MMUSIC Working Group
Internet Draft
Expires: April 2007

F. Andreasen
Cisco Systems

October 20, 2006

SDP Capability Negotiation
draft-andreasen-mmusic-sdp-capability-negotiation-01.txt

Status of this Memo

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 address that by identifying a set of requirements, evaluate existing work in this area, and provide a recommended solution for extending SDP with 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 was defined by the offer/answer model in RFC 3264. It defines how two entities, an offerer and an answerer, exchange SDPs 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 to define a mechanism that enables SDP to provide limited support for indicating potential configurations (capabilities) and negotiate the use of those potential configurations as actual configurations. It is not the intent to provide a full-fledged capability indication and negotiation mechanism along the lines of SDPng or ITU-T H.245. Instead, the focus is on addressing a set of well-known real-life limitations.

As mentioned above, SDP is used by several protocols, and hence the mechanism should be usable by all of these. One particularly important protocol for this problem however is the Session Initiation Protocol (SIP) [RFC3261]. SIP uses the offer/answer model (which is not specific to SIP) to negotiate sessions and hence any mechanism must at least consider how it either interacts with offer/answer, or how it should extend it.

The rest of the document is structured as follows. We first provide a set of requirements for the solution in Section 2. In Section 3. we review relevant existing work in this area, how a solution based on that might look, and the pros and cons associated with each. In Section 4. we present our proposed solution in more detail followed examples in Section 5. and security considerations in Section 6.

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 defines new attributes that may need to be conveyed as part of a capability.

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

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 Indication of mandatory and optional capabilities.

- o Constraints on combinations of configurations, e.g. inability to use a video codec together with a particular audio codec, parameter X values that can only be used with parameter Y values, etc.

- 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 let to its removal.

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 all 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:

- 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:

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. Proposed Solution

Based on the requirements provided in Section 2. and the alternatives examined in Section 3. the solution based on the Session Description Protocol (SDP) Simple Capability Declaration (simcap) [RFC3407] with extensions as outlined in Section 3.2. is preferred. In this section we present that solution in detail.

4.1. Solution Overview

The solution consists of the following:

o The capability declaration mechanism defined by simcap [RFC3407] with a few extensions.

o A new attribute ("a=capar") similar to the "a=cpar" attribute defined by simcap, except with a handle that enables referencing individual attribute capabilities (and for attributes only).

o A new attribute ("a=ctrpr") that defines how to list transport protocols as capabilities.

o A new attribute ("a=pcfg") that lists the potential configurations supported by the entity that generated the SDP. This is done by reference to the extended simcap capabilities from the SDP in question, and optionally one or more of the transport protocol capabilities. The potential configurations are listed in order of preference.

o A new attribute ("a=acfg") to be used in an answer SDP. The attribute identifies which of the potential configurations from an offer SDP were used as actual configurations to form the answer SDP. This is done by listing the potential configurations that were used from the offer SDP.

o Extensions to the offer/answer model that allow for potential configurations to be included in an offer, where they constitute offers that may be accepted by the answerer instead of the actual configuration(s) included in the "m=" line(s).

The mechanism is illustrated by the offer/answer exchange below, where Alice sends an offer to Bob:

```
             Alice                               Bob

               | (1) Offer (SRTP and RTP)         |
               |--------------------------------->|
               |                                  |
               | (2) Answer (RTP)                 |
               |<---------------------------------|
               |                                  |
```

Alice's offer includes RTP and SRTP as alternatives. RTP is the default, but SRTP is the preferred one:

```
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0 18
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 audio RTP/SAVP 0 18
a=capar: 1 a=crypto:1 AES_CM_128_HMAC_SHA1_32
   inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
a=pcfg: c=3,4 a=1
a=pcfg: c=1,2
```

The "m=" line indicates that Alice is offering to use plain RTP with PCMU or G.729. The extended simcap capability declaration is provided by the "a=sqn" and "a=cdsc" attributes as defined in [RFC3407], and the new "a=capar" attribute defined in this document. The capabilities indicate that PCMU and G.729 are supported with either RTP or secure RTP. The "capar" attribute provides a capability parameter with a handle of 1. The capability parameter is a "crypto" attribute in the capability set, which provides the keying material for SRTP using SDP security descriptions [SDES]. The new "a=pcfg" attribute provides the potential configurations included in the offer by reference to the simcap capability declarations. Two alternatives are provided; the first one, and hence the preferred one is using capabilities 3 and 4, i.e. PCMU and G.729 under the RTP/SAVP profile (secure RTP) together with the attribute capability parameter 1, i.e. the crypto attribute provided. The second one is using capabilities 1 and 2, i.e. PCMU and G.729 under the RTP/AVP profile.

Bob receives the SDP offer from Alice. Bob supports RTP, but not SRTP, and hence he accepts the potential configuration for RTP provided by Alice:

```
v=0
o=- 24351 621814 IN IP4 128.96.41.2
s=
c=IN IP4 128.96.41.2
t=0 0
m=audio 4567 RTP/AVP 0 18
a=acfg: c=1,2
```

Bob includes the new "a=acfg" attribute in the answer to inform Alice that he based his answer on an offer containing the potential configuration with capabilities 1 and 2 from the offer SDP (i.e. PCMU and G.729 under the RTP/AVP profile). Note that in this particular example, the answerer supported the capability extensions defined here, however had he not, everything would still have worked fine since the actual configuration is what was being used. Consequently, the answer would simply have omitted the "a=acfg" attribute line.

4.2. Extensions to Simcap

Simcap [RFC3407] defines capability descriptions to be on the form:

```
a=cdsc: <cap-num> <media> <transport> <fmt list>
```

where <cap-num> is an integer between 1 and 255 (both included) used to number the capabilities, and <media>, <transport>, and <fmt list> are defined as in the SDP "m=" line. We extend that definition here

to allow for wild-carding of the <media>, <transport> and <fmt list> parameters at the session level only. The wild-card character to use is asterisk ("*"). This enables us to provide session level capability parameters that are not specific to any particular media stream, or applies only to certain types of media streams. Such capability parameters apply to all media streams that match the combined <media>, <transport> and <fmt list> provided.

An example use case is to allow for negotiation of MIKEY at the session level outside of a specific simcap capability description (and hence media type) by use of the key management framework [KMGMT].

This is illustrated by the following examples:

```
a=cdsc: 1 * * *

a=cdsc: 2 audio * *
```

In the first example, the capability description applies to all media stream. In the second example, the capability description applies to media streams of type audio only.

Simcap capability descriptions start with a sequence number ("a=sqn") and, as specified in [RFC3407], require that a capability description as defined by simcap, i.e. an "a=cdsc" line, follows immediately after the sequence number. We remove that requirement here. As a result of that, we enable the new "a=capar" attribute (and other parameters) to follow after the sequence number. There is however not a requirement that it follows immediately after the sequence number.

4.3. Attribute Definitions

4.3.1. The Attribute Parameter Capability Attribute

Attributes can be expressed as negotiable parameters by use of a new attribute parameter capability attribute ("a=capar") similar to the "a=cpar" attribute defined by simcap, except with a handle that enables referencing it and supporting attributes only (the "cpar" attribute defined in RFC 3407 supports bandwidth information as well). The attribute is defined as follows:

```
a=capar: <att-cap-num> <att-par>
```

where <att-cap-num> is an integer between 1 and 255 (both included) used to number the attribute parameter capability and <att-par> is an attribute ("a=") in its full '<type>=<value>' form (see [SDP])

The "capar" attribute can be provided at the session level or the media level. Each occurrence of the attribute MUST use a different value of <app-cap-num>, with the first one being 1, the second one being 2, etc. The <att-cap-num> values provided are independent of similar <cap-num> values provided for other attributes, i.e., they form a separate name-space for attribute parameter capabilities.

TO DO: There is a need to clarify the relationship between this one, the simcap cpar values, and regular attributes (actual configuration attributes). The basic idea is that attributes that can only be used with certain potential configurations should be provided here and then included by reference in those potential configurations.

The following example illustrates use of the "capar" attribute:

```
a=capar: 1 a=ptime:20

a=capar: 2 a=ptime:30

a=capar: 3 a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAAAAsAyONQ6gAA
AAAGEEoo2pee4hp2UaDX8ZE22YwKAAAPZG9uYWxkQGR1Y2suY29tAQAAAAAAAQAk0
JKpgaVkDaawi9whVBtBt0KZ14ymNuu62+Nv3ozPLygwK/GbAV9iemnGUIZ19fWQUO
SrzKTAv9zV

a=capar: 4 a=crypto:1 AES_CM_128_HMAC_SHA1_32
      inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
```

The first two ones provide attribute values for the ptime attribute. The third one provides SRTP parameters by using MIKEY with the key-mgmt attribute [KMGMT]. The fourth one provides SRTP parameters by use of security descriptions with the crypto attribute [SDES].

4.3.2. The Transport Protocol Capability Attribute

Transport Protocols can be expressed as capabilities by use of a new Transport Protocol Capability attribute ("a=ctrpr") defined as follows:

```
a=ctrpr: <trpr-cap-num> <proto>
```

where <trpr-cap-num> is an integer between 1 and 255 (both included) used to number the transport address capability for later reference, and <proto> is defined as in the SDP "m=" line.

The "ctrpr" attribute can be provided at either the session or media-level. Each occurrence of the attribute MUST use a different value of <trpr-cap-num>, with the first one being 1, the second one being 2, etc. The <trpr-cap-num> values provided are independent of similar <cap-num> values provided for other attributes, i.e., they form a separate name-space for transport protocol capabilities.

Below, we provide examples of the "a=ctrpr" attribute:

```
a=ctrpr: 1 RTP/AVP
a=ctrpr: 2 RTP/AVPF
```

The first one provides a capability for the "RTP/AVP" profile defined in [RFC3551] and the second one provides a capability for the RTP with RTCP-Based Feedback profile defined in [AVPF].

Note that the simcap extensions already provide similar functionality by inclusion in the "cdsc" attribute (as illustrated by the example in Section 4.1.), however having this as a separate capability indication can provide significant message size reduction when negotiating alternative profiles (of which there can be many). In particular, there is no need to repeat supported payload types. Also, use of this attribute combined with the potential configuration attribute (see Section 4.3.3.) provides for more expressive power.

4.3.3. The Potential Configuration Attribute

Potential Configurations can be expressed by use of a new Potential Configuration Attribute ("a=pcfg") defined as follows:

```
a=pcfg:     <simcap-capabilities>
            [<attribute-parameter-capabilities>]
            [<transport-protocol-capabilities>]
```

The potential configuration attribute includes one or more sets of simcap-capabilities. A list of attribute parameter capabilities and a list of transport protocol capabilities can optionally be included as well. Together, these values define a set of potential configurations. There can be one or more potential configuration attributes provided at the session level as well as for each media stream. The attributes are provided in order of preference.

TO DO: Clean up capability and configuration terminology.

<simcap-capabilities> is defined by the following ABNF:

```
simcap-capabilities = "c=" cap-list *("|" cap-list)
cap-list            = cap-num *("," cap-num)
cap-num             = 1*3DIGIT   ; defined in [RFC4234]
```

Each capability list is a comma-separated list of simcap capability numbers where cap-num refers to simcap capability numbers and hence MUST be between 1 and 255 (both included). Alternative potential simcap configurations are separated by a vertical bar ("|"). The alternatives are ordered by preference.

<attribute-parameter-capabilities> is defined by the following ABNF:

```
attribute-parameter-capabilities
                  = "a=" capar-cap-list *("|" capar-cap-list)
capar-cap-list    = att-cap-num *("," att-cap-num)
att-cap-num       = 1*3DIGIT  ;defined in [RFC4234]
```

Each attribute parameter capability list is a comma-separated list of attribute capability parameter numbers where att-cap-num refers to attribute parameter capability numbers defined above and hence MUST be between 1 and 255 (both included). Alternative attribute parameter capabilities are separated by a vertical bar ("|"). The alternatives are ordered by preference.

<transport-protocol-capabilities> is defined by the following ABNF:

```
transport-protocol-capabilities =
                     "p=" trpr-cap-num *("|" trpr-cap-num)
trpr-cap-num        = 1*3DIGIT   ; defined in [RFC4234]
```

The trpr-cap-num refers to transport protocol capability numbers defined above and hence MUST be between 1 and 255 (both included). Alternative transport protocol capabilities are separated by a vertical bar ("|"). When transport protocol capabilities are not included, the transport protocol information from the media description ("m=" line) will be used.

The potential configuration ("a=pcfg") attribute can be provided at the session level and the media-level. Each occurrence of the attribute within a given media description ("m=" line) defines a set of potential configurations that can be used for that media description.

TO DO: Need to decide on relationship between session-level and media-level (how should conflicts, overlap, etc. be handled - simplicity at the possible expense of expressive power is preferable in the editor's opinion).

Below, we provide an example of the "a=pcfg" attribute in a complete media description in order to properly indicate the supporting attributes:

```
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/SAVPF 0 18
a=crypto:1 AES_CM_128_HMAC_SHA1_32
   inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
a=sqn: 0
a=cdsc: 1 audio RTP/SAVP 0 4 18
a=ctrpr: 1 RTP/AVP
a=ctrpr: 2 RTP/AVPF
a=ctrpr: 3 RTP/SAVP
a=ctrpr: 4 RTP/SAVPF
a=pcfg: c=1|3 p=1|2|3|4
a=pcfg: c=2 p=1
```

We have two potential configurations listed here. The first one indicates that PCMU (payload type number 0 referenced by simcap capability number 1) or G.729 (payload type number 18 referenced by simcap capability number 3) can be supported with either of the profiles RTP/AVP, RTP/AVPF, RTP/SAVP, or RTP/SAVPF (specified by the transport protocol capability numbers 1,2,3 and 4). The second potential configuration indicates that G.723 (payload type number 4 referenced by simcap capability number 2) can be supported with the RTP/AVP profile only (transport protocol capability number 1).

4.3.4. The Actual Configuration Attribute

The actual configuration attribute identifies which of the potential configurations from an offer SDP were used as actual configurations in an answer SDP. This is done by reference to the simcap capabilities, and the transport protocol (if included) capabilities from the offer that were actually used by the answerer in his offer/answer procedure.

The Actual Configuration Attribute ("a=acfg") is defined as follows:

```
a=acfg:     <simcap-capability-list>
            [<attribute-parameter-capabilities>]
            [<transport-protocol-capability]
```

<simcap-capability-list> is defined by the following ABNF:

```
simcap-capability-list = "c=" cap-list
cap-list               = cap-num *("," cap-num)
cap-num                = 1*3DIGIT   ; defined in [RFC4234]
```

Each capability list is a comma-separated list of simcap capability numbers where cap-num refers to simcap capability numbers and hence MUST be between 1 and 255 (both included).

<attribute-parameter-capabilities> is defined by the following ABNF:

```
attribute-parameter-capabilities = "a=" capar-cap-list
capar-cap-list    = att-cap-num *("," att-cap-num)
att-cap-num       = 1*3DIGIT  ;defined in [RFC4234]
```

Each attribute parameter capability list is a comma-separated list of attribute capability parameter numbers where att-cap-num refers to attribute parameter capability numbers defined above and hence MUST be between 1 and 255 (both included).

<transport-protocol-capabilities> is defined by the following ABNF:

```
transport-protocol-capability = "p=" trpr-cap-num
trpr-cap-num        = 1*3DIGIT   ; defined in [RFC4234]
```

The trpr-cap-num refers to transport protocol capability numbers defined above and hence MUST be between 1 and 255 (both included). When a transport protocol capability is not included, the transport protocol information from the media description ("m=" line) in the offer is being used.

The actual configuration ("a=acfg") attribute can be provided at the session level and the media-level. There MUST NOT be more than one occurrence of an actual configuration attribute at the session level, and there MUST NOT be more than one occurrence of an actual configuration attribute within a given media description.

Below, we provide an example of the "a=acfg" attribute (building on the previous example with the potential configuration attribute):

```
v=0
o=- 24351 621814 IN IP4 128.96.41.2
s=
c=IN IP4 128.96.41.2
t=0 0
m=audio 4567 RTP/AVPF 0
a=acfg: c=1 p=2
```

It indicates that the answerer used an offer consisting of simcap capability 1 from the offer (PCMU) and transport protocol capability 2 from the offer (RTP/AVPF).

4.4. Offer/Answer Model Extensions

In this section, we define extensions to the offer/answer model defined in [RFC3264] to allow for potential configurations to be included in an offer, where they constitute offers that may be accepted by the answerer instead of the actual configuration(s) included in the "m=" line(s).

[Editor's Note: Multicast considerations have been omitted for now.]

TO DO: Elaborate and firm up offer/answer procedures.

4.4.1. Generating the Initial Offer

An offerer that wants to use capability negotiation extensions defined in this document MUST include the following in the offer:

- one or more simcap capability descriptions (as defined in [RFC3407] and extended above) for each of the capabilities.

- optionally, one or more attribute parameter capability attributes (as defined in Section 4.3.1.) if one or more alternative attribute parameter values is to be negotiated.

- optionally, one or more transport protocol capability attributes (as defined in Section 4.3.2.) if one or more alternative transport protocols is to be negotiated.

- one or more potential configuration attributes (as defined in Section 4.3.3. which define the potential configurations supported by the offerer.

Each of the potential configurations listed constitutes an alternative offer which may be used to negotiate and establish the session. The current actual configuration is included in the "m=" line (as defined by [RFC3264]).

4.4.2. Generating the Answer

When the answerer receives an offer with one or more valid potential configuration information attributes present, it may use any of the potential configurations as an alternative offer. A potential

configuration information attribute is valid if all of the capabilities (simcap, attribute capabilities and transport protocol) it references are present and valid themselves.

The actual configuration is contained in the media description's "m=" line. The answerer can send media to the offerer in accordance with the actual configuration, however if it chooses to use one of the alternative potential configurations, media sent to the offerer may be discarded by the offerer until the answer is received.

If the answerer chooses to accept one of the alternative potential configurations instead of the actual configuration, the answerer MUST generate an answer as if the offer contained that potential configuration instead of the actual configuration included. The answerer MUST also include an actual configuration attribute in the answer that identifies the potential configuration from the offer used by the answerer. The actual configuration attribute in the answer MUST include information about the capabilities. Furthermore, if the offered potential configuration included attribute capability parameters and/or transport protocol capabilities, those parameters MUST be included in the actual configuration attribute in the answer as well.

4.4.3. Offerer Processing of the Answer

When the offerer included potential configurations for a media stream, it MUST examine the answer for the presence of an actual configuration attribute for each such media stream. If the attribute is missing, offerer processing of the answer MUST proceed as defined by [RFC3264]. If the attribute is present, processing continues as follows:

The actual configuration attribute specifies which of the potential configurations was used by the answerer to generate the answer. This includes all the types of capabilities from the potential configuration offered, i.e. the media formats ("cdsc" capabilities), attribute capability parameters ("capar") and transport protocol capabilities ("ctrpr")

The offerer MUST now process the answer as if the offer had contained the potential configuration as the actual configuration in the media description ("m=" line) and relevant attributes in the offer.

4.4.4. Modifying the Session

Potential configurations may be included in subsequent offers as defined in [RFC3264, Section 8]. The procedure for doing so is

similar to that described above with the answer including an indication of the actual configuration used by the answerer.

5. Examples

TBD.

6. Security Considerations

TBD.

7. IANA Considerations

TBD.

8. To Do and Open Issues

o Capability descriptions, potential configurations and actual configurations can be provided at both the session level and media level. It needs to be decided what the relationship between the session level and media level parameters are.

o We are currently capping all capability numbers at 255. Is this a concern, not least considering the limits apply to the session, not just individual media streams.

9. Acknowledgments

Thanks to Francois Audet and Dan Wing for comments on this document.

10. Change Log

10.1. Changes since -00

o Added requirements to allow for alternative attribute values to be negotiated at the session level.

o Removed requirements to support unicast/multicast as alternatives and IPv4/IPv6 as alternatives.

o Updated section 3.6. on opportunistic encryption using a session attribute

o Added new section 3.7. on best-effort secure real-time transport protocol.

- Updated solution to align with updated requirements. More specifically

  - Added minor extensions to simcap in new Section 4.2.

  - Removed the "a=ctrad" attribute that supported transport addresses as capabilities and updated the rest of the attributes and procedures accordingly.

  - Allowed for the "a=ctrpr" to be specified at the session level as well.

  - Updated semantics for the "a=pcfg" attribute to specify that potential configurations are listed in order of preference.

  - Defined a new attribute "a=capar" that enables the offerer to determine which of several possible alternative attributes from an offer was chosen by the answerer.

  - Updated example in Section 4.1. to illustrate backwards compatibility with a non-SRTP capable endpoint.

  - Updated open issues section and in particular noted issue around session level and media level parameter semantics overlap.

11. References

11.1. 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.

11.2. Informative References

[RFC2046] Freed, N., and N. Borensteain, "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.

Author's Addresses

Flemming Andreasen
Cisco Systems
Edison, NJ

Email: fandreas@cisco.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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