SIPREC	L. Portman
Internet-Draft	NICE Systems
Intended status: Standards Track	H. Lum, Ed.
Expires: January 14, 2013	Genesys
	C. Eckel
	Cisco
	A. Johnston
	Avaya
	A. Hutton
	Siemens Enterprise
	Communications
	July 13, 2012

Session Recording Protocol draft-ietf-siprec-protocol-06

Abstract

This document specifies the use of the Session Initiation Protocol (SIP), the Session Description Protocol (SDP), and the Real Time Protocol (RTP) for delivering real-time media and metadata from a Communication Session (CS) to a recording device. The Session Recording Protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) between the Session Recording Client (SRC), which is on the path of the CS, and a Session Recording Server (SRS) at the recording device.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

TOC

This document specifies the mechanism to record a Communication Session (CS) by delivering real-time media and metadata from the CS to a recording device. In accordance to the architecture **[I-D.ietf-siprec-architecture]**, the Session Recording Protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) between the Session Recording Client (SRC), which is on the path of the CS, and a Session Recording Server (SRS) at the recording device.

SIP is also used to deliver metadata to the recording device, as specified in **[I-D.ietf-siprec-metadata]**. Metadata is information that describes recorded media and the CS to which they relate.

The Session Recording Protocol intends to satisfy the SIP-based Media Recording requirements listed in **[RFC6341]**.

In addition to the Session Recording Protocol, this document specifies extensions for user agents that are participants in a CS to receive recording indications and to provide preferences for recording.

2. Terminology

TOC

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]**.

3. Definitions

TOC

This document refers to the core definitions provided in the architecture document **[I-D.ietf-siprec-architecture]**.

The RTP Handling section uses the definitions provided in "RTP: A Transport Protocol for Real-Time Application" **[RFC3550]**.

4. Scope

TOC

The scope of the Session Recording Protocol includes the establishment of the recording sessions and the reporting of the metadata. The scope also includes extensions supported by User Agents participating in the CS such as indication of recording. The user agents need not be recording-aware in order to participate in a CS being recorded.

The following items, which are not an exhaustive list, do not represent the protocol itself and are considered out of the scope of the Session Recording Protocol:

- Delivering recorded media in real-time as the CS media
- Specifications of criteria to select a specific CS to be recorded or triggers to record a certain CS in the future
- Recording policies that determine whether the CS should be recorded and whether parts of the CS are to be recorded
- Retention policies that determine how long a recording is stored
- Searching and accessing the recorded media and metadata
- Policies governing how CS users are made aware of recording
- Delivering additional recording session metadata through non-SIP mechanism

5. Overview of operations

TOC

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

Section 5 provides the procedures for establishing a recording session between a SRC and a SRS. Section 6 describes the SDP in a recording session. Section 7 describes the RTP handling in a recording session. Section 8 describes the mechanism to deliver recording metadata from the SRC to the SRS.

Section 10 describes the procedures for user agents participating in a CS to receive recording indications and to provide preferences for recording.

As mentioned in the architecture document **[I-D.ietf-siprec-architecture]**, there are a number 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.

5.1. Delivering recorded media

TOC

When a SIP Back-to-back User Agent (B2BUA) with SRC functionality routes a call from UA(A) to UA(B), 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 can cause the B2BUA to 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 may also have SRC functionality, where the endpoint itself establishes 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 SRC is establishing the Recording Session (message #5) after the Communication Session is established.

```
UA A           SRC                        UA B                       SRS
 |(1)CS INVITE  |                           |                           |
 |------------->|                           |                           |
 |              |(2)CS INVITE               |                           |
 |              |-------------------------->|                           |
 |              |               (3) 200 OK  |                           |
 |              |<--------------------------|                           |
 |    (4) 200 OK |                          |                           |
 |<-------------|                           |                           |
 |              |(5)RS INVITE with SDP      |                           |
 |              |------------------------------------------------------>|
 |              |                           |  (6) 200 OK with SDP      |
 |              |<------------------------------------------------------|
 |(7)CS RTP     |                           |                           |
 |=============>|==========================>|                           |
 |<=============|<==========================|                           |
 |              |(8)RS RTP                 |                           |
 |              |======================================================>|
 |              |======================================================>|
 |(9)CS BYE     |                           |                           |
 |------------->|                           |                           |
 |              |(10)CS BYE                 |                           |
 |              |-------------------------->|                           |
 |              |(11)RS BYE                 |                           |
 |              |------------------------------------------------------>|
```

```
 |                  |                            |                          |
```

Figure 1: Basic recording call flow

The above call flow can also apply to the case of a centralized conference with a mixer. For clarity, ACKs to INVITEs and 200 OKs to BYEs are not shown. The conference focus can provide the SRC functionality since the conference focus has access to all the media from each conference participant. When a recording is requested, the SRC delivers the metadata and the media streams to the SRS. Since the conference focus has access to a mixer, the SRC may choose to mix the media streams from all participants as a single mixed media stream towards the SRS.

An SRC can use a single recording session to record multiple communication sessions. Every time the SRC wants to record a new call, the SRC updates the recording session with a new SDP offer to add new recorded streams to the recording session, and correspondingly also update the metadata for the new call.

An SRS can also establish a recording session to an SRC, although it is beyond the scope of this document to define how an SRS would specify which calls to record.

5.2. Delivering recording metadata

TOC

The SRC is responsible for the delivery of metadata to the SRS. 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.

Metadata is transported in the body of INVITE or UPDATE messages. Certain metadata, such as the attributes of the recorded media stream are located in the SDP of the recording session.

The SRS has the ability to send a request to the SRC to request for a new metadata snapshot update from the SRC. This can happen when the SRS fails to understand the current stream of incremental updates for whatever reason, for example, when SRS loses the current state due to internal failure. The SRS may optionally attach a reason along with the the snapshot request. This request allows both SRC and SRS to synchronize 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 SRS in the recording session.

```
SRC                                                     SRS
 |                                                       |
 |(1) INVITE (metadata snapshot)                         |
 |------------------------------------------------------>|
 |                                           (2)200 OK   |
 |<------------------------------------------------------|
 |(3) ACK                                                |
 |------------------------------------------------------>|
 |(4) RTP                                                |
 |======================================================>|
 |======================================================>|
 |(5) UPDATE (metadata update 1)                         |
 |------------------------------------------------------>|
 |                                          (6) 200 OK   |
 |<------------------------------------------------------|
 |(7) UPDATE (metadata update 2)                         |
 |------------------------------------------------------>|
 |                                          (8) 200 OK   |
```

```
|<--------------------------------------------------------|
|              (9) UPDATE (metadata snapshot request)     |
|<--------------------------------------------------------|
|                                        (10) 200 OK      |
|-------------------------------------------------------->|
|      (11) INVITE (metadata snapshot 2 + SDP offer)      |
|-------------------------------------------------------->|
|                            (12) 200 OK (SDP answer)     |
|<--------------------------------------------------------|
| (13) UPDATE (metadata update 1 based on snapshot 2)     |
|-------------------------------------------------------->|
|                                         (14) 200 OK     |
|<--------------------------------------------------------|
```

Figure 2: Delivering metadata via SIP UPDATE

5.3. Receiving recording indications and providing recording preferences

The SRC is responsible to provide recording indications to the participants in the CS. User agents that are recording-aware supports receiving recording indications with new SDP attribute a=record and the recording-aware UA can also set recording preference in the CS with a new SDP attribute a=recordpref. The recording attribute is a declaration by the SRC in the CS to indicate whether recording is taking place. The recording preference attribute is a declaration by the recording-aware UA in the CS to indicate the recording preference.

To illustrate how the attributes are used, 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. Since UA (A) is the SRC, UA (A) receives the recording indication from the SRC directly. When UA (B) receives the SDP offer, UA (B) will see that recording is happening on the other endpoint of this session. Since UA (B) is not an SRC and does not provide any recording preference, the SDP answer does not contain a=record nor a=recordpref.

```
UA A                                                    UA B
(SRC)                                                    |
  |                                                      |
  |                [SRC recording starts]                |
  |(1) INVITE (SDP offer + a=record:on)                  |
  |----------------------------------------------------->|
  |                             (2) 200 OK (SDP answer)  |
  |<-----------------------------------------------------|
  |(3) ACK                                               |
  |----------------------------------------------------->|
  |(4) RTP                                               |
  |<====================================================>|
  |                                                      |
  |   [UA B wants to set preference to no recording]     |
  |           (5) INVITE (SDP offer + a=recordpref:off)  |
  |<-----------------------------------------------------|
  |   [SRC honors the preference and stops recording]    |
  |(6) 200 OK (SDP answer + a=record:off)                |
  |----------------------------------------------------->|
  |                                             (7) ACK  |
  |<-----------------------------------------------------|
```

Figure 3: Recording indication and recording preference

After the call is established and recording is in progress, UA (B) later decides to change the recording preference to no recording and sends a reINVITE with the a=recordpref attribute. It is up to the SRC to honor the preference, and in this case SRC decides to stop the recording and updates the recording indication in the SDP answer.

TOC

6. Initiating a Recording Session

A recording session is a SIP session with specific extensions applied, and these extensions are listed in the procedures below. When an SRC or an SRS receives a SIP session that is not a recording session, it is up to the SRC or the SRS to determine what to do with the SIP session.

TOC

6.1. Procedures at the SRC

The SRC can initiate a recording session by sending a SIP INVITE request to the SRS. The SRC and the SRS are identified in the From and To headers, respectively.

The SRC MUST include the '+sip.src' feature tag in the Contact URI, defined in this specification as an extension to **[RFC3840]**, for all recording sessions. An SRS uses the presence of the '+sip.src' 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, when an SRC sends a REGISTER request to a registrar, the SRC MUST include the '+sip.src' feature tag to indicate the that it is a SRC.

Since SIP Caller Preferences extensions are optional to implement for routing proxies, there is no guarantee that a recording session will be routed to an SRC or SRS. A new options tag is introduced: "siprec". As per **[RFC3261]**, only an SRC or an SRS can accept this option tag in a recording session. An SRC MUST include the "siprec" option tag in the Require header when initiating a Recording Session so that UA's which do not support the session recording protocol extensions will simply reject the INVITE request with a 420 Bad Extension.

When an SRC receives a new INVITE, the SRC MUST only consider the SIP session as a recording session when both the '+sip.srs' feature tag and 'siprec' option tag are included in the INVITE request.

TOC

6.2. Procedures at the SRS

When an SRS receives a new INVITE, the SRS MUST only consider the SIP session as a recording session when both the '+sip.src' feature tag and 'siprec' option tag are included in the INVITE request.

The SRS can initiate a recording session by sending a SIP INVITE request to the SRC. The SRS and the SRC are identified in the From and To headers, respectively.

The SRS MUST include the '+sip.srs' feature tag in the Contact URI, as per **[RFC3840]**, for all recording sessions. An SRC 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, when an SRS sends a REGISTER request to a registrar, the SRS MUST include the '+sip.srs' feature tag to indicate that it is a SRS.

An SRS MUST include the "siprec" option tag in the Require header as per **[RFC3261]** when initiating a Recording Session so that UA's which do not support the session recording protocol extensions will simply reject the INVITE request with a 420 Bad Extension.

TOC

7. SDP Handling

The SRC and SRS follows the SDP offer/answer model in **[RFC3264]**. The rest of this section describes conventions used in a recording session.

7.1. Procedures at the SRC

TOC

Since the SRC does not expect to receive media from the SRS, the SRC typically sets each media stream of the SDP offer to only send media, by qualifying them with the a=sendonly attribute, according to the procedures in **[RFC3264]**.

The SRC sends recorded streams of participants to the SRS, and the SRC MUST provide a label attribute (a=label), as per **[RFC4574]**, on each media stream in order to identify the recorded stream with the rest of the metadata. The a=label attribute identifies each recorded media stream, and the label name is mapped to the Media Stream Reference in the metadata as per **[I-D.ietf-siprec-metadata]**. The scope of the label name only applies to the same SIP message as the SDP, meaning that the label name can be reused by another media stream within the same recording session. Note that a recorded stream is distinct from a CS stream; the metadata provides a list of participants that contributes to each recorded stream.

The following is an example of SDP offer from SRC with both audio and video recorded streams. Note that the following example contain unfolded lines longer than 72 characters. These are captured between <allOneLine> tags.

```
v=0
o=SRC 2890844526 2890844526 IN IP4 198.51.100.1
s=-
c=IN IP4 198.51.100.1
t=0 0
m=audio 12240 RTP/AVP 0 4 8
a=sendonly
a=label:1
m=video 22456 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=sendonly
a=label:2
m=audio 12242 RTP/AVP 0 4 8
a=sendonly
a=label:3
m=video 22458 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=sendonly
a=label:4
```

Figure 4: Sample SDP offer from SRC with audio and video streams

7.1.1. Handling media stream updates

TOC

Over the lifetime of a recording session, the SRC can add and remove recorded streams from the recording session for various reasons. For example, when a CS stream is added or removed from the CS, or when a CS is created or terminated if a recording session handles multiple CSes. To remove a recorded stream from the recording session, the SRC sends a new SDP offer where the port of the media stream to be removed is set to zero, according to the procedures in **[RFC3264]**. To add a recorded stream to the recording session, the SRC sends a new SDP offer by adding a new media stream description or by reusing an old media stream which had been previously disabled, according to the procedures in **[RFC3264]**.

The SRC can temporarily discontinue streaming and collection of recorded media from the SRC to the SRS for reason such as masking the recording. In this case, the SRC sends a new SDP offer and sets the media stream to inactive (a=inactive) for each recorded stream to be paused, as per the procedures in **[RFC3264]**. To resume streaming and collection of recorded media, the SRC sends a new SDP offer and sets the media streams with a=sendonly attribute. Note that when a CS stream is muted/unmuted, this information is conveyed in the metadata by the SRC. The SRC SHOULD NOT modify the media stream with a=inactive for mute since this operation is reserved for pausing the RS media.

TOC

7.2. Procedures at the SRS

The SRS only receives RTP streams from the SRC, the SDP answer normally sets each media stream to receive media, by setting them with the a=recvonly attribute, according to the procedures of **[RFC3264]**. When the SRS is not ready to receive a recorded stream, the SRS sets the media stream as inactive in the SDP offer or answer by setting it with a=inactive attribute, according to the procedures of **[RFC3264]**. When the SRS is ready to receive recorded streams, the SRS sends a new SDP offer and sets the media streams with a=recvonly attribute.

The following is an example of SDP answer from SRS for the SDP offer from the above sample. Note that the following example contain unfolded lines longer than 72 characters. These are captured between <allOneLine> tags.

```
v=0
o=SRS 0 0 IN IP4 198.51.100.20
s=-
c=IN IP4 198.51.100.20
t=0 0
m=audio 10000 RTP/AVP 0 4 8
a=recvonly
a=label:1
m=video 10002 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=recvonly
a=label:2
m=audio 10004 RTP/AVP 0 4 8
a=recvonly
a=label:3
m=video 10006 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=recvonly
a=label:4
```

Figure 5: Sample SDP answer from SRS with audio and video streams

Over the lifetime of a recording session, the SRS can remove recorded streams from the recording session for various reasons. To remove a recorded stream from the recording session, the SRS sends a new SDP offer where the port of the media stream to be removed is set to zero, according to the procedures in **[RFC3264]**.

The SRS SHOULD NOT add recorded streams in the recording session when SRS sends a new SDP offer. Similarly, when the SRS starts a recording session, the SRS SHOULD initiate the INVITE without an SDP offer to let the SRC generate the SDP offer with recorded streams.

The following sequence diagram shows an example where the SRS is initially not ready to receive recorded streams, and later updates the recording session when the SRS is ready to record.

```
SRC                                                      SRS
 |                                                        |
 |(1) INVITE (SDP offer)                                  |
 |------------------------------------------------------->|
 |                                             [not ready to record]
 |                         (2)200 OK with SDP inactive    |
 |<-------------------------------------------------------|
 |(3) ACK                                                 |
 |------------------------------------------------------->|
 |                      ...                               |
 |                                               [ready to record]
 |                     (4) re-INVITE with SDP recvonly    |
 |<-------------------------------------------------------|
 |(5)200 OK with SDP sendonly                             |
 |------------------------------------------------------->|
 |                                               (6) ACK  |
 |<-------------------------------------------------------|
 |(7) RTP                                                 |
 |=======================================================>|
 |                      ...                               |
 |(8) BYE                                                 |
 |------------------------------------------------------->|
 |                                               (9) OK   |
 |<-------------------------------------------------------|
```

Figure 6: SRS responding to offer with a=inactive

TOC

8. RTP Handling

This section provides recommendations and guidelines for RTP and RTCP in the context of SIPREC. In order to communicate most effectively, the Session Recording Client (SRC), the Session Recording Server (SRS), and any Recording aware User Agents (UAs) SHOULD utilize the mechanisms provided by RTP in a well-defined and predicable manner. It is the goal of this document to make the reader aware of these mechanisms and provide recommendations and guidelines.

TOC

8.1. RTP Mechanisms

This section briefly describes important RTP/RTCP constructs and mechanisms that are particularly useful within the content of SIPREC.

TOC

8.1.1. RTCP

The RTP data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery. RTCP, as defined in **[RFC3550]**, is based on the periodic transmission of control packets to all participants in the RTP session, using the same distribution mechanism as the data packets. Support for RTCP is REQUIRED, per **[RFC3550]**, and it provides, among other things, the following important functionality in relation to SIPREC:

1) Feedback on the quality of the data distribution

This feedback from the receivers may be used to diagnose faults in the distribution. As such, RTCP is a well-defined and efficient mechanism for the SRS to inform the SRC, and for the SRC to inform Recording aware UAs, of issues that arise with respect to the reception of media that is to be recorded.

2) Carries a persistent transport-level identifier for an RTP source called the canonical name or CNAME

The SSRC identifier may change if a conflict is discovered or a program is restarted; in which case receivers can use the CNAME to keep track of each participant. Receivers may also use the CNAME to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video. Synchronization of media streams is also facilitated by the NTP and RTP timestamps included in RTCP packets by data senders.

TOC

8.1.2. RTP Profile

The RECOMMENDED RTP profiles for the SRC, SRS, and Recording aware UAs are "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", **[RFC5124]** when using encrypted RTP streams, and "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", **[RFC4585]** when using non encrypted media streams. However, as this is not a requirement, some implementations may use "The Secure Real-time Transport Protocol (SRTP)", **[RFC3711]** and "RTP Profile for Audio and Video Conferences with Minimal Control", AVP **[RFC3551]**. Therefore, it is RECOMMENDED that the SRC, SRS, and Recording aware UAs not rely entirely on SAVPF or AVPF for core functionality that may be at least partially achievable using SAVP and AVP.

AVPF and SAVPF provide an improved RTCP timer model that allows more flexible transmission of RTCP packets in response to events, rather than strictly according to bandwidth. AVPF based codec control messages provide efficient mechanisms for an SRC, SRS, and Recording aware UAs to handle events such as scene changes, error recovery, and dynamic bandwidth adjustments. These messages are discussed in more detail later in this document.

SAVP and SAVPF provide media encryption, integrity protection, replay protection, and a limited form of source authentication. They do not contain or require a specific keying mechanism.

TOC

8.1.3. SSRC

The synchronization source (SSRC), as defined in **[RFC3550]** is carried in the RTP header and in various fields of RTCP packets. It is a random 32-bit number that is required to be globally unique within an RTP session. It is crucial that the number be chosen with care in order that participants on the same network or starting at the same time are not likely to choose the same number. Guidelines regarding SSRC value selection and conflict resolution are provided in **[RFC3550]**.

The SSRC may also be used to separate different sources of media within a single RTP session. For this reason as well as for conflict resolution, it is important that the SRC, SRS, and Recording aware UAs handle changes in SSRC values and properly identify the reason of the change. The CNAME values carried in RTCP facilitate this identification.

TOC

8.1.4. CSRC

The contributing source (CSRC), as defined in **[RFC3550]**, identifies the source of a stream of RTP packets that has contributed to the combined stream produced by an RTP mixer. The mixer inserts a list of the SSRC identifiers of the sources that contributed to the generation of a particular packet into the RTP header of that packet. This list is called the CSRC list. It is RECOMMENDED that a SRC or Recording aware UA, when acting a mixer, sets the CSRC list accordingly, and that the SRC and SRS interpret the CSRC list appropriately when received.

TOC

8.1.5. SDES

The Source Description (SDES), as defined in **[RFC3550]**, contains an SSRC/CSRC identifier followed by a list of zero or more items, which carry information about the SSRC/CSRC. End systems send one SDES packet containing their own source identifier (the same as the SSRC in the fixed RTP header). A mixer sends one SDES packet containing a chunk for each contributing source from which it is receiving SDES information, or multiple complete SDES packets if there are more than 31 such sources.

TOC

8.1.5.1. CNAME

The Canonical End-Point Identifier (CNAME), as defined in **[RFC3550]**, provides the binding from the SSRC identifier to an identifier for the source (sender or receiver) that remains constant. It is important the SRC and Recording aware UAs generate CNAMEs appropriately and that the SRC and SRS interpret and use them for this purpose. Guidelines for generating CNAME values are provided in "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)" **[RFC6222]**.

TOC

8.1.6. Keepalive

It is anticipated that media streams in SIPREC may exist in an inactive state for extended periods of times for any of a number of valid reasons. In order for the bindings and any pinholes in NATs/firewalls to remain active during such intervals, it is RECOMMENDED that the SRC, SRS, and Recording aware UAs follow the keep-alive procedure recommended in "Application Mechanism for Keeping Alive the NAT Mappings Associated to RTP/RTP Control Protocol (RTCP) Flows" **[RFC6263]** for all RTP media streams.

TOC

8.1.7. RTCP Feedback Messages

"Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)" **[RFC5104]** specifies extensions to the messages defined in AVPF **[RFC4585]**. Support for and proper usage of these messages is important to SRC, SRS, and Recording aware UA implementations. Note that these messages are applicable only when using the AVFP or SAVPF RTP profiles

---

TOC

8.1.7.1. Full Intra Request

A Full Intra Request (FIR) Command, when received by the designated media sender, requires that the media sender sends a Decoder Refresh Point at the earliest opportunity. Using a decoder refresh point implies refraining from using any picture sent prior to that point as a reference for the encoding process of any subsequent picture sent in the stream.

Decoder refresh points, especially Intra or IDR pictures for H.264 video codecs, are in general several times larger in size than predicted pictures. Thus, in scenarios in which the available bit rate is small, the use of a decoder refresh point implies a delay that is significantly longer than the typical picture duration.

8.1.7.1.1. SIP INFO for FIR

TOC

"XML Schema for Media Control" **[RFC5168]** defines an Extensible Markup Language (XML) Schema for video fast update. Implementations are discouraged from using the method described except for backward compatibility purposes. Implementations SHOULD use FIR messages instead.

8.1.7.2. Picture Loss Indicator

TOC

Picture Loss Indication (PLI), as defined in **[RFC4585]**, informs the encoder of the loss of an undefined amount of coded video data belonging to one or more pictures. Using the FIR command to recover from errors is explicitly disallowed, and instead the PLI message SHOULD be used. FIR SHOULD be used only in situations where not sending a decoder refresh point would render the video unusable for the users. Examples where sending FIR is appropriate include a multipoint conference when a new user joins the conference and no regular decoder refresh point interval is established, and a video switching MCU that changes streams.

8.1.7.3. Temporary Maximum Media Stream Bit Rate Request

TOC

A receiver, translator, or mixer uses the Temporary Maximum Media Stream Bit Rate Request (TMMBR) to request a sender to limit the maximum bit rate for a media stream to the provided value. Appropriate use of TMMBR facilitates rapid adaptation to changes in available bandwidth.

8.1.7.3.1. Renegotiation of SDP bandwidth attribute

TOC

If it is likely that the new value indicated by TMMBR will be valid for the remainder of the session, the TMMBR sender is expected to perform a renegotiation of the session upper limit using the session signaling protocol. Therefore for SIPREC, implementations are RECOMMENDED to use TMMBR for temporary changes, and renegotiation of bandwidth via SDP offer/answer for more permanent changes.

8.1.8. Symmetric RTP/RTCP for Sending and Receiving

TOC

Within an SDP offer/answer exchange, RTP entities choose the RTP and RTCP transport addresses (i.e., IP addresses and port numbers) on which to receive packets. When sending packets, the RTP entities may use the same source port or a different source port as those signaled for receiving packets. When the transport address used to send and receive RTP is the same, it is termed "symmetric RTP" **[RFC4961]**. Likewise, when the transport address used to send and receive RTCP is the same, it is termed "symmetric RTCP" **[RFC4961]**.

When sending RTP, it is REQUIRED to use symmetric RTP. When sending RTCP, it is REQUIRED to use symmetric RTCP. Although an SRS will not normally send RTP, it will send RTCP as well as receive RTP and RTCP. Likewise, although an SRC will not normally receive RTP from the SRS, it will receive RTCP as well as send RTP and RTCP.

> Note: Symmetric RTP and symmetric RTCP are different from RTP/RTCP multiplexing **[RFC5761]**.

8.2. Roles

TOC

An SRC has the task of gathering media from the various UAs in one or more Communication Sessions (CSs) and forwarding the information to the SRS within the context of a corresponding Recording Session (RS). There are numerous ways in which an SRC may do this is, including but not limited to, appearing as a UA within a CS, or as a B2BUA between UAs within a CS.

```
           (Recording Session)   +---------+
         +------------SIP------->|         |
         |  +------RTP/RTCP----->|   SRS   |
         |  |     +-- Metadata -->|         |
         |  |     |              +---------+
         v  v     |
        +---------+
        |   SRC   |
        |---------| (Communication Session) +---------+
        |         |<----------SIP---------->|         |
        |  UA-A   |                         |  UA-B   |
        |         |<-------RTP/RTCP-------->|         |
        +---------+                         +---------+
```

Figure 7: UA as SRC

```
                           (Recording Session)   +---------+
                         +------------SIP------->|         |
                         |  +------RTP/RTCP----->|   SRS   |
                         |  |     +-- Metadata -->|         |
                         |  |     |              +---------+
                         v  v     |
                        +---------+
                        |   SRC   |
+---------+             |---------|              +---------+
|         |<----SIP----->|         |<----SIP----->|         |
|  UA-A   |             |  B2BUA  |              |  UA-B   |
|         |<--RTP/RTCP-->|         |<--RTP/RTCP-->|         |
+---------+             +---------+              +---------+
      |_______________________________________________|
                     (Communication Session)
```

Figure 8: B2BUA as SRC

The following subsections define a set of roles an SRC may choose to play based on its position with respect to a UA within a CS, and an SRS within an RS. A CS and a corresponding RS are independent sessions; therefore, an SRC may play a different role within a CS than it does within the corresponding RS.

8.2.1. SRC acting as an RTP Translator

The SRC may act as a translator, as defined in **[RFC3550]**. A defining characteristic of a translator is that it forwards RTP packets with their SSRC identifier intact. There are two types of translators, one that simply forwards, and another that performs transcoding (e.g., from one codec to another) in addition to forwarding.

8.2.1.1. Forwarding Translator

When acting as a forwarding translator, RTP received as separate streams from different sources (e.g., from different UAs with different SSRCs) cannot be mixed by the SRC and MUST be sent separately to the SRS. All RTCP reports MUST be passed by the SRC between the UAs and the SRS, such that the UAs and SRS are able to detect any SSRC collisions.

RTCP Sender Reports generated by a UA sending a stream MUST be forwarded to the SRS. RTCP Receiver Reports generated by the SRS MUST be forwarded to the relevant UA.

UAs may receive multiple sets of RTCP Receiver Reports, one or more from other UAs participating in the CS, and one from the SRS participating in the RS. A Recording aware UA SHOULD be prepared to process the RTCP Receiver Reports from the SRS, whereas a recording unaware UA may discard such RTCP packets as not of relevance.

If SRTP is used on both the CS and the RS, decryption and/or re-encryption may occur. For example, if different keys are used, it will occur. If the same keys are used, it need not occur. **Section 13** provides additional information on SRTP and keying mechanisms.

If packet loss occurs, either from the UA to the SRC or from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss. The SRC does not play a role in this other than forwarding the associated RTP and RTCP packets.

TOC

8.2.1.2. Transcoding Translator

When acting as a transcoding translator, an SRC MAY perform transcoding (e.g., from one codec to another), and this may result in a different rate of packets between what the SRC receives and what the SRC sends. As when acting as a forwarding translator, RTP received as separate streams from different sources (e.g., from different UAs with different SSRCs) cannot be mixed by the SRC and MUST be sent separately to the SRS. All RTCP reports MUST be passed by the SRC between the UAs and the SRS, such that the UAs and SRS are able to detect any SSRC collisions.

RTCP Sender Reports generated by a UA sending a stream MUST be forwarded to the SRS. RTCP Receiver Reports generated by the SRS MUST be forwarded to the relevant UA. The SRC may need to manipulate the RTCP Receiver Reports to take account of any transcoding that has taken place.

UAs may receive multiple sets of RTCP Receiver Reports, one or more from other UAs participating in the CS, and one from the SRS participating in the RS. A Recording aware UA SHOULD be prepared to process the RTCP Receiver Reports from the SRS, whereas a recording unaware UA may discard such RTCP packets as not of relevance.

If SRTP is used on both the CS and the RS, decryption and/or re-encryption may occur. For example, if different keys are used, it will occur. If the same keys are used, it need not occur. **Section 13** provides additional information on SRTP and keying mechanisms.

If packet loss occurs, either from the UA to the SRC or from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss. The SRC does not play a role in this other than forwarding the associated RTP and RTCP packets.

TOC

8.2.2. SRC acting as an RTP Mixer

In the case of the SRC acting as a RTP mixer, as defined in **[RFC3550]**, the SRC combines RTP streams from different UA and sends them towards the SRS using its own SSRC. The SSRCs from the contributing UA SHOULD be conveyed as CSRCs identifiers within this stream. The SRC may make timing adjustments among the received streams and generate its own timing on the stream sent to the SRS. Optionally an SRC acting as a mixer can perform transcoding, and can even cope with different codings received from different UAs. RTCP Sender Reports and Receiver Reports are not forwarded by an SRC acting as mixer, but there are requirements for forwarding RTCP Source Description (SDES) packets. The SRC generates its own RTCP Sender and Receiver reports toward the associated UAs and SRS.

The use of SRTP between the SRC and the SRS for the RS is independent of the use of SRTP between the UAs and SRC for the CS. **Section 13** provides additional information on SRTP and keying mechanisms.

If packet loss occurs from the UA to the SRC, the SRC SHOULD detect and attempt to recover from the loss. If packet loss occurs from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss.

TOC

8.2.3. SRC acting as an RTP Endpoint

The case of the SRC acting as an RTP endpoint, as defined in **[RFC3550]**, is similar to the mixer case, except that the RTP session between the SRC and the SRS is considered completely independent from the RTP session that is part of the CS. The SRC can, but need not, mix RTP streams from different participants prior to sending to the SRS. RTCP between the SRC and the SRS is completely independent of RTCP on the CS.

The use of SRTP between the SRC and the SRS for the RS is independent of the use of SRTP between the UAs and SRC for the CS. **Section 13** provides additional information on SRTP and keying mechanisms.

If packet loss occurs from the UA to the SRC, the SRC SHOULD detect and attempt to recover from the loss. If packet loss occurs from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss.

TOC

8.3. RTP Session Usage by SRC

There are multiple ways that an SRC may choose to deliver recorded media to an SRS. In some cases, it may use a single RTP session for all media within the RS, whereas in others it may use multiple RTP sessions. The following subsections provide examples of basic RTP session usage by the SRC, including a discussion of how the RTP constructs and mechanisms covered previously are used. An SRC may choose to use one or more of the RTP session usages within a single RS. The set of RTP session usages described is not meant to be exhaustive.

TOC

8.3.1. SRC Using Multiple m-lines

When using multiple m-lines, an SRC includes each m-line in an SDP offer to the SRS. The SDP answer from the SRS MUST include all m-lines, with any rejected m-lines indicated with a zero port, per **[RFC3264]**. Having received the answer, the SRC starts sending media to the SRS as indicated in the answer. Alternatively, if the SRC deems the level of support indicated in the answer to be unacceptable, it may initiate another SDP offer/answer exchange in which an alternative RTP session usage is negotiated.

In order to preserve the mapping of media to participant within the CSs in the RS, the SRC SHOULD map each unique CNAME within the CSs to a unique CNAME within the RS. Additionally, the SRC SHOULD map each unique combination of CNAME/SSRC within the CSs to a unique CNAME/SSRC within the RS. In doing to, the SRC may act as an RTP translator or as an RTP endpoint.

The following figure illustrates a case in which each UA represents a participant contributing two RTP sessions (e.g. one for audio and one for video), each with a single SSRC. The SRC acts as an RTP translator and delivers the media to the SRS using four RTP sessions, each with a single SSRC. The CNAME and SSRC values used by the UAs within their media streams are preserved in the media streams from the SRC to the SRS.

```
                                                          +---------+
```

```
                          +------------SSRC Aa--->|         |
                          |  + --------SSRC Av--->|         |
                          |  |  +------SSRC Ba--->|   SRS   |
                          |  |  |  +---SSRC Bv--->|         |
                          |  |  |  |              +---------+
                          |  |  |  |
                          |  |  |  |
+---------+             +-----------+             +---------+
|         |---SSRC Aa-->|   SRC     |<--SSRC Ba---|         |
|  UA-A   |             |(CNAME-A,  |             |  UA-B   |
|(CNAME-A)|---SSRC Av-->| CNAME-B)  |<--SSRC Bv---|(CNAME-B)|
+---------+             +-----------+             +---------+
```

Figure 9: SRC Using Multiple m-lines

8.3.2. SRC Using SSRC Multiplexing

TOC

When using SSRC multiplexing, an SRC multiplexes RTP packets of the same media type from multiple RTP sessions into a single RTP session with multiple SSRC values. The SRC includes one m-line for each RTP session in an SDP offer to the SRS. The SDP answer from the SRS MUST include all m-lines, with any rejected m-lines indicated with the zero port, per **[RFC3264]**. Having received the answer, the SRC starts sending media to the SRS as indicated in the answer.

In order to preserve the mapping of media to participant within the CSs in the RS, the SRC SHOULD map each unique combination of CNAME/SSRC within the CSs to a unique SSRC within the RS. The CNAMEs used in the CSs are not preserved within the RS. The SRS relies on the SIPREC metadata to determine the participants included within each multiplexed stream. The SRC MUST avoid SSRC collisions, rewriting SSRCs if necessary. In doing to, the SRC acts as an RTP endpoint.

In the event the SRS does not support SSRC multiplexing, the SRC becomes aware of this when it receives RTCP receiver reports from the SRS indicating the absence of any packets for one or more of the multiplexed SSRC values. If the SRC deems the level of support indicated in the RTCP receiver report to be unacceptable, it may initiate another SDP offer/answer exchange in which an alternative RTP session usage is negotiated.

The following figure illustrates a case in which each UA represents a participant contributing two RTP sessions (e.g. one for audio and another for video), each with a single SSRC. The SRC delivers the media to the SRS using two RTP sessions, multiplexing one stream with the same media type from each participant into a single RTP session containing two SSRCs. The SRC uses its own CNAME and SSRC values, but it preserves the mapping of unique CNAME/SSRC used by the UAs within their media streams in the media streams from the SRC to the SRS.

```
                                                  +---------+
                                                  |         |
                            +-----SSRC SAa,SBa--->|         |
                            |   +-SSRC SAv,SBv--->|   SRS   |
                            |   |                 |         |
                            |   |                 +---------+
                            |   |
                            |   |
+---------+             +-----------+             +---------+
|         |---SSRC Aa-->|   SRC     |<--SSRC Ba---|         |
|  UA-A   |             |(CNAME-S)  |             |  UA-B   |
|(CNAME-A)|---SSRC Av-->|           |<--SSRC Bv---|(CNAME-B)|
+---------+             +-----------+             +---------+
```

Figure 10: SRC Using SSRC Multiplexing

8.3.3. SRC Using Mixing

TOC

When using mixing, the SRC combines RTP streams from different participants and sends them towards the SRS using its own SSRC. The SSRCs from the contributing participants SHOULD be conveyed as CSRCs identifiers. The SRC includes one m-line for each RTP session in an SDP offer to the SRS. The SDP answer from the SRS MUST include all m-lines, with any rejected m-lines indicated with the zero port, per **[RFC3264]**. Having received the answer, the SRC starts sending media to the SRS as indicated in the answer.

In order to preserve the mapping of media to participant within the CSs in the RS, the SRC SHOULD map each unique CNAME within the CSs to a unique CNAME within the RS. Additionally, the SRC SHOULD map each unique combination of CNAME/SSRC within the CSs to a unique CNAME/SSRC within the RS. The SRC MUST avoid SSRC collisions, rewriting SSRCs if necessary when used as CSRCs in the RS. In doing to, the SRC acts as an RTP mixer.

In the event the SRS does not support this usage of CSRC values, it relies entirely on the SIPREC metadata to determine the participants included within each mixed stream.

The following figure illustrates a case in which each UA represents a participant contributing two RTP sessions (e.g. one for audio and one for video), each with a single SSRC. The SRC acts as an RTP mixer and delivers the media to the SRS using two RTP sessions, mixing media from each participant into a single RTP session containing a single SSRC and two CSRCs.

```
                                   SSRC Sa       +---------+
                           +-------CSRC Aa,Ba--->|         |
                           |                     |         |
                           |       SSRC Sa       |   SRS   |
                           |   +---CSRC Av,Bv--->|         |
                           |   |                 +---------+
                           |   |
                        +----------+
+---------+             |   SRC    |             +---------+
|         |---SSRC Aa-->|(CNAME-S, |<--SSRC Ba---|         |
|  UA-A   |             | CNAME-A, |             |  UA-B   |
|(CNAME-A)|---SSRC Aa-->| CNAME-B) |<--SSRC Bv---|(CNAME-B)|
+---------+             +----------+             +---------+
```

Figure 11: SRC Using Mixing

9. Metadata

TOC

9.1. Procedures at the SRC

TOC

The SRC MUST deliver metadata to the SRS in a recording session; the timing of which SRC sends the metadata depends on when the metadata becomes available. Metadata SHOULD be provided by the SRC in the initial INVITE request when establishing the recording session, and subsequent metadata updates can be provided by the SRC in reINVITE and UPDATE requests (**[RFC3311]**) and responses in the recording session. There are cases that metadata is not available in the initial INVITE request sent by the SRC, for example, when a recording session is established in the absence of a communication session, and the SRC would update the recording session with metadata whenever metadata becomes available.

Certain metadata attributes are contained in the SDP, and others are contained in a new content type "application/rs-metadata". The format of the metadata is described as part of the mechanism in **[I-D.ietf-siprec-metadata]**. A new "disposition-type" of Content-Disposition is defined for the purpose of carrying metadata and the value is "recording-session". The "recording-session" value indicates that the "application/rs-metadata" content contains metadata to be handled by the SRS, and the disposition can be carried in either INVITE or UPDATE requests or responses sent by the SRC.

Metadata sent by the SRC can be categorized as either a full metadata snapshot or partial update. A full metadata snapshot describes all the recorded streams and all metadata associated with the recording session. When the SRC sends a full metadata snapshot, the SRC MUST send an INVITE or an UPDATE request (**[RFC3311]**) with an SDP offer and the "recording-session" disposition. A partial update represents an incremental update since the last metadata update sent by the SRC. A partial update sent by the SRC can be an INVITE request or response with an SDP offer, or an INVITE/UPDATE request or response containing a "recording-session" disposition, or an INVITE request containing both an SDP offer and the "recording-session" disposition.

The following is an example of a full metadata snapshot sent by the SRC in the initial INVITE request:

```
INVITE sip:recorder@example.com SIP/2.0
Via: SIP/2.0/TCP src.example.com;branch=z9hG4bKdf6b622b648d9
From: <sip:2000@example.com>;tag=35e195d2-947d-4585-946f-098392474
To: <sip:recorder@example.com>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
CSeq: 101 INVITE
Max-Forwards: 70
Require: siprec
Accept: application/sdp, application/rs-metadata,
  application/rs-metadata-request
Contact: <sip:2000@src.example.com>;+sip.src
Content-Type: multipart/mixed;boundary=foobar
Content-Length: [length]

--foobar
Content-Type: application/sdp

v=0
o=SRS 2890844526 2890844526 IN IP4 198.51.100.1
s=-
c=IN IP4 198.51.100.1
t=0 0
m=audio 12240 RTP/AVP 0 4 8
a=sendonly
a=label:1

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

[metadata content]
```

Figure 12: Sample INVITE request for the recording session

9.2. Procedures at the SRS

TOC

The SRS receives metadata updates from the SRC in INVITE and UPDATE requests. Since the SRC can send partial updates based on the previous update, the SRS needs to keep track of the sequence of updates from the SRC.

In the case of an internal failure at the SRS, the SRS may fail to recognize a partial update from the SRC. The SRS may be able to recover from the internal failure by requesting for a full metadata snapshot from the SRC. Certain errors, such as syntax errors or semantic errors in the metadata information, are likely caused by an error on the SRC side, and it is likely the same error will occur again even when a full metadata snapshot is requested. In order to avoid repeating the same error, the SRS can simply terminate the recording session when a syntax error or semantic error is detected in the metadata.

When the SRS explicitly requests for a full metadata snapshot, the SRS MUST send an UPDATE request without an SDP offer. A metadata snapshot request contains a content with the content disposition type "recording-session". Note that the SRS MAY generate an INVITE request without an SDP offer but this MUST NOT include a metadata snapshot request. The format of the content is "application/rs-metadata-request", and the body format is chosen to be a simple text-based format. The following shows an example:

```
UPDATE sip:2000@src.exmaple.com SIP/2.0
Via: SIP/2.0/UDP srs.example.com;branch=z9hG4bKdf6b622b648d9
To: <sip:2000@exmaple.com>;tag=35e195d2-947d-4585-946f-098392474
From: <sip:recorder@example.com>;tag=1234567890
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
CSeq: 1 UPDATE
Max-Forwards: 70
Require: siprec
Contact: <sip:recorder@srs.example.com>;+sip.srs
Accept: application/sdp, application/rs-metadata
Content-Disposition: recording-session
Content-Type: application/rs-metadata-request
Content-Length: [length]

SRS internal error
```

Figure 13: Metadata Request

The SRS MAY include the reason why a metadata snapshot request is being made to the SRC in the reason line. This reason line is free form text, mainly designed for logging purposes on the SRC side. The processing of the content by the SRC is entirely optional since the content is for logging only, and the snapshot request itself is indicated by the use of the application/rs-metadata-request content type.

When the SRC receives the request for a metadata snapshot, the SRC MUST provide a full metadata snapshot in a separate INVITE or UPDATE transaction, along with an SDP offer. All subsequent metadata updates sent by the SRC MUST be based on the new metadata snapshot.

TOC

9.2.1. 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 **[RFC5234]**.

snapshot-request = srs-reason-line CRLF

srs-reason-line = [TEXT-UTF8-TRIM]

TOC

10. Persistent Recording

Persistent recording is a specific use case outlined in REQ-005 or Use Case 4 in [RFC6341], where a recording session can be established in the absence of a communication session. The SRC continuously records media in a recording session to the SRS even in the absence of a CS for all user agents that are part of persistent recording. By allocating recorded streams and continuously sending recorded media to the SRS, the SRC does not have to prepare new recorded streams with new SDP offer when a new communication session is created and also does not impact the timing of the CS. The SRC only needs to update the metadata when new communication sessions are created.

When there is no communication sessions running on the devices with persistent recording, there is no recorded media to stream from the SRC to the SRS. In certain environments where Network Address Translator (NAT) is used, typically a minimum of flow activity is required to maintain the NAT binding for each port opened. Agents that support Interactive Connectivity Establishment (ICE) solves this problem. For non-ICE agents, in order not to lose the NAT bindings for the RTP/RTCP ports opened for the recorded streams, the SRC and SRS SHOULD follow the recommendations provided in [RFC6263] to maintain the NAT bindings.

TOC

11. Extensions for Recording-aware User Agents

The following sections describe the SIP and SDP extensions for recording-aware user agents. A recording-aware user agent is a participant in the CS that supports the SIP and SDP extensions for receiving recording indication and for requesting recording preferences for the call.

TOC

11.1. Procedures at the record-aware user agent

A recording-aware UA MUST indicate that it can accept reporting of recording indication provided by the SRC with a new option tag "record-aware" when initiating or establishing a CS, meaning including the "record-aware" tag in the Supported header in the initial INVITE request or response. A recording-aware UA that has indicated recording awareness MUST provide at recording indication to the end user through an appropriate user interface an indication whether recording is on or off for a given medium based on the most recently received a=record SDP attribute for that medium.

Some user agents that are automatons (e.g. IVR, media server, PSTN gateway) may not have a user interface to render recording indication. When such user agent indicates recording awareness, the UA SHOULD render 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.

TOC

11.1.1. Recording preference

A participant in a CS MAY set the recording preference in the CS to be recorded or not recorded at session establishment or during the session. The recording-aware UA sets the indication of recording preference in a new SDP attribute a=recordpref in the CS in any SDP offer/answer. This indication of recording preference can be sent at session establishment time or during the session. The SRC is not required to honor the recording preference from a participant based on local policies at the SRC; the participant gets the recording indication through the a=record SDP attribute described in the next section.

The SDP a=recordpref attribute can appear at the media level or session level and can appear in an SDP offer or answer. When the attribute is applied at the session level, the recording preference applies to all media stream in the SDP. When the attribute is applied at the media level, the recording preference applies to the media stream only, and that overrides the recording preference if also set at the session level. The user agent can change the recording preference by changing the a=recordpref attribute in subsequent SDP offer or answer. If the a=recordpref attribute is omitted, then the recording preference would be assumed to be the recording preference set in a previous SDP offer or answer.

The following is the ABNF of the recordpref attribute:

```
attribute /= recordpref-attr
; attribute defined in RFC 4566
recordpref-attr = "a=recordpref:" pref
pref = "on" / "off" / "pause" / "nopreference"
```

on
: Sets the preference to record if it has not already been started. If the recording is currently paused, the preference is to resume recording.

off
: Sets the preference for no recording. If recording has already been started, then the preference is to stop the recording.

pause
: If the recording is currently in progress, sets the preference to pause the recording.

nopreference
: To indicate that the UA has no preference on recording.

TOC

11.2. Procedures at the SRC

The SRC MUST provide recording indication to all participants in the CS. When a UA has indicated that it is recording-aware through the "record-aware" option tag, the SRC MUST provide recording indications in the new SDP a=record attribute described in the following section. In the absence of the "record-aware" option tag, meaning that the UA is not recording-aware, an SRC MUST provide recording indications through other means such as playing a tone inband, if the SRC is required to do so (e.g. based on policies).

TOC

11.2.1. 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 or session level in either SDP offer or answer. When the attribute is applied at the session level, the indication applies to all media streams in the SDP. When the attribute is applied at the media level, the indication applies to the media stream only, and that overrides the indication if also set at the session level. Whenever the recording indication needs to change, such as termination of recording, then the SRC MUST initiate a reINVITE or UPDATE to update the SDP a=record attribute.

The following is the ABNF of the 'record' attribute:

```
attribute /= record-attr
; attribute defined in RFC 4566
record-attr = "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.

If a call is traversed through one or more SIP B2BUA, and it happens that there are more than one SRC in the call path, the recording indication attribute does not provide any hint as to which SRC is performing the recording, meaning the endpoint only knows that the call is being recorded. This attribute is also not used as an indication to negotiate which SRC in the call path will perform recording and is not used as a request to start/stop recording if there are multiple SRCs in the call path.

11.2.2. Recording preference

TOC

When the SRC receives the a=recordpref SDP in an SDP offer or answer, the SRC chooses to honor the preference to record based on local policy at the SRC. When the SRC honors the preference, the SRC MUST also update the a=record attribute to indicate the current state of the recording (on/off/paused).

12. IANA Considerations

TOC

12.1. Registration of Option Tags

TOC

This specification registers two option tags. The required information for this registration, as specified in **[RFC3261]**, is as follows.

12.1.1. siprec Option Tag

TOC

Name: siprec

Description: This option tag is for identifying the SIP session for the purpose of recording session only. This is typically not used in a Supported header. When present in a Require header in a request, it indicates that the UAS MUST be either a SRC or SRS capable of handling the contexts of a recording session.

12.1.2. record-aware Option Tag

TOC

Name: record-aware

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

12.2. Registration of media feature tags

TOC

This document registers two new media feature tags in the SIP tree per the process defined in **[RFC2506]** and **[RFC3840]**

12.2.1. src feature tag

TOC

Media feature tag name: sip.src

ASN.1 Identifier: 25

Summary of the media feature indicated by this tag: This feature tag indicates that the user agent is a Session Recording Client for the purpose for Recording Session.

Values appropriate for use with this feature tag: boolean

The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is only useful for a Recording Session.

Examples of typical use: Routing the request to a Session Recording Server.

Security Considerations: Security considerations for this media feature tag are discussed in Section 11.1 of RFC 3840.

TOC

12.2.2. srs feature tag

Media feature tag name: sip.srs

ASN.1 Identifier: 26

Summary of the media feature indicated by this tag: This feature tag indicates that the user agent is a Session Recording Server for the purpose for Recording Session.

Values appropriate for use with this feature tag: boolean

The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is only useful for a Recording Session.

Examples of typical use: Routing the request to a Session Recording Client.

Security Considerations: Security considerations for this media feature tag are discussed in Section 11.1 of RFC 3840.

TOC

12.3. 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

TOC

12.4. Media Type Registration

TOC

12.4.1. 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

TOC

12.4.2. 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

12.5. SDP Attributes

TOC

This document registers the following new SDP attributes.

12.5.1. 'record' SDP Attribute

TOC

Contact names: Leon Portman leon.portman@nice.com, Henry Lum henry.lum@genesyslab.com

Attribute name: record

Long form attribute name: Recording Indication

Type of attribute: session or media level

Subject to charset: no

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

Allowed attribute values: on, off, paused

12.5.2. 'recordpref' SDP Attribute

TOC

Contact names: Leon Portman leon.portman@nice.com, Henry Lum henry.lum@genesyslab.com

Attribute name: recordpref

Long form attribute name: Recording Preference

Type of attribute: session or media level

Subject to charset: no

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

Allowed attribute values: on, off, pause, nopreference

13. Security Considerations

TOC

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. Other security considerations are outlined in the use cases and requirements document **[RFC6341]**.

13.1. RTP handling

TOC

In many scenarios it will be critical that the media transported between the SRC and SRS to be protected. Media encryption is an important element in the overall SIPREC solution, therefore, it is RECOMMENDED that SRC and SRS support RTP/SAVP **[RFC3711]** and RTP/SAVPF **[RFC5124]**. RTP/SAVP and RTP/SAVPF provide media encryption, integrity protection, replay protection, and a limited form of source authentication. They do not contain or require a specific keying mechanism.

TOC

13.2. 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.

TOC

14. Acknowledgements

We want to thank John Elwell, Paul Kyzivat, Partharsarathi R, Ram Mohan R, Hadriel Kaplan, Adam Roach, Miguel Garcia, Thomas Stach, Muthu Perumal, Dan Wing, and Magnus Westerlund for their valuable comments and inputs to this document.

TOC

15. References

TOC

15.1. Normative References

[I-D.ietf-siprec-metadata] R, R., Ravindran, P., and P. Kyzivat, "Session Initiation Protocol (SIP) Recording Metadata," draft-ietf-siprec-metadata-07 (work in progress), July 2012 (TXT).

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).

[RFC2506] Holtman, K., Mutz, A., and T. Hardie, "Media Feature Tag Registration Procedure," BCP 31, RFC 2506, March 1999 (TXT).

[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 (TXT).

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)," RFC 3264, June 2002 (TXT).

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)," RFC 3840, August 2004 (TXT).

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute," RFC 4574, August 2006 (TXT).

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," STD 68, RFC 5234, January 2008 (TXT).

TOC

15.2. Informative References

[I-D.ietf-siprec-architecture] Hutton, A., Portman, L., Jain, R., and K. Rehor, "An Architecture for Media Recording using the Session Initiation Protocol," draft-ietf-siprec-architecture-05 (work in progress), May 2012 (TXT).

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method," RFC 3311, October 2002 (TXT).

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," STD 64, RFC 3550, July 2003 (TXT, PS, PDF).

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," STD 65, RFC 3551, July 2003 (TXT, PS, PDF).

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," RFC 3711, March 2004 (TXT).

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)," RFC 4585, July 2006 (TXT).

[RFC4961] Wing, D., "**Symmetric RTP / RTP Control Protocol (RTCP)**," BCP 131, RFC 4961, July 2007 (**TXT**).

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "**Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)**," RFC 5104, February 2008 (**TXT**).

[RFC5124] Ott, J. and E. Carrara, "**Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)**," RFC 5124, February 2008 (**TXT**).

[RFC5168] Levin, O., Even, R., and P. Hagendorf, "**XML Schema for Media Control**," RFC 5168, March 2008 (**TXT**).

[RFC5761] Perkins, C. and M. Westerlund, "**Multiplexing RTP Data and Control Packets on a Single Port**," RFC 5761, April 2010 (**TXT**).

[RFC6222] Begen, A., Perkins, C., and D. Wing, "**Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)**," RFC 6222, April 2011 (**TXT**).

[RFC6263] Marjou, X. and A. Sollaud, "**Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows**," RFC 6263, June 2011 (**TXT**).

[RFC6341] Rehor, K., Portman, L., Hutton, A., and R. Jain, "**Use Cases and Requirements for SIP-Based Media Recording (SIPREC)**," RFC 6341, August 2011 (**TXT**).

Authors' Addresses

TOC

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

Henry Lum (editor)
Genesys
1380 Rodick Road, Suite 201
Markham, Ontario L3R4G5
Canada
Email: henry.lum@genesyslab.com

Charles Eckel
Cisco
170 West Tasman Drive
San Jose, CA 95134
United States
Email: eckelcu@cisco.com

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

Andrew Hutton
Siemens Enterprise Communications
Brickhill Street
Milton Keynes MK15 0DJ
United Kingdom
Email: andrew.hutton@siemens-enterprise.com