                      SDP Capability Negotiation:
                 Requirements and Review of Existing Work

           draft-andreasen-mmusic-sdp-capability-negotiation-reqts-00.txt

Status of this Memo

    By submitting this Internet-Draft, each author represents that
    any applicable patent or other IPR claims of which he or she is
    aware have been or will be disclosed, and any of which he or she
    becomes aware will be disclosed, in accordance with Section 6 of
    BCP 79.

    Internet-Drafts are working documents of the Internet Engineering
    Task Force (IETF), its areas, and its working groups. Note that
    other groups may also distribute working documents as Internet-
    Drafts.

    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."

    The list of current Internet-Drafts can be accessed at
         http://www.ietf.org/ietf/1id-abstracts.txt

    The list of Internet-Draft Shadow Directories can be accessed at
         http://www.ietf.org/shadow.html

    This Internet-Draft will expire on June 10, 2007.

Abstract

    The Session Description Protocol (SDP) was intended for describing
    multimedia sessions for the purposes of session announcement, session
    invitation, and other forms of multimedia session initiation. SDP was
    not intended to provide capability indication or capability
    negotiation, however over the years, SDP has seen widespread adoption
    and as a result it has been gradually extended to provide limited
    support for these. SDP and its current extensions however do not have

the ability to negotiate one or more alternative transport protocols
(e.g. RTP profiles) which makes it particularly difficult to deploy
new RTP profiles such as secure RTP and RTP with RTCP-based feedback.
The purpose of this document is to identify a set of requirements for
SDP Capability Negotiation and evaluate existing work in this area.
The document does not provide any solutions to SDP Capability
Negotiation

Conventions used in this document

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].

Table of Contents

1. Introduction

The Session Description Protocol (SDP) was intended for describing
multimedia sessions for the purposes of session announcement, session
invitation, and other forms of multimedia session initiation. The SDP

contains one or more media stream descriptions with information such
as IP-address and port, type of media stream (e.g. audio or video),
transport protocol (possibly including profile information, e.g.
RTP/AVP or RTP/SAVP), media formats (e.g. codecs), and various other
session and media stream parameters that define the session.

Simply providing media stream descriptions is sufficient for session
announcements for a broadcast application, where the media stream
parameters are fixed for all participants. When a participant wants
to join the session, he obtains the session announcement and uses the
media descriptions provided, e.g., joins a multicast group and
receives media packets in the encoding format specified. If the
media stream description is not supported by the participant, he is
unable to receive the media.

Such restrictions are not generally acceptable to multimedia session
invitations, where two or more entities attempt to establish a media
session using a set of media stream parameters acceptable to all
participants. First of all, each entity must inform the other of its
receive address, and secondly, the entities need to agree on the
media stream parameters to use for the session, e.g. transport
protocols and codecs. We here make a distinction between the
capabilities supported by each participant and the parameters that
can actually be used for the session. More generally, we can say that
we have the following:

o A set of capabilities, or potential configurations of the media
   stream components, supported by each side.

o A set of actual configurations of the media stream components,
   which specifies which media stream components to use and with what
   parameters.

o A negotiation process that takes the set of potential
   configurations (capabilities) as input and provides the actual
   configurations as output.

SDP by itself was designed to provide only the second of these, i.e.,
the actual configurations, however over the years, use of SDP has
been extended beyond its original scope. Session negotiation
semantics were defined by the offer/answer model in RFC 3264. It
defines how two entities, an offerer and an answerer, exchange
session descriptions to negotiate a session. The offerer can include
one or more media formats (codecs) per media stream, and the answerer
then selects one or more of those offered and returns them in an
answer. Both the offer and the answer contain actual configurations -

potential configurations are not supported. The answer however may reduce the set of actual configurations from the offer.

Other relevant extensions have been defined. Simple capability declarations, which defines how to provide a simple and limited set of capability descriptions in SDP was defined in RFC 3407. Grouping of media lines, which defines how media lines in SDP can have other semantics than the traditional "simultaneous media streams" semantics, was defined in RFC 3388, etc.

Each of these extensions was designed to solve a specific limitation of SDP. Since SDP had already been stretched beyond its original intent, a more comprehensive capability declaration and negotiation process was intentionally not defined. Instead, work on a "next generation" of a protocol to provide session description and capability negotiation was initiated [SDPng]. SDPng however has not gained traction and has remained as work in progress for an extended period of time. Existing real-time multimedia communication protocols such as SIP, RTSP, Megaco, and MGCP continue to use SDP. SDP and its current extensions however do not address an increasingly important problem: the ability to negotiate one or more alternative transport protocols (e.g., RTP profiles). This makes it difficult to deploy new RTP profiles such as secure RTP (SRTP) [SRTP], RTP with RTCP-Based Feedback [AVPF], etc. This particular problem is exacerbated by the fact that RTP profiles are defined independently. When a new profile is defined and N other profiles already exist, there is a potential need for defining N additional profiles, since profiles cannot be combined automatically. For example, in order to support the plain and secure RTP version of RTP with and without RTCP-based feedback, four separate profiles (and hence profile definitions) are needed: RTP/AVP [RFC3551], RTP/SAVP [SRTP], RTP/AVPF [AVPF], and RTP/SAVPF [SAVPF]. In addition to the pressing profile negotiation problem, other important real-life constraints have been found as well.

The purpose of this document is two-fold.

1. Identify a set of requirements for a SDP capability negotiation mechanism that will enable SDP to provide limited support for indicating potential configurations (capabilities) and negotiate the use of those potential configurations as actual configurations.

2. Review relevant existing work in the are of SDP capability negotiation and see how it aligns with the proposed requirements.

It should be noted, that it is not the intent to provide requirements for a full-fledged capability indication and negotiation mechanism

along the lines of SDPng [SDPng] or ITU-T H.245 [H245]. Rather, the focus is on identifying requirements for addressing a set of well-known real-life limitations.

The rest of the document is structured as follows. In Section 2, we provide a set of requirements for SDP capability negotiation, and in Section 3, we review relevant existing work in this area, how a solution based on that work might look, and the pros and cons associated with each. An actual solution to the proposed requirements will be provided separately.

2. Requirements

REQ-10: It MUST be possible to indicate and negotiate alternative media formats on a per media stream basis.

> For example, many implementations support multiple codecs, but only one at a time. Changes between codecs cannot be done on-the-fly, e.g. when receiving a simple RTP payload type change.

REQ-20: It MUST be possible to indicate and negotiate alternative attribute values ("a=") on a per media stream basis.

> For example, T.38 [T38] defines new attributes that may need to be conveyed as part of a capability. Also, alternative SRTP keying mechanisms (e.g. [SDES] and [KMGMT]) may use SDP attributes to negotiate SRTP keying material.

REQ-25: It MUST be possible to indicate and negotiate alternative attribute values ("a=") at the session level.

> For example, [KMGMT] may indicate alternative key management material for MIKEY [MIKEY] at the session level.

REQ-30: It MUST be possible to indicate and negotiate alternative media format parameter values ("a=fmtp") per media format on a per media stream basis.

> For example, a media format (codec) indicated as an alternative capability may include fmtp parameters.

REQ-40: It MUST be possible to indicate and negotiate alternative transport protocols, e.g. different RTP profiles, on a per media stream basis.

> For example, "RTP/AVP" and "RTP/SAVP" may be alternatives.

REQ-50: It MUST be possible to indicate and negotiate alternative transport protocol and media type combinations on a per media stream basis.

> For example, an entity may support a fax call using either T.38 fax relay ("m=image <port> udptl t38") or PCMU ("m=audio <port> RTP/AVP 0").

REQ-80: The mechanism MUST be backwards compatible for SIP. Ideally, the mechanism should be completely transparent to entities that do not support it, without the need for any further signaling.

REQ-90: The mechanism MUST either be backwards compatible for Megaco and MGCP or it MUST be possible to interwork it with Megaco and MGCP without any additional signaling between the MGC and its peer (e.g. another SIP UA as opposed to a media gateway).

> For example, if a media gateway controller (MGC) uses SIP to communicate with peers, and the MGC uses Megaco or MGCP to control a media gateway, it must be possible to translate between the mechanism and normal SDP. Avoiding interworking requirements in the MGC is desirable.

REQ-100: The mechanism MUST work within the context of the offer/answer model [RFC3264]. Specifically, it MUST be possible to negotiate alternatives within a single offer/answer exchange.

REQ-110: The offer/answer model requires the offerer to be able to receive media for any media streams listed as either "recvonly" or "sendrecv" in an offer, as soon as that offer is generated. The mechanism MUST preserve this capability for all actual configurations included in an offer.

> Potential configurations do not have such a requirement.

REQ-120: The mechanism MUST enable inclusion of potential configurations (alternative capabilities) in the offer - the answer would then indicate which, if any of these potential configurations were accepted. The offerer is not required to process media for a specific potential configuration until the offerer receives an answer showing that potential configuration was accepted.

> Note that this implies that it may not be possible for the offerer to process early media generated using a potential configuration (as opposed to the actual configuration) until the answer has been received.

REQ-130: The mechanism MUST work in the presence of SIP forking.

REQ-140: The mechanism SHOULD be reasonably efficient in terms of overall message size.

   This is a relative requirement to evaluate alternative solutions as opposed to an absolute and quantifiable requirement. Use of compression techniques can help reduce the size of text-based messages, however it is still considered important to try and keep the message size reasonably small.

REQ-150: It MUST be possible to specify valid combinations of media lines.

   For example, an entity may be able to support audio and video or audio and IM, but not video and IM (or all three).

REQ-160: It MUST be possible to specify valid combinations of media formats between media streams.

   For example, there may be constraints on which combinations of audio and video codecs can be supported.

REQ-170: The mechanism MUST be extensible and allow for new types of capabilities to be specified and used in potential and actual configurations.

   For example, the mechanism could be extended to negotiate unicast or multicast addresses as alternatives.

REQ-300: The mechanism provided MUST be modular inasmuch as it can be divided into a required core set of functionality that all entities MUST support and an optional set of enhancements that entities MAY support. Entities that implement different sets of enhancements MUST be fully interoperable, albeit possibly with reduced functionality in terms of the actual negotiation performed.

   For example, not all entities may implement support for REQ-150 and REQ-160

REQ-310: The following requirements are considered enhancements as defined in REQ-300:

  * REQ-50 (alternative combinations of transport protocol and media type)

  * REQ-150 (valid combinations of media lines)

* REQ-160 (valid combinations of media formats between media
  streams)

[Editor's Note: Should we have any security requirements - see e.g.
Section 4.

Above, we presented the requirements for the capability negotiation
mechanism. Below, we provide a set of features that were considered
and then explicitly deemed to be out-of-scope:

o Support for negotiation of unicast and multicast addresses as
   alternatives. It was suggested as a requirement initially, but
   subsequent discussion led to its removal.

o Support for negotiation of IPv4 and IPv6 addresses as
   alternatives. It was suggested as a requirement initially, but
   subsequent discussion led to its removal.

If necessary, it should be possible to define such capabilities in
the form of extensions.

3. Review of Existing Work

In this section, we provide an overview of existing relevant work
that has either been completed or is work in progress. For each
item, we outline how/if it can be used to address the requirements
provided and the pros and cons of doing so.

3.1. Grouping of Media Lines

Grouping of Media Lines is defined in [RFC3388]. RFC 3388 defines a
framework that enables two or media lines to be grouped together for
different purposes. Each media line is assigned an identifier and one
or more group attributes then references two or more of those
identifiers. Associated with each group attribute is a semantics
indication. One semantic indication is the Alternative Network
Address Types ("ANAT") [RFC4091] which allows for IPv4 and IPv6
addresses to be specified as alternatives. The requirements presented
above go beyond that, however a new semantic to simply indicate
alternative media lines and associated negotiation rules could easily
be defined.

The main advantages of such an approach would be:

o Mechanism Reuse: Several semantics have already been defined
   which increases the likelihood of an implementation supporting the
   framework.

The disadvantages of such an approach would be:

o Backwards Compatibility: The mechanism is not transparently
   backwards compatible. If an entity that does not support the
   mechanism receives it, the entity may incorrectly interpret the
   SDP as consisting of multiple media streams. While RFC 3388
   defines procedures for recognizing and recover from this when
   using offer/answer, it can still lead to unintended behavior with
   endpoints that do not support the mechanism.

   In practice, it is not clear how much of an issue this is, at
   least for intelligent SIP endpoints. Most current
   implementations generally accept only one media stream of a
   given type (e.g. audio). Use of alternatives with different
   media stream types (e.g. a fax call using "audio" for voice-
   band data or "image" for T.38) makes it less clear though.
   Also, Media Gateway Controllers and Media Gateways that do not
   support grouping of media lines have been known to encounter
   problems.

o Semantics Combination Issues: Multiple semantics may be provided
   by use of grouping, however they may interact with each other
   unintentionally. For example, the "FID" semantics defined in RFC
   3388 forbids grouping of media lines with the same transport
   address, however that would be needed for alternative
   capabilities. Thus, using "FID" and alternative capabilities
   together would require special consideration.

o Some Combinatoric Explosion: The mechanism is not ideal to
   indicate alternative capabilities for multiple parameters or media
   formats within a particular media stream. For example, alternative
   attribute values and media format parameters for several codecs
   would lead to combinatoric explosion.

   [Editor's note: In practice, it is not clear this is a huge issue
   though.]

o Message Size: Each alternative requires full duplication of all
   the relevant media stream parameters.

   [Editor's note: In practice, it is not clear this is a huge issue
   though.]

3.2. **Session Description Protocol (SDP) Simple Capability Declaration**

SDP Simple Capability Declaration (simcap) is defined in [RFC3407]. It defines a set of SDP attributes that enables capabilities to be described at a session level or on a per media stream basis. RFC 3407 defines capability declaration only - actual negotiation procedures taking advantage of such capabilities have not been defined. Such rules however could easily be defined - the negotiation part would extend the offer/answer model to examine alternative configurations (capabilities). In conjunction with that, attributes to indicate the alternative configurations accepted would likely be needed as well.

The main advantages of this approach are:

o Satisfies most of the requirements provided above. In particular, by relying solely on SDP attributes, transparent backwards compatibility is always ensured.

The disadvantages of this approach are:

o Offered Capabilities Hidden in Attributes: An offer may be accepted by the answerer and a media stream established based on SDP parameters contained in SDP attributes not known to intermediaries. Such intermediaries may be back-to-back user agents, or proxies that need to inspect the SDP, e.g., to authorize Quality of Service, add transcoders, etc.

o Maximum of 255 alternative media formats per SDP: RFC 3407 currently allows a maximum of 255 alternative media formats (codecs) per SDP. This may be too restrictive.

3.3. **Session Description and Capability Negotiation (SDPng)**

The Session Description and Capability Negotiation protocol [SDPng] was intended as a replacement for SDP [SDP]. SDPng includes a full capability indication and negotiation framework that would address the shortcomings of SDP and satisfy the requirements provided above. However, SDPng has not gained traction, in large part due to existing widespread adoption of SDP. As a consequence, SDPng has remained as work in progress with limited progress for an extended period of time.

SDPng consists of two things: an SDPng description, which is an XML document that describes the actual and/or potential configurations as well as an optional negotiation process (an offer/answer compliant

process is included as part of SDPng). The SDPng description consists
of up to five parts:

o Capabilities: An optional list of capabilities (potential
   configurations) to be matched with the other parties'
   capabilities.

o Definitions: An optional set of definitions of commonly used
   parameters for later referencing.

o Configurations: A mandatory description of the conference
   components, each of which can provide a list of alternative
   configurations.

o Constraints: An optional set of constraints of combinations
   of configurations. Constraints are not defined as part of the
   base SDPng specification.

o Session Information: Optional meta information on the
   conferences and individual components.

SDPng is application-agnostic with the base specification defining a
basic XML schema supporting the above. In order to actually use
SDPng, application-specific packages are needed. Packages define
things such as media types, codecs and their configuration
parameters, etc. The base SDPng specification includes a couple of
example packages to support audio, video, and RTP.

One approach to extending SDP with capability indication and
negotiation capabilities could be to adopt the mechanisms defined by
SDPng that are necessary to satisfy the requirements provided above.
Those areas could then be included within SDP itself, e.g. in the
form of one or more SDP attributes ("a=") containing the actual SDPng
description. The areas to consider here include:

o Capabilities: This would be needed to describe alternative media
   formats and media format parameters.

o Configurations: This would be needed to define alternative
   configurations

The constraints and session information parts of SDPng would not be
used.

The main advantages of such an approach would be:

o SDPng was designed and intended to solve the general capability
   negotiation problems faced by SDP. A considerable amount of work
   has already gone into it and it was originally targeted as the
   long-term direction (replacement for SDP).

The disadvantages of such an approach would be:

o Duplicate Encoding and Specification Work: SDPng uses a
   different coding format than SDP and hence all SDP parameters
   (incl. codecs and transports) that need to be provided will need
   to have an equivalent SDPng definition. There is currently no
   automatic process for translating all SDP parameters or values
   into corresponding SDPng parameters or values; many existing SDP
   parameters and values currently have no corresponding SDPng
   definition.

o SDPng is Work in Progress: SDPng is currently work in progress but
   has seen limited interest and progress for a while. Adoption of a
   subset of its current definition may end up differing from the
   final specification. Also, the current SDPng specification needs
   further clarification and semantic tightening in a number of areas
   that would be of relevance to this approach.

o Negotiation of Transport Parameters: SDPng currently does not
   support negotiation of transport parameters as individual
   capabilities. It is however still possible to negotiate different
   transport parameters by providing alternative configurations.

o Verbose Encoding and Large Message Size: SDPng descriptions are
   XML documents, which are fairly verbose and result in descriptions
   that are substantially larger than existing SDP.

3.4. Multipart/alternative

In [I-D.jenning-sipping-multipart], the use of multipart/alternative
MIME is proposed as a way to support multiple alternative offers.
Each alternative offer has an id associated with it by use of a new
MIME header field called Content-Answering-CID. The answerer chooses
one of the offers and performs normal offer/answer operation on that
offer, and then sends back a single answer which includes the
Content-Answering-CID value of the offer chosen.

The main advantages of this approach are:

o It allows for use of alternative encodings of the offer, e.g. SDP
   and SDPng, as well as varying levels of confidentiality and
   integrity by use of S/MIME [RFC3851].

Use of multipart/alternative to solve the SDP capability negotiation
problems however has several shortcomings:

o Backwards Compatibility: Neither SIP nor RTSP mandate support
   for Multipart MIME. In the case of SIP, multipart/alternative is
   generally incompatible with existing SIP proxies, firewalls,
   Session Border Controllers, and endpoints.

o Heterogeneous Error Response Forking Problem (HERFP): When a SIP
   proxy forks a request to multiple Contacts, each of which generate
   a response, the proxy only forwards the "best" of these responses
   to the request originator. If one or more of the Contacts do not
   support multipart/alternative, the request originator may never
   discover this. Instead, only a Contact that supports
   multipart/alternative will be able to generate an answer that
   reaches the request originator.

o Combinatoric Explosions: Use of multipart/alternative to convey
   alternatives on a per media stream basis or even per media format
   parameter basis quickly leads to combinatoric explosions.

o Message Size: Each alternative requires full duplication of all
   the relevant SDP parameters (one complete SDP per alternative).

It should be noted, that use of multipart/alternative has been
discussed several times before and, in large part due to the problems
mentioned above as well as the semantics defined for
multipart/alternative [RFC2046], has met with opposition when it
comes to addressing the above types of requirements.

3.5. Sharing Ports Between "m=" Lines

SDP [SDP] does not state whether two "m=" lines can share the same
transport address or not but rather leaves this explicitly undefined.
It has been suggested that alternative capabilities for a media
stream could be indicated by including multiple media stream
descriptions sharing the same transport address (i.e. using the same
port number in the "m=" line and sharing the same IP-address).

   Such practice was not defined in [RFC2327], however it was
   suggested in an Internet-Draft version of [SDP]. Following
   discussion of the potential problems it introduced, it was removed.

The main advantages of this approach would be:

   o May not require any additional extensions to SDP - only additional
      semantics.

      [Editor's note: It is somewhat unclear how it would work without
      extensions if we allow for alternative attributes and media format
      parameters and the offerer needs to always know which ones were
      accepted]

   The disadvantages of this approach would be:

   o Backwards Compatibility Issues: Since sharing of transport
      addresses between multiple streams was never specified as part of
      SDP, backwards compatibility is likely to be an issue. Some
      implementations may support it whereas others may not. The lack of
      an explicit signaling indication to indicate the desired operation
      may lead to ungraceful failure scenarios. Offer/answer semantics
      would be unclear here as well.

   o Some Combinatoric Explosion: The mechanism is not ideal to
      indicate alternative capabilities for multiple parameters or media
      formats within a particular media stream. For example, alternative
      attribute values and media format parameters for several codecs
      would lead to combinatoric explosion.

   o Message Size: Each alternative requires full duplication of all
      the relevant media stream parameters.

      [Editor's note: In practice, it is not clear this is a huge issue
      though.]

3.6. Opportunistic Encryption Using a Session Attribute

   This approach was suggested to address the specific scenario of
   negotiating either RTP or SRTP. The endpoints signal their desire to
   do SRTP by listing RTP (RTP/AVP) as the transport protocol in the
   "m=" line in the offer together with an attribute ("a=") that
   indicates whether SRTP is supported or not. If the answerer supports
   SRTP and wants to use it, the answer then includes SRTP (RTP/SAVP) as
   the transport protocol in the "m=" line.

   The main advantages of this approach are:

   o Compatible with non-SRTP-aware endpoints.

   The disadvantages of this approach are:

   o Does not allow the offerer to indicate alternatives other than
      SRTP (including vanilla RTP as an alternative to SRTP).

   o Addresses only a small subset of the requirements provided above.

3.7. Best-Effort Secure Real-Time Transport Protocol

   This approach is documented in [BESRTP]. The approach is similar to
   the one described above, except it does not actually include any
   explicit signaling indication as to the transport protocols
   supported. Instead, support for the Secure RTP profile [SRTP] is
   inferred based on the presence of the crypto attribute defined in
   [SDES] and/or the key-mgmt attribute defined in [KMGMT].

   The main advantages of this approach are:

   o Compatible with non-SRTP-aware endpoints.

   The disadvantages of this approach are:

   o Defines new semantics above and beyond those defined by RFC 3264,
      RFC 4567, and RFC 4568 without any explicit signaling in the offer
      to that effect. This in turn may lead to unintended side-effects.

         Without explicit signaling indication, it is questionable to
         infer that presence of e.g. a crypto parameter in the offer
         indeed indicates that the offer wants to use the mechanism
         defined by the proposal. Furthermore, Section 5.1.2 of [SDES]
         defines generic operation where presence of a crypto attribute
         without e.g. SRTP as the offered transport protocol could
         result in the media stream being rejected.

   o Does not allow the offerer to indicate alternatives other than the
      inferred SRTP (including vanilla RTP as an alternative to SRTP).

   o Addresses only a small subset of the requirements provided above.

3.8. Opportunistic Encryption using Probing

   This is another approach suggested to address the specific scenario
   of negotiating either RTP or SRTP. In this case, the endpoints first
   establish an RTP session using RTP (RTP/AVP). The endpoints send
   probe messages, over the media path, to determine if the remote
   endpoint supports their keying technique.

   The main advantages of this approach are:

o Compatible with non-SRTP-aware endpoints.

The disadvantages of this approach are:

o Addresses only a small subset of the requirements provided above.

[4](#). **Security Considerations**

One of the motivations for SDP capability negotiation is to enable
best-effort SRTP negotiation, i.e. an offer/answer exchange offering
both a secure and a non-secure version of RTP. The answerer in turn
will select one of these. Such a negotiation where the offerer is
willing to accept either a secure or insecure RTP profile, and
possible with more or less strong security algorithms as a result of
the negotiation opens up for a range of possible security attacks. It
is important that any solution for SDP capability negotiation
properly addresses such security risks and/or notes any security
threats inherent in the proposed solution.

   [Editor's note: This almost sounds like we should have some
   specific requirements around all of this].

[5](#). **IANA Considerations**

There are no IANA considerations in this document.

[6](#). **Acknowledgments**

Thanks to the MMUSIC WG for comments on the requirements in this
document. Also, Francois Audet, Paul Jones and Dan Wing provided
specific comments on a precursor to this document.

[7](#). **Change Log**

Version -00 is the initial version. The requirements provided in this
initial version were taken from an earlier version of [SDPCapNeg]
with additional requirements added (from REQ-150 and up).

The ability to indicate capabilities as either mandatory or optional
is no longer explicitly out of scope (in order to support modularity
and extensibility per the newly added requirements), and neither is
the ability to indicate constraints on combinations of
configurations.

8. References

8.1. Normative References

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

   [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for
             Syntax Specifications: ABNF", RFC 2234, Internet Mail
             Consortium and Demon Internet Ltd., November 1997.

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

   [RFC3407] F. Andreasen, "Session Description Protocol (SDP) Simple
             Capability Declaration", RFC 3407, October 2002.

   [RFC3605] C. Huitema, "Real Time Control Protocol (RTCP) attribute in
             Session Description Protocol (SDP)", RFC 3605, October
             2003.

   [RFC4234] Crocker, D., and P. Overell, "Augmented BNF for Syntax
             Specifications: ABNF", RFC 4234, October 2005.

   [SDP]   Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
             Description Protocol", RFC 4566, July 2006.

8.2. Informative References

   [RFC2046] Freed, N., and N. Borenstein, "Multipurpose Internet Mail
             Extensions (MIME) Part Two: Media Types", RFC 2046,
             November 1996.

   [RFC2327] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
             Description Protocol", RFC 2327, April 1998.

   [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.

   [RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H.
             Schulzrinne, "Grouping of Media Lines in the Session
             Description Protocol (SDP)", RFC 3388, December 2002.

    [RFC3551] Schulzrinne, H., and S. Casner, "RTP Profile for Audio and
              Video Conferences with Minimal Control", RFC 3551, July
              2003.

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

    [RFC3851] B. Ramsdell, "Secure/Multipurpose Internet Mail Extensions
              (S/MIME) Version 3.1 Message Specification", RFC 3851, July
              2004.

    [RFC4091] Camarillo, G., and J. Rosenberg, The Alternative Network
              Address Types (ANAT) Semantics for the Session Description
              Protocol (SDP) Grouping Framework, RFC 4091, June 2005.

    [AVPF]   Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey,
              "Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF)",
              Work in Progress, August 2004.

    [I-D.jennings-sipping-multipart] Wing, D., and C. Jennings, "Session
              Initiation Protocol (SIP) Offer/Answer with Multipart
              Alternative", Work in Progress, March 2006.

    [SAVPF]   Ott, J., and E Carrara, "Extended Secure RTP Profile for
              RTCP-based Feedback (RTP/SAVPF)", Work in Progress,
              December 2005.

    [SDES]   Andreasen, F., Baugher, M., and D. Wing, "Session
              Description Protocol Security Descriptions for Media
              Streams", RFC 4568, July 2006.

    [SDPng]   Kutscher, D., Ott, J., and C. Bormann, "Session Description
              and Capability Negotiation", Work in Progress, February
              2005.

    [BESRTP] Kaplan, H., and F. Audet, "Session Description Protocol
              (SDP) Offer/Answer Negotiation for Best-Effort Secure Real-
              Time Transport Protocol, Work in progress, August 2006.

    [KMGMT]   Arkko, J., Lindholm, F., Naslund, M., Norrman, K., and E.
              Carrara, "Key Management Extensions for Session Description
              Protocol (SDP) and Real Time Streaming Protocol (RTSP)",
              RFC 4567, July 2006.

    [SDPCapNeg] Andreasen, F. "SDP Capability Negotiation", work in
              progress, December 2006.

   [MIKEY]   J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K.
              Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830,
              August 2004.

   [T38]   ITU-T Recommendation T.38, "Procedures for real-time Group
              3 facsimile communication over IP networks", September
              2005.

   [H245]   ITU-T Recommendation H.245, "Control protocol for
              multimedia communication", May 2006.

Author's Addresses

   Flemming Andreasen
   Cisco Systems
   Edison, NJ

   Email: fandreas@cisco.com