only for the case where the users are aware of that they are being recorded and the architecture provides the means for the Session Recording Client to notify users that they are being recorded.

This architectural solution is not intended to support lawful intercept which in contrast requires that users are not informed.

The Session Recording Client fact that the Recording Client decrypts and re-encrypts media, means the Session Recording Client must take precautions to prevent disclosure of media in the clear. Also the Session Recording Client has to be trusted not to manipulate or suppress media.

It is the responsibility of the Session Recording Server to protect the Replicated Media and Media Recording Metadata once it has been received and archived.

7. References

7.1. Normative References

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation

Protocol (SIP): Locating SIP Servers", [RFC 3263](#), June 2002.

- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3911] Mahy, R. and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header", [RFC 3911](#), October 2004.

7.2. Informative References

- [I-D.ietf-siprec-req]
Rehor, K., Jain, R., Portman, L., and A. Hutton,
"Requirements for SIP-based Media Recording (SIPREC)",
[draft-ietf-siprec-req-03](#) (work in progress), October 2010.
- [I-D.ietf-sipcore-info-events]
Holmberg, C., Burger, E., and H. Kaplan, "Session
Initiation Protocol (SIP) INFO Method and Package
Framework", [draft-ietf-sipcore-info-events-10](#) (work in
progress), October 2010.
- [I-D.ietf-xcon-event-package]
Camarillo, G., Srinivasan, S., Even, R., and J.
Urpalainen, "Conference Event Package Data Format
Extension for Centralized Conferencing (XCON)",
[draft-ietf-xcon-event-package-01](#) (work in progress),
September 2008.
- [RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific
Event Notification", [RFC 3265](#), June 2002.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat,
"Indicating User Agent Capabilities in the Session
Initiation Protocol (SIP)", [RFC 3840](#), August 2004.
- [RFC4353] Rosenberg, J., "A Framework for Conferencing with the
Session Initiation Protocol (SIP)", [RFC 4353](#),
February 2006.
- [RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session
Initiation Protocol (SIP) Event Package for Conference
State", [RFC 4575](#), August 2006.
- [RFC5567] Melanchuk, T., "An Architectural Framework for Media

Server Control", [RFC 5567](#), June 2009.

[RFC5369] Camarillo, G., "Framework for Transcoding with the Session Initiation Protocol (SIP)", [RFC 5369](#), October 2008.

[RFC2804] IAB and IESG, "IETF Policy on Wiretapping", [RFC 2804](#), May 2000.

Authors' Addresses

Andrew Hutton
Siemens Enterprise Communications
Hofmannstrasse 51
Munich D-81379
Germany

Email: andrew.hutton@siemens-enterprise.com

Leon Portman
Nice Systems
8 Hapnina
Ra'anana 43017
Israel

Email: leon.portman@nice.com

Rajnish Jain
IPC Systems
777 Commerce Drive
Fairfield, CT 06825
USA

Email: rajnish.jain@ipc.com

Ken Rehor
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Email: krehor@cisco.com

