Session Initiation Protocol (SIP) Recording Metadata

Abstract

   Session recording is a critical requirement in many communications
   environments, such as call centers and financial trading
   organizations. In some of these environments, all calls must be
   recorded for regulatory, compliance, and consumer protection reasons.
   The recording of a session is typically performed by sending a copy
   of a media stream to a recording device. This document describes the
   metadata model as viewed by the Session Recording Server (SRS) and
   the recording metadata format.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7865.

Copyright Notice

Table of Contents

1. Introduction

   Session recording is a critical requirement in many communications
   environments, such as call centers and financial trading
   organizations. In some of these environments, all calls must be
   recorded for regulatory, compliance, and consumer protection reasons.
   The recording of a session is typically performed by sending a copy
   of a media stream to a recording device. This document focuses on
   the recording metadata, which describes the Communication Session
   (CS). The document describes a metadata model as viewed by the
   Session Recording Server (SRS) and the recording metadata format, the
   requirements for which are described in [RFC6341] and the
   architecture for which is described in [RFC7245].

2. Terminology

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC2119]. This
   document only uses these key words when referencing normative
   statements in existing RFCs.

3. Definitions

   Metadata model: A metadata model is an abstract representation of
      metadata using a Unified Modeling Language (UML) [UML] class
      diagram.

   Metadata classes: Each block in the model represents a class. A
      class is a construct that is used as a blueprint to create
      instances (called "objects") of itself. The description of each
      class also has a representation of its attributes in a second
      compartment below the class name.

   Attributes: Attributes represent the elements listed in each of the
      classes. The attributes of a class are listed in the second
      compartment below the class name. Each instance of a class
      conveys values for the attributes of that class. These values get
      added to the recording's metadata.

   Linkages: Linkages represent the relationship between the classes in
      the model. Each linkage represents a logical connection between
      classes (or objects) in class diagrams (or object diagrams). The
      linkages used in the metadata model of this document are
      associations.

   This document also refers to the terminology defined in [RFC6341].

4. Metadata Model

   Metadata is the information that describes recorded media and the CS
   to which they relate. The diagram below shows a model for metadata
   as viewed by an SRS.

```
                    +------------------------------+ 1..*
                    |     Recording Session (RS)   |----+
                    +------------------------------+    |
                         | 1..*         | 1..*          |
                         |              |               |
                         |              | 0..*          |
                         |         +----------------+   |
      +-----------+      |         | Communication  |   |
      | CS-RS     |      |         | Session Group  |   |
      | Association|--+  |         |   (CS-Group)   |   |
      |           |   |  |         +----------------+   |
      +-----------+   |  |              | 0..1          |
                      |  |              |               |
                      |  | 0..*         | 1..*          |
      +------------------------------+  |               |
      |   Communication Session (CS) |  |               |
      |                              |  |               |
      +------------------------------+  |               |
          | 1..*               | 0..1   |               |
      +-----+                  |        |               |
      |     | 0..*             | 0..*   |          | 0..*
      | +-------------+ receives +----------------+ |
      | | Participant |----------|  Media Stream  |--+
      | |             |0..*  0..*|                |
      | |             |          |                |
      | |             |          |                |
      | |             |  sends   |                |
      | |             |----------|                |
      | |             |1.*   0..*|                |
      | +-------------+          +----------------+
      |        |                    |
      |        |                    |
      |        +--------------------+-----------+
      |                                         |
      |                                         |
      |   +-----------------+   +--------------------+
      |   |Participant-CS   |   |  Participant-Stream |
      +---|  Association    |   |    Association      |
          |                 |   |                     |
          +-----------------+   +--------------------+
```
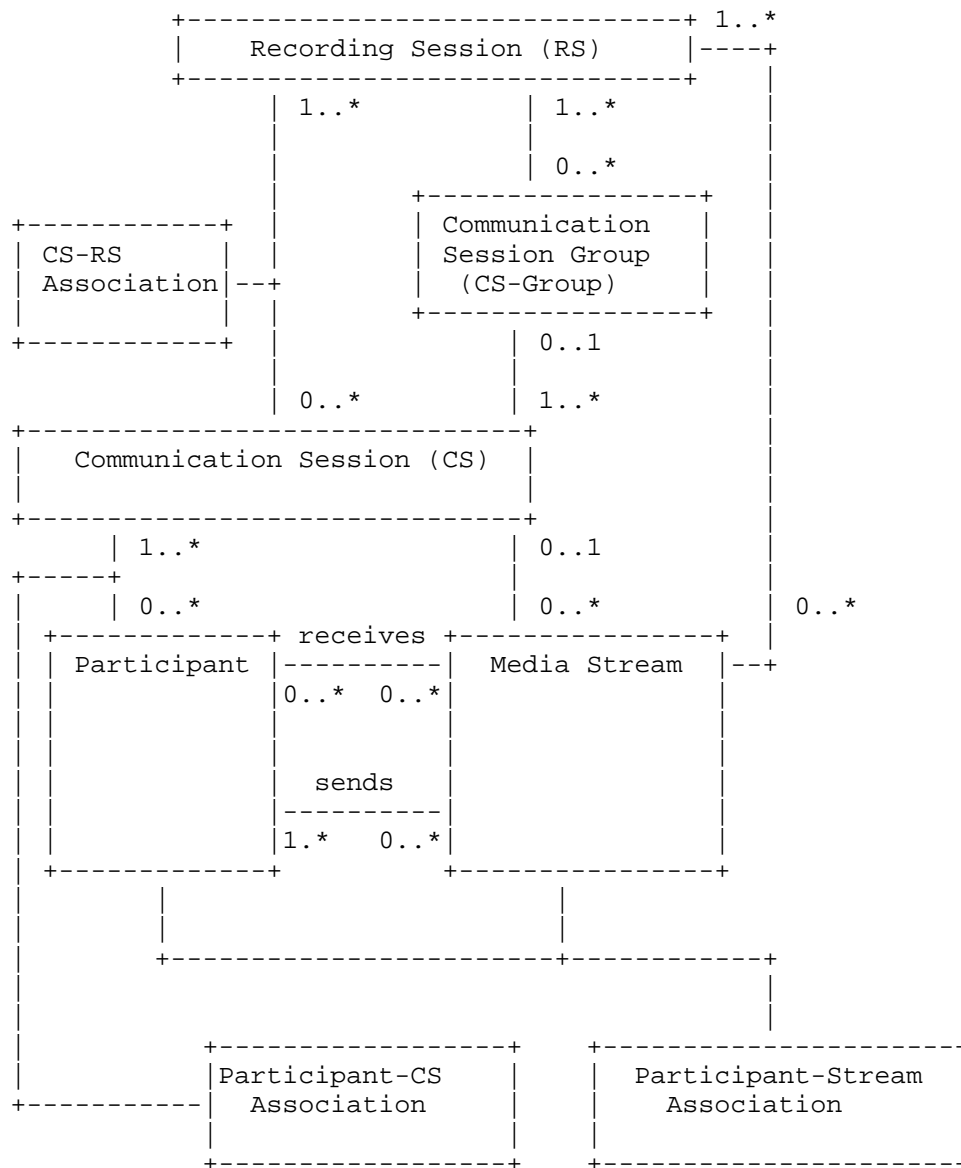
The metadata model is a class diagram in UML. The model describes
the structure of metadata in general by showing the classes, their
attributes, and the relationships among the classes. Each block in
the model above represents a class. The linkages between the classes
represent the relationships, which can be associations or
compositions. The metadata is conveyed from the Session Recording
Client (SRC) to the SRS.

The model allows metadata describing CSs to be communicated to the
SRS as a series of snapshots, each representing the state as seen by
a single SRC at a particular instant in time. Metadata changes from
one snapshot to another reflect changes in what is being recorded.
For example, if a participant joins a conference, then the SRC sends
the SRS a snapshot of metadata having that participant information
(with attributes like (Name, AoR) tuple and associate-time). (Note:
"AoR" means "Address-of-Record".)

Some of the metadata is not required to be conveyed explicitly from
the SRC to the SRS, if it can be obtained contextually by the SRS
(e.g., from SIP or Session Description Protocol (SDP) signaling).
For example, the 'label' attribute within the 'stream' XML element
references an SDP "a=label" attribute that identifies an m-line
within the Recording Session (RS) SDP. The SRS would learn the media
properties from the media line.

5. Recording Metadata Format from SRC to SRS

   This section gives an overview of the recording metadata format.
   Some data from the metadata model is assumed to be made available to
   the SRS through SDP [RFC4566], and therefore this data is not
   represented in the XML document format specified in this document.
   SDP attributes describe different media formats like audio and video.
   The other metadata attributes, such as participant details, are
   represented in a new recording-specific XML document of type
   'application/rs-metadata+xml'. The SDP "label" attribute [RFC4574]
   provides an identifier by which a metadata XML document can refer to
   a specific media description in the SDP sent from the SRC to the SRS.

   The XML document format can be used to represent either the complete
   metadata or a partial update to the metadata. The latter includes
   only elements that have changed compared to the previously reported
   metadata.

5.1. XML Data Format

   Every recording metadata XML document sent from the SRC to the SRS
   contains a 'recording' element. The 'recording' element acts as a
   container for all other elements in this XML document. A 'recording'
   object is an XML document. It has the XML declaration and contains
   an encoding declaration in the XML declaration, e.g.,
   "<?xml version="1.0" encoding="UTF-8"?>". If the charset parameter
   of the MIME content type declaration is present and it is different
   from the encoding declaration, the charset parameter takes
   precedence.

   Every application conforming to this specification MUST accept the
   UTF-8 character encoding to ensure minimal interoperability.

   Syntax and semantic errors in an XML document should be reported to
   the originator, using application-specific mechanisms.

5.1.1. Namespace

   With the following URN, this document defines a new namespace URI for
   elements defined herein:

   urn:ietf:params:xml:ns:recording:1

5.1.2. 'recording' Element

   The 'recording' element MUST contain an xmlns namespace attribute
   with a value of urn:ietf:params:xml:ns:recording:1. Exactly one
   'recording' element MUST be present in every recording metadata XML
   document.

   A 'recording' element MAY contain a 'dataMode' element indicating
   whether the XML document is a complete document or a partial update.
   If no 'dataMode' element is present, then the default value is
   "complete".

6. Recording Metadata Classes

   This section describes each class of the metadata model and the
   attributes of each class. This section also describes how different
   classes are linked and the XML element for each of them.

6.1. Recording Session

```
      +-----------------------------+
      | Recording Session (RS)      |
      +-----------------------------+
      |                             | 1..*      0..*
      |      start-time             |------------- Media Stream
      |      end-time               |
      |                             |
      |                             |
      +-----------------------------+
          | 1..*            | 1..*
          |                 |
          | 0..*            | 0..*
       Communication     Communication
       Session (CS)      Session Group (CS-Group)
```
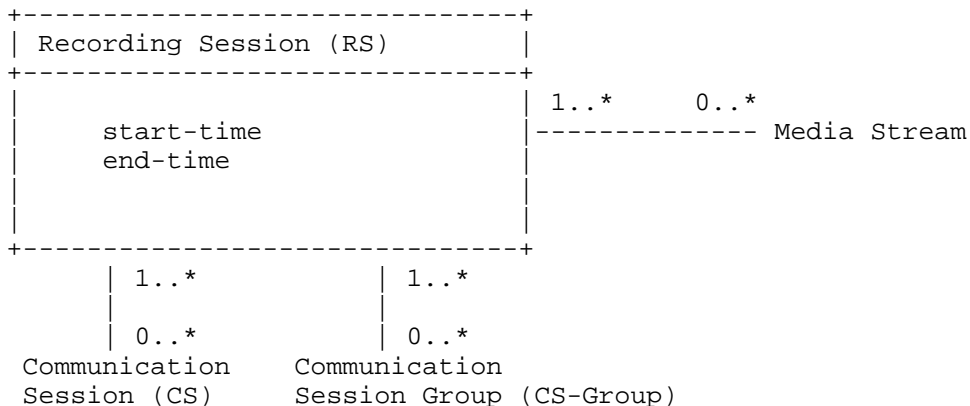
   Each instance of an RS class, namely the RS object, represents a SIP
   session created between an SRC and SRS for the purpose of recording
   a CS.

   The RS object is represented in the XML schema using the 'recording'
   element, which in turn relies on the SIP/SDP session with which the
   XML document is associated to provide the attributes of the RS
   element.

6.1.1. Attributes

   An RS class has the following attributes:

   o start-time - Represents the start time of an RS object.

   o end-time - Represents the end time of an RS object.

   'start-time' and 'end-time' attribute values are derivable from the
   Date header (if present in the SIP message) in the RS. In cases
   where the Date header is not present, 'start-time' is derivable from
   the time at which the SRS receives the notification of the SIP
   message to set up the RS, and 'end-time' is derivable from the time
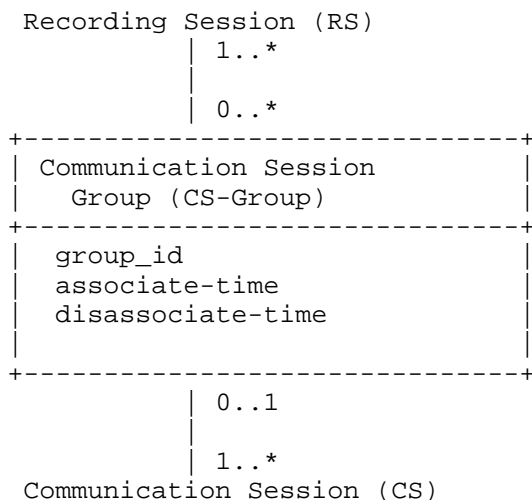   at which the SRS receives a disconnect on the RS SIP dialog.

6.1.2. Linkages

   Each instance of an RS has:

   o Zero or more instances of CS-Groups.

   o Zero or more instances of CS objects.

   o Zero or more instances of MediaStream objects.

   Zero instances of CSs and CS-Groups in a 'recording' element are
   allowed to accommodate persistent recording scenarios. A persistent
   RS is a SIP dialog that is set up between the SRC and the SRS, even
   before any CS is set up. The metadata sent from the SRC to the SRS
   when the persistent RS SIP dialog is set up may not have any CS (and
   the related CS-Group) elements in the XML, as there may not be a
   session that is associated to the RS yet. For example, a phone
   acting as an SRC can set up an RS with the SRS, possibly even before
   the phone is part of a CS. Once the phone joins a CS, the same RS
   would be used to convey the CS metadata.

6.2. Communication Session Group

```
     Recording Session (RS)
               | 1..*
               |
               | 0..*
     +-------------------------------+
     | Communication Session         |
     |    Group (CS-Group)           |
     +-------------------------------+
     |   group_id                    |
     |   associate-time              |
     |   disassociate-time           |
     |                               |
     +-------------------------------+
               | 0..1
               |
               | 1..*
      Communication Session (CS)
```

   One instance of a CS-Group class, namely the CS-Group object,
   provides association or grouping of all related CSs. For example, in
   a contact center flow, a call can get transferred to multiple agents.
   Each of these can trigger the setup of a new CS. In cases where the
   SRC knows the related CSs, it can group them using the CS-Group
   element. The CS-Group object is represented in the XML schema using
   the 'group' element.

6.2.1. Attributes

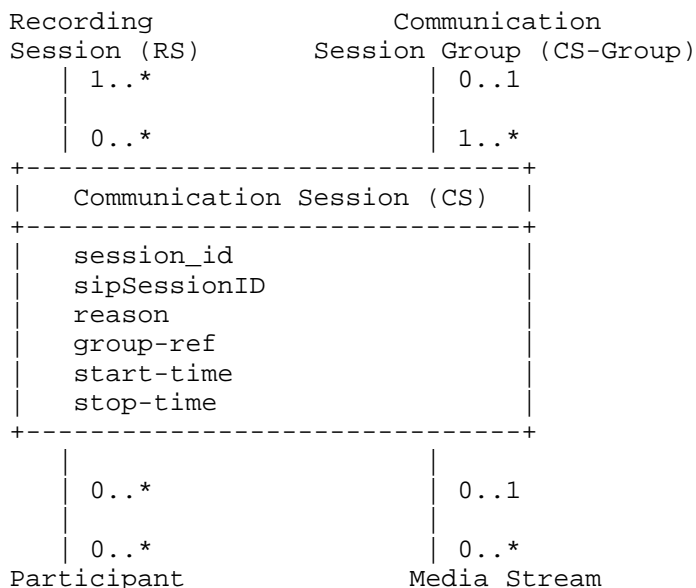   A CS-Group has the following attributes:

   o group_id - This attribute groups different CSs that are related.
      The SRC (or the SRS) is responsible for ensuring the uniqueness of
      'group_id' in cases where multiple SRCs interact with the same
      SRS. The mechanism by which the SRC groups the CS is outside the
      scope of this document.

   o associate-time - This is the time when a grouping is formed. The
      rules that determine how a grouping of different CS objects is
      done by the SRC are outside the scope of this document.

   o disassociate-time - 'disassociate-time' for the CS-Group is
      calculated by the SRC as the time when the grouping ends.

6.2.2. Linkages

   The linkages between a CS-Group class and other classes are
   associations. A CS-Group is associated with the RS and CS in the
   following manner:

   o There are one or more RS objects per CS-Group.

   o Each CS-Group object has to be associated with one or more RSs.
      Here, each RS can be set up by the potentially different SRCs.

   o There are one or more CSs per CS-Group (for example, in cases
      where the call is transferred). A CS cannot be associated with
      more than one CS-Group.

6.3. Communication Session

```
   Recording                    Communication
   Session (RS)           Session Group (CS-Group)
      | 1..*                      | 0..1
      |                          |
      | 0..*                      | 1..*
    +-------------------------------+
    |    Communication Session (CS)  |
    +-------------------------------+
    |    session_id                 |
    |    sipSessionID               |
    |    reason                     |
    |    group-ref                  |
    |    start-time                 |
    |    stop-time                  |
    +-------------------------------+
      |                          |
      | 0..*                      | 0..1
      |                          |
      | 0..*                      | 0..*
   Participant                Media Stream
```

   A CS class and its object in the metadata model represent the CS and
   its properties as seen by the SRC. The CS object is represented in
   the XML schema using the 'session' element.

6.3.1. Attributes

   A CS class has the following attributes:

   o session_id - This attribute is used to uniquely identify an
      instance of a CS object, namely the 'session' XML element within
      the metadata XML document. 'session_id' is generated using the
      rules mentioned in Section 6.10.

   o reason - This represents the reason why a CS was terminated. The
      value for this attribute is derived from the SIP Reason header
      [RFC3326] of the CS. There MAY be multiple instances of the
      'reason' XML element inside a 'session' element. The 'reason' XML
      element has 'protocol' as an attribute, which indicates the
      protocol from which the reason string is derived. The default
      value for the 'protocol' attribute is "SIP". The 'reason' element
      can be derived from a SIP Reason header in the CS.

   o sipSessionID - This attribute carries a SIP Session-ID as defined
      in [SessionID]. Each CS object can have zero or more
      'sipSessionID' elements. More than one 'sipSessionID' attribute
      may be present in a CS. For example, if three participants -- A,
      B, and C -- are in a conference that has a focus acting as an SRC,
      the metadata sent from the SRC to the SRS will likely have three
      'sipSessionID' elements that correspond to the SIP dialogs that
      the focus has with each of the three participants.

   o group-ref - A 'group-ref' attribute MAY be present to indicate the
      group (identified by 'group_id') to which the enclosing session
      belongs.

   o start-time - This optional attribute represents the start time of
      the CS as seen by the SRC.

   o stop-time - This optional attribute represents the stop time of
      the CS as seen by the SRC.

   This document does not specify attributes relating to what should
   happen to a recording of a CS after it has been delivered to the SRS
   (e.g., how long to retain the recording, what access controls to
   apply). The SRS is assumed to behave in accordance with its local
   policy. The ability of the SRC to influence this policy is outside
   the scope of this document. However, if there are implementations
   where the SRC desires to specify its own policy preferences, this
   information could be sent as extension data attached to the CS.

6.3.2. Linkages

   A CS is linked to the CS-Group, participant, MediaStream (MS), and
   RS classes by using the association relationship. The association
   between the CS and the participant allows the following:

   o A CS will have zero or more participants.

   o A participant is associated with zero or more CSs. This includes
      participants who are not directly part of any CS. An example of
      such a case is participants in a pre-mixed media stream. The SRC
      may have knowledge of such participants but not have any signaling
      relationship with them. This might arise if one participant in a
      CS is a conference focus. To summarize, even if the SRC does not
      have direct signaling relationships with all participants in a CS,
      it should nevertheless create a participant object for each
      participant that it knows about.

   o The model also allows participants in a CS that are not
      participants in the media. An example is the identity of a Third
      Party Call Control (3pcc) that has initiated a CS to two or more
      participants in the CS. Another example is the identity of a
      conference focus. Of course, a focus is probably in the media,
      but since it may only be there as a mixer, it may not report
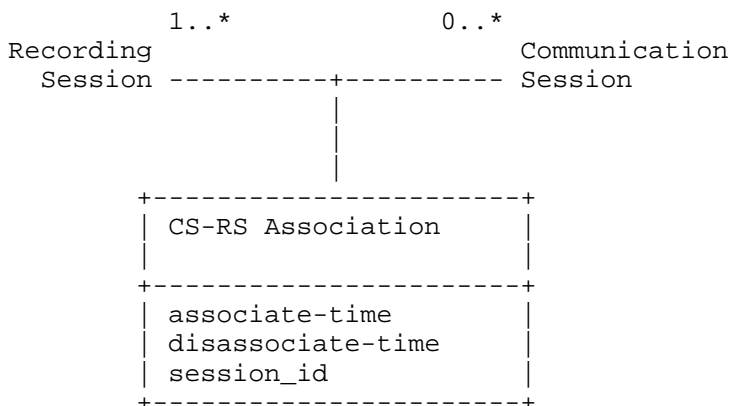   itself as a participant in any of the media streams.

   The association between the CS and the media stream allows the
   following:

   o A CS will have zero or more streams.

   o A stream can be associated with at most one CS. A stream in a
      persistent RS is not required to be associated with any CS before
      the CS is created, and hence the zero association is allowed.

   The association between the CS and the RS allows the following:

   o Each instance of an RS has zero or more instances of CS objects.

   o Each CS has to be associated with one or more RSs. Each RS can be
      potentially set up by different SRCs.

6.4. CS-RS Association

```
                        1..*               0..*
              Recording                          Communication
                Session ----------+---------- Session
                                  |
                                  |
                                  |
                                  |
                  +----------------------+
                  | CS-RS Association    |
                  |                      |
                  +----------------------+
                  | associate-time       |
                  | disassociate-time    |
                  | session_id           |
                  +----------------------+
```

   The CS-RS Association class describes the association of a CS to an
   RS for a period of time. A single CS may be associated with
   different RSs (perhaps by different SRCs) and may be associated and
   dissociated several times.

   The CS-RS Association class is represented in XML using the
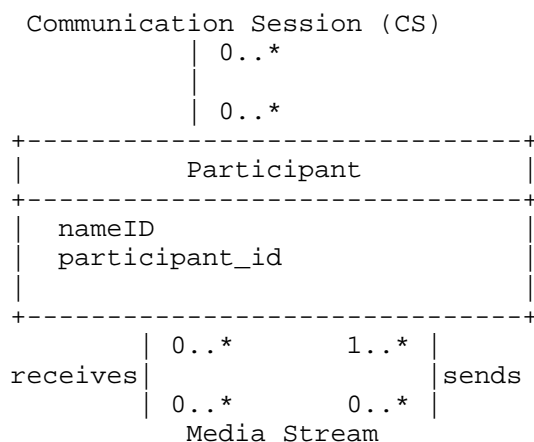   'sessionrecordingassoc' XML element.

6.4.1. Attributes

   The CS-RS Association class has the following attributes:

   o associate-time - associate-time is calculated by the SRC as the
      time it sees a CS associated to an RS.

   o disassociate-time - disassociate-time is calculated by the SRC as
      the time it sees a CS disassociate from an RS.

   o session_id - Each instance of this class MUST have a 'session_id'
      attribute that identifies the CS to which this association
      belongs.

6.4.2. Linkages

   The CS-RS Association class is linked to the CS and RS classes.

6.5. Participant

```
    Communication Session (CS)
              | 0..*
              |
              | 0..*
    +------------------------------+
    |           Participant        |
    +------------------------------+
    |   nameID                     |
    |   participant_id             |
    |                              |
    +------------------------------+
           | 0..*        1..* |
    receives|                 |sends
           | 0..*        0..* |
              Media Stream
```

   A participant class and its objects have information about a device
   that is part of a CS and/or contributes/consumes media stream(s)
   belonging to a CS.

   The participant object is represented in the XML schema using the
   'participant' element.

6.5.1. Attributes

   A participant class has two attributes:

   o nameID - This attribute is a list of (Name, AoR) tuples. An AoR
      (Section 6 of [RFC3261]) can be either a SIP/SIPS/tel URI ("SIPS"
      means "SIP Secure"; the tel URI is discussed in [RFC3966]), a
      Fully Qualified Domain Name (FQDN), or an IP address. For
      example, the AoR may be drawn from the From header field or the
      P-Asserted-Identity header [RFC3325] field. The SRC's local
      policy is used to decide where to draw the AoR from. The Name
      parameter represents the participant name (SIP display name) or
      dialed number (when known). Multiple tuples are allowed for cases
      where a participant has more than one AoR. For example, a
      P-Asserted-Identity header can have both SIP and tel URIs.

   o participant_id - This attribute is used to identify the
      'participant' XML element within the XML document. It is
      generated using the rules mentioned in Section 6.10. This
      attribute MUST be used for all references to a participant within
      a CS-Group, and MAY be used to reference the same participant more
      globally.

   This document does not specify other attributes relating to
   participants (e.g., participant role, participant type). An SRC that
   has information regarding these attributes can provide this
   information as part of extension data to the 'participant' XML
   element from the SRC to the SRS.
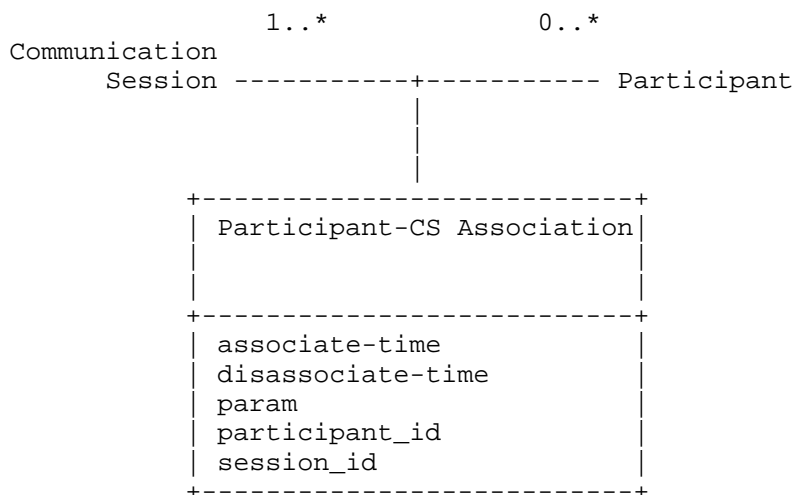
6.5.2. Linkages

   The participant class is linked to the MS and CS classes by using an
   association relationship. The association between the participant
   and the MS allows the following:

   o A participant will receive zero or more media streams.

   o A participant will send zero or more media streams. (The same
      participant provides multiple streams, e.g., audio and video.)

   o A media stream will be received by zero or more participants. It
      is possible, though perhaps unlikely, that a stream is generated
      but sent only to the SRC and SRS, not to any participant -- for
      example, in conferencing where all participants are on hold and
      the SRC is collocated with the focus. Also, a media stream may be
      received by multiple participants (e.g., "whisper" calls, side
      conversations).

   o A media stream will be sent by one or more participants (pre-mixed
      streams).

   An example of a case where a participant receives zero or more
   streams is where a supervisor may have a side conversation with an
   agent while the agent converses with a customer.

6.6. Participant-CS Association

```
                        1..*              0..*
         Communication
             Session ----------+---------- Participant
                               |
                               |
                               |
                               |
                 +---------------------------+
                 | Participant-CS Association|
                 |                           |
                 |                           |
                 +---------------------------+
                 | associate-time            |
                 | disassociate-time         |
                 | param                     |
                 | participant_id            |
                 | session_id                |
                 +---------------------------+
```

   The Participant-CS Association class describes the association of a
   participant to a CS for a period of time. A participant may be
   associated to and dissociated from a CS several times (for example,
   connecting to a conference, then disconnecting, then connecting
   again).

   The Participant-CS Association object is represented in the XML
   schema using the 'participantsessionassoc' element.
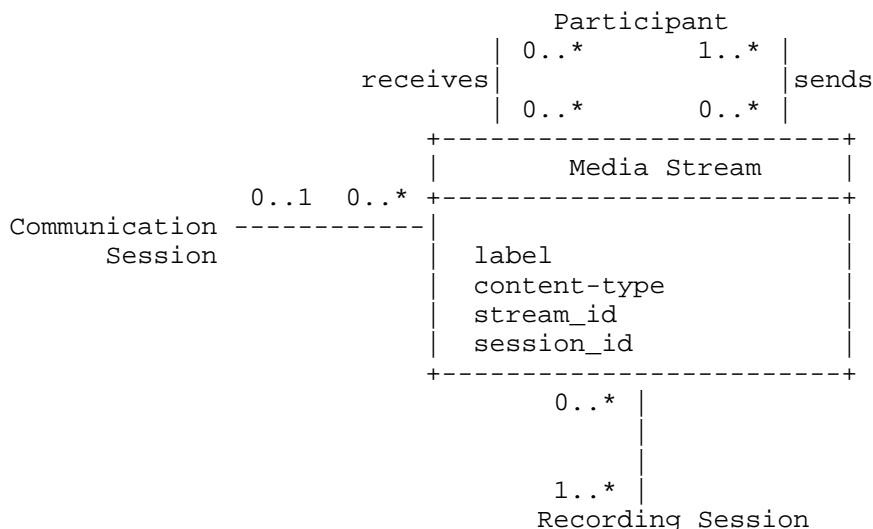
6.6.1. Attributes

   The Participant-CS Association class has the following attributes:

   o associate-time - associate-time is calculated by the SRC as the
      time it sees a participant associated to a CS.

   o disassociate-time - disassociate-time is calculated by the SRC as
      the time it sees a participant disassociate from a CS. It is
      possible that a given participant can have multiple associate
      times / disassociate times within a given communication session.

   o param - The capabilities here are those that are indicated in the
      Contact header as defined in Section 9 of [RFC3840]. For example,
      in a CS (which can be a conference), you can have participants who
      are playing the role of "focus". These participants do not
      contribute to media in the CS; however, they switch the media
      received from one participant to every other participant in the
      CS. Indicating the capabilities of the participants (here,
      "focus") would be useful for the recorder to learn about these
      kinds of participants. The capabilities are represented using the
      'param' XML element in the metadata. The 'param' XML element
      encoding defined in [RFC4235] is used to represent the capability
      attributes in metadata. Each participant may have zero or more
      capabilities. A participant may use different capabilities,
      depending on the role it plays at a particular instance -- for
      example, if a participant moves across different CSs (e.g., due to
      transfer) or is simultaneously present in different CSs with
      different roles.

   o participant_id - This attribute identifies the participant to
      which this association belongs.

   o session_id - This attribute identifies the session to which this
      association belongs.

6.6.2. Linkages

   The Participant-CS Association class is linked to the participant and
   CS classes.

6.7. Media Stream

```
                                 Participant
                                │ 0..*        1..* │
                          receives│                  │sends
                                │ 0..*        0..* │
                         +-------------------------+
                         │      Media Stream       │
              0..1  0..* +-------------------------+
       Communication ------------│                         │
           Session              │   label                 │
                                │   content-type          │
                                │   stream_id             │
                                │   session_id            │
                         +-------------------------+
                                  0..* │
                                       │
                                       │
                                  1..* │
                          Recording Session
```

   A MS class (and its objects) has the properties of media as seen by
   the SRC and sent to the SRS. Different snapshots of MS objects may
   be sent whenever there is a change in media (e.g., a direction
   change, like pause/resume, codec change, and/or participant change).

   The MS object is represented in the XML schema using the 'stream'
   element.

6.7.1. Attributes

   A MS class has the following attributes:

   o label - The 'label' attribute within the 'stream' XML element
      references an SDP "a=label" attribute that identifies an m-line
      within the RS SDP. That m-line carries the media stream from the
      SRC to the SRS.

   o content-type - The content of a MS element will be described in
      terms of the "a=content" attribute defined in Section 5 of
      [RFC4796]. If the SRC wishes to convey the content-type to the
      SRS, it does so by including an "a=content" attribute with the
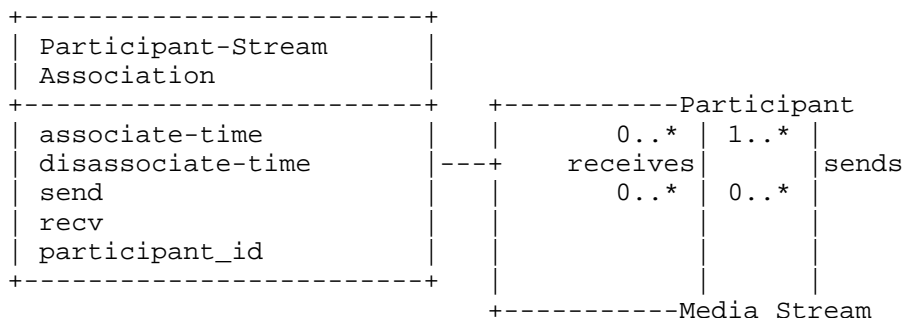   m-line in the RS SDP.

   o stream_id - Each 'stream' element has a unique 'stream_id'
      attribute that helps to uniquely identify the stream. This
      identifier is generated using the rules mentioned in Section 6.10.

   o session_id - This attribute associates the stream with a specific
      'session' element.

   The metadata model can include media streams that are not being
   delivered to the SRS. For example, an SRC offers audio and video
   towards an SRS that accepts only audio in response. The metadata
   snapshots sent from the SRC to the SRS can continue to indicate the
   changes to the video stream as well.

6.7.2. Linkages

   A MS class is linked to the participant and CS classes by using the
   association relationship. Details regarding associations with the
   participant are described in Section 6.5. Details regarding
   associations with the CS are mentioned in Section 6.3.

6.8. Participant-Stream Association

```
    +------------------------+
    | Participant-Stream     |
    | Association            |
    +------------------------+   +----------Participant
    | associate-time         |   |      0..* | 1..* |
    | disassociate-time      |---+   receives|      |sends
    | send                   |   |      0..* | 0..* |
    | recv                   |   |           |      |
    | participant_id         |   |           |      |
    +------------------------+   |           |      |
                                 +----------Media Stream
```

   A Participant-Stream Association class describes the association of a
   participant to a MS for a period of time, as a sender or as a
   receiver, or both.

   This class is represented in XML using the 'participantstreamassoc'
   element.

6.8.1. Attributes

   A Participant-Stream Association class has the following attributes:

   o associate-time - This attribute indicates the time a participant
      started contributing to a MS.

   o disassociate-time - This attribute indicates the time a
      participant stopped contributing to a MS.

   o send - This attribute indicates whether a participant is
      contributing to a stream or not. This attribute has a value that
      points to a stream represented by its unique_id. The presence of
      this attribute indicates that a participant is contributing to a
      stream. If a participant stops contributing to a stream due to
      changes in a CS, a snapshot MUST be sent from the SRC to the SRS
      with no 'send' element for that stream.

   o recv - This attribute indicates whether a participant is receiving
      a media stream or not. This attribute has a value that points to
      a stream represented by its unique_id. The presence of this
      attribute indicates that a participant is receiving a stream. If
      the participant stops receiving a stream due to changes in a CS
      (like hold), a snapshot MUST be sent from the SRC to the SRS with
      no 'recv' element for that stream.

   o participant_id - This attribute points to the participant with
      which a 'stream' element is associated.

   The 'participantstreamassoc' XML element is used to represent a
   participant association with a stream. The 'send' and 'recv' XML
   elements MUST be used to indicate whether a participant is
   contributing to a stream or receiving a stream. There MAY be
   multiple instances of the 'send' and 'recv' XML elements inside a
   'participantstreamassoc' element. If a metadata snapshot is sent
   with a 'participantstreamassoc' element that does not have any 'send'
   and 'recv' elements, it means that the participant is neither
   contributing to any streams nor receiving any streams.

6.8.2. Linkages

   The Participant-Stream Association class is linked to the participant
   and MS classes.

6.9. Syntax of XML Elements for Date and Time

   The XML elements 'associate-time', 'disassociate-time', 'start-time',
   and 'stop-time' contain strings representing the date and time. The
   value of these elements MUST follow the Instant Messaging and
   Presence Protocol (IMPP) date-time format [RFC3339]. Timestamps that
   contain "T" or "Z" MUST use the capitalized forms.

   As a security measure, the 'timestamp' element MUST be included in
   all tuples, unless the exact time of the status change cannot be
   determined.

6.10. Format of Unique ID

   A unique_id is generated in two steps:

   o The Universally Unique Identifier (UUID) is created using any of
      the procedures mentioned in Sections 4.3, 4.4, and 4.5 of
      [RFC4122]. The algorithm MUST ensure that it does not use any
      potentially personally identifying information to generate the
      UUIDs. If implementations are using a Name-Based UUID as defined
      in Section 4.3 of [RFC4122], a namespace ID generated using the
      guidance in Section 4.2 or 4.5 of [RFC4122] might be a good
      choice.

   o The UUID is encoded using base64 as defined in [RFC4648].

   The above-mentioned unique_id mechanism SHOULD be used for each
   metadata element. Multiple SRCs can refer to the same element/UUID
   (how each SRC learns the UUID here is beyond the scope of this
   document). If two SRCs use the same UUID, they MUST retain the
   UUID/element mapping. If the SRS detects that a UUID is mapped to
   more than one element at any point in time, it MUST treat this as an
   error. For example, the SRS may choose to reject or ignore the
   portions of metadata where it detects that the same UUID is mapped to
   an element that is different than the expected element (the SRS
   learns the mapped UUID when it sees an element for the first time in
   a metadata instance).

6.11. Metadata Version Indicator

   The Metadata version is defined to help the SRC and SRS know the
   version of metadata XML schema used. SRCs and SRSs that support this
   specification MUST use version 1 in the namespace
   (urn:ietf:params:xml:ns:recording:1) in all the XML documents.
   Implementations may not interoperate if the version implemented by
   the sender is not known by the receiver. No negotiation of versions
   is provided. The version number has no significance, although

documents that update or obsolete this document (possibly including
drafts of such documents) should include a higher version number if
the metadata XML schema changes.

7. Recording Metadata Snapshot Request Format

   The SRS can explicitly request a metadata snapshot from the SRC. To
   request a metadata snapshot, the SRS MUST send a SIP request message
   with an XML document having the namespace
   urn:ietf:params:xml:ns:recording:1. The XML document has the
   following elements:

   o A 'requestsnapshot' XML element MUST be present as the top-level
      element in the XML document.

   o A 'requestreason' XML element that indicates the reason (as a
      string) for requesting the snapshot MAY be present as a child XML
      element of 'requestsnapshot'.

   The example below shows a metadata snapshot request from the SRS.

   <?xml version="1.0" encoding="UTF-8"?>
     <requestsnapshot xmlns='urn:ietf:params:xml:ns:recording:1'>
     <requestreason xml:lang="it">SRS internal error</requestreason>
   </requestsnapshot>

              Example Metadata Snapshot Request from SRS to SRC

8. SIP Recording Metadata Examples

8.1. Complete SIP Recording Metadata Example

   The following example provides all the tuples involved in the
   recording metadata XML body.

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording:1'>
  <datamode>complete</datamode>
  <group group_id="7+OTCyoxTmqmqyA/1weDAg==">
    <associate-time>2010-12-16T23:41:07Z</associate-time>
    <!-- Standardized extension -->
    <call-center xmlns='urn:ietf:params:xml:ns:callcenter'>
         <supervisor>sip:alice@atlanta.com</supervisor>
    </call-center>
    <mydata xmlns='http://example.com/my'>
         <structure>FOO!</structure>
         <whatever>bar</whatever>
    </mydata>
  </group>
  <session session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <sipSessionID>ab30317f1a784dc48ff824d0d3715d86;
        remote=47755a9de7794ba387653f2099600ef2</sipSessionID>
        <group-ref>7+OTCyoxTmqmqyA/1weDAg==</group-ref>
        <!-- Standardized extension -->
    <mydata xmlns='http://example.com/my'>
         <structure>FOO!</structure>
          <whatever>bar</whatever>
        </mydata>
  </session>
  <participant participant_id="srfBElmCRp2QB23b7Mpk0w==">
        <nameID aor="sip:bob@biloxi.com">
           <name xml:lang="it">Bob</name>
        </nameID>
        <!-- Standardized extension -->
        <mydata xmlns='http://example.com/my'>
              <structure>FOO!</structure>
              <whatever>bar</whatever>
        </mydata>
  </participant>
```

```
     <participant participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
           <nameID aor="sip:Paul@biloxi.com">
             <name xml:lang="it">Paul</name>
           </nameID>
           <!-- Standardized extension -->
           <mydata xmlns='http://example.com/my'>
              <structure>FOO!</structure>
              <whatever>bar</whatever>
           </mydata>
     </participant>
     <stream stream_id="UAAMm5GRQKSCMVvLyl4rFw=="
             session_id="hVpd7YQgRW2nD22h7q60JQ==">
           <label>96</label>
     </stream>
     <stream stream_id="i1Pz3to5hGk8fuXl+PbwCw=="
             session_id="hVpd7YQgRW2nD22h7q60JQ==">
            <label>97</label>
     </stream>
     <stream stream_id="8zc6e0lYTlWIINA6GR+3ag=="
             session_id="hVpd7YQgRW2nD22h7q60JQ==">
           <label>98</label>
     </stream>
     <stream stream_id="EiXGlc+4TruqqoDaNE76ag=="
             session_id="hVpd7YQgRW2nD22h7q60JQ==">
           <label>99</label>
     </stream>
     <sessionrecordingassoc session_id="hVpd7YQgRW2nD22h7q60JQ==">
                  <associate-time>2010-12-16T23:41:07Z</associate-time>
     </sessionrecordingassoc>
     <participantsessionassoc
         participant_id="srfBElmCRp2QB23b7Mpk0w=="
         session_id="hVpd7YQgRW2nD22h7q60JQ==">
          <associate-time>2010-12-16T23:41:07Z</associate-time>
     </participantsessionassoc>
     <participantsessionassoc
         participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
         session_id="hVpd7YQgRW2nD22h7q60JQ==">
             <associate-time>2010-12-16T23:41:07Z</associate-time>
     </participantsessionassoc>
```

```
      <participantstreamassoc
            participant_id="srfBElmCRp2QB23b7Mpk0w==">
                <send>i1Pz3to5hGk8fuXl+PbwCw==</send>
                <send>UAAMm5GRQKSCMVvLyl4rFw==</send>
                <recv>8zc6e0lYTlWIINA6GR+3ag==</recv>
                <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
      </participantstreamassoc>
      <participantstreamassoc
            participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
                <send>8zc6e0lYTlWIINA6GR+3ag==</send>
                <send>EiXGlc+4TruqqoDaNE76ag==</send>
                <recv>UAAMm5GRQKSCMVvLyl4rFw==</recv>
                <recv>i1Pz3to5hGk8fuXl+PbwCw==</recv>
      </participantstreamassoc>
    </recording>
```

                Example Metadata Snapshot from SRC to SRS

8.2. Partial Update of Recording Metadata XML Body

   The following example provides a partial update in the recording
   metadata XML body for the above example. The example has a snapshot
   that carries the disassociate-time for a participant from a session.

```
      <?xml version="1.0" encoding="UTF-8"?>
        <recording xmlns='urn:ietf:params:xml:ns:recording:1'>
          <datamode>partial</datamode>
            <participant
              participant_id="srfBElmCRp2QB23b7Mpk0w==">
              <nameID aor="sip:bob@biloxi.com">
                        <name xml:lang="it">Bob</name>
                  </nameID>
            </participant>
            <participantsessionassoc
               participant_id="srfBElmCRp2QB23b7Mpk0w=="
               session_id="hVpd7YQgRW2nD22h7q60JQ==">
              <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
            </participantsessionassoc>
        </recording>
```

              Partial Update of SIP Recording Example XML Body

9. XML Schema Definition for Recording Metadata

    This section defines the XML schema for the recording metadata
    document.

```
<?xml version="1.0" encoding="UTF-8"?>
   <xs:schema targetNamespace="urn:ietf:params:xml:ns:recording:1"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:tns="urn:ietf:params:xml:ns:recording:1"
      elementFormDefault="qualified"
      attributeFormDefault="unqualified">
    <!-- This import brings in the XML language attribute xml:lang -->
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
                schemaLocation="https://www.w3.org/2001/xml.xsd"/>
    <xs:element name="recording" type="tns:recording"/>
    <xs:complexType name="recording">
        <xs:sequence>
            <xs:element name="datamode" type="tns:dataMode"
                  minOccurs="0"/>
            <xs:element name="group" type="tns:group"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="session" type="tns:session"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="participant" type="tns:participant"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="stream" type="tns:stream"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="sessionrecordingassoc"
                  type="tns:sessionrecordingassoc"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="participantsessionassoc"
                  type="tns:participantsessionassoc"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="participantstreamassoc"
                  type="tns:participantstreamassoc"
                  minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace='##other'
                    minOccurs='0'
                    maxOccurs='unbounded'
                    processContents='lax'/>
        </xs:sequence>
    </xs:complexType>
```

```
      <xs:complexType name="group">
          <xs:sequence>
             <xs:element name="associate-time" type="xs:dateTime"
               minOccurs="0"/>
             <xs:element name="disassociate-time" type="xs:dateTime"
               minOccurs="0"/>
           <xs:any namespace='##other'
                      minOccurs='0'
                      maxOccurs='unbounded'
                      processContents='lax'/>
          </xs:sequence>
          <xs:attribute name="group_id" type="xs:base64Binary"
                use="required"/>
      </xs:complexType>
      <xs:complexType name="session">
          <xs:sequence>
             <xs:element name="sipSessionID" type="xs:string"
                   minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="reason" type="tns:reason"
                   minOccurs="0" maxOccurs="unbounded"/>
             <xs:element name="group-ref" type="xs:base64Binary"
                   minOccurs="0" maxOccurs="1"/>
             <xs:element name="start-time" type="xs:dateTime"
                   minOccurs="0" maxOccurs="1"/>
             <xs:element name="stop-time" type="xs:dateTime"
                   minOccurs="0" maxOccurs="1"/>
               <xs:any namespace='##other'
                      minOccurs='0'
                      maxOccurs='unbounded'
                      processContents='lax'/>
          </xs:sequence>
          <xs:attribute name="session_id" type="xs:base64Binary"
                use="required"/>
      </xs:complexType>
      <xs:complexType name="sessionrecordingassoc">
          <xs:sequence>
             <xs:element name="associate-time" type="xs:dateTime"
                   minOccurs="0"/>
             <xs:element name="disassociate-time" type="xs:dateTime"
                   minOccurs="0"/>
             <xs:any namespace='##other'
                      minOccurs='0'
                      maxOccurs='unbounded'
                      processContents='lax'/>
          </xs:sequence>
          <xs:attribute name="session_id" type="xs:base64Binary"
                use="required"/>
      </xs:complexType>
```

```
    <xs:complexType name="participant">
        <xs:sequence>
            <xs:element name="nameID" type="tns:nameID"
                    maxOccurs='unbounded'/>
        <xs:any namespace='##other'
                    minOccurs='0'
                    maxOccurs='unbounded'
                    processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="participant_id" type="xs:base64Binary"
            use="required"/>
    </xs:complexType>
    <xs:complexType name="participantsessionassoc">
        <xs:sequence>
        <xs:element name="associate-time" type="xs:dateTime"
                minOccurs="0"/>
        <xs:element name="disassociate-time" type="xs:dateTime"
                minOccurs="0"/>
        <xs:element name="param" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:attribute name="pname" type="xs:string"
                  use="required"/>
                <xs:attribute name="pval" type="xs:string"
                  use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:any namespace='##other'
                minOccurs='0'
                maxOccurs='unbounded'
                processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="participant_id" type="xs:base64Binary"
            use="required"/>
        <xs:attribute name="session_id" type="xs:base64Binary"
            use="required"/>
    </xs:complexType>
```

```
    <xs:complexType name="participantstreamassoc">
        <xs:sequence>
            <xs:element name="send" type="xs:base64Binary"
               minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="recv" type="xs:base64Binary"
               minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="associate-time" type="xs:dateTime"
                    minOccurs="0"/>
            <xs:element name="disassociate-time" type="xs:dateTime"
                   minOccurs="0"/>
                <xs:any namespace='##other'
                     minOccurs='0'
                     maxOccurs='unbounded'
                     processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="participant_id" type="xs:base64Binary"
             use="required"/>
    </xs:complexType>
    <xs:complexType name="stream">
        <xs:sequence>
           <xs:element name="label" type="xs:string"
              minOccurs="0" maxOccurs="1"/>
            <xs:any namespace='##other'
                     minOccurs='0'
                     maxOccurs='unbounded'
                     processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="stream_id" type="xs:base64Binary"
             use="required"/>
        <xs:attribute name="session_id" type="xs:base64Binary"/>
    </xs:complexType>
    <xs:simpleType name="dataMode">
        <xs:restriction base="xs:string">
            <xs:enumeration value="complete"/>
                <xs:enumeration value="partial"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="nameID">
      <xs:sequence>
        <xs:element name="name" type ="tns:name" minOccurs="0"
                          maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="aor" type="xs:anyURI" use="required"/>
    </xs:complexType>
```

```
   <xs:complexType name="name">
    <xs:simpleContent>
       <xs:extension base="xs:string">
          <xs:attribute ref="xml:lang" use="optional"/>
       </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="reason">
      <xs:simpleContent>
       <xs:extension base="xs:string">
          <xs:attribute type="xs:short" name="cause" use="required"/>
          <xs:attribute type="xs:string" name="protocol" default="SIP"/>
       </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
   <xs:element name="requestsnapshot" type="tns:requestsnapshot"/>
    <xs:complexType name="requestsnapshot">
       <xs:sequence>
          <xs:element name="requestreason" type="tns:name"
            minOccurs="0"/>
          <xs:any namespace='##other'
            minOccurs='0'
            maxOccurs='unbounded'
            processContents='lax'/>
       </xs:sequence>
    </xs:complexType>
</xs:schema>
```

10. Security Considerations

   This document describes an extensive set of metadata that may be
   recorded by the SRS. Most of the metadata could be considered
   private data. The procedures mentioned in the Security
   Considerations section of [RFC7866] MUST be followed by the SRC and
   the SRS for mutual authentication and to protect the content of the
   metadata in the RS.

   An SRC MAY, by policy, choose to limit the parts of the metadata sent
   to the SRS for recording. Also, the policy of the SRS might not
   require recording all the metadata it receives. For the sake of data
   minimization, the SRS MUST NOT record additional metadata that is not
   explicitly required by local policy. Metadata in storage needs to be
   provided with a level of security that is comparable to that of the
   recording session.

11. IANA Considerations

   This specification registers a new XML namespace and a new XML
   schema.

11.1. SIP Recording Metadata Schema Registration

   URI: urn:ietf:params:xml:ns:recording:1

   Registrant Contact: IETF SIPREC working group, Ram Mohan R
      (rmohanr@cisco.com)

   XML: The registered XML schema is contained in Section 9.

   Its first line is <?xml version="1.0" encoding="UTF-8"?>, and its
   last line is </xs:schema>.

12. References

12.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
              A., Peterson, J., Sparks, R., Handley, M., and E.
              Schooler, "SIP: Session Initiation Protocol", RFC 3261,
              DOI 10.17487/RFC3261, June 2002,
              <http://www.rfc-editor.org/info/rfc3261>.

   [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
              Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
              <http://www.rfc-editor.org/info/rfc3339>.

   [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat,
              "Indicating User Agent Capabilities in the Session
              Initiation Protocol (SIP)", RFC 3840,
              DOI 10.17487/RFC3840, August 2004,
              <http://www.rfc-editor.org/info/rfc3840>.

   [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally
              Unique IDentifier (UUID) URN Namespace", RFC 4122,
              DOI 10.17487/RFC4122, July 2005,
              <http://www.rfc-editor.org/info/rfc4122>.

   [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
              Description Protocol", RFC 4566, DOI 10.17487/RFC4566,
              July 2006, <http://www.rfc-editor.org/info/rfc4566>.

   [RFC4574] Levin, O. and G. Camarillo, "The Session Description
              Protocol (SDP) Label Attribute", RFC 4574,
              DOI 10.17487/RFC4574, August 2006,
              <http://www.rfc-editor.org/info/rfc4574>.

   [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data
              Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006,
              <http://www.rfc-editor.org/info/rfc4648>.

   [RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description
              Protocol (SDP) Content Attribute", RFC 4796,
              DOI 10.17487/RFC4796, February 2007,
              <http://www.rfc-editor.org/info/rfc4796>.

   [RFC7866] Portman, L., Lum, H., Ed., Eckel, C., Johnston, A., and A.
              Hutton, "Session Recording Protocol", RFC 7866,
              DOI 10.17487/RFC7866, May 2016,
              <http://www.rfc-editor.org/info/rfc7866>.

12.2. Informative References

   [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private
              Extensions to the Session Initiation Protocol (SIP) for
              Asserted Identity within Trusted Networks", RFC 3325,
              DOI 10.17487/RFC3325, November 2002,
              <http://www.rfc-editor.org/info/rfc3325>.

   [RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason
              Header Field for the Session Initiation Protocol (SIP)",
              RFC 3326, DOI 10.17487/RFC3326, December 2002,
              <http://www.rfc-editor.org/info/rfc3326>.

   [RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers",
              RFC 3966, DOI 10.17487/RFC3966, December 2004,
              <http://www.rfc-editor.org/info/rfc3966>.

   [RFC4235] Rosenberg, J., Schulzrinne, H., and R. Mahy, Ed., "An
              INVITE-Initiated Dialog Event Package for the Session
              Initiation Protocol (SIP)", RFC 4235,
              DOI 10.17487/RFC4235, November 2005,
              <http://www.rfc-editor.org/info/rfc4235>.

   [RFC6341] Rehor, K., Ed., Portman, L., Ed., Hutton, A., and R. Jain,
              "Use Cases and Requirements for SIP-Based Media Recording
              (SIPREC)", RFC 6341, DOI 10.17487/RFC6341, August 2011,
              <http://www.rfc-editor.org/info/rfc6341>.

   [RFC7245] Hutton, A., Ed., Portman, L., Ed., Jain, R., and K. Rehor,
              "An Architecture for Media Recording Using the Session
              Initiation Protocol", RFC 7245, DOI 10.17487/RFC7245,
              May 2014, <http://www.rfc-editor.org/info/rfc7245>.

   [SessionID]
              Jones, P., Salgueiro, G., Pearce, C., and P. Giralt,
              "End-to-End Session Identification in IP-Based Multimedia
              Communication Networks", Work in Progress,
              draft-ietf-insipid-session-id-22, April 2016.

   [UML]   Object Management Group, "OMG Unified Modeling Language
              (UML)", 2011, <http://www.omg.org/spec/UML/2.4/>.

Authors' Addresses

   Ram Mohan Ravindranath
   Cisco Systems
   Cessna Business Park
   Bangalore, Karnataka
   India

   Email: rmohanr@cisco.com

   Parthasarathi Ravindran
   Nokia Networks
   Bangalore, Karnataka
   India

   Email: partha@parthasarathi.co.in

   Paul Kyzivat
   Huawei
   Hudson, MA
   United States

   Email: pkyzivat@alum.mit.edu