

SIPREC
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2013

Ram Mohan. Ravindranath
Cisco Systems, Inc.
Parthasarathi. Ravindran
Nokia Siemens Networks
Paul. Kyzivat
Huawei
October 17, 2012

Session Initiation Protocol (SIP) Recording Metadata
draft-ietf-siprec-metadata-08

Abstract

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

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 20, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
2. Terminology	4
3. Definitions	4
4. Metadata Model	5
5. Recording Metadata Format	6
5.1. XML data format	6
5.1.1. Namespace	7
5.1.2. recording	7
6. Recording Metadata classes	7
6.1. Recording Session	7
6.1.1. Attributes	8
6.1.2. Linkages	8
6.1.3. XML element	8
6.2. Communication Session Group	9
6.2.1. Attributes	9
6.2.2. Linkages	9
6.2.3. XML element	10
6.3. Communication Session	10
6.3.1. Attributes	11
6.3.2. Linkages	11
6.3.3. XML element	12
6.4. CSRSAssociation	12
6.4.1. Attributes	13
6.4.2. Linkages	13
6.4.3. XML element	13
6.5. Participant	13
6.5.1. Attributes	14
6.5.2. Linkages	14
6.5.3. XML element	14
6.6. ParticipantCSAssociation	15
6.6.1. Attributes	15
6.6.2. Linkages	16
6.6.3. XML element	16
6.7. Media Stream	16
6.7.1. Attributes	17
6.7.2. Linkages	17
6.7.3. XML element	17
6.8. ParticipantStream Association	17

6.8.1. Attributes	18
6.8.2. Linkages	18
6.8.3. XML element	18
6.9. associate-time/disassociate-time	18
6.10. Unique ID format	19
7. SIP Recording Metadata Example	19
7.1. Complete SIP Recording Metadata Example	19
7.2. Partial Update of Recording metadata XML body	21
8. XML Schema definition for Recording metadata	21
9. Security Considerations	25
9.1. Connection Security	25
10. IANA Considerations	25
10.1. SIP recording metadata Schema Registration	25
11. Acknowledgement	26
12. Appendix A: Metadata Model Object Instances	26
12.1. Use case 1: Basic Call	26
12.2. Use case 2: Hold/Resume	27
12.3. Use case 3: Basic call with Transfer	29
12.4. Conference Use Cases	30
12.4.1. Case 1:	31
12.4.2. Case 2:	33
12.4.3. Case 3:	35
12.4.4. Case 4:	36
13. Appendix B: Metadata XML schema Instances	37
13.1. Use case 1: Basic Call	37
13.2. Use case 2: Hold/resume	39
13.3. Use case 3: Basic Call with transfer	41
13.4. Use Case 4: Call disconnect	45
14. References	46
14.1. Normative References	46
14.2. Informative References	47
Authors' Addresses	47

1. Introduction

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

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: An abstract representation of metadata using a Unified Modelling Language(UML) class diagram.

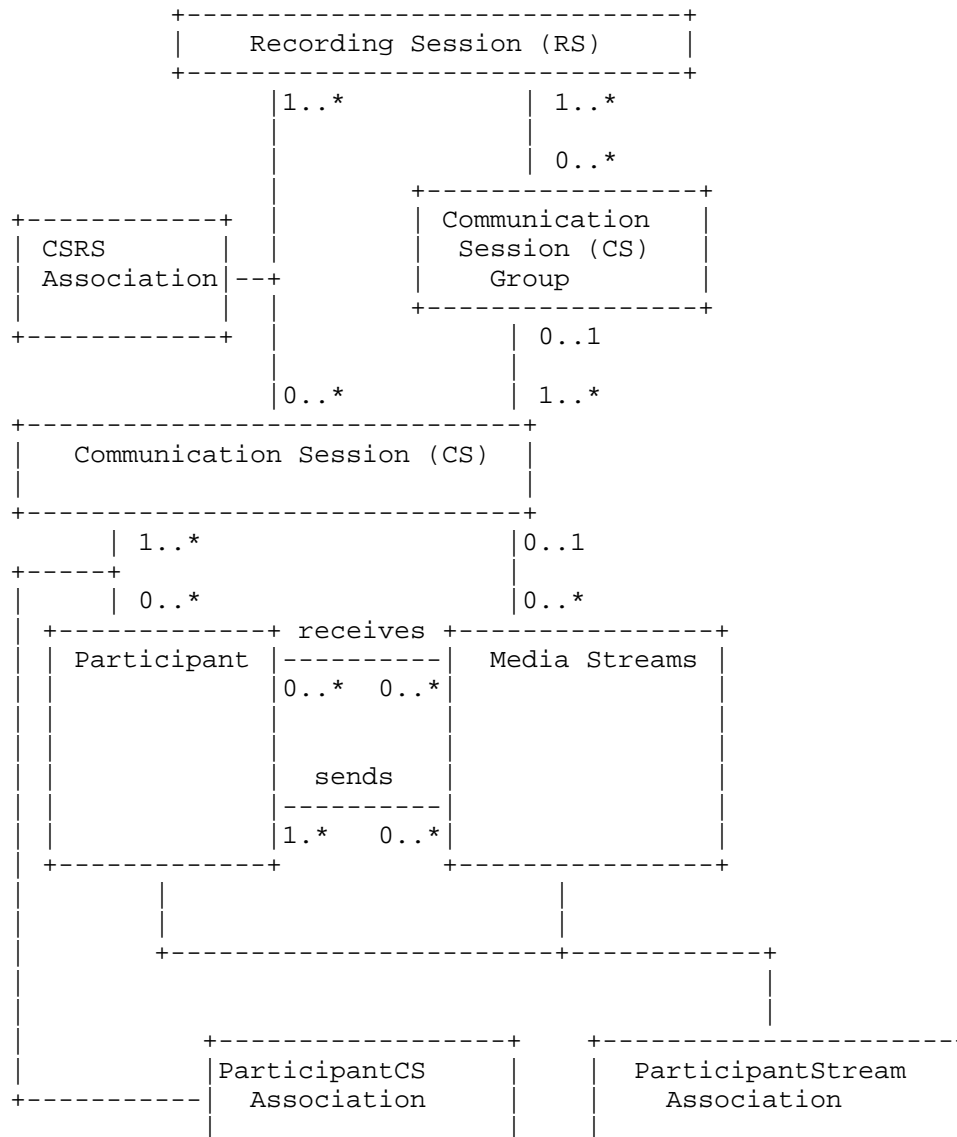
Metadata classes: Each block in the model represents a class. A class is a construct that is used as a blueprint to create instances(called objects) of itself. The description of each class also has representation of its attributes in a second compartment below the class name.

Attributes: Attributes represents the attributes listed in each of the classes. The attributes of a class are listed in the second compartment below the class name. Each instance of class conveys values for these attributes which adds to the recording's Metadata.

Linkages: Linkages represents the relationship between the classes in the model. It represents the logical connections between classes(or objects) in class diagrams/ object diagrams. The linkages used in the Metadata model of this document are associations.

4. Metadata Model

Metadata is the information that describes recorded media and the CS to which they relate. Below diagram shows a model for Metadata as viewed by Session Recording Server (SRS).



+-----+ +-----+

The Metadata model is a class diagram in Unified Modelling Language(UML). The model describes the structure of a 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 represents the relationships which can be associations or Composition. The metadata is conveyed from SRC to SRS.

The model allows the capture of a snapshot of a recording's Metadata at a given instant in time. Metadata changes to reflect changes in what is being recorded. For example, if in a conference a participant joins SRC sends a snapshot of metadata having that participant information (with attributes like name/AoR pair and associate-time) to the SRS.

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 SDP signalling).

5. Recording Metadata Format

This section gives an overview of Recording Metadata Format. Some data from the metadata model is assumed to be made available to the SRS through Session Description Protocol (SDP)[RFC4566], and therefore this data is not represented in the XML document format specified in this document. SDP attributes describes about different media formats like audio, video. The other metadata attributes like participant details are represented in a new Recording specific XML document namely 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

Recording Metadata document is an XML document. recording element MUST be present in all recording metadata XML document. recording acts as container for all other elements in this XML document.

Recording object is a XML document. It MUST have the XML declaration and it SHOULD contain 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 the minimal interoperability.

Syntax and semantics error in recording XML document has to be informed to the originator using application specific mechanism.

5.1.1. Namespace

The namespace URI for elements defined by this specification is a Uniform Resource Namespace (URN) [RFC2141], using the namespace identifier 'ietf' defined by [RFC2648] and extended by [RFC3688].

The URN is as follows: urn:ietf:params:xml:ns:recording

5.1.2. recording

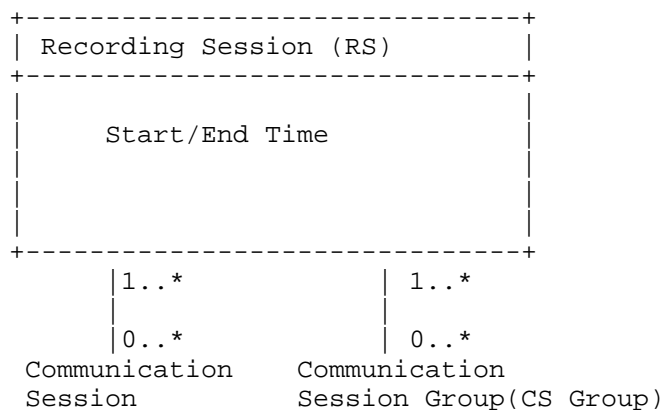
recording element MUST contain an xmlns namespace attribute with value as urn:ietf:params:xml:ns:recording. One recording element MUST be present in the all recording metadata XML document.

dataMode element shows whether the XML document is complete document or partial update. 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



Each instance of a Recording Session class (namely the Recording Session Object) represents a SIP session created between an SRC and SRS for the purpose of recording a Communication Session.

6.1.1. Attributes

A Recording Session class has the following attributes:

- o Start/End Time - Represents the Start/End time of a Recording Session object.

6.1.2. Linkages

Each instance of Recording Session has:

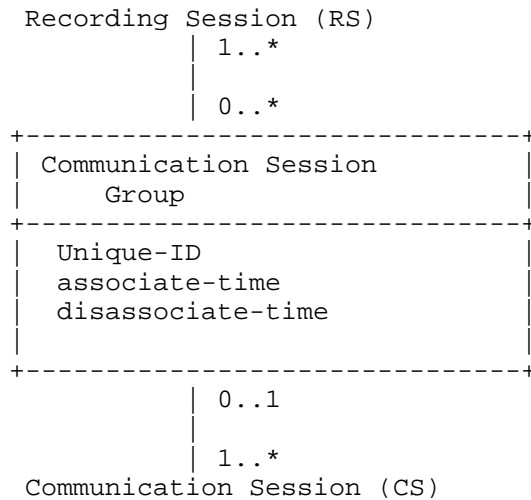
- o Zero or more instances of Communication Session Group. CSG may be zero because it is optional metadata object. Also the allowance of zero instances is to accommodate persistent recording, where there may be none.
- o Zero or more instances of Communication Session objects.

6.1.3. XML element

Recording Session object is represented by recording XML element. That in turn relies on the SIP/SDP session with which the XML document is associated to provide some of the attributes of the Recording Session element.

Start and End time value are derivable from Date header(if present in SIP message) in RS. In cases where Date header is not present, Start/End time are derivable from the time at which SRS receives the notification of SIP message to setup RS / disconnect RS.

6.2. Communication Session Group



One instance of a Communication Session Group class (namely the Communication Session Group object) provides association or linking of Communication Sessions.

6.2.1. Attributes

A CS Group has the following attributes:

- o Unique-ID - This Unique-ID is to group different CSs that are related. SRC (or SRS) is responsible for ensuring the uniqueness of Unique-ID in case multiple SRC interacts with the same SRS. The mechanism by which SRC groups the CS is outside the scope of SIPREC.
- o Associate-time - Associate-time for CS-Group shall be calculated by SRC as the time when a grouping is formed. The rules that determine how a grouping of different Communication Session objects is done by SRC is outside the scope of SIPREC.
- o Disassociate-time - Disassociate-time for CS-Group shall be calculated by SRC as the time when the grouping ends

6.2.2. Linkages

The linkages between Communication Session Group class and other classes is association. A communication Session Group is associated with RS and CS in the following manner:

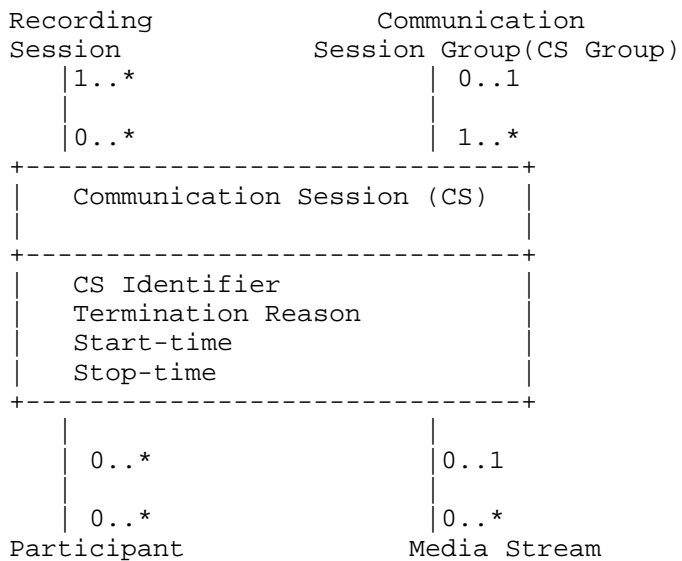
- o There is one or more Recording Session objects per Communication Session Group.
- o Each Communication Session Group object has to be associated with one or more RS [Here each RS can be setup by the potentially different SRCs]
- o There is one or more Communication Sessions per CS Group [e.g. Consult Transfer]

6.2.3. XML element

Group element is an optional element provides the information about the communication session group

Each communication session group (CSG) object is represented using one group element. Each group element has unique Base 64 URN UUID attribute which helps to uniquely identify CSG.

6.3. Communication Session



A Communication Session class and its object in the metadata model represents Communication Session and its properties needed as seen by SRC.

6.3.1. Attributes

A communication Session class has the following attributes:

- o Termination Reason - This represents the reason why a CS was terminated. The communication session MAY contain a Call Termination Reason. This MAY be derived from SIP Reason header [RFC3326] of CS.
- o CS Identifier - This attribute is used to uniquely identify a CS.
- o Start-time - This optional attribute represents start time of CS as seen by SRC
- o Stop-time - This optional attribute represents stop time of CS as seen by 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 policy. The ability for the SRC to influence this policy is outside the scope of this document. However if there are implementations where SRC has enough information, this could be sent as Extension Data attached to CS

6.3.2. Linkages

A Communication Session is linked to CS-Group, Participant, Media Stream and Recording Session classes using the association relationship. Association between CS and Participant allows:

- o CS to have atleast zero or more participants
- o 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 premixed media stream. The SRC may have knowledge of such Participants, yet not have any signaling relationship with them. This might arise if one participant in CS is a conf focus. To summarize even if SRC does not have direct signalling 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 CS that are not participants in the media. An example is the identity of a 3pcc controller that has initiated a CS to two or more participants of 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.

Association between CS and Media Stream allows:

- o A CS to have zero or more Streams
- o A stream can be associated with at most one CS. Stream in persistent RS is not required to be associated with any CS before CS is created and hence the zero association is allowed.

Association between CS and RS allows:

- o Each instance of RS has Zero or more instances of Communication Session objects.
- o Each CS has to be associated with one more RS [Here each RS can be potentially setup by different SRCs]

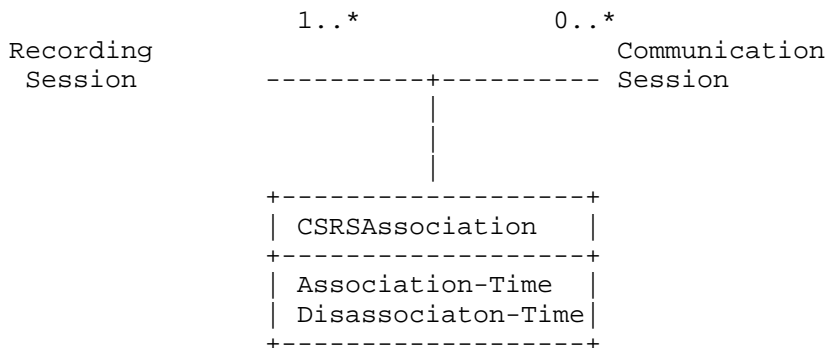
6.3.3. XML element

Session element provides the information about the communication session

Each communication session(CS) object is represented by one session element. Each session element has unique Base 64 URN UUID attribute which helps to uniquely identify CS.

Reason element MAY be included to represent the Termination Reason attribute. group-ref element MAY exist to indicate the group where the mentioned session belongs.

6.4. CSRSAssociation



A CSRS Association class and its objects has attributes of CS object which are attributes of association of a session to a RS.

6.4.1. Attributes

CSRS association class has the following attributes:

- o Associate-time - associate-time is calculated by SRC as the time it sees a CS is associated to a RS
- o Disassociate-time- Disassociate-time is calculated by SRC as the time it see a CS disassociate from a RS.

It is possible that a given CS can have multiple associate/disassociate times within given RS.

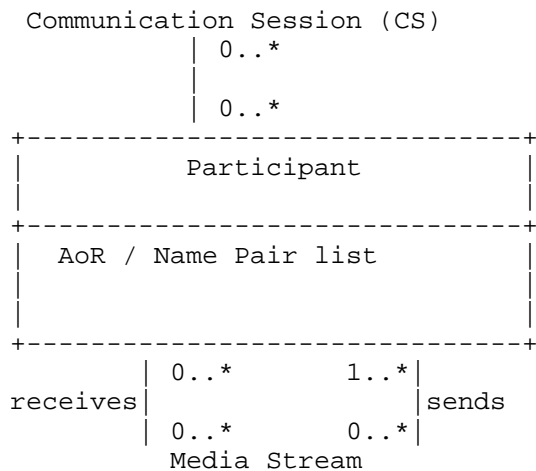
6.4.2. Linkages

CSRS association class is linked to CS and RS classes. There are no cardinalties for this linkage.

6.4.3. XML element

sessionrecordingassoc is the XML element to represent CSRS association object. session URN UUID is used to uniquely identify this element and link with the specific session.

6.5. Participant



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

6.5.1. Attributes

Participant has attributes like:

- o AoR / Name pair list - This attribute is a list of Name/AoR tuple. An AoR MAY be SIP/SIPS/TEL URI. Name represents Participant name(SIP display name) or DN number (in case it is known). There are cases where a participant can have more than one AoR [e.g. P-Asserted-identity header [RFC3325] which can have both SIP and TEL URIs]

This document does not specify other attributes relating to participant e.g. Participant Role, Participant type. An SRC which has information of these attributes can indicate the same as part of extension data to Participant from SRC to SRS.

6.5.2. Linkages

The participant class is linked to MS and CS class using association relationship. The association between participant and Media Stream allows:

- o Participant to receives zero or more media streams
- o Participant to send zero or more media streams. (Same participant provides multiple streams e.g. audio and video)
- o Media stream to be received by zero or more participants. Its possible, though perhaps unlikely, that a stream is generated but sent only to the SRC and SRS, not to any participant. E.g. 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 Media stream to be sent by one or more participants (pre-mixed streams).

Example of a case where a participant receives Zero or more streams - a Supervisor may have side conversation with Agent, while Agent converses with customer.

6.5.3. XML element

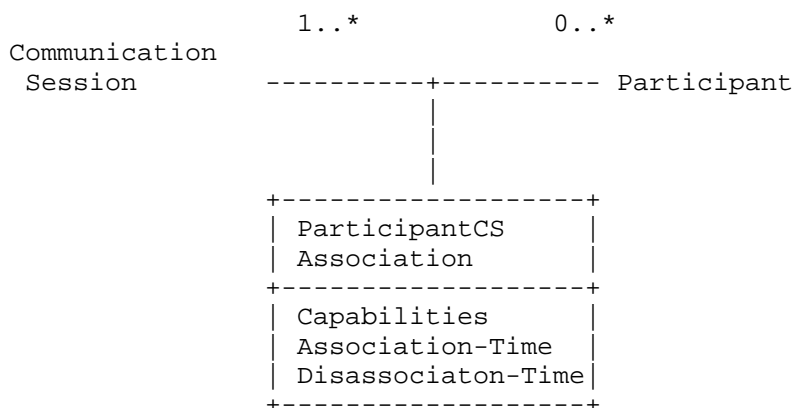
A participant element represents a Participant object.

Participant MUST have a NameID complex element which contains AoR as attribute and Name as element. AOR element is SIP/SIPS URI FQDN or IP address which represents the user. name is an optional element to represent display name.

Each participant element has unique ID (Base 64 URN UUID) attribute

which helps to uniquely identify participant and session Base 64 URN UUID to associate participant with specific session element. Base 64 URN UUID of participant MUST used in the scope of CSG and no new Base 64 URN UUID has to be created for the same element (participant, stream) between different CS in the same CSG. In case Base 64 URN UUID has to be used permanent, careful usage of Base 64 URN UUID to original AoR has to be decided by the implementers and it is implementer's choice.

6.6. ParticipantCSAssociation



A participantCS Association class and its objects has attributes of participant object which are attributes of association of a participant to a Session.

6.6.1. Attributes

ParticipantCS association class has the following attributes:

- o Associate-time - associate-time is calculated by SRC as the time it sees a participant is associated to CS
- o Disassociate-time- Disassociate-time is calculated by SRC as the time it see a participant disassociate from a CS. It is possible that a given participant can have multiple associate/disassociate times within given communication session.
- o Capabilities - A participant capabilities as defined in [RFC3840] which is an optional attribute that includes the capabilities of a participant in a CS. Each participant shall have Zero or more capabilities. A participant may use different capabilities depending on the role it plays at a particular instance. IOW if a

participants moves across different CSs (due to transfer e.t.c)
OR is simultaneously present in different CSs its role may be
different and hence the capability used.

- o "send" or "recv" element in each participant is associating SDP m-lines with the participant. send element indicates that participant is sending the stream of media with the mentioned media description. recv element indicates that participant is receiving the stream and by default all participant will receive the stream. recv element has relevance in case whisper call scenario wherein few of the participant in the session receives the stream and not others.

6.6.2. Linkages

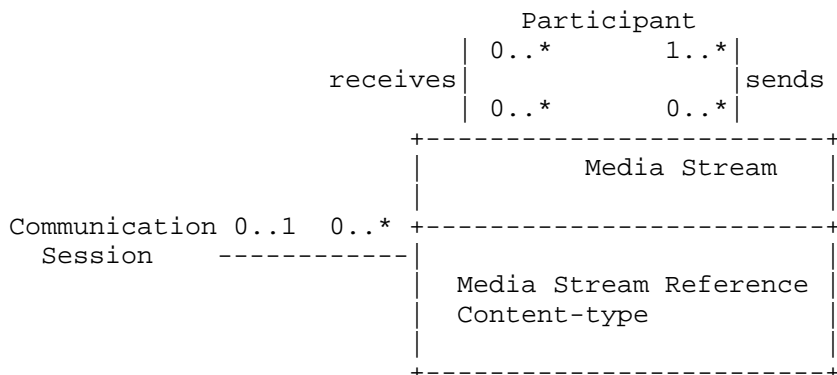
The participantCS association class is linked to participant and CS classes. There are no cardinalities for this linkage.

6.6.3. XML element

participantsessionassoc XML element represent participantCS association object. participant and session id is used to uniquely identify this element

NOTE: RFC 4235 encoding shall be used to represent capabilities attribute in XML.

6.7. Media Stream



A Media Stream class (and its objects) has the properties of media as seen by SRC and sent to SRS. Different snapshots of media stream object may be sent whenever there is a change in media (e.g. dir

change like pause/resume and/or codec change and/or participant change.)).

6.7.1. Attributes

A Media Stream class has the the following attributes:

- o Media Stream Reference - In implementations this can reference to m-line
- o Content - The content of an MS element will be described in terms of value from the [RFC4796] registry.

The metadata model should include media streams that are not being delivered to the SRS. Examples include cases where SRC offered certain media types but SRS chooses to accept only a subset of them OR an SRC may not even offer a certain media type due it its restrictions to record

6.7.2. Linkages

A Media Stream is linked to participant and CS classes using the association relationship. The details of association with the Participant are described in the Participant class section. The details of association with CS is mentioned in the CS section.

6.7.3. XML element

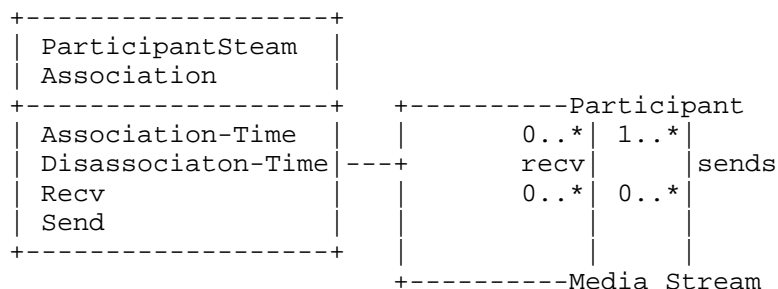
stream element represents a Media Stream object. Stream element indicates SDP media lines associated with the session and participants.

This element indicates the SDP m-line properties like label attributes. Label attribute is used to link m-line SDP body using label attribute in SDP m-line.

Each stream element has unique Base 64 URN UUID attribute which helps to uniquely identify stream and session Base 64 URN UUID to associate stream with specific session element.

The content attribute if an SRC wishes to send is conveyed in RS SDP.

6.8. ParticipantStream Association



A ParticipantStream association class and its object has attributes that are attributes of association of a Participant to a Stream.

6.8.1. Attributes

A participantStream association class has the following attributes:

- o Associate-Time: This attributes indicates the time a Participant started contributing to a Media Stream
- o Disassociate-Time: This attribute indicates the time a Participant stopped contributing to a Media Stream

6.8.2. Linkages

The participantStream association class is linked to participant and Stream classes. There are no cardinalities for this linkage.

6.8.3. XML element

ParticipantStreamAssoc XML element represents participant to stream association object. participant element is used to uniquely identify this element and related with stream using stream unique URN id..

6.9. associate-time/disassociate-time

associate-time/disassociate-time contains a string indicating the date and time of the status change of this tuple. The value of this element MUST follow the IMPP datetime format [RFC3339]. Timestamps that contain 'T' or 'Z' MUST use the capitalized forms. At a time, any of the time tuple associate-time or disassociate-time MAY exist in the element namely group, session, participant and not both timestamp at the same time.

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

determined.

6.10. Unique ID format

Unique id is generated in two steps:

- o UUID is created using [RFC4122])
- o UUID is encoded using base64 as defined in [RFC4648]

The above mentioned unique-id mechanism SHOULD be used for each metadata element.

7. SIP Recording Metadata Example

7.1. Complete SIP Recording Metadata Example

The following example provides all the tuples involved in Recording Metadata XML body.

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <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>
        <uniqueid>FaXHlc+3WruaroDaNE87am==</uniqueid>
      </call-center>
      <mydata xmlns='http://example.com/my'>
        <structure>FOO!</structure>
        <whatever>bar</whatever>
      </mydata>
    </group>
    <session session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <group-ref>7+OTCyoxTmqmqyA/1weDag==
      </group-ref>

      <!-- Standardized extension -->
      <structure>FOO!</structure>
      <whatever>bar</whatever>
    </session>
    <sessionrecordingassoc session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <associate-time>2010-12-16T23:41:07Z</associate-time>
    </sessionrecordingassoc>
    <participant
      participant_id="srfBEImCRp2QB23b7Mpk0w==">
```

```
<nameID aor=sip:bob@biloxi.com>
  <name xml:lang="it">Bob B</name>
</nameID>
<!-- Standardized extension -->
<structure>FOO!</structure>
<whatever>bar</whatever>
</participant>
<participantsessionassoc
  participant_id="srfBEImCRp2QB23b7Mpk0w=="
  session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <associate-time>2010-12-16T23:41:07Z</associate-time>
</participantsessionassoc>
<participantstreamassoc
  participant_id="srfBEImCRp2QB23b7Mpk0w==">
  <send>ilPz3to5hGk8fuXl+PbwCw==</send>
  <send>UAAMm5GRQKSCMVvLyl4rFw==</send>
  <recv>8zc6e0lYTlWIINA6GR+3ag==</recv>
  <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
</participantstreamassoc>
<participant
  participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
  <nameID aor=sip:Paul@biloxi.com>
    <name xml:lang="it">Paul</name>
  </nameID>
  <!-- Standardized extension -->
  <structure>FOO!</structure>
  <whatever>bar</whatever>
</participant>
<participantsessionassoc
  participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
  session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <associate-time>2010-12-16T23:41:07Z</associate-time>
</participantsessionassoc>
<participantstreamassoc
  participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
  <send>8zc6e0lYTlWIINA6GR+3ag==</send>
  <send>EiXGlc+4TruqqoDaNE76ag==</send>
  <recv>UAAMm5GRQKSCMVvLyl4rFw==</recv>
  <recv>ilPz3to5hGk8fuXl+PbwCw==</recv>
</participantstreamassoc>
<stream stream_id="UAAMm5GRQKSCMVvLyl4rFw=="
  session session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <label>96</label>
</stream>
<stream stream_id="ilPz3to5hGk8fuXl+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>
  </recording>

```

SIP Recording Metadata Example XML body

7.2. Partial Update of Recording metadata XML body

The following example provides partial update in 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'>
    <dataMode>partial</dataMode>
    <participant
      participant_id="srfBElmCRp2QB23b7Mpk0w==">
      <name ID=sip:bob@biloxi.com>
        <name xml:lang="it">Bob R</name>
      </nameID>
      <structure>FOO!</structure>
      <whatever>bar</whatever>
    </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

8. XML Schema definition for Recording metadata

This section defines XML schema for Recording metadata document

```

<?xml version="1.0" encoding="UTF-8"?>

```

```
<xs:schema targetNamespace="urn:ietf:params:xml:ns:recording"
  xmlns:tns="urn:ietf:params:xml:ns:recording"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <!-- This import brings in the XML language attribute xml:lang-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:element name="recording" type="recording"/>
  <xs:complexType name="recording">
    <xs:sequence>
      <xs:element name="datamode" type="dataMode"
        minOccurs="0"/>
      <xs:element name="group" type="group"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="session" type="session"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="participant" type="participant"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="stream" type="stream"
        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="reason" type="xs:string"
        minOccurs="0"/>
      <xs:element name="group-ref" type="xs:base64Binary"
        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="dateTime"
            minOccurs="0" />
        <xs:element name="disassociate-time" type="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="nameID"
            maxOccurs="1" />
        <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: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="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:schema>
```

9. Security Considerations

The metadata information sent from SRC to SRS MAY reveal sensitive information about different participants in a session. For this reason, it is RECOMMENDED that a SRC use a strong means for authentication and metadata information protection and that it apply comprehensive authorization rules when using the metadata format defined in this document. The following sections will discuss each of these aspects in more detail.

9.1. Connection Security

It is RECOMMENDED that a SRC authenticate SRS using the normal SIP authentication mechanisms, such as Digest as defined in Section 22 of [RFC3261]. The mechanism used for conveying the metadata information MUST ensure integrity and SHOULD ensure confidentiality of the information. In order to achieve these, an end-to-end SIP encryption mechanism, such as S/MIME described in [RFC3261], SHOULD be used.

If a strong end-to-end security means (such as above) is not available, it is RECOMMENDED that a SRC use mutual hop-by-hop Transport Layer Security (TLS) authentication and encryption mechanisms described in "SIPS URI Scheme" and "Interdomain Requests" of [RFC3261].

10. IANA Considerations

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

10.1. SIP recording metadata Schema Registration

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

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

XML: the XML schema to be registered is contained in Section 6.

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

11. Acknowledgement

We wish to thank John Elwell(Siemens-Enterprise), Henry Lum(Alcatel-Lucent), Leon Portman(Nice), De Villers, Andrew Hutton(Siemens-Enterprise), Deepanshu Gautam(Huawei), Charles Eckel(Cisco), Muthu Arul(Cisco), Michael Benenson(Cisco), Hadriel Kaplan (ACME), Brian Rosen(Neustar), Scott Orton(Broadsoft), Ofir Roth for their valuable comments and inputs.

We wish to thank Joe Hildebrand(Cisco), Peter Saint-Andre(Cisco) for the valuable XML related guidance and Martin Thompson for validating the XML schema and providing comments on the same.

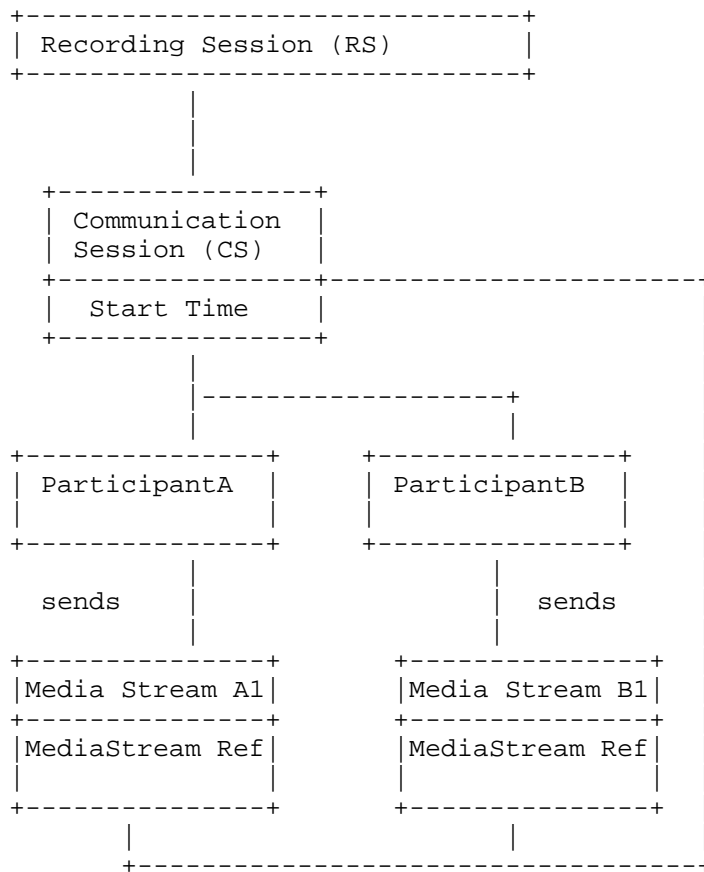
12. Appendix A: Metadata Model Object Instances

Note: This Appendix has to be moved to callflow document after the discussion in the mailing alias

This section describes the metadata model object instances for different use cases of SIPREC. For the sake of simplicity as the media streams sent by each of the participants is received by every other participant in these use cases, it is NOT shown in the object instance diagrams below. Also for the sake of ease not all attributes of each object are shown in these instance diagrams.

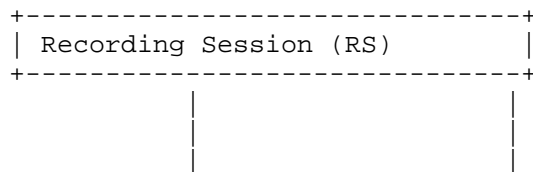
12.1. Use case 1: Basic Call

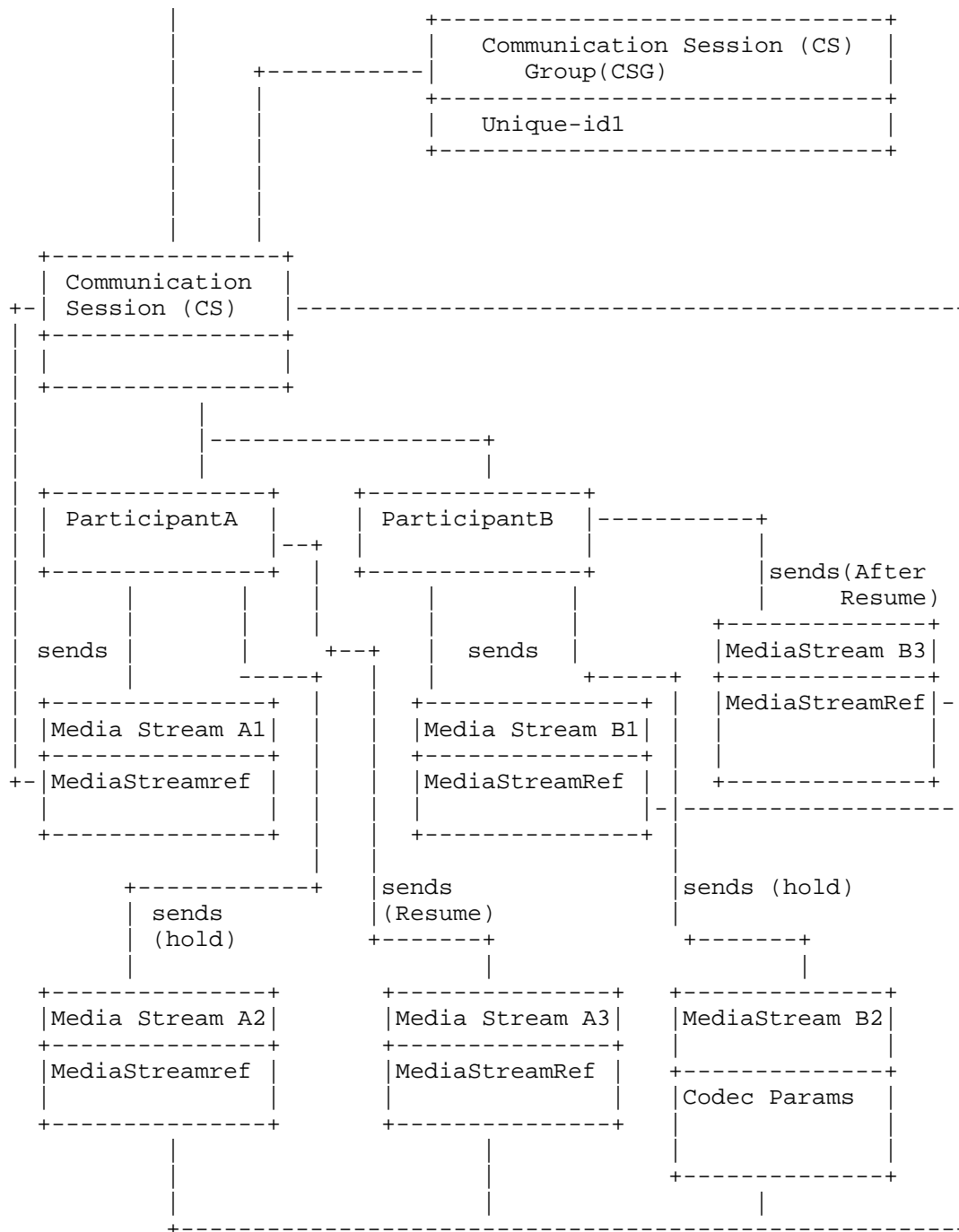
Basic call between two Participants A and B. In this use case each participant sends one Media Stream. For the sake of simplicity "receives" lines are not shown in this instance diagram. Media Streams sent by each participant is received all other participants of that CS.



12.2. Use case 2: Hold/Resume

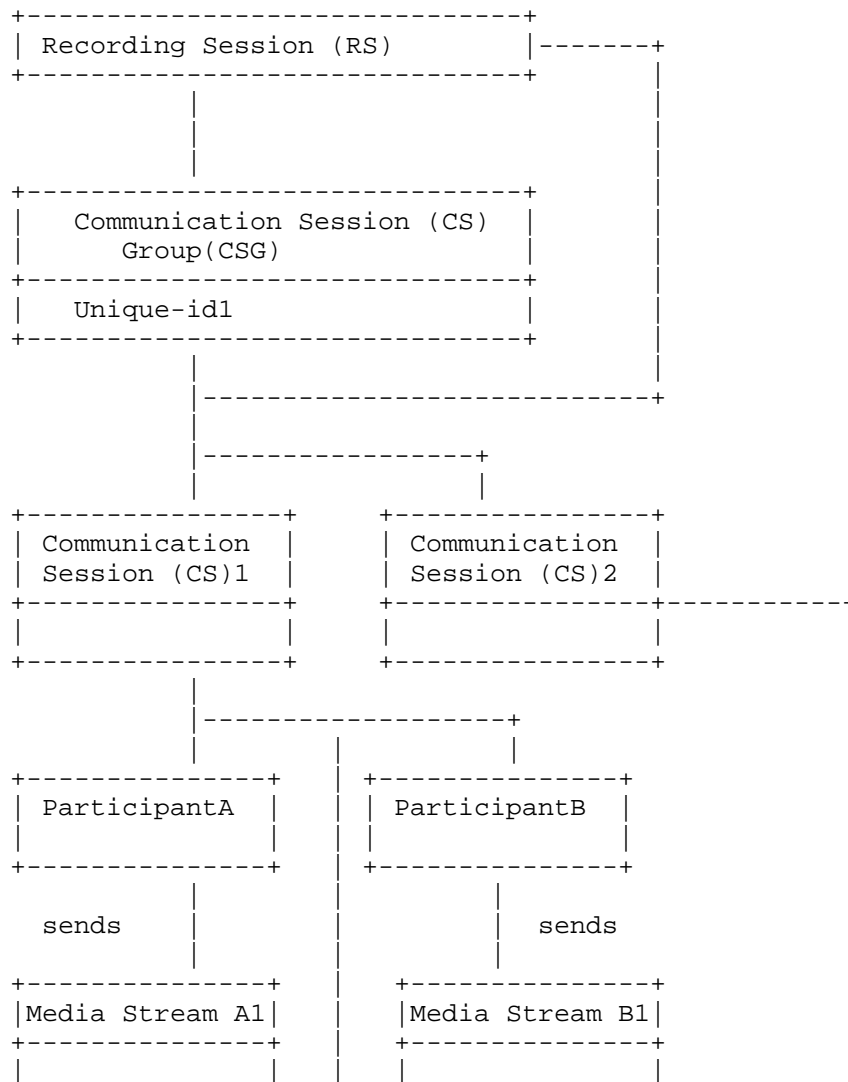
Basic call between two Participants A and B and with Participant A or B doing a Hold/Resume. In this use case each participant sends one Media Stream. After Hold/Resume the properties of Media can change. For the sake of simplicity "receives" lines are not shown in this instance diagram. Media Streams sent by each participant is received all other participants of that CS.

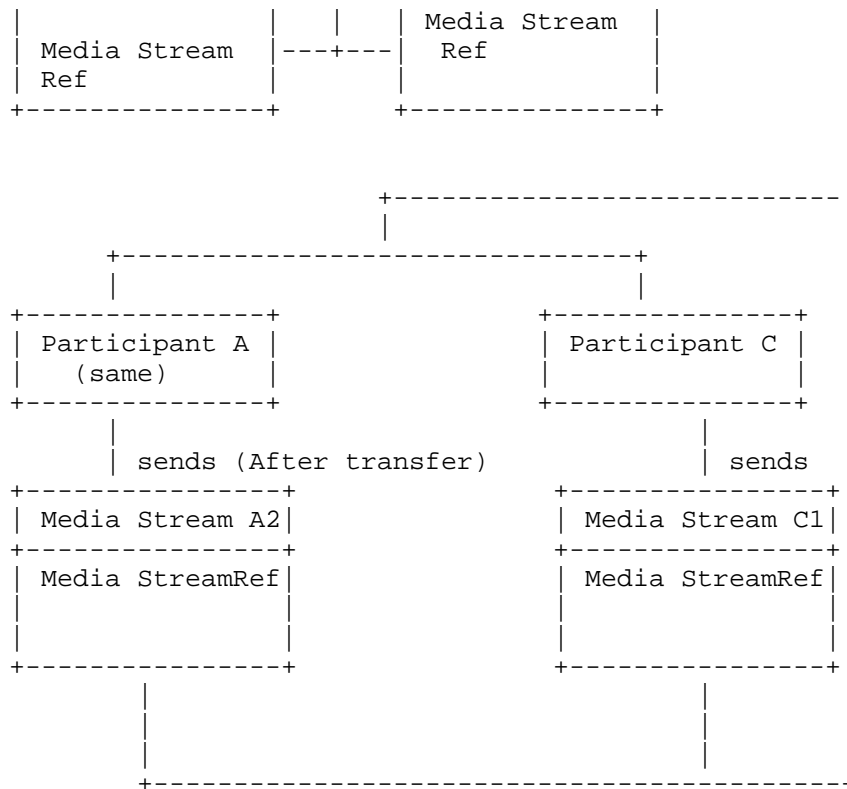




12.3. Use case 3: Basic call with Transfer

Basic call between two Participants A and B and with Participant A transfer(consult transfer) to Participant C. In this use case each participant sends one Media Stream. After transfer the properties of Participant A Media can change. For the sake of simplicity "receives" lines are not shown in this instance diagram. Media Streams sent by each participant is received all other participants of that CS.





12.4. Conference Use Cases

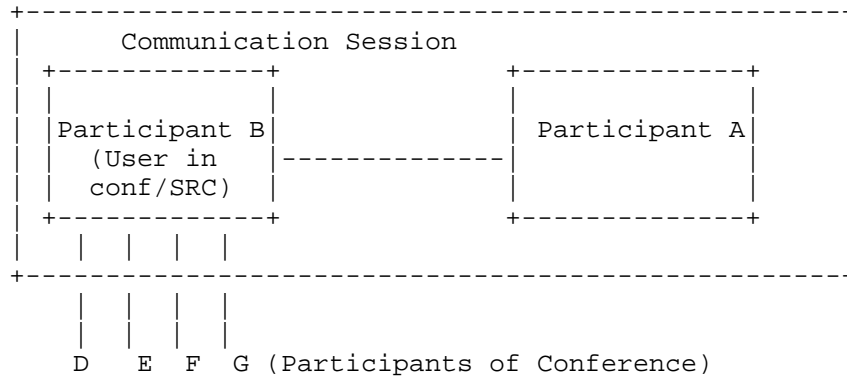
Depending on who act as SRC and the information that an SRC has there can be several ways to model conference use cases. This section has instance diagrams for the following cases:

- o A CS where one of the participant (which is also SRC) is a user in a conference
- o A CS where one of the participant is focus (which is also SRC)
- o A CS where one of the participant is user and the SRC is a different entity like B2BUA
- o A CS where one of the participant is focus and the SRC is a different entity like B2BUA

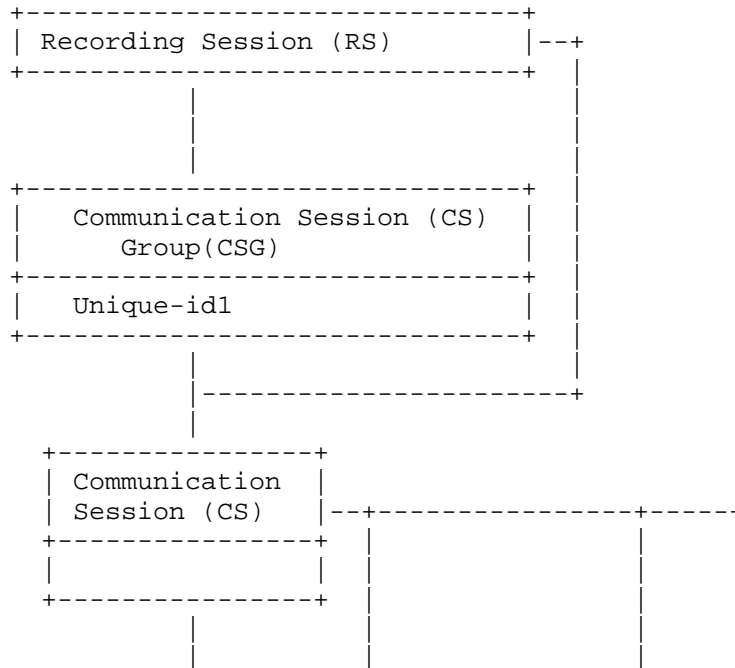
NOTE: There MAY be other ways to model the same use cases depending on what information the SRC has.

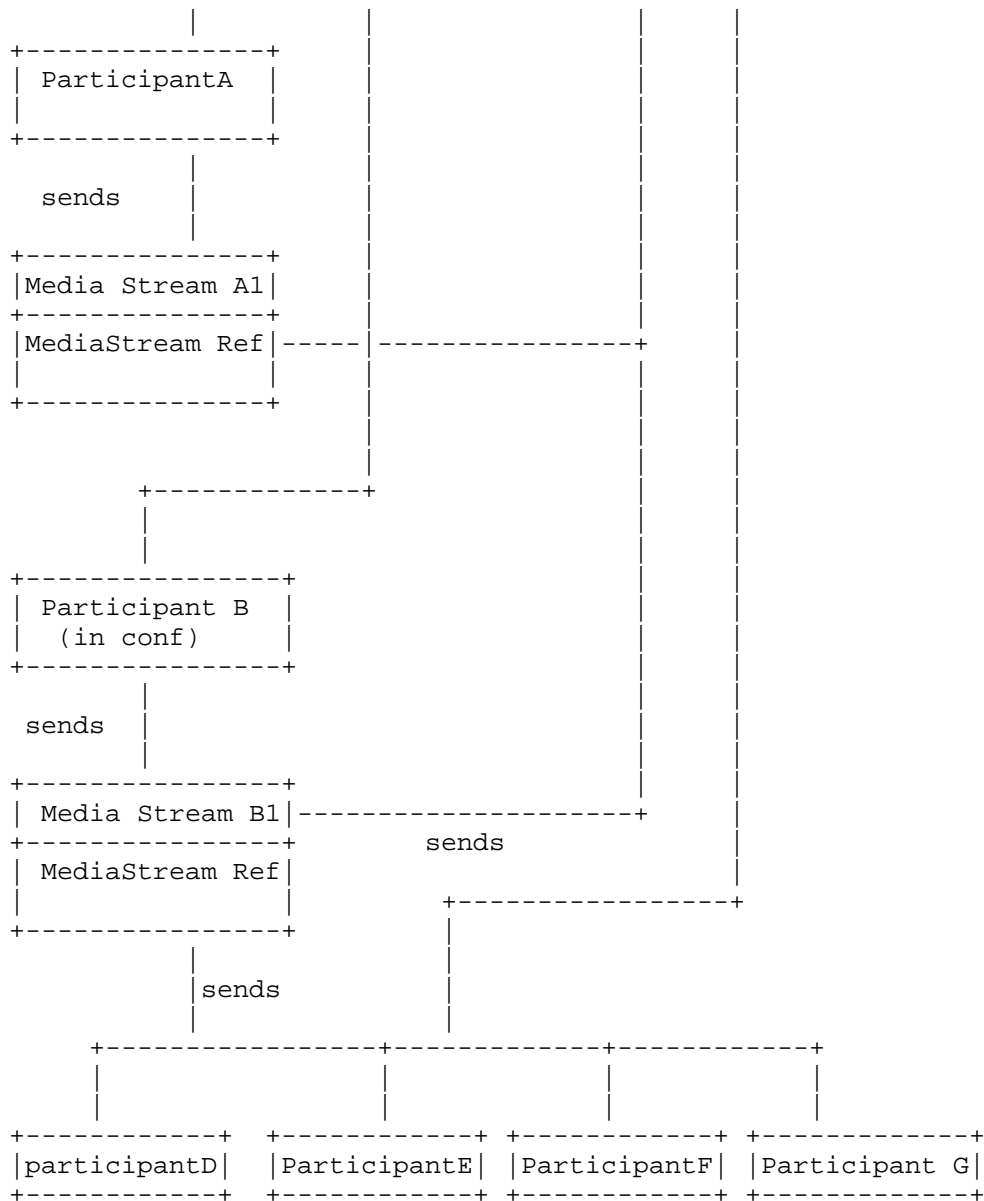
12.4.1. Case 1:

This is the usecase where there is a CS with one of the participant (who is also SRC) as a user in a conference. For the sake of simplicity the receive lines for each of the participant is not shown.



Instance Diagram:



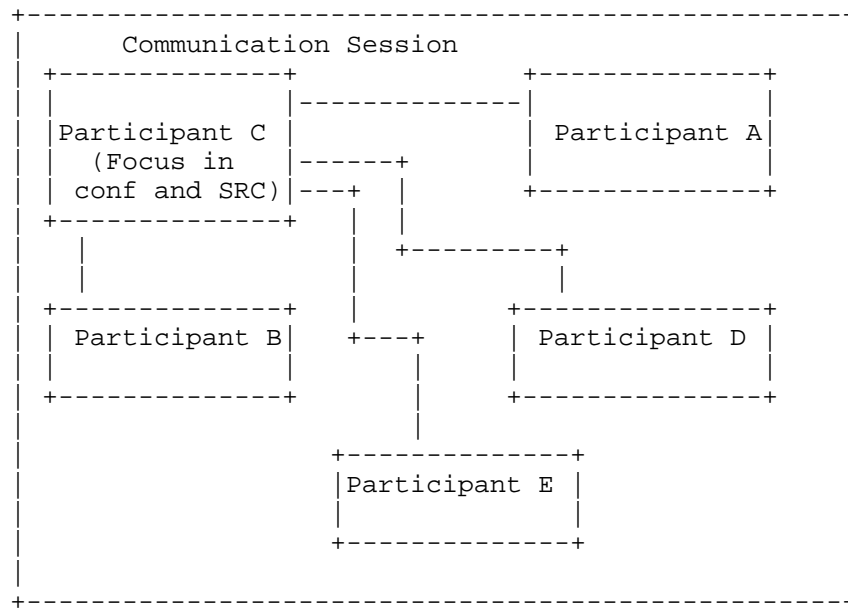


In this example we have two participants A and B who are part of a Communication Session(CS). One of the participants B is part of a conference and also acts as SRC. There can be two cases here. B can be a participant of the conference or B can be a focus. In this

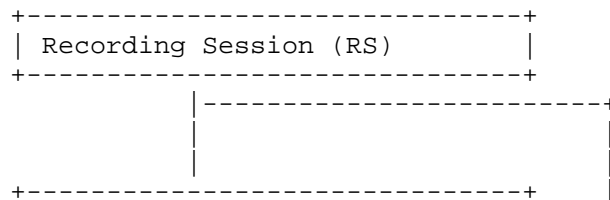
instance diagram Participant B is a user in a conference. The SRC (Participant B) subscribes to conference event package to get the details of other participants. Participant B(SRC) sends the same through the metadata to SRS. In this instance diagram the Media Stream(mixed stream) sent from Participant B has media streams contributed by conference participants (D,E,F and G). For the sake of simplicity the "receives" line is not shown here. In this example the media stream sent by each participant(A or B) of CS is received by all other participant(A or B).

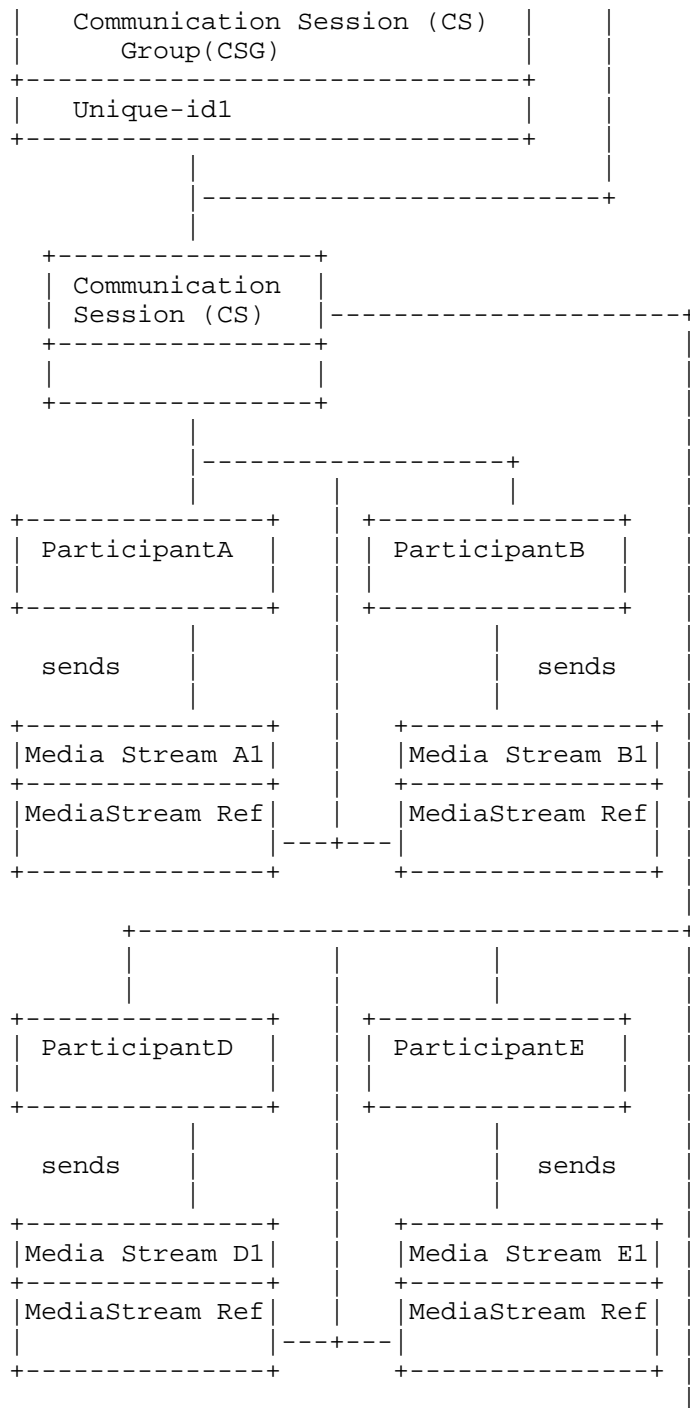
12.4.2. Case 2:

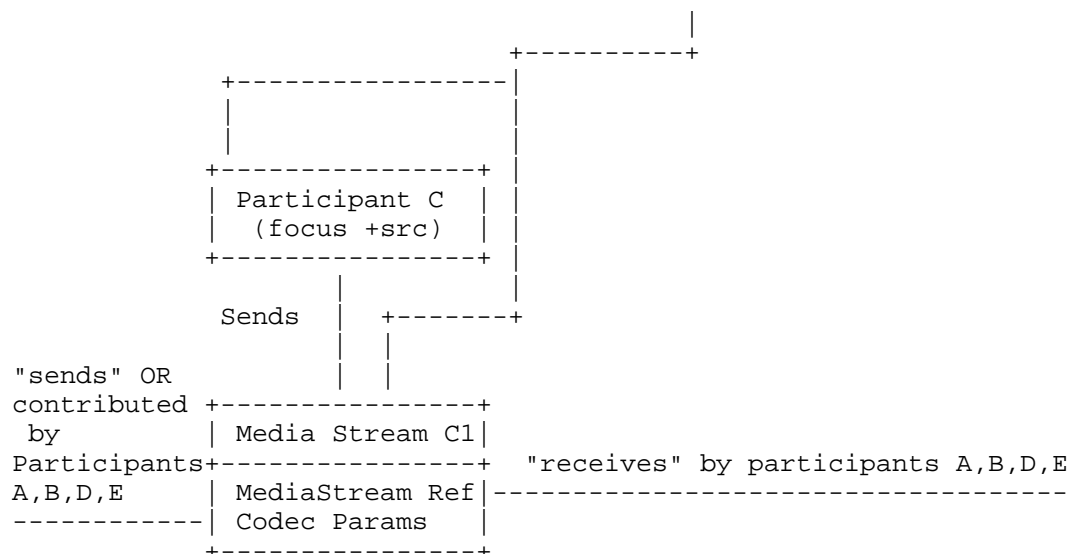
This is the usecase where there is a CS where one of the participant is focus (which is also SRC).



Instance Diagram:





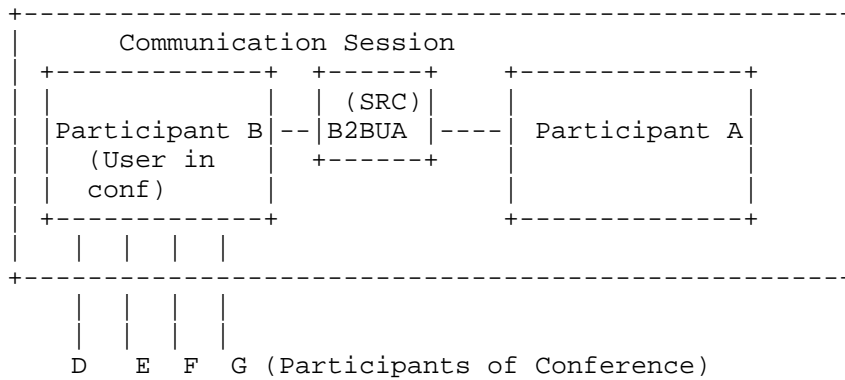


In this example we have two participants A and B who are part of a Communication Session(CS). One of the participants (C) is focus of a conference and also acts as SRC. The SRC (Participant C) being the Focus of the conference has access to the details of other participants. SRC (Participant C) sends the same through the metadata to SRS. In this instance diagram the Media Stream(mixed stream) sent by C has media streams contributed by conference participants (A, B, D and E). Participants A, B,D and E sends Media Streams A1, B1, D1 and E1 respectively. The media stream sent by Participant C(Focus) is received by all other participants of CS. For the sake of simplicity the "receives" line is not shown linked to all other participants.

NOTE: SRC (Participant C) can send mixed stream or seperate streams to SRS

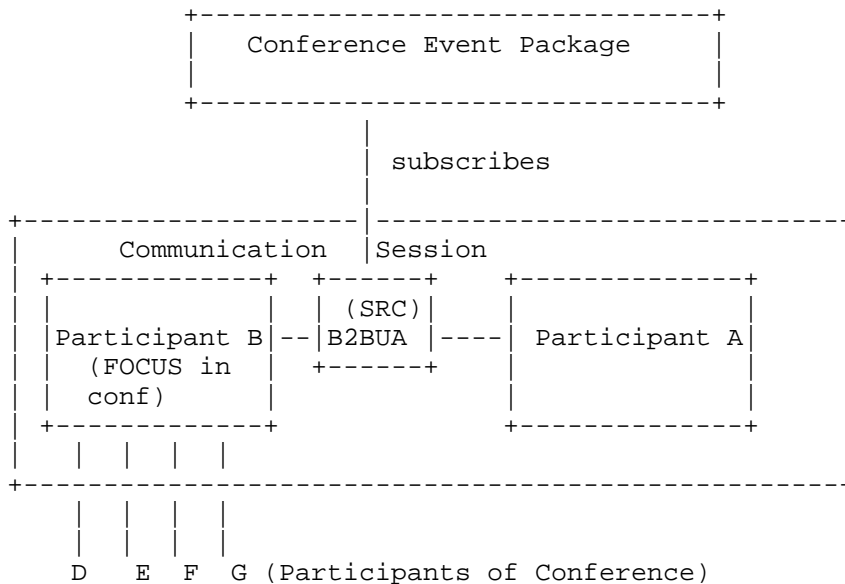
12.4.3. Case 3:

A CS where one of the participant is user and the SRC is a different entity like B2BUA. In this case the SRC may not know that one of the user is part of conference. Hence the instance diagram will not have information about the conference participants.



12.4.4. Case 4:

A CS where one of the participant is focus and the SRC is a different entity like B2BUA. In this case the participant which is focus sends "isfocus" in SIP message to SRC. The SRC subscribe to conference event package on seeing this "isfocus". SRC learns the details of other participants of conference from the conference package and send the same in metadata to SRS. The instance diagram for this use case is same as Case 1.



13. Appendix B: Metadata XML schema Instances

Note: This Appendix has to be moved to callflow document after the discussion in the mailing alias

This section describes the metadata model XML instances for different use cases of SIPREC. For the sake of simplicity the complete SIP messages are NOT shown here.

13.1. Use case 1: Basic Call

Basic call between two Participants A(Alice) and B(Bob) who are part of one session. In this use case each participant sends two Media Streams. Media Streams sent by each participant is received all other participants of that CS in this use-case. Below is the initial snapshot sent by SRC that has complete metadata. For the sake of completeness even snippets of SDP is shown. For the sake of simplicity these use-cases assume the RS stream is unmixed.

```
Content-Type: application/SDP
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>complete</dataMode>
    <group group_id="7+OTCyoxTmqmqyA/lweDAg==">
      <associate-time>2010-12-16T23:41:07Z</associate-time>
      <!-- Standardized extension -->
```

```
<call-center xmlns='urn:ietf:params:xml:ns:callcenter'>
  <supervisor>sip:alice@cisco.com</supervisor>
</call-center>
<mydata xmlns='http://example.com/my'>
  <structure>FOO!</structure>
  <whatever>bar</whatever>
</mydata>
</group>
<session session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <group-ref>7+OTCyoxTmqmqyA/lweDAg==
</group-ref>
  <start-time>2010-12-16T23:41:07Z</start-time>
  <!-- Standardized extension -->
  <structure>FOO!</structure>
  <whatever>bar</whatever>
</session>
<participant
  participant_id="srfBEImCRp2QB23b7Mpk0w=="
  <nameID aor=sip:alice@atlanta.com>
  <name xml:lang="it">Alice</name>
</nameID>
  <!-- Standardized extension -->
  <structure>FOO!</structure>
  <whatever>bar</whatever>
</participant>
<participantsessionassoc
  participant_id="srfBEImCRp2QB23b7Mpk0w=="
  session_id="hVpd7YQgRW2nD22h7q60JQ=="
  <associate-time>2010-12-16T23:41:07Z</associate-time>
</participantsessionassoc>
<participantstreamassoc
  participant_id="srfBEImCRp2QB23b7Mpk0w=="
  <send>i1Pz3to5hGk8fuXl+PbwCw==</send>
  <send>UAAMm5GRQKSCMVvLyl4rFw==</send>
  <recv>8zc6e0lYTlWIINA6GR+3ag==</recv>
  <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
</participantstreamassoc>
<participant
  participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
  <nameID aor=sip:bob@biloxi.com>
  <name xml:lang="it">Bob</name>
</nameID>
  <!-- Standardized extension -->
  <structure>FOO!</structure>
  <whatever>bar</whatever>
</participant>
<participantsessionassoc
  participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
```

```

        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <associate-time>2010-12-16T23:41:07Z</associate-time>
    </participantsessionassoc>
    <participantstreamassoc
        participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <send>8zc6e0lYTlWIINA6GR+3ag==</send>
        <send>EiXGlc+4TruqqoDaNE76ag==</send>
        <recv>UAAMm5GRQKSCMVvLyl4rFw==</recv>
        <recv>ilPz3to5hGk8fuXl+PbwCw==</recv>
    </participantstreamassoc>
    <stream id="UAAMm5GRQKSCMVvLyl4rFw=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>96</label>
    </stream>
    <stream id="ilPz3to5hGk8fuXl+PbwCw=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>97</label>
    </stream>
    <stream id="8zc6e0lYTlWIINA6GR+3ag=="
        session_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <label>98</label>
    </stream>
    <stream id="EiXGlc+4TruqqoDaNE76ag=="
        session_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <label>99</label>
    </stream>
</recording>

```

13.2. Use case 2: Hold/resume

Basic call between two Participants A and B. This is the continuation of above use-case. One of the participants(say A) goes on hold and then resumes as part of the same session. The metadata snapshot looks as below

During hold

```
Content-Type: application/SDP
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=inactive
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=inactive
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....

<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>partial</dataMode>
    <participantstreamassoc
      participant_id="srfBElmCRp2QB23b7Mpk0w=="
      session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <recv>8zc6e0lYtLWIINA6GR+3ag==</recv>
      <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
    </participantstreamassoc>
    <participantstreamassoc
      participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
      session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <send>8zc6e0lYtLWIINA6GR+3ag==</send>
      <send>EiXGlc+4TruqqoDaNE76ag==</send>
    </participantstreamassoc>
  </recording>
```

During resume

The snapshot will look pretty much same as Use-case 1.

13.3. Use case 3: Basic Call with transfer

Basic call between two Participants A and B is connected as in Use-case 1. Transfer is initiated by one of the participants or by other entity(3PCC case). SRC sends a snapshot of the participant changes to SRS. In this instance participant A(Alice) drops out during the transfer and Participant C(Paul) joins the session. There can be two cases here, same session continues after transfer or a new session (e.g. REFER based transfer) is created

Transfer with same session retained - (.e.g. RE-INVITE based transfer). Participant A drops out and C is added to the same session. No change to session/group element. C will be new stream element which maps to RS SDP using the same labels in this instance.

```
Content-Type: application/SDP
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>partial</dataMode>
    <participantstreamassoc
      participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
      <send>8zc6e0lYtLWIINA6GR+3ag==</send>
      <send>EiXGlc+4TruqqoDaNE76ag==</send>
      <recv>60JAJm9UTvik0Ltlih/Gzw==</recv>
      <recv>AcR5FUd3Edi8cACQJy/3JQ==</recv>
    </participantstreamassoc>
```

```

<participant
  participant_id="AtnmlZRnOC6Pm5MApkrDzQ==">
    <nameID aor=sip:paul@example.com>
      <name xml:lang="it">Paul</name>
    </nameID>
    <!-- Standardized extension -->
    <structure>FOO!</structure>
    <whatever>bar</whatever>
  </participant>
  <participantsessionassoc
    participant_id="AtnmlZRnOC6Pm5MApkrDzQ=="
    session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <associate-time>2010-12-16T23:41:07Z</associate-time>
    </participantsession>
    <participantstreamassoc
      participant_id="AtnmlZRnOC6Pm5MApkrDzQ==">
        <send>60JAJm9UTvik0Ltlih/Gzw==</send>
        <send>AcR5FUd3Edi8cACQJy/3JQ==</send>
        <recv>8zc6e0lYTlWIINA6GR+3ag==</recv>
        <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
      </participantstreamassoc>
      <stream stream_id="60JAJm9UTvik0Ltlih/Gzw=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>96</label>
      </stream>
      <stream stream_id="AcR5FUd3Edi8cACQJy/3JQ=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>97</label>
      </stream>
      <stream stream_id="8zc6e0lYTlWIINA6GR+3ag=="
        session_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <label>98</label>
      </stream>
      <stream stream_id="EiXGlc+4TruqqoDaNE76ag=="
        session_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <label>99</label>
      </stream>
    </recording>

```

Transfer with new session - (.e.g. REFER based transfer). In this case new session is part of same grouping (done by SRC).

SRC may send an optional snapshot indicating stop for the old session.

```

    <?xml version="1.0" encoding="UTF-8"?>
    <recording xmlns='urn:ietf:params:xml:ns:recording'>
      <dataMode>Partial</dataMode>
      <session session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <group-ref>7+OTCyoxTmqmqyA/lweDAg==
        </group-ref>
        <stop-time>2010-12-16T23:41:07Z</stop-time>
        <!-- Standardized extension -->
        <structure>FOO!</structure>
        <whatever>bar</whatever>
      </session>
      <participantsessionassoc
        participant_id="srfBEImCRp2QB23b7Mpk0w=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
      </participantsessionassoc>
      <participantsessionassoc
        participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
      </participantsessionassoc>
    </recording>

```

SRC sends a snapshot to indicate the participant change and new session information after transfer. In this example the same RS is used.

Content-Type: application/SDP

```

...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000

```

```

a=label:99
a=sendonly
....
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>partial</dataMode>
    <session session_id="bfLZ+NTFEeCNxQTuRyQBmw==">
      <group-ref>7+OTCyoxTmqmqyA/lweDAg==
      </group-ref>
      <start-time>2010-12-16T23:41:07Z</start-time>
      <!-- Standardized extension -->
      <structure>FOO!</structure>
      <whatever>bar</whatever>
    </session>
    <participant
      participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
      <nameID aor=sip:Alice@atlanta.com/>
      <!-- Standardized extension -->
      <structure>FOO!</structure>
      <whatever>bar</whatever>
    </participant>
    <participantsessionassoc
      participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
      session_id="bfLZ+NTFEeCNxQTuRyQBmw=="
      <associate-time>2010-12-16T23:32:03Z</associate-time>
    </participantsessionassoc>
    <participantstreamassoc
      participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
      <send>8zc6e0lYTlWIINA6GR+3ag==</send>
      <send>EiXGlc+4TruqqoDaNE76ag==</send>
      <recv>60JAJm9UTvik0Ltlih/Gzw==</recv>
      <recv>AcR5FUd3Edi8cACQJy/3JQ==</recv>
    </participantstreamassoc>
    <participant
      participant_id="AtnmlZRnOC6Pm5MApkrDzQ=="
      <nameID aor=sip:Bob@biloxi.com/>
      <!-- Standardized extension -->
      <structure>FOO!</structure>
      <whatever>bar</whatever>
    </participant>
    <participantsessionassoc
      participant_id="AtnmlZRnOC6Pm5MApkrDzQ=="
      session_id="bfLZ+NTFEeCNxQTuRyQBmw=="
      <associate-time>2010-12-16T23:41:07Z</associate-time>
    </participantsessionassoc>
    <participantstreamassoc
      participant_id="AtnmlZRnOC6Pm5MApkrDzQ=="
      <send>60JAJm9UTvik0Ltlih/Gzw==</send>

```

```
<send>AcR5FUd3Edi8cACQJy/3JQ==</send>
<recv>8zc6e0lYtLWIINA6GR+3ag==</recv>
<recv>EiXGlc+4TruqqoDaNE76ag==</recv>
</participantstreamassoc>
<stream stream_id="60JAJm9UTvik0Ltlih/Gzw=="
  session_id="bfLZ+NTFEeCNxQTuRyQBmw==">
  <label>96</label>
</stream>
<stream stream_id="AcR5FUd3Edi8cACQJy/3JQ=="
  session_id="bfLZ+NTFEeCNxQTuRyQBmw==">
  <label>97</label>
</stream>
<stream stream_id="8zc6e0lYtLWIINA6GR+3ag=="
  session_id="bfLZ+NTFEeCNxQTuRyQBmw==">
  <label>98</label>
</stream>
<stream stream_id="EiXGlc+4TruqqoDaNE76ag=="
  session_id="bfLZ+NTFEeCNxQTuRyQBmw==">
  <label>99</label>
</stream>
</recording>
```

13.4. Use Case 4: Call disconnect

This example shows a snapshot of metadata sent by an SRC at CS disconnect where the participants of CS are Alice and Bob

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>Partial</dataMode>
    <session session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <group-ref>7+OTCyoxTmqmqyA/lweDAg==</group-ref>
      <stop-time>2010-12-16T23:41:07Z</stop-time>
      <!-- Standardized extension -->
      <structure>FOO!</structure>
      <whatever>bar</whatever>
    </session>
    <participant
      participant_id="srfBEImCRp2QB23b7Mpk0w=="
      <nameID aor=sip:Alice@atlanta.com/>
      <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
    </participant>

    <participant
      participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
      <nameID aor=sip:Bob@biloxxy.com/>
      <disasociate-time>2010-12-16T23:41:07Z</disasociate-time>
    </participant>
  </recording>
```

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2141] Moats, R., "URN Syntax", RFC 2141, May 1997.
- [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.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session

Description Protocol", RFC 4566, July 2006.

- [RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.
- [RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", RFC 4796, February 2007.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

14.2. Informative References

- [RFC6341] Rehor, K., Portman, L., Hutton, A., and R. Jain, "Use Cases and Requirements for SIP-Based Media Recording (SIPREC)", RFC 6341, August 2011.
- [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-06 (work in progress), September 2012.
- [RFC2648] Moats, R., "A URN Namespace for IETF Documents", RFC 2648, August 1999.
- [RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, December 2002.
- [RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

Authors' Addresses

Ram Mohan Ravindranath
Cisco Systems, Inc.
Cessna Business Park,
Kadabeesanahalli Village, Varthur Hobli,
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: rmohanr@cisco.com

Parthasarathi Ravindran
Nokia Siemens Networks
Bangalore, Karnataka
India

Email: partha@parthasarathi.co.in

Paul Kyzivat
Huawei
Hudson, MA
USA

Email: pkyzivat@alum.mit.edu

