       Grouping of Media Lines in the Session Description Protocol (SDP)

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   This document defines two Session Description Protocol (SDP)
   attributes: "group" and "mid". They allow to group together several
   "m" lines for two different purposes: for lip synchronization and for
   receiving media from a single flow (several media streams) that are
   encoded in different formats during a particular session, on
   different ports and host interfaces.

Table of Contents

1. Introduction

   An SDP session description typically contains one or more media lines
   - they are commonly known as "m" lines. When a session description
   contains more than one "m" line, SDP does not provide any means to
   express a particular relationship between two or more of them. When
   an application receives an SDP session description with more than one
   "m" line, it is up to the application what to do with them. SDP does
   not carry any information about grouping media streams.

   While in some environments this information can be carried out of
   band, it would be desirable to have extensions to SDP that allow the
   expression of how different media streams within a session
   description relate to each other. This document defines such
   extensions.

2. Terminology

   In this document, the key words "MUST", "MUST NOT", "REQUIRED",
   "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY",
   and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119
   [1] and indicate requirement levels for compliant implementations.

3. Media Stream Identification Attribute

   A new "media stream identification" media attribute is defined. It
   is used for identifying media streams within a session description.
   Its formatting in SDP [2] is described by the following BNF:

        mid-attribute   = "a=mid:" identification-tag
        identification-tag = token

   The identification tag MUST be unique within an SDP session
   description.

4. Group Attribute

   A new "group" session-level attribute is defined. It is used for
   grouping together different media streams. Its formatting in SDP is
   described by the following BNF:

        group-attribute   = "a=group:" semantics
                              *(space identification-tag)
        semantics   = "LS" | "FID"

   This document defines two standard semantics: LS (Lip
   Synchronization) and FID (Flow Identification). Further semantics
   need to be defined in a standards-track document. However, defining
   new semantics apart from LS and FID is discouraged. Instead, it is
   RECOMMENDED to use other session description mechanisms such as
   SDPng.

5. Use of "group" and "mid"

   All the "m" lines of a session description that uses "group" MUST be
   identified with a "mid" attribute whether they appear in the group
   line(s) or not. If a session description contains at least one "m"
   line that has no "mid" identification the application MUST NOT
   perform any grouping of media lines.

   "a=group" lines are used to group together several "m" lines that are
   identified by their "mid" attribute. "a=group" lines that contain
   identification-tags that do not correspond to any "m" line within the
   session description MUST be ignored. The application acts as if the
   "a=group" line did not exist. The behavior of an application
   receiving an SDP with grouped "m" lines is defined by the semantics
   field in the "a=group" line.

There MAY be several "a=group" lines in a session description. All
the "a=group" lines of a session description MAY or MAY NOT use the
same semantics. An "m" line identified by its "mid" attribute MAY
appear in more than one "a=group" line as long as the "a=group" lines
use different semantics. An "m" line identified by its "mid"
attribute MUST NOT appear in more than one "a=group" line using the
same semantics.

6. Lip Synchronization (LS)

An application that receives a session description that contains "m"
lines that are grouped together using LS semantics MUST synchronize
the playout of the corresponding media streams. Note that LS
semantics not only apply to a video stream that has to be
synchronized with an audio stream. The playout of two streams of the
same type can be synchronized as well.

For RTP streams synchronization is typically performed using RTCP,
which provides enough information to map time stamps from the
different streams into a wall clock. However, the concept of media
stream synchronization MAY also apply to media streams that do not
make use of RTP. If this is the case, the application MUST recover
the original timing relationship between the streams using whatever
available mechanism.

6.1 Example of LS

   The following example shows a session description of a conference
   that is being multicast. The first media stream (mid:1) contains the
   voice of the speaker who speaks in English. The second media stream
   (mid:2) contains the video component and the third (mid:3) media
   stream carries the translation to Spanish of what he is saying. The
   first and the second media streams MUST be synchronized.

```
v=0
o=Laura 289083124 289083124 IN IP4 one.example.com
t=0 0
c=IN IP4 224.2.17.12/127
a=group:LS 1 2
m=audio 30000 RTP/AVP 0
a=mid:1
m=video 30002 RTP/AVP 31
a=mid:2
m=audio 30004 RTP/AVP 0
i=This media stream contains the Spanish translation
a=mid:3
```

   Note that although the third media stream is not present in the group
   line, it still MUST contain a mid attribute (mid:3), as stated
   before.

7. Flow Identification (FID)

   An "m" line in an SDP session description defines a media stream.
   However, SDP does not define what a media stream is. This definition
   can be found in the RTSP specification. The RTSP RFC [5] defines a
   media stream as "a single media instance, e.g., an audio stream or a
   video stream as well as a single whiteboard or shared application
   group. When using RTP, a stream consists of all RTP and RTCP packets
   created by a source within an RTP session".

   This definition assumes that a single audio (or video) stream maps
   into an RTP session. The RTP RFC [6] defines an RTP session as
   follows: "For each participant, the session is defined by a
   particular pair of destination transport addresses (one network
   address plus a port pair for RTP and RTCP)".

   While the previous definitions cover the most common cases, there are
   situations where a single media instance, (e.g., an audio stream or a
   video stream) is sent using more than one RTP session. Two examples
   (among many others) of this kind of situation are cellular systems
   using SIP [3] and systems receiving DTMF tones on a different host
   than the voice.

7.1 SIP and Cellular Access

   Systems using a cellular access and SIP as a signalling protocol need
   to receive media over the air. During a session the media can be
   encoded using different codecs. The encoded media has to traverse
   the radio interface. The radio interface is generally characterized
   by being bit error prone and associated with relatively high packet
   transfer delays. In addition, radio interface resources in a
   cellular environment are scarce and thus expensive, which calls for
   special measures in providing a highly efficient transport. In order
   to get an appropriate speech quality in combination with an efficient
   transport, precise knowledge of codec properties are required so that
   a proper radio bearer for the RTP session can be configured before
   transferring the media. These radio bearers are dedicated bearers
   per media type, i.e., codec.

   Cellular systems typically configure different radio bearers on
   different port numbers. Therefore, incoming media has to have
   different destination port numbers for the different possible codecs
   in order to be routed properly to the correct radio bearer. Thus,
   this is an example in which several RTP sessions are used to carry a
   single media instance (the encoded speech from the sender).

7.2 DTMF Tones

   Some voice sessions include DTMF tones. Sometimes the voice handling
   is performed by a different host than the DTMF handling. It is
   common to have an application server in the network gathering DTMF
   tones for the user while the user receives the encoded speech on his
   user agent. In this situations it is necessary to establish two RTP
   sessions: one for the voice and the other for the DTMF tones. Both
   RTP sessions are logically part of the same media instance.

7.3 Media Flow Definition

   The previous examples show that the definition of a media stream in
   [5] do not cover some scenarios. It cannot be assumed that a single
   media instance maps into a single RTP session. Therefore, we
   introduce the definition of a media flow:

   Media flow consists of a single media instance, e.g., an audio stream
   or a video stream as well as a single whiteboard or shared
   application group. When using RTP, a media flow comprises one or
   more RTP sessions.

7.4 FID Semantics

   Several "m" lines grouped together using FID semantics form a media
   flow. A media agent handling a media flow that comprises several "m"
   lines MUST send a copy of the media to every "m" line part of the
   flow as long as the codecs and the direction attribute present in a
   particular "m" line allow it.

   It is assumed that the application uses only one codec at a time to
   encode the media produced. This codec MAY change dynamically during
   the session, but at any particular moment only one codec is in use.

   The application encodes the media using the current codec and checks
   one by one all the "m" lines that are part of the flow. If a
   particular "m" line contains the codec being used and the direction
   attribute is "sendonly" or "sendrecv", a copy of the encoded media is
   sent to the address/port specified in that particular media stream.
   If either the "m" line does not contain the codec being used or the
   direction attribute is neither "sendonly" nor "sendrecv", nothing is
   sent over this media stream.

   The application typically ends up sending media to different
   destinations (IP address/port number) depending on the codec used at
   any moment.

7.4.1 Examples of FID

   The session description below might be sent by a SIP user agent using
   a cellular access. The user agent supports GSM on port 30000 and AMR
   on port 30002. When the remote party sends GSM, it will send RTP
   packets to port number 30000. When AMR is the codec chosen, packets
   will be sent to port 30002. Note that the remote party can switch
   between both codecs dynamically in the middle of the session.
   However, in this example, only one media stream at a time carries
   voice. The other remains "muted" while its corresponding codec is
   not in use.

```
     v=0
     o=Laura 289083124 289083124 IN IP4 two.example.com
     t=0 0
     c=IN IP4 131.160.1.112
     a=group:FID 1 2
     m=audio 30000 RTP/AVP 3
     a=rtpmap:3 GSM/8000
     a=mid:1
     m=audio 30002 RTP/AVP 97
     a=rtpmap:97 AMR/8000
     a=fmtp:97 mode-set=0,2,5,7; mode-change-period=2;
  mode-change-neighbor; maxframes=1
     a=mid:2
```

   (The linebreak in the fmtp line accommodates RFC formatting
   restrictions; SDP does not have continuation lines.)

   In the previous example, a system receives media on the same IP
   address on different port numbers. The following example shows how a
   system can receive different codecs on different IP addresses.

```
     v=0
     o=Laura 289083124 289083124 IN IP4 three.example.com
     t=0 0
     c=IN IP4 131.160.1.112
     a=group:FID 1 2
     m=audio 20000 RTP/AVP 0
     c=IN IP4 131.160.1.111
     a=rtpmap:0 PCMU/8000
     a=mid:1
     m=audio 30002 RTP/AVP 97
     a=rtpmap:97 AMR/8000
     a=fmtp:97 mode-set=0,2,5,7; mode-change-period=2;
   mode-change-neighbor; maxframes=1
     a=mid:2
```

   (The linebreak in the fmtp line accomodates RFC formatting
   restrictions; SDP does not have continuation lines.)

   The cellular terminal of this example only supports the AMR codec.
   However, many current IP phones only support PCM (payload 0). In
   order to be able to interoperate with them, the cellular terminal
   uses a transcoder whose IP address is 131.160.1.111. The cellular
   terminal includes in its SDP support for PCM at that IP address.
   Remote systems will send AMR directly to the terminal but PCM will be
   sent to the transcoder. The transcoder will be configured (using
   whatever method) to convert the incoming PCM audio to AMR and send it
   to the terminal.

   The next example shows how the "group" attribute used with FID
   semantics can indicate the use of two different codecs in the two
   directions of a bidirectional media stream.

```
     v=0
     o=Laura 289083124 289083124 IN IP4 four.example.com
     t=0 0
     c=IN IP4 131.160.1.112
     a=group:FID 1 2
     m=audio 30000 RTP/AVP 0
     a=mid:1
     m=audio 30002 RTP/AVP 8
     a=recvonly
     a=mid:2
```

   A user agent that receives the SDP above knows that at a certain
   moment it can send either PCM u-law to port number 30000 or PCM A-law
   to port number 30002. However, the media agent also knows that the
   other end will only send PCM u-law (payload 0).

The following example shows a session description with different "m"
lines grouped together using FID semantics that contain the same
codec.

```
v=0
o=Laura 289083124 289083124 IN IP4 five.example.com
t=0 0
c=IN IP4 131.160.1.112
a=group:FID 1 2 3
m=audio 30000 RTP/AVP 0
a=mid:1
m=audio 30002 RTP/AVP 8
a=mid:2
m=audio 20000 RTP/AVP 0 8
c=IN IP4 131.160.1.111
a=recvonly
a=mid:3
```

At a particular point in time, if the media agent is sending PCM u-
law (payload 0), it sends RTP packets to 131.160.1.112 on port 30000
and to 131.160.1.111 on port 20000 (first and third "m" lines). If
it is sending PCM A-law (payload 8), it sends RTP packets to
131.160.1.112 on port 30002 and to 131.160.1.111 on port 20000
(second and third "m" lines).

The system that generated the SDP above supports PCM u-law on port
30000 and PCM A-law on port 30002. Besides, it uses an application
server whose IP address is 131.160.1.111 that records the
conversation. That is why the application server always receives a
copy of the audio stream regardless of the codec being used at any
given moment (it actually performs an RTP dump, so it can effectively
receive any codec).

Remember that if several "m" lines grouped together using FID
semantics contain the same codec the media agent MUST send media over
several RTP sessions at the same time.

   The last example of this section deals with DTMF tones. DTMF tones
   can be transmitted using a regular voice codec or can be transmitted
   as telephony events. The RTP payload for DTMF tones treated as
   telephone events is described in RFC 2833 [7]. Below, there is an
   example of an SDP session description using FID semantics and this
   payload type.

```
       v=0
       o=Laura 289083124 289083124 IN IP4 six.example.com
       t=0 0
       c=IN IP4 131.160.1.112
       a=group:FID 1 2
       m=audio 30000 RTP/AVP 0
       a=mid:1
       m=audio 20000 RTP/AVP 97
       c=IN IP4 131.160.1.111
       a=rtpmap:97 telephone-events
       a=mid:2
```

   The remote party would send PCM encoded voice (payload 0) to
   131.160.1.112 and DTMF tones encoded as telephony events to
   131.160.1.111. Note that only voice or DTMF is sent at a particular
   point of time. When DTMF tones are sent, the first media stream does
   not carry any data and, when voice is sent, there is no data in the
   second media stream. FID semantics provide different destinations
   for alternative codecs.

7.5 Scenarios that FID does not Cover

   It is worthwhile mentioning some scenarios where the "group"
   attribute using existing semantics (particularly FID) might seem to
   be applicable but is not.

7.5.1 Parallel Encoding Using Different Codecs

   FID semantics are useful when the application only uses one codec at
   a time. An application that encodes the same media using different
   codecs simultaneously MUST NOT use FID to group those media lines.
   Some systems that handle DTMF tones are a typical example of parallel
   encoding using different codecs.

   Some systems implement the RTP payload defined in RFC 2833, but when
   they send DTMF tones they do not mute the voice channel. Therefore,
   in effect they are sending two copies of the same DTMF tone: encoded
   as voice and encoded as a telephony event. When the receiver gets
   both copies, it typically uses the telephony event rather than the
   tone encoded as voice. FID semantics MUST NOT be used in this
   context to group both media streams since such a system is not using

   alternative codecs but rather different parallel encodings for the
   same information.

7.5.2 Layered Encoding

   Layered encoding schemes encode media in different layers. Quality
   at the receiver varies depending on the number of layers received.
   SDP provides a means to group together contiguous multicast addresses
   that transport different layers. The "c" line below:

       c=IN IP4 224.2.1.1/127/3

   is equivalent to the following three "c" lines:

       c=IN IP4 224.2.1.1/127
       c=IN IP4 224.2.1.2/127
       c=IN IP4 224.2.1.3/127

   FID MUST NOT be used to group "m" lines that do not represent the
   same information. Therefore, FID MUST NOT be used to group "m" lines
   that contain the different layers of layered encoding scheme.
   Besides, we do not define new group semantics to provide a more
   flexible way of grouping different layers because the already
   existing SDP mechanism covers the most useful scenarios.

7.5.3 Same IP Address and Port Number

   If several codecs have to be sent to the same IP address and port,
   the traditional SDP syntax of listing several codecs in the same "m"
   line MUST be used. FID MUST NOT be used to group "m" lines with the
   same IP address/port. Therefore, an SDP like the one below MUST NOT
   be generated.

       v=0
       o=Laura 289083124 289083124 IN IP4 six.example.com
       t=0 0
       c=IN IP4 131.160.1.112
       a=group:FID 1 2
       m=audio 30000 RTP/AVP 0
       a=mid:1
       m=audio 30000 RTP/AVP 8
       a=mid:2

The correct SDP for the session above would be the following one:

```
v=0
o=Laura 289083124 289083124 IN IP4 six.example.com
t=0 0
c=IN IP4 131.160.1.112
m=audio 30000 RTP/AVP 0 8
```

If two "m" lines are grouped using FID they MUST differ in their
transport addresses (i.e., IP address plus port).

8. Usage of the "group" Attribute in SIP

   SDP descriptions are used by several different protocols, SIP among
   them. We include a section about SIP because the "group" attribute
   will most likely be used mainly by SIP systems.

   SIP [3] is an application layer protocol for establishing,
   terminating and modifying multimedia sessions. SIP carries session
   descriptions in the bodies of the SIP messages but is independent
   from the protocol used for describing sessions. SDP [2] is one of
   the protocols that can be used for this purpose.

   At session establishment SIP provides a three-way handshake (INVITE-
   200 OK-ACK) between end systems. However, just two of these three
   messages carry SDP, as described in [4].

8.1 Mid Value in Answers

   The "mid" attribute is an identifier for a particular media stream.
   Therefore, the "mid" value in the offer MUST be the same as the "mid"
   value in the answer. Besides, subsequent offers (e.g., in a re-
   INVITE) SHOULD use the same "mid" value for the already existing
   media streams.

   RFC 3264 [4] describes the usage of SDP in relation to SIP. The
   offerer and the answerer align their media description so that the
   nth media stream ("m=" line) in the offerer's session description
   corresponds to the nth media stream in the answerer's description.

   The presence of the "group" attribute in an SDP session description
   does not modify this behavior.

   Since the "mid" attribute provides a means to label "m" lines, it
   would be possible to perform media alignment using "mid" labels
   rather than matching nth "m" lines. However this would not bring any

gain and would add complexity to implementations. Therefore SIP
systems MUST perform media alignment matching nth lines regardless of
the presence of the "group" or "mid" attributes.

If a media stream that contained a particular "mid" identifier in the
offer contains a different identifier in the answer the application
ignores all the "mid" and "group" lines that might appear in the
session description. The following example illustrates this
scenario.

8.1.1 Example

Two SIP entities exchange SDPs during session establishment. The
INVITE contains the SDP below:

```
v=0
o=Laura 289083124 289083124 IN IP4 seven.example.com
t=0 0
c=IN IP4 131.160.1.112
a=group:FID 1 2
m=audio 30000 RTP/AVP 0 8
a=mid:1
m=audio 30002 RTP/AVP 0 8
a=mid:2
```

The 200 OK response contains the following SDP:

```
v=0
o=Bob 289083122 289083122 IN IP4 eigth.example.com
t=0 0
c=IN IP4 131.160.1.113
a=group:FID 1 2
m=audio 25000 RTP/AVP 0 8
a=mid:2
m=audio 25002 RTP/AVP 0 8
a=mid:1
```

Since alignment of "m" lines is performed based on matching of nth
lines, the first stream had "mid:1" in the INVITE and "mid:2" in the
200 OK. Therefore, the application MUST ignore every "mid" and
"group" lines contained in the SDP.

A well-behaved SIP user agent would have returned the SDP below in
the 200 OK:

```
v=0
o=Bob 289083122 289083122 IN IP4 nine.example.com
t=0 0
c=IN IP4 131.160.1.113
a=group:FID 1 2
m=audio 25002 RTP/AVP 0 8
a=mid:1
m=audio 25000 RTP/AVP 0 8
a=mid:2
```

8.2 Group Value in Answers

A SIP entity that receives an offer that contains an "a=group" line
with semantics that it does not understand MUST return an answer
without the "group" line. Note that, as it was described in the
previous section, the "mid" lines MUST still be present in the
answer.

A SIP entity that receives an offer that contains an "a=group" line
with semantics that are understood MUST return an answer that
contains an "a=group" line with the same semantics. The
identification-tags contained in this "a=group" lines MUST be the
same that were received in the offer or a subset of them (zero
identification-tags is a valid subset). When the identification-tags
in the answer are a subset, the "group" value to be used in the
session MUST be the one present in the answer.

SIP entities refuse media streams by setting the port to zero in the
corresponding "m" line. "a=group" lines MUST NOT contain
identification-tags that correspond to "m" lines with port zero.

Note that grouping of m lines MUST always be requested by the
offerer, never by the answerer. Since SIP provides a two-way SDP
exchange, an answerer that requested grouping would not know whether
the "group" attribute was accepted by the offerer or not. An
answerer that wants to group media lines SHOULD issue another offer
after having responded to the first one (in a re-INVITE for
instance).

8.2.1 Example

The example below shows how the callee refuses a media stream offered
by the caller by setting its port number to zero. The "mid" value
corresponding to that media stream is removed from the "group" value
in the answer.

SDP in the INVITE from caller to callee:

```
v=0
o=Laura 289083124 289083124 IN IP4 ten.example.com
t=0 0
c=IN IP4 131.160.1.112
a=group:FID 1 2 3
m=audio 30000 RTP/AVP 0
a=mid:1
m=audio 30002 RTP/AVP 8
a=mid:2
m=audio 30004 RTP/AVP 3
a=mid:3
```

SDP in the INVITE from callee to caller:

```
v=0
o=Bob 289083125 289083125 IN IP4 eleven.example.com
t=0 0
c=IN IP4 131.160.1.113
a=group:FID 1 3
m=audio 20000 RTP/AVP 0
a=mid:1
m=audio 0 RTP/AVP 8
a=mid:2
m=audio 20002 RTP/AVP 3
a=mid:3
```

8.3 Capability Negotiation

A client that understands "group" and "mid" but does not want to make
use of them in a particular session MAY want to indicate that it
supports them. If a client decides to do that, it SHOULD add an
"a=group" line with no identification-tags for every semantics it
understands.

If a server receives an offer that contains empty "a=group" lines, it
SHOULD add its capabilities also in the form of empty "a=group" lines
to its answer.

8.3.1 Example

   A system that supports both LS and FID semantics but does not want to
   group any media stream for this particular session generates the
   following SDP:

       v=0
       o=Bob 289083125 289083125 IN IP4 twelve.example.com
       t=0 0
       c=IN IP4 131.160.1.113
       a=group:LS
       a=group:FID
       m=audio 20000 RTP/AVP 0 8

   The server that receives that offer supports FID but not LS. It
   responds with the SDP below:

       v=0
       o=Laura 289083124 289083124 IN IP4 thirteen.example.com
       t=0 0
       c=IN IP4 131.160.1.112
       a=group:FID
       m=audio 30000 RTP/AVP 0

8.4 Backward Compatibility

   This document does not define any SIP "Require" header. Therefore,
   if one of the SIP user agents does not understand the "group"
   attribute the standard SDP fall back mechanism MUST be used
   (attributes that are not understood are simply ignored).

8.4.1 Offerer does not Support "group"

   This situation does not represent a problem because grouping requests
   are always performed by offerers, not by answerers. If the offerer
   does not support "group" this attribute will just not be used.

8.4.2 Answerer does not Support "group"

   The answerer will ignore the "group" attribute, since it does not
   understand it (it will also ignore the "mid" attribute). For LS
   semantics, the answerer might decide to perform or to not perform
   synchronization between media streams.

   For FID semantics, the answerer will consider that the session
   comprises several media streams.

   Different implementations would behave in different ways.

In the case of audio and different "m" lines for different codecs an
implementation might decide to act as a mixer with the different
incoming RTP sessions, which is the correct behavior.

An implementation might also decide to refuse the request (e.g., 488
Not acceptable here or 606 Not Acceptable) because it contains
several "m" lines. In this case, the server does not support the
type of session that the caller wanted to establish. In case the
client is willing to establish a simpler session anyway, he SHOULD
re-try the request without "group" attribute and only one "m" line
per flow.

9. Security Considerations

   Using the "group" parameter with FID semantics, an entity that
   managed to modify the session descriptions exchanged between the
   participants to establish a multimedia session could force the
   participants to send a copy of the media to any particular
   destination.

   Integrity mechanism provided by protocols used to exchange session
   descriptions and media encryption can be used to prevent this attack.

10. IANA Considerations

   This document defines two SDP attributes: "mid" and "group".

   The "mid" attribute is used to identify media streams within a
   session description and its format is defined in Section 3.

   The "group" attribute is used for grouping together different media
   streams and its format is defined in Section 4.

   This document defines a framework to group media lines in SDP using
   different semantics. Semantics to be used with this framework are
   registered by the IANA when they are published in standards track
   RFCs.

   The IANA Considerations section of the RFC MUST include the following
   information, which appears in the IANA registry along with the RFC
   number of the publication.

      o A brief description of the semantics.

      o Token to be used within the group attribute. This token may be
         of any length, but SHOULD be no more than four characters long.

      o Reference to an standards track RFC.

The only entries in the registry for the time being are:

```
Semantics           Token  Reference
-------------------  -----  -----------
Lip synchronization  LS     RFC 3388
Flow identification  FID    RFC 3388
```

11. Acknowledgments

The authors would like to thank Jonathan Rosenberg, Adam Roach, Orit Levin and Joerg Ott for their feedback on this document.

12. References

12.1 Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[3] 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.

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

12.2 Informative References

[5] Schulzrinne, H., Rao, A. and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[6] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[7] Schulzrinne, H. and S. Petrack, "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 2833, May 2000.

13. Authors' Addresses

Gonzalo Camarillo
Ericsson
Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland

Phone: +358 9 299 3371
Fax: +358 9 299 3052
EMail: Gonzalo.Camarillo@ericsson.com

Jan Holler
Ericsson Research
S-16480 Stockholm
Sweden

Phone: +46 8 58532845
Fax: +46 8 4047020
EMail: Jan.Holler@era.ericsson.se

Goran AP Eriksson
Ericsson Research
S-16480 Stockholm
Sweden

Phone: +46 8 58531762
Fax: +46 8 4047020
EMail: Goran.AP.Eriksson@era.ericsson.se

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA

EMail: schulzrinne@cs.columbia.edu

14. Full Copyright Statement

Acknowledgement