INTERNET-DRAFT
Expires: January 2003

J. Ott/C. Perkins
TZI/ISI

July 2002

SDPng Transition
draft-ietf-mmusic-sdpng-trans-01.txt

Status of this memo

Abstract

The Session Description Protocol (SDP) is today widely used in the Internet to announce as well as negotiate multimedia sessions and exchange capabilities. Having originally been designed for session announcements only, as opposed to announcements and capabilities negotiation announcements, native SDP lacks numerous features to be applicable in many session scenarios. Numerous extensions have been developed to circumvent SDP's shortcomings -- but they have also repeatedly shown its inherent limitations. A successor protocol -- termed "SDPng" for the time being -- is developed to address the aforementioned needs of Internet applications in a more structured manner. With the huge installed base of SDP-based applications, a migration path needs to be developed to move from SDP to SDPng over time. This document outlines how this migration can be achieved: in general as well as for the various IETF control protocols that potentially make use of SDP and SDPng.

This document is a product of the Multiparty Multimedia Session Control (MMUSIC) working group of the Internet Engineering Task

Force. Comments are solicited and should be addressed to the working group's mailing list at mmusic@ietf.org and/or the authors.

1. Introduction

SDP is now widely used within the Internet community to describe media sessions and, in a limited fashion, system capabilities relating to (multi)media sessions, for a variety of application scenarios: session announcements, interactive session setup, capability assessment and remote control of media streams. All but the first of these are rather different from what SDP was originally designed for -- but all of them share the idea of setting up and configuring media streams. Over time, its wide range of uses has revealed numerous shortcomings -- most of which stem from the fact that SDP has been used for lack of a better alternative and its semantics have been re-interpreted to make it fit the respective scenarios' needs. In many cases, workrounds (typically called "extensions") for those shortcomings could be found which are often rather cumbersome. While this practice has extended SDP's lifetime and provided at least a suitable basis for numerous applications, in parallel, a successor protocol -- currently referred to as "SDPng" -- has been developed.

> It is worthwhile noting that the aforementioned applications' needs are sufficiently similar for a single description protocol to take care of them if it was designed for this purpose from the beginning. As a lesson learned from SDP, any further expansion in scope should be avoided where no clear fit can be seen -- and specific (different) solutions should be developed instead.

The design of SDPng takes into account the requirements arising from the above application scenarios and puts particular emphasis on protocol extensibility and modularization of extensions, at the same time keeping the core description format simple. SDPng uses a different (more expressive) syntax than SDP does and hence is not backward compatible at the syntax level. Nevertheless, the concepts of SDPng take into account the migration issues from SDP to SDPng by providing straightforward mappings between the two formats where possible and try to maximize compatibility from a semantics perspective.

The current revisions of SDP and SDPng are documented in

[1] draft-ietf-mmusic-sdp-new-10.txt and

[2] draft-ietf-mmusic-sdpng-05.txt.

For SDP, numerous additional documents need to be taken into account:

[3] RFC 3108

[4] draft-ietf-mmusic-fid-06.txt

[5] draft-andreasen-mmusic-sdp-simcap-05.txt

[6] draft-ietf-mmusic-sdp-offer-answer-02.txt

[7] draft-fairlie-mmusic-sdp-sctp-00.txt

[8] draft-ietf-mmusic-sdp-comedia-03.txt

[9] draft-taylor-mmusic-sdp-tdm-01.txt

[10] draft-ietf-mmusic-sdp4nat-02.txt

[11] draft-ietf-mmusic-kmgmt-ext-05.txt

[12] draft-ietf-mmusic-sdp-ipv6-03.txt

This document outlines a migration path from SDP to SDPng, starting from a short overview of the current application scenarios. In the next step, we highlight which design decisions taken for SDPng should simplify a smooth migration and describe how mappings between the two description formats can be performed at an abstract level. We then address procedural issues for integrating SDP and SDPng into the various protocols relying on those media description formats. Finally, we summarize work items on the agenda for SDPng.

2. Application Scenarios

The following session control protocols that make use of SDP have been standardized in the IETF so far:

1. SDP was originally developed to announce (Mbone-based) multimedia sessions via session directories using the Session Announcement Protocol (SAP) -- but other mechanisms for disseminating the session descriptions (such as HTTP, SMTP, NNTP, etc.) are conceivable as well.

   The major property of this application scenario is that the creator of the session description defines a (set of) fixed choice(s) for all media types in a conference and the conference partipants have no way to influence these. If they support at least one of the codecs for a particular media type they can participate in this media session, otherwise they cannot. There is no interaction between sender(s) and receiver(s) to negotiate the media stream codecs and parameters.

   This scenario is referred to as "announcement".

2. Another use of SDP is in conjunction with the Real-Time Streaming Protocol (RTSP). In RTSP, SDP is used to convey descriptions of a media stream interactively requested to be

played from a server (or recorded by a server). SDP itself is not used for capability negotiation, not even for the addresses to be used; those are negotiated within RTSP and may override the addresses specified as part of SDP.

This scenario is referred to as "retrieval".

3. With SIP, SDP is used to propose media stream configurations and choose out of these (i.e. enable a subset of these). By proposing and accepting media stream configurations, endpoints use SDP to implicitly describe their capabilities and carry out a negotiation procedure on the media streams to use.

   In the context of SIP, specific meanings (including required extensions) have been defined for use of SDP with unicast addresses, for connection-oriented transports, and for certain media level attributes (such as the direction attribute send-only, receive-only, and inactive).

   Numerous extensions have been proposed to extend SDP to better suit SIP's needs. Besides a description of the offer/answer model, these extensions particularly include the ability to describe simultaneous capabilites and to group media stream semantically.

   This scenario is referred to as "offer/answer".

4. SDP is used to convey the capability descriptions of a MEGACO media gateway (MG) to its media gateway controller (MGC) as well as for the MGC to instruct the MG where to send media streams to and from where to receive media streams, including codec and parameter choice.

   For this purpose, SDP has been modified/extended to some degree to fit the MEGACO needs.

   This scenario is referred to as "gateway control".

It should noted that the original SDP concept already provided an extension mechanism to cover other network types than IPv4 and IPv6; however, specific extensions have only been defined recently for ATM and are now under discussion for TDM. Extensions to other transport (including radio interfaces or next generation wireless networks) as well as to new types of session descriptions (e.g. electronic program guides) are conceivable.

3. Mapping SDP to SDPng

On a transition path from SDP to SDPng, allowing for a somewhat straightforward mapping of (parts of) one description format onto the other is of crucial importance. SDPng has been designed in way that allows many of the session description features of SDP to be easily mapped onto the SDPng format and vice versa -- except that SDPng is more expressive than SDP and hence information loss is not unlikely to occur when doing the reverse mapping. The final mapping rules between SDP and SDPng to be drawn up shall ensure that when mapping

SDP to SDPng and then back to SDP will produce an SDP that is functionally identical to the one originally fed into the mapping process. Note that the use of a number of SDP extensions (FID, SIMCAP) may be implied in this mapping process, depending on the use of SDP. The mapping rules will ensure that no information loss will occur when translating from SDP to SDPng.

The SDPng design uses a structure of four sections: definitions, potential or actual configurations, constraints and session attributes. Of these, the "Configurations" and "Session Attributes" sections map well onto the current SDP. The "Definitions" and "Constraints" sections provide additional structure which is not directly expressible in SDP.

o At the media description level, the Potential and Actual Configurations specified in the "Configurations" section maps well to media descriptions ("m=", possibly "c=", and associated attributes ("a=") lines).

o At the session description level, the SDP session parameters are largely reflected in the "Session Attributes" section of SDPng. The attributes proven suitable for session announcements have been used as the basis when defining SDPng.

In SDPng, media descriptions are explicitly tagged with identifiers and thus are easily referenced for semantically grouping media streams (e.g. to describe alternative audio in different languages, media streams to be synchronized, or media streams to carry the same information simultaneously but with different encodings) -- as has been defined for SDP in a limited fashion by the "fid" attribute set. SDPng allows even to more formally describe the syntax of individual or compound media streams in the "Session Attributes" section. Furthermore, SDPng supports a superset of additional constraints that may be realized by the "simcap" extensions for SDP in the "Constraints" section.

Additional address families such as ATM or TDM bearers, next generation wireless network bearers, DVB channels, etc. can be incorporated into SDPng by defining the appropriate extensions for the SDPng transports.

Similarly, new codecs can be added by just defining new codec specifications or defining entire new classes of applications to be described as new content types ("codec") to be carried in a media session (including e.g. text, fax, slide presentations, shared editors, etc). If necessary, sophisticated parameter structures can be supported (even though the authors believe that simplicity is key to interoperability here). This is similar to, but more structured than, the definition of new codecs attributes/MIME registrations in

SDP.

By means of its conceptual differentiation into Potential and Actual Configurations, SDPng supports both indicating a system's capabilities (without specifying transport addresses) separately from the instantiation of a particular media stream as well as conveying capability descriptions and instantiation proposals at the same time -- thereby providing a good fit for all the above session control scenarios: the "announcement" and "retrieval" scenarios will just use rather fixed Actual Configurations. The "offer/answer" model will use use Actual Configuration but use them to negotiate media strams in a two-way handshake but may in addition use Potential Configurations to indicate capabilities that shall not be used immediately. The "gateway control" scenario will use both: Potential Configurations to describe an MG's capabilities and Actual Configurations for setting up media sessions at MGs as well as retrieving information about currently active media sessions. This differentiation is not directly expressible in SDP, although various extensions can be used to overload SDP semantics to achieve at least part of this effect.

Finally, SDPng is also intended to allow for content-independent negotiation of session parameters by defining collapsing/intersection rules. In particular, SDPng tries to take the need for multicast-based distributed calculation of joint capabilities into account for those rules (but note that it is *not* intended as a generic format for describing conference state information). Such functionality is not covered by current SDP.

4. Integration with Session Control Protocols

This section outlines for each of the session control protocols described above how SDP and SDPng can be used in parallel and indicates how a suitable transition could be achieved.

4.1. Session Announcement Protocol (SAP)

There are two revision of SAP specified, version 0 which is implemented in a number of experimental tools, and version 1 which is defined in RFC 2972.

SAPv0:SAPv0 does not support a mechanism to identify the content type of a session announcement but implicitly assumes SDP. Proper parsers will note that the contents of the SAPv0 message does not begin with a "v=" line and hence will ignore the entire announcement. SDPng contents MAY be identified by the character sequence "<sdpng" in the beginning of the announcement body -- but such content is not strictly legal in SAPv0.

SAPv1:In SAPv1, an explicit payload type field (containing a MIME type) is available and SHOULD used to differentiate between SDP anf SDPng contents. two approaches are conceivable: Either multipart MIME message is used with two parts containing the same session descriptions -- one expressing it in SDP and the other in SDPng. Alternatively, two alternate session announcements may be used (being properly distinguished by the SDP "o=" field and the SDPng equivalent).

> It is RECOMMENDED that implementations recognize the MIME multipart/alternative type in SAPv1 announcements, allowing for a simple transition to SDPng.

It should be noted that current session directory implementations only support SDP. Nevertheless, using the SAP Message Identifier Hash and the source address, they should be able to perform session deletions and modifications properly -- even without understanding the format contained in the SAP message body.

For the introduction of SDPng, session announcements SHOULD be made "bi-lingual", i.e. in SDP and SDPng. If a SAP announcer for some reason knows that all its potential audience will support SDPng, the SDP announcement SHOULD be omitted.

It should be noted that, for IPv4-based multicast sessions, session directories still may rely on parsing the session specifications to avoid clashes in the multicast address space. Introducing a new session description language will prevent older implementations from continuing this practice successfully -- assuming that only SDPng announcements are used and/or that old implementations do not support MIME multipart/alternative message bodies. This use of SAP is deprecated, of course.

4.2. Real-Time Streaming Protocol (RTSP)

RTSP uses SDP to provide presentation descriptions (with a presentation comprising one or more media sessions), typically communicated from the server to the client (for playing) and in the opposite direction for recording. The presentation description may also include initialization data for the various media streams and URLs to be used for controlling the entire presentation as well as the individual media sessions. Transport parameters -- such as IP addresses, port numbers, etc. -- are conveyed as part of RTSP header fields.

RTSP uses the Content-Type: header field to indicate the format of the enclosed entity. This provides a straightforward means for distinguishing SDP and SDPng-based presentation descriptions. In addition, the Accept: header SHOULD be used by the client, to indicate which content types it supports. If the client specifies both SDP and SDPng as acceptable, the server SHOULD provide only the SDPng-based presentation description.

If the client does not indicate a particular Content-Type: the server can, theoretically, use MIME multipart bodies to convey both description types simultaneously.

> [Editors note: can implementors comment on their ability to

parse such content?]

In general, it would be preferrable to have the servers migrate to always supporting both description formats, thus enabling the clients to choose.

Finally, RTSP makes special provision to interworking with firewalls by including the crucial transport parameters in a separate RTSP header field _in_addition_ to the presentation description. This practice in principle allows to change the presentation description format without having to worry about the operation of firewalls and similar devices.

4.3. Session Initiation Protocol (SIP)

The use of SDP with SIP follows the offer/anser model and is described in [6]. It is key to the (efficiency of the) offer/answer model that a complete capability exchange and media stream instantiation be carried out in one round-trip -- which is supported by SDP. While SDPng allows to separate capability exchange from media sesssion instantiation, those two pieces are also easily integrated in a single step.

SIP also uses a Content-Type: header to indicate the nature of data carried in its message body; and SIP explicitly calls for supporting MIME multipart message bodies. While, again, the use of MIME multipart/alternative would in principle be possible (from a theoretical perspective), issues regarding the actual implementation of multipart/alternative in SIP entities have been raised. As backward compatibility has to be achieved, a different approach is suggested:

A SIP UAC MAY use an SDPng message body in a SIP INVITE (or other) message. If the SIP UAS does not support SDPng, it will return a "415 Unsupported Media Type" response to the UAC and indicate acceptable content types in the Accept: header (probably including "application/sdp"). The SIP UAC MUST then retry INVITE (or other) message using the indicated session description language. The SIP UAC SHOULD cache knowledge about which peers did not understand SDPng as session description formats for a limited amount of time (e.g. several days) so that extra round-trips for session setup are only incurred infrequently. Whenever a peer has sent an SDPng description (or it is known from other means that the peer supports SDPng), this information SHOULD also be cached.

The SIP Accept: header can be exploited to determine the capability of a peer to understand SDPng in addition (or instead) of plain SDP. Methods such as OPTIONS MAY be used to determine a peer's support for SDPng. However, a peer's capabilities may not be known when the first message is sent which may introduce an extra round-trip if including SDP and SDPng in the initial INVITE message is not an option. Further approaches to make a UA's support for SDPng known ahead of time should be explored.

A number of SDP extensions have been motivated by SIP-based applications and these need to be accommodated in SDPng as well. Features such as "simcap" and "FID" are inherently supported by

SDPng; proper definitions for connection-oriented media need to be fully understood and then incorporated. Key management attributes as defined in [11] need to be included (not just for SIP) and so may need to be general mechanisms to signal security capabilities [11] [13] and indicate their optional or mandatory use. The same applies to quality of service parameters [13] (which are largely also motivated by SIP but are also useful with control protocols).

In the context of SIP, a number of special rules to deal with certain SDP fields are set up (e.g. to work with NATs). The SDPng development needs to make sure that similar definitions are provided (as need be the handling of those in SIP).

[Editor's note: This section needs more work on details.]

4.4. Media Gateway Control Protocol (MEGACOP)

The MEGACO specification already supports two different encodings for capability and media stream descriptions: a text-based variant based upon (a modified) SDP and a binary representation of the same information set. MGCs are required to implement both encodings while MGs have the choice to pick either or both. Differentiation between the protocol encoding variants is done using different port numbers: 2944 for the text-based and 2945 for the binary encoding.

Unfortunately, within the text-based encoding, there is no means to differentiate several description formats. SDP messages are carried as an "octet string" without any type identifier. Defining a third port number for this further differentiation does not seem to be appropriate, particularly since the message encoding is still a text format.

The remaining means for distinction is that an SDP specification would start with a "v=0" line while an SDPng document would begin with an "<sdpng" part.

> [Editor's Note: MEGACOP also supports a binary encoding for SDP messages; we can assume that not all of the SDPng messages will be expressible this binary encoding. How shall we handle these?]

5. SDPng and Middleboxes

TBD.

6. Directing the Evolution of SDP

With the transition from SDP to SDPng, there is the question of the evolution of SDP, and legacy systems which use it.

The SDP specification (draft-ietf-mmusic-sdp-new-10.txt) is stable, and mostly corrects errors in the original specification, with the addition of very few new features. This is expected to be published as a draft standard RFC shortly.

A number of extensions to SDP for use in offer/answer scenarios are also close to completion. These include grouping (draft-ietf-mmusic-fid-05.txt) and capability negotiation (draft-andreasen-mmusic-sdp-simcap-04.txt). We expect these to be completed, and published as proposed standard RFCs, to bring minimal capability/alternative descriptions to SDP.

Related is the SDP offer/answer model for SDP, currently under development as draft-rosenberg-mmusic-sdp-offer-answer-01.txt). This defines the model used to complete the steps of a negotiation using SDP.

All these are subsumed into SDPng, so there should be no further need for development in these areas; applications with requirements that are not met by these specifications should use SDPng.

There have recently been proposals to add quality of service negotiation for SDP and, similarly, we expect other extensions to be proposed over time. Due to the well-known limitations of SDP, we do not believe it appropriate to continue development of more elaborate extensions: for negotiation, for QoS, for security, and for other general-purpose or application-specific needs.

Instead, such new work should be done in the framework of SDPng where applications and their requirements for (new) expressiveness in end-to-end exchanges to negotiate and configure media sessions will hopefully act as a driver for that process.

7. Author's Addresses

Joerg Ott <jo@tzi.org>
Universitaet Bremen
MZH 5180
Bibliothekstr. 1
D-28359 Bremen
Germany
tel:+49-421-201-7028
sip:jo@tzi.org

Colin Perkins <csp@isi.edu>
USC Information Sciences Institute
3811 North Fairfax Drive, Suite 200
Arlington, VA 22203
USA
tel:+1-703-812-3705

8. References

[1] Mark Handley, Van Jacobson, Colin Perkins, "SDP: Session Description Protocol," Internet Draft draft-ietf-mmusic-sdp-new-10.txt, Work in Progress, May 2002.

[2] Dirk Kutscher, Joerg Ott, Carsten Bormann, "Session Description and Capability Negotiation", Internet Draft draft-ietf-mmusic-sdpng-05.txt, Work in Progress, July 2002.

For SDP, numerous additional documents need to be taken into account:

[3] 3108 R. Kumar, M. Mostafa, "Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections," RFC 3108, May 2001.

[4] Gonzalo Camarillo, Jan Holler, Goran AP Eriksson, Henning Schulzrinne, "Grouping of media lines in SDP," Internet Draft draft-ietf-mmusic-fid-06.txt, Work in Progress, February 2002.

[5] F. Andreasen, "SDP Simple Capability Negotiation," Internet Draft draft-andreasen-mmusic-sdp-simcap-05.txt, Work in Progress, February 2002.

[6] Jonathan Rosenberg, Henning Schulzrinne, "An Offer/Answer Model with SDP," Internet Draft draft-ietf-mmusic-sdp-offer-answer-02.txt, Work in Progress, February 2002.

[7] Robert Fairlie-Cuninghame, "Guidelines for specifying SCTP-based media transport using SDP," Internet Draft draft-fairlie-mmusic-sdp-sctp-00.txt, Work in Progress, May 2001.

[8] David Yon, "Connection-Oriented Media Transport in SDP," Internet Draft draft-ietf-mmusic-sdp-comedia-03.txt, Work in Progress, June 2002.

[9] Tom Taylor, "Conventions for the use of the Session Description Protocol (SDP) for Digital Circuit Connections," Internet Draft draft-taylor-mmusic-sdp-tdm-01.txt, Work in Progress, April 2002.

[10] Christian Huitema, "RTCP attribute in SDP," Internet Draft draft-ietf-mmusic-sdp4nat-02.txt, Work in Progress, February 2002.

[11] Jari Arkko, Elisabette Carrarra, Fredrik Lindholm, Mats Naslund, Karl Norrman, "Key Management Extensions for SDP and RTSP," Internet Draft draft-ietf-mmusic-kmgmt-ext-05.txt, Work in Progress, June 2002.

[12] Sean Olson, Gonzalo Camarillo, Adam Roach, "Support for IPv6 in SDP," Internet Draft draft-ietf-mmusic-sdp-ipv6-03.txt, Work in

Progress, February 2002.

[13] G. Camarillo (ed), W. Marshall (ed), J. Rosenberg, "Integration of Resource Management and SIP", Internet Draft draft-ietf-sip-manyfolks-resource-07.txt, Work in Progress, April 2002.

9. Full Copyright Statement