MMUSIC Working Group
Internet Draft
Expires: September 2007

F. Andreasen
Cisco Systems
March 4, 2007

SDP Capability Negotiation: Requirements and Review of Existing Work

draft-ietf-mmusic-sdp-capability-negotiation-reqts-01.txt

Status of this Memo

Copyright Notice

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, for example, 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 conversational 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:

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

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

- 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 [RFC3264]. 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. The answerer may also include additional media formats in the answer, however those media formats can only be used in the answerers receive direction. Such additional media formats would be actual configurations as well.

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 area 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 endpoints; an entity that does not support the mechanism should behave as if the mechanism was not present in the signaling the first place.

The above is referred to as "best-case" backwards compatibility. Another form of backwards compatibility could result in an error message indicating the extension is not supported, however this has undesirable side-effects for SIP (heterogeneous error response forking problem - HERFP) and leads to additional signaling. Another form of backwards compatibility would have an entity that does not support the extension behaving differently than it normally would (but not failing), possibly followed by additional signaling to remedy this.

REQ-85: The mechanism SHOULD attempt to be as backwards compatible and friendly as reasonably possible for SIP middle-boxes. A SIP middle-box is here defined as a SIP entity between two SIP endpoints in a session, that may or may not have the media associated with the SIP session traverse through it. In either case, the middle box may have a need to understand the details of the media streams for the session.

For example, a SIP middle-box may inspect the SDP exchanged to determine what Quality of Service to authorize for the associated media streams. While such middle boxes are not part of the SIP architecture or the Internet per se, they do exist in many real life scenarios, and we have a desire to provide a solution for those. A SIP middle-box such as the above, may be affected by a solution to the requirements provided above. For example, if the SDP exchanged appears to negotiate use of plain RTP, but in

reality, extensions are being used that result in Secure RTP with an added authentication tag being used, then the middle-box above may authorize less bandwidth than necessary. Similarly, if it tries to process media, unexpected results may occur. This is why such middle-boxes in theory should not pass SDP through them, if said SDP contains one or more attributes they do not understand or support. However, in practice, many SIP middle-boxes nevertheless do in an attempt to be as minimally invasive as possible. The alternative would be for a new extensions to either never make it through such middle-boxes or have SIP methods using them be rejected. The requirement provided here is to try and aid such middle-boxes under the assumption that they will be upgraded to support the extensions that will be defined, and hence the aid provided need not be optimal.

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-121: The mechanism SHOULD enable inclusion of potential configurations for each offered media stream in the answer. The offerer may wish to use such potential configurations as input to generating additional offers subsequently.

REQ-122: The mechanism SHOULD enable inclusion of additional media streams beyond those offered in the offer or accepted in the answer as latent potential configurations, provided this does not unduly complicate the mechanism. Latent potential configurations cannot be promoted to actual configurations in the current offer/answer exchange.

For example, an offerer that offers only audio, but is also able to support video, may want to indicate this. Similarly, an answerer that receives an offer for audio only, but is able to do video may want to indicate this. Such capabilities are purely information in the current offer/answer exchange - use of them cannot be negotiated until a new offer/exchange is performed.

REQ-130: The mechanism MUST work as well as existing offer/answer procedures in the presence of SIP forking. The mechanism MUST NOT introduce any new failure scenarios for 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 SHOULD 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 SHOULD 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-301: The mechanism MUST provide a way for both the offerer and the answerer to indicate the current version(s) of the mechanism supported as well as the enhancements supported.

REQ-302: The mechanism MUST provide a way for the offerer to indicate the version and enhancements that are used by the offerer and must be supported by the answerer (and vice versa), in order to correctly process the SDP capability negotiation in the offer (or answer).

Note that this requirement applies to the SDP capability negotiation part of the SDP only - other SDP extensions are unaffected by this.

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

- REQ-10 (alternative media formats)
- REQ-30 (alternative fmtp parameters)
- REQ-50 (alternative combinations of transport protocol and media type)
- REQ-122 (additional media streams as latent configurations)
- REQ-150 (valid combinations of media lines)
- REQ-160 (valid combinations of media formats between media streams)

[Editor's Note: Need to consider layered codecs and how they may impact the requirements]

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:

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

- 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 more 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:

- 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:

- 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.

- 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.

- 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.]

- 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:

- 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:

- 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.

- 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:

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

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

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

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

- 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:

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

- 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:

- 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:

- 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.

- 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.

- 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.

- 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.jennings-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:

- 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:

- 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.

- 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.

- 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.

- 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:

- 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:

- 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.

- 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.

- 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:

- Compatible with non-SRTP-aware endpoints.

The disadvantages of this approach are:

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

- 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 draft also proposes the use of separate payload types for codecs being used under different profiles as a way to enable the offerer to process early media packets (especially non-secure ones) prior to receiving an answer (which includes the selected profiles and, in some cases, information about SRTP keying material).

The main advantages of this approach are:

- Compatible with non-SRTP-aware endpoints.

- Provides a (separable) solution to disambiguating secure from non-secure RTP packets before receiving an answer.

The disadvantages of this approach are:

- 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.

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

- 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 profile (e.g. RTP/SAVP) and keying technique.

The main advantages of this approach are:

- Compatible with non-SRTP-aware endpoints.

The disadvantages of this approach are:

- 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 possibly 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.

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

7.1. draft-ietf-mmusic-sdp-capability-negotiation-reqts-01.txt

- Moved REQ-10, REQ-30, and REQ-122 from core to enhancements

- Elaborated on note around middle-boxes passing unknown attributes through them, and the potential implication of doing so with respect to this framework.

7.2. draft-ietf-mmusic-sdp-capability-negotiation-reqts-00.txt

Version -00 is based on draft-andreasen-mmusic-sdp-capability-negotiation-reqts-00.txt with the following changes:

- Noted that REQ-50 may have interactions with ICE
- Clarified requirements REQ-80, REQ-130.
- Added new requirements REQ-85, REQ-121, REQ-122, REQ-301, and REQ-302.
- Reduced requirements REQ-150 and REQ-160 to "SHOULD" strength.
- Minor updates to Section 3.7. and 3.8.

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

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.

[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

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.