

SIPREC	L.P. Portman, Ed.
Internet-Draft	NICE Systems
Intended status: Informational	H. Lum, Ed.
Expires: January 12, 2012	Genesys, Alcatel-Lucent
	A. Johnston
	Avaya
	A. Hutton
	Siemens Enterprise Communications
	July 11, 2011

Session Recording Protocol
draft-portman-siprec-protocol-05

Abstract

The Session Recording Protocol is used for establishing recording session and reporting of the metadata of the communication session. This document specifies the Session Recording Protocol. The protocol is used between Session Recording Client (SRC) and Session Recording Server (SRS).

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 January 12, 2012.

Copyright Notice

Copyright (c) 2011 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. Introduction
- *2. Definitions
- *3. Scope
- *4. Overview of operations
 - *4.1. Delivering recorded media
 - *4.2. Conference focus as an SRC
 - *4.3. Delivering recording metadata
- *5. SIP Extensions for Recording Session
 - *5.1. Callee Capabilities Extensions for SIP Recording
 - *5.1.1. src Feature Tag
 - *5.1.2. srs Feature Tag
 - *5.2. SDP handling
 - *5.3. RTP handling
 - *5.4. Metadata
 - *5.5. Requesting for metadata snapshot
 - *5.5.1. Formal Syntax
 - *5.6. Recording Pause and Resume
- *6. SIP Extensions for Recording-aware User Agents
 - *6.1. Providing recording indication
 - *6.2. Recording awareness
 - *6.3. Recording preference
- *7. IANA Considerations
 - *7.1. New Content-Disposition Parameter Registrations
 - *7.2. Media Type Registration
 - *7.2.1. Registration of MIME Type application/rs-metadata

*7.2.2. Registration of MIME Type application/rs-metadata-request

*7.3. Registration of record-aware Option Tag

*7.4. SDP Attributes

*7.4.1. 'record' SDP Attribute

*7.4.2. 'recordpref' SDP Attribute

*8. Security Considerations

*8.1. Authentication and Authorization

*9. References

*9.1. Normative References

*9.2. Informative References

*Authors' Addresses

1. Introduction

Communication Session (CS) recording requires establishment of the recording session between communication system and recording system. In order to allow access to such recordings, the metadata about the CS shall be sent from the SRC to the SRS.

The SIP-based Media Recording Requirements [I-D.ietf-siprec-req] list a set of requirements that need to be met by session recording protocols. The Session Recording Protocol, which is specified in this document, meets these requirements.

The Session Recording Protocol uses SIP as the protocol for session establishment with special attention to reducing size of the required SIP messages. In addition, it is designed for future extendability and protocol version management to ensure backward compatibility.

The remainder of this document is organized as follows: Section 2 defines the terminology used throughout this document, Section 3 discusses the scope of the Session Recording Protocol, Section 4 provides a non-normative overview of recording operations, Section 5 provides normative description of SIP extensions for the Recording Session, Section 6 provides normative description of SIP extensions for recording-aware user agents.

2. Definitions

The core definitions are taken from the requirements document [I-D.ietf-siprec-req].

Session Recording Server (SRS): A Session Recording Server (SRS) is a SIP User Agent (UA) that is a specialized media server or collector 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.

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 SRS. An SRC is a logical function. Its capabilities may be implemented across one or more physical devices. In practice, an SRC could be a personal device (such as a SIP phone), a SIP Media Gateway (MG), a Session Border Controller (SBC) or a SIP Media Server (MS) integrated with an Application Server (AS). This specification defines the term SRC such that all such SIP entities can be generically addressed under one definition. The SRC itself or another entity working on its behalf (such as a SIP Application Server) may act as the source of the recording metadata.

Communication Session (CS) A session created between two or more SIP User Agents (UAs) that is the target for recording.

Recording Session (RS) The SIP session created between an SRC and SRS for the purpose of recording a Communication Session.

Metadata Information that describes recorded media and the CS to which they relate.

Session Recording Protocol The set of SIP extensions that supports recording of Communication Sessions.

Pause during a Communication Session: The action of temporarily discontinuing the transmission and collection of RS media.

Resume during a Communication Session: The action of recommencing the transmission and collection of RS media.

Figure 1 shows the relationship between the definitions.

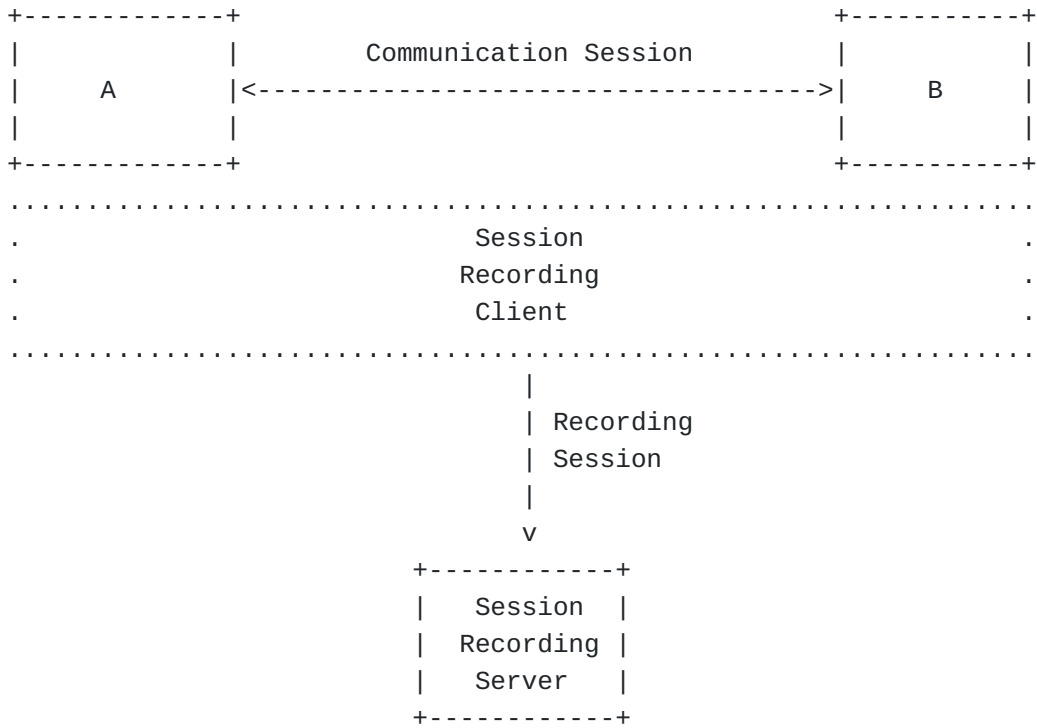


Figure 1: Relationship between CS, SRC, SRS, and RS

3. Scope

The scope of the Session Recording Protocol includes the establishment of the recording sessions and the reporting of the metadata. The following items, which is not an exhaustive list, do not represent the protocol itself and are considered out of the scope of the Session Recording Protocol:

- *Recording policies that determine whether the CS should be recorded
- *Retention policies that determine how long a recording is stored
- *Searching and accessing the recorded media and metadata
- *Delivering recording session metadata through non-SIP mechanism

4. Overview of operations

This section is informative and provides a description of recording operations.

As mentioned in the architecture document [[I-D.ietf-siprec-architecture](#)], there are a couple of types of call flows based on the location of the Session Recording Client. The following sample call

flows provide a quick overview of the operations between the SRC and the SRS.

4.1. Delivering recorded media

When the SRC is deployed as a B2BUA, the SRC can route call requests from UA(A) to UA(B). As a SIP B2BUA, the SRC has access to the media path between the user agents. When the SRC is aware that it should be recording the conversation, the SRC may bridge the media between UA(A) and UA(B). The SRC then establishes the Recording Session with the SRS and sends replicated media towards the SRS.

An endpoint can also be acting as the SRC, and the endpoint itself will be establishing the Recording Session to the SRS. Since the endpoint has access to the media in the Communication Session, the endpoint can send replicated media towards the SRS.

The following is a sample call flow that shows the SRC establishing a recording session towards the SRS. The call flow is essentially identical when the SRC is a B2BUA or as the endpoint itself. Note that the SRC can choose when to establish the Recording Session independent of the Communication Session, even though the following call flow suggests that the Recording Session is established after the Communication Session is established.

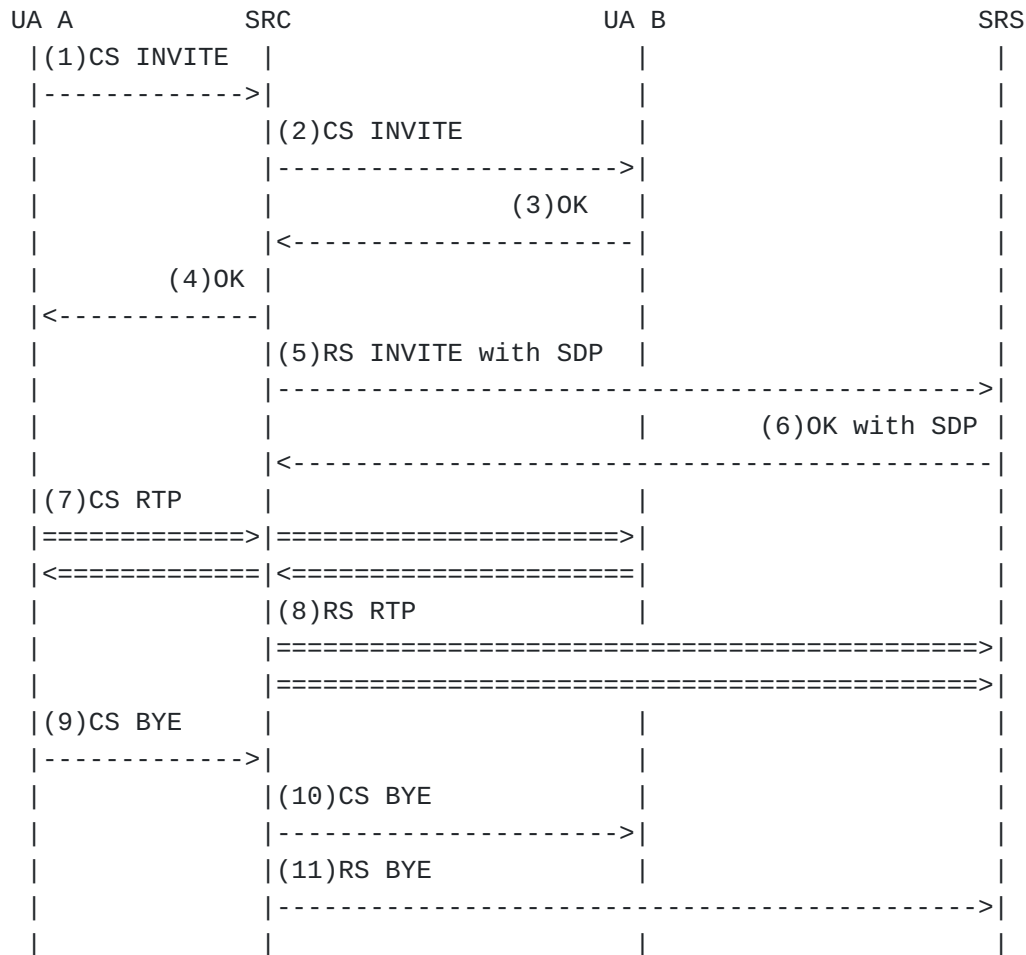


Figure 2: Basic Recording Call flow

4.2. Conference focus as an SRC

A conference focus may also act as an SRC since it has access to all the media from each conference participant. In this example, a user agent may REFER the conference focus to the SRS, and the SRC may choose to mix media streams from all participants as a single media stream towards the SRS. In order to tell the conference focus to start a recording session to the SRS, the user agent can include the srs feature tag in the Refer-To header as per [RFC4508].

UA A	Focus	UA B	SRS
	(SRC)		
	(already in a conference)		
<=====	<=====		
(1)REFER sip:Conf-ID	Refer-To:<SRS>;srs		
----->			
(2)202 Accepted			
<-----			
(3)NOTIFY (Trying)			
<-----			
(4)200 OK			
----->			
	(5)RS INVITE Contact:Conf-ID;isfocus		
	----->		
		(6)200 OK	
	<-----		
	(7)RTP (mixed or unmixed)		
	=====		
(8)NOTIFY (OK)			
<-----			
(9)200 OK			
----->			

Figure 3: Recording call flow - SRC as a conference focus

4.3. Delivering recording metadata

Certain metadata, such as the attributes of the recorded media stream, are already included in the SDP of the recording session. This information is reused as part of the metadata. The SRC may provide an initial metadata snapshot about recorded media streams in the initial INVITE content in the recording session. Subsequent metadata updates can be represented as a stream of events in UPDATE or reINVITE requests sent by the SRC. These metadata updates are normally incremental updates to the initial metadata snapshot to optimize on the size of updates, however, the SRC may also decide to send a new metadata snapshot anytime.

The SRS also has the ability to sent a request to the SRC to request to receive a new metadata snapshot update when the SRS fails to understand the current stream of incremental updates for whatever reason (ie. SRS gets a syntax/semantic error in metadata update, the SRS crashes and restarts), and the SRS may attach a reason along with the snapshot request. This request allows both SRC and SRS to restart the states with a new metadata snapshot so that further metadata incremental updates will be based on the latest metadata snapshot. Similar to the

metadata content, the metadata snapshot request is transported as content in UPDATE or INVITE sent by the SRC in the recording session.

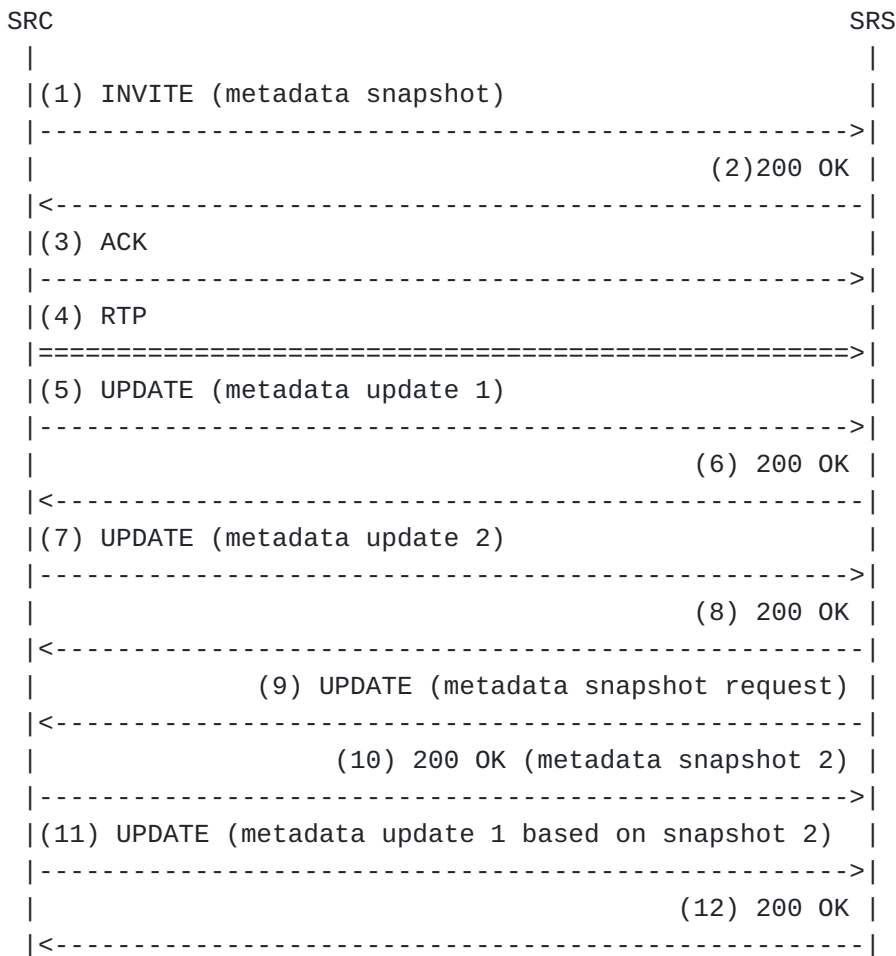


Figure 4: Delivering metadata via SIP UPDATE

In some cases session metadata can be conveyed through non-SIP mechanism such as HTTP or JTAPI. These non-SIP mechanisms are considered out of the scope of the Session Recording Protocol, however, it is envisioned that a link with a URI can be provided in the recording session INVITE message so that the SRS can access the session metadata via the URI provided that the SRS supports the type of URI.

5. SIP Extensions for Recording Session

The following sections describe SIP extensions for the Recording Session.

The From header must contain the identity of the SRC. Participants information is not recorded in the From or To header; they are included in the metadata information.

Note that a recording session does not have to live within the scope of a single communication session. As outline in REQ-005 of [\[I-D.ietf-siprec-req\]](#), the recording session can be established in the absence of a communication session. In this case, the SRC must pre-allocate a recorded media stream and offer an SDP with at least one m= line to establish a persistent recording session. When the actual call arrives, the SRC can map recorded media stream to participant media and minimize media clipping.

Recorded media from multiple communication sessions may be handled in a single recording session. The SRC provides a reference of each recorded media stream to the metadata described in the next section.

5.1. Callee Capabilities Extensions for SIP Recording

This section discusses how the callee capabilities defined in [\[RFC3840\]](#) can be extended for SIP call recording.

SIP Callee Capabilities defines feature tags which are used to represent characteristics and capabilities of a UA. From RFC 3840:

"Capability and characteristic information about a UA is carried as parameters of the Contact header field. These parameters can be used within REGISTER requests and responses, OPTIONS responses, and requests and responses that create dialogs (such as INVITE)."

Note that feature tags are also used in dialog modifying requests and responses such as re-INVITE and responses to a re-INVITE, and UPDATE. The 'isfocus' feature tag, defined in [\[RFC4579\]](#) is similar semantically to this case: it indicates that the UA is acting as a SIP conference focus, and is performing a specific action (mixing) on the resulting media stream. This information is available from OPTIONS queries, dialog package notifications, and the SIP registration event package. We propose the definition of two new feature tags: 'src' and 'srs'.

5.1.1. src Feature Tag

The 'src' feature tag is used in Contact URIs by the Session Recording Client (SRC) related to recording sessions. A Session Recording Server uses the presence of this feature tag in dialog creating and modifying requests and responses to confirm that the dialog being created is for the purpose of a Recording Session. In addition, a registrar could discover that a UA is an SRC based on the presence of this feature tag in a registration. Other SIP Recording extensions and behaviors can be triggered by the presence of this feature tag.

Note that we could use a single feature tag, such as 'recording' used by either an SRC or SRS to identify that the session is a recording session. However, due to the differences in functionality and behavior between an SRC and SRS, using only one feature tag for both is not ideal. For instance, if a routing mistake resulted in a request from a

SRC being routed back to another SRC, if only one feature tag were defined, they would not know right away about the error and could become confused. With separate feature tags, they would realize the error immediately and terminate the session. Also, call logs would clearly show the routing error.

5.1.2. srs Feature Tag

The 'srs' feature tag is used in Contact URIs by the Session Recording Server (SRS) related to recording sessions. A Session Recording Client uses the presence of this feature tag in dialog creating and modifying requests and responses to confirm that the dialog being created is for the purpose of a Recording Session (REQ-30). In addition, a registrar could discover that a UA is an SRS based on the presence of this feature tag in a registration. Other SIP Recording extensions and behaviors can be triggered by the presence of this feature tag. To ensure a recording session is redirected to an SRS, an SRC can utilize the SIP Caller Preferences extensions, defined in [\[RFC3841\]](#). The presence of a `Accept-Contact: *;sip.srs` allows a UA to request that the INVITE be routed to an SRS. Note that to be completely sure, the SRC would need to include a `Require: prefs` header field in the request.

5.2. SDP handling

Following the SDP offer/answer model in [\[RFC3264\]](#), this section describes the conventions used in the recording session for SDP handling.

SRC must provide an SDP offer in the initial INVITE to the SRS. SRC can include one or more media streams to the SRS. The SRS must respond with the same number of media descriptors in the SDP body of the 200 OK. The SRC should use `a=sendonly` attribute as the SRC does not expect to receive media from the SRS. As SRS only receives RTP streams from SRC, the 200 OK response will normally contain SDP with `a=recvonly` attribute.

Since the SRC may send recorded media of different participants (or even mixed streams) to the SRS, the SDP must provide a label on each media stream in order to identify the recorded stream with the rest of the metadata. The `a=label` attribute [\[RFC4574\]](#) will be used to identify each recorded media stream, and the label name is mapped to the Media Stream Reference in the metadata in [\[I-D.ietf-siprec-metadata\]](#). Note that a participant may have multiple streams (audio and video) and each stream is labeled separately.

```
v=0
o=SRS 0 0 IN IP4 172.22.3.8
s=SRS
c=IN IP4 172.22.3.8
t=0 0
m=audio 12241 RTP/AVP 0 4 8
a=sendonly
a=label:1
m=audio 12242 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 ...
a=sendonly
a=label:2
m=audio 12243 RTP/AVP 0 4 8
a=sendonly
a=label:3
m=audio 12244 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 ...
a=sendonly
a=label:4
```

Figure 6: Sample SDP with audio and video streams

To remove a recorded media stream from the recording session, send a reINVITE and set the port to zero in the m= line.

To add a recorded media stream, send a reINVITE and add a new m= line.

The SRS may respond with a=inactive attribute as part of the SDP in the 200 OK response when the SRS is not ready to receive recorded media.

The SRS can send re-INVITE to update the SDP with a=recvonly when it is ready to receive media.

The following sequence diagram shows an example of SRS responds with SDP that contain a=inactive, and then later update media information update with re-INVITE.

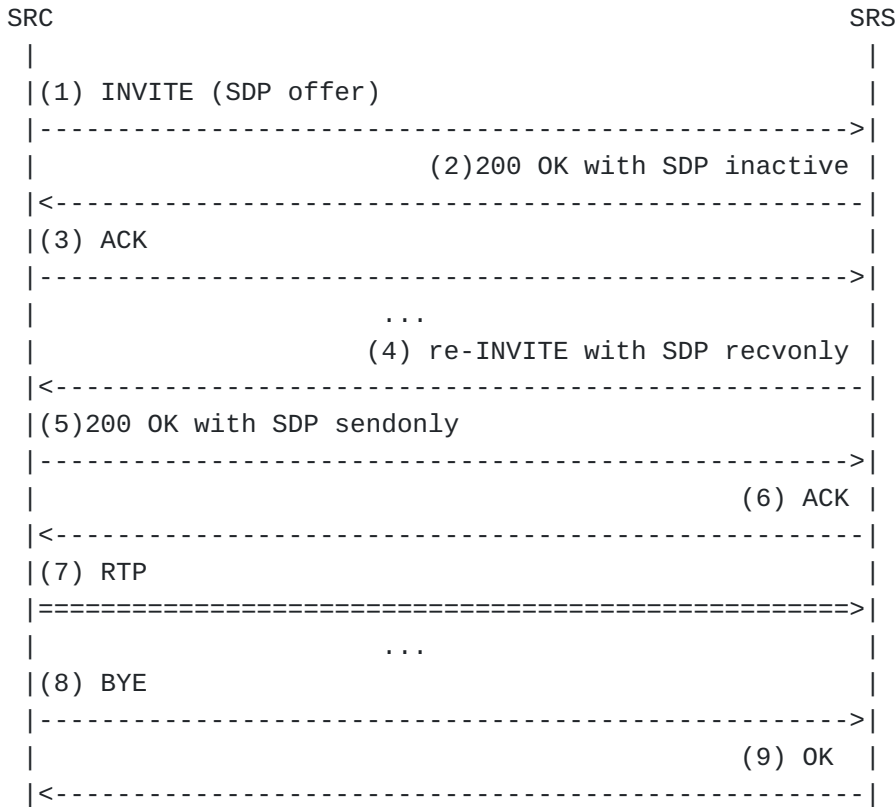


Figure 7: SRS to offer with a=inactive

5.3. RTP handling

[This is a placeholder section to specify any protocol impacts or recommendations for RTP usage in the session recording protocol. The details are listed in [\[I-D.eckel-siprec-rtp-rec\]](#)]

5.4. Metadata

The format of the full metadata will be described as part of the mechanism in [\[I-D.ietf-siprec-metadata\]](#).

As mentioned in the previous section, the SDP of the recording session is the metadata for all recorded media streams. The label attribute contains a reference to the rest of the metadata information.

For all basic metadata information such as communication session, participants, call identifiers and direction, they can be included in the initial INVITE request sent by the SRC. Metadata can be included as content in the INVITE or UPDATE request. A new "disposition-type" of Content-Disposition is defined for this purpose and the value is "recording-session".

The following SIP example for RS establishment between SRC and SRS with metadata as content.

```
INVITE sip:97753210@10.240.3.10:5060 SIP/2.0
From: <sip:2000@10.226.240.3>;tag=35e195d2-947d-4585-946f-098392474
To: <sip:Recorder@10.240.3.10>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a@10.226.240.3
CSeq: 101 INVITE
Date: Thu, 26 Nov 2009 02:38:49 GMT
Supported: timer
Supported: replaces
User-Agent: B2BUA
Max-Forwards: 70
Allow: INVITE,OPTIONS,INFO,BYE,CANCEL,ACK,PRACK,UPDATE,
REFER,SUBSCRIBE,NOTIFY,PUBLISH
Allow-Events: presence,kpml
Min-SE: 90
Contact: <sip:2000@10.226.240.3:5060;transport=tcp>;isfocus;src
Via: SIP/2.0/TCP 10.226.240.3:5060;branch=z9hG4bKdf6b622b648d9
Session-Expires: 1800
Content-Type: multipart/mixed;boundary=foobar
Content-Length: [length]

--foobar
Content-Type: application/sdp

v=0
o=SRS 0 0 IN IP4 10.226.240.3
c=IN IP4 10.226.240.3
t=0 0
m=audio 12241 RTP/AVP 0 4 8
a=sendonly
a=label:1

--foobar
Content-Type: application/rs-metadata
Content-Disposition: recording-session

[metadata content]
```

Figure 8: Sample INVITE request for the recording session

Further updates to recording metadata can be delivered as a sequence events reported in SIP UPDATE or reINVITE requests and the SRS must receive the sequence of events in order. Since there can only be a single INVITE or UPDATE transaction happening at a time within a SIP dialog, using sequence number CSeq in the dialog can be a reliable way for the SRS to identify the receipt of the next metadata update.

At any time during Recording Session, the SRC may send a new metadata snapshot in SIP UPDATE or reINVITE request. All subsequent metadata updates will be based on the new metadata snapshot.

5.5. Requesting for metadata snapshot

The SRS may send a request for metadata snapshot any time after the Recording Session has been established. Typically, the SRS sends such as request in the case where the SRS is failing to process further metadata incremental updates. Failure scenarios can include failure to parse metadata information (syntax error), failure to match metadata information with the current metadata snapshot (semantic error), or failure at the SRS.

Similar to delivering metadata, the SRS sends the metadata snapshot request as content in UPDATE or INVITE requests or responses. The same disposition type "recording-session" is used to note that the content represents content sent by the SRS. The format of the content is application/rs-metadata-request, and the body format is chosen to be a simple text-based format with header and values. The following shows an example:

```
SRS-Status: SRS failure
```

The SRS MUST include the reason why a metadata snapshot request is being made to the SRC in the SRS-Status header. This header is free form text to allow the SRS to provide a descriptive reason. The body format also allows additional extension headers to be included by the SRS in the snapshot request to convey additional information to the SRC.

When the SRC receives the request for a metadata snapshot, the SRC may provide the metadata snapshot in the response or as a separate INVITE/UPDATE transaction. All subsequent metadata updates sent by the SRC MUST be based on the new metadata snapshot.

5.5.1. Formal Syntax

The formal syntax for the application/rs-metadata-request MIME is described below using the augmented Backus-Naur Form (BNF) as described in [\[RFC2234\]](#).

```
snapshot-request = srs-status-line CRLF [ *opt-srs-headers ]
srs-status-line = "SRS-Status" HCOLON srs-status
srs-status = [TEXT-UTF8-TRIM]
opt-srs-headers = CRLF 1*(extension-header CRLF)
```

5.6. Recording Pause and Resume

To temporarily discontinue streaming and collection of recorded media from the SRC to the SRS, the SRC must send a reINVITE and set `a=inactive` for each recorded media stream to be paused.

To resume streaming and collection of recorded media, the SRC must send a reINVITE and set `a=sendonly` for each recorded media stream to resume. Note that when a media stream in the CS is muted/unmuted, this information may be conveyed in the metadata by the SRC. The SRC should not modify the recorded media stream with `a=inactive` for mute since this operation is reserved for pausing the RS media.

6. SIP Extensions for Recording-aware User Agents

The following sections describe SIP extensions for recording-aware UA.

6.1. Providing recording indication

While there are existing mechanisms for providing an indication that a CS is being recorded, these mechanisms are usually delivered on the CS media streams such as playing an in-band tone or an announcement to the participants. A new SDP attribute is introduced to allow a recording-aware UA to render recording indication at the user interface.

The 'record' SDP attribute appears at the media level, and may appear in either SDP offer or answer. The recording indication applies to the specified media stream only, for example, the audio portion of the call may be recorded in a audio/video call. The following is the ABNF of the 'record' attribute:

```
record-attr = "a=record:" indication
```

```
indication = "on" / "off" / "paused"
```

on Recording is in progress.

off No recording is in progress.

paused Recording is in progress by media is paused.

The recording attribute is a declaration by the endpoints in the session to indicate whether recording is taking place. For example, if a UA (A) is initiating a call to UA (B) and UA (A) is also an SRC that is performing the recording, then UA (A) provides the recording indication in the SDP offer with `a=record:on`. When UA (B) receives the SDP offer, UA (B) will see that recording is happening on the other endpoint of this session. If UA (B) does not wish to perform recording itself, UA (B) provides the recording indication as `a=record:off` in the SDP answer.

Whenever the recording indication needs to change, such as termination of recording, then the UA must initiate a reINVITE to update the SDP

recording indication through other means, such as passing an inband tone on the PSTN gateway, putting the recording indication in a log file, or raising an application event in a VoiceXML dialog. These user agents may also choose not to indicate recording awareness, thereby relying on whatever mechanism an SRC chooses to indicate recording, such as playing a tone inband.

When a UA has not indicated that it is recording aware, an SRC must provide recording indications, where SRC is required to do so based on policies, through other means such as playing a tone inband.

6.3. Recording preference

A recording-aware UA involved in a CS may request the CS to be recorded or not recorded. This indication of recording preference may be sent at session establishment time or during the session.

A new SDP attribute "recordpref" is introduced. The SDP attribute appears at the media level and can only appear in an SDP offer. The recording indication applies to the specified media stream only. The following is the ABNF of the recordpref attribute:

```
recordpref-attr = "a=recordpref:" pref
```

```
pref = "on" / "off" / "pause" / "nopreference"
```

on Request for recording if it has not already been started. If the recording is currently paused, request to resume recording.

off Request for no recording. If recording has already been started, then this preference indicates a request to stop recording.

pause Request to pause recording if recording is currently in progress.

nopreference To indicate that the UA has no preference on recording. While the absence of this attribute indirectly implies the lack of preference, using this value allows the UA to explicitly state no preference to being recorded.

7. IANA Considerations

7.1. New Content-Disposition Parameter Registrations

This document registers a new "disposition-type" value in Content-Disposition header: recording-session.

recording-session the body describes the metadata information about the recording session

7.2. Media Type Registration

7.2.1. Registration of MIME Type application/rs-metadata

This document registers the application/rs-metadata MIME media type in order to describe the recording session metadata. This media type is defined by the following information:

Media type name: application
Media subtype name: rs-metadata
Required parameters: none
Options parameters: none

7.2.2. Registration of MIME Type application/rs-metadata-request

This document registers the application/rs-metadata-request MIME media type in order to describe a recording session metadata snapshot request. This media type is defined by the following information:

Media type name: application
Media subtype name: rs-metadata-request
Required parameters: none
Options parameters: none

7.3. Registration of record-aware Option Tag

This document registers the "record-aware" option tag.

Name: record-aware

Description: This option tag is to indicate the ability for the user agent to receive recording indicators in media level SDP. When present in a Supported header, it indicates that the UA can receive recording indicators in media level SDP.

7.4. SDP Attributes

This document registers the following new SDP attributes.

7.4.1. 'record' SDP Attribute

Attribute name: record
Long form attribute name: Recording Indication
Type of attribute: media level
Subject to charset: no
This attribute provides the recording indication for the session or media stream.
Allowed attribute values: on, off, paused

7.4.2. 'recordpref' SDP Attribute

Attribute name: recordpref
Long form attribute name: Recording Preference
Type of attribute: media level

Subject to charset: no

This attribute provides the recording indication for the session or media stream.

Allowed attribute values: on, off, pause, nopreference

8. Security Considerations

The recording session is fundamentally a standard SIP dialog [[RFC3261](#)], therefore, the recording session can reuse any of the existing SIP security mechanism available for securing the recorded media as well as metadata.

8.1. Authentication and Authorization

The recording session reuses the SIP mechanism to challenge requests that is based on HTTP authentication. The mechanism relies on 401 and 407 SIP responses as well as other SIP header fields for carrying challenges and credentials.

The SRS may have its own set of recording policies to authorize recording requests from the SRC. The use of recording policies is outside the scope of the Session Recording Protocol.

9. References

9.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels" , BCP 14, RFC 2119, March 1997.
[RFC2234]	Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF" , RFC 2234, November 1997.
[RFC2804]	IABIESG, " IETF Policy on Wiretapping ", RFC 2804, May 2000.
[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.
[RFC3264]	Rosenberg, J. and H. Schulzrinne, " An Offer/Answer Model with Session Description Protocol (SDP) ", RFC 3264, June 2002.
[RFC4574]	Levin, O. and G. Camarillo, " The Session Description Protocol (SDP) Label Attribute ", RFC 4574, August 2006.
[RFC3840]	Rosenberg, J., Schulzrinne, H. and P. Kyzivat, " Indicating User Agent Capabilities in the Session Initiation Protocol (SIP) ", RFC 3840, August 2004.
[RFC3841]	Rosenberg, J., Schulzrinne, H. and P. Kyzivat, " Caller Preferences for the Session Initiation Protocol (SIP) ", RFC 3841, August 2004.

[I-D.ietf-siprec-req]	Rehor, K, Portman, L, Hutton, A and R Jain, " <u>Use Cases and Requirements for SIP-based Media Recording (SIPREC)</u> ", Internet-Draft draft-ietf-siprec-req-12, June 2011.
[I-D.ietf-siprec-metadata]	R, R, Ravindran, P and P Kyzivat, " <u>Session Initiation Protocol (SIP) Recording Metadata</u> ", Internet-Draft draft-ietf-siprec-metadata-05, October 2011.

9.2. Informative References

[I-D.ietf-siprec-architecture]	Hutton, A, Portman, L, Jain, R and K Rehor, " <u>An Architecture for Media Recording using the Session Initiation Protocol</u> ", Internet-Draft draft-ietf-siprec-architecture-03, October 2011.
[I-D.eckel-siprec-rtp-rec]	Eckel, C, " <u>Real-time Transport Protocol (RTP) Recommendations for SIPREC</u> ", Internet-Draft draft-eckel-siprec-rtp-rec-03, October 2011.
[RFC4508]	Levin, O. and A. Johnston, " <u>Conveying Feature Tags with the Session Initiation Protocol (SIP) REFER Method</u> ", RFC 4508, May 2006.
[RFC4579]	Johnston, A. and O. Levin, " <u>Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents</u> ", BCP 119, RFC 4579, August 2006.

Authors' Addresses

Leon Portman editor Portman NICE Systems 8 Hapnina Ra'anana, 43017 Israel EMail: leon.portman@nice.com

Henry Lum editor Lum Genesys, Alcatel-Lucent 1380 Rodick Road, Suite 200 Markham, Ontario L3R4G5 Canada EMail: henry.lum@genesyslab.com

Alan Johnston Johnston Avaya St. Louis, MO 63124 EMail: alan.b.johnston@gmail.com

Andrew Hutton Hutton Siemens Enterprise Communications EMail: andrew.hutton@siemens-enterprise.com