

Session Description Protocol (SDP) Simple Capability Declaration

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This document defines a set of Session Description Protocol (SDP) attributes that enables SDP to provide a minimal and backwards compatible capability declaration mechanism. Such capability declarations can be used as input to a subsequent session negotiation, which is done by means outside the scope of this document. This provides a simple and limited solution to the general capability negotiation problem being addressed by the next generation of SDP, also known as SDPng.

1. 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 RFC-2119 [2].

2. Introduction

The Session Description Protocol (SDP) [3] describes multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP was not intended to provide capability negotiation. However, as the need for this has become increasingly important, work has begun on a "next generation SDP" (SDPng) [4,5] that supports both session description and capability negotiation. SDPng is not anticipated to be backwards compatible with SDP and work on SDPng is currently in the early stages. However, several other protocols, e.g. SIP [6] and Media Gateway Control Protocol (MGCP) [7], use SDP and are likely to

continue doing so for the foreseeable future. Nevertheless, in many cases these signaling protocols have an urgent need for some limited form of capability negotiation.

For example, an endpoint may support G.711 audio (over RTP) as well as T.38 fax relay (over UDP or TCP). Unless the endpoint is willing to support two media streams at the same time, this cannot currently be expressed in SDP. Another example involves support for multiple codecs. An endpoint indicates this by including all the codecs in the "m=" line in the session description. However, the endpoint thereby also commits to simultaneous support for each of these codecs. In practice, Digital Signal Processor (DSP) memory and processing power limitations may not make this feasible.

As noted in [4], the problem with SDP is that media descriptions are used to describe session parameters as well as capabilities without a clear distinction between the two.

In this document, we define a minimal and backwards compatible capability declaration feature in SDP by defining a set of new SDP attributes. Together, these attributes define a capability set, which consists of a capability set sequence number followed by one or more capability descriptions. Each capability description in the set contains information about supported media formats, but the endpoint is not committing to use any of these. In order to actually use a declared capability, session negotiation will have to be done by means outside the scope of this document, e.g., using the offer/answer model [8].

It should be noted that the mechanism is not intended to solve the general capability negotiation problem targeted by SDPng. It is merely intended as a simple and limited solution to the most urgent problems facing current users of SDP.

3. Simple Capability Declaration Attributes

The SDP Simple Capability Declaration (simcap) is defined by a set of SDP attributes. Together, these attributes form a capability set which describes the complete media capabilities of the endpoint. Any previous capability sets issued by the endpoint for the session in question no longer apply. The capability set consists of a sequence number and one or more capability descriptions. Each such capability description describes the media type and media formats supported and may include one or more capability parameters to further define the capability. A session description MUST NOT contain more than one capability set, however the capability set can describe capabilities at both the session and media level. Capability descriptions provided at the session level apply to all media streams of the media

type indicated, whereas capability descriptions provided at the media level apply to that particular media stream only. We refer to these respectively as session capabilities and media stream capabilities. A media stream capability may or may not be of the same media type as the media stream to which it applies.

The capability set MUST begin with a single sequence number followed by one or more capability descriptions listing all media formats the endpoint is currently able and willing to support. More specifically, if a media format is included in a media ("m=") line, then by definition the media format MUST be included in either a session capability or a media stream capability for that media line. The endpoint MAY include additional media formats in a capability if it is capable of supporting those media formats in a session with its peer. An endpoint MUST NOT include capabilities it knows it cannot use in a particular session. An endpoint receiving a capability set from another endpoint MAY use any of the media formats included in that capability set in a later attempt to negotiate media streams with the other endpoint, e.g., using the offer/answer model [8]. If a new capability set is received from the other endpoint, the old capability set MUST NOT be used any longer. Session capabilities can be used for any media streams of the indicated media type, whereas media stream capabilities can only be used for their associated media line. However, an endpoint receiving a capability set with a given media format MUST NOT assume that a subsequent attempt to negotiate a media stream using just this media format will succeed.

The individual capability descriptions in a capability set can be provided contiguously or they can be scattered throughout the session description. The first capability description MUST, however, follow immediately after the sequence number.

The sequence number is on the form:

```
a=sqn: <sqn-num>
```

where <sqn-num> is an integer between 0 and 255 (both included). The initial sequence number MUST be 0 (zero) and it MUST be incremented by 1 modulo 256 with each new capability set issued by the endpoint. Receivers may not necessarily see all capability sets issued and hence MUST NOT reject a capability set due to gaps in sequence numbers. The sequence number MUST either be provided as a session-level or media-level attribute, however there MUST NOT be more than one occurrence of the sequence number attribute in the session description (since there cannot be more than one capability set).

Each capability description in the capability set is on the form:

```
a=cdisc: <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. The capability description refers to a send and receive capability by default. When generating a capability set, the capability number MUST start with 1 in the first capability description, and be incremented by the number of media formats in the <fmt list> for each subsequent capability description. The media formats in the <fmt list> are numbered from left to right. Receivers of a capability set MUST NOT, however, reject capability descriptions due to gaps in the capability numbers. The capability number provides a convenient handle within the context of the capability set (as referenced by the sequence number) which may be used to reference a particular capability by means outside of this specification.

A capability description can include one or more capability parameter lines on the form:

```
a=cpar: <cap-par>
a=cparmin: <cap-par>
a=cparmax: <cap-par>
```

where <cap-par> is either bandwidth information ("b=") or an attribute ("a=") in its full '<type>=<value>' form (see [3]). A capability parameter line provides additional parameters for the preceding "cdisc" attribute line. Capability parameter lines for a capability description SHOULD immediately follow the "cdisc" line they refer to. Nevertheless, the capability description includes all capability parameter lines until the next capability description ("cdisc") or media ("m=") line in the session description.

The "cpar" attribute should normally be used when capability parameter values are to be specified. When provided, it means that the endpoint is declaring that it supports the media formats in the preceding "cdisc" line in accordance with the <cap-par> value specified. This can, for example, be used to specify "fmtp" parameters. If a session negotiation is attempted without considering the <cap-par> value, it may fail due to lack of endpoint support. A capability description may contain zero, one, or more "cpar" attribute lines describing either the same or different parameters. Describing the same parameter more than once can be used to specify alternatives.

Where a minimum numerical value is to be specified, the "cparmin" attribute should be used. There may be zero, one, or more "cparmin" attribute lines in a capability description, however a given parameter MUST NOT be described by a "cparmin" attribute more than once.

Where a maximum numerical value is to be specified, the "cparmax" attribute should be used. There may be zero, one, or more "cparmax" attribute lines in a capability description, however a given parameter MUST NOT be described by a "cparmax" attribute more than once.

Ranges of numerical values can be expressed by using a "cparmin" and a "cparmax" attribute for a given parameter. It follows from the previous rules, that only a single range can be specified for a given parameter.

Capability descriptions may be provided at both the session-level and media-level. A capability description provided at the session-level applies to all the media streams of the indicated media type in the session description. A capability description provided at the media-level only applies to that particular media stream (regardless of media type). If a capability description with media type X is provided at the session-level, and there are no media streams of type X in the session description, then it is undefined which of the media streams the capability description applies to (except if there is only one media stream). It is therefore RECOMMENDED, that such capabilities are provided at the media-level instead.

Below we show an example session description using the above simple capability declaration mechanism:

```
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 18 96
a=rtpmap:96 telephone-event
a=fmtp:96 0-15,32-35
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 0 18 96
a=cpar: a=fmtp:96 0-16,32-35
a=cdsc: 4 image udptl t38
a=cdsc: 5 image tcp t38
```

The sender of this session description is currently prepared to send and receive G.729 audio as well as telephone-events 0-15 and 32-35. The sender is furthermore capable of supporting:

- * PCMU encoding for the audio media stream,
- * telephone events 0-16 and 32-35,
- * T.38 fax relay using udp or tcp (see [9]).

Note, that the first capability number specified is 1, whereas the next is 4 since three media formats were included in the first capability description. Also note that the rtpmap for payload type 96 was not included in the capability description, as it was already specified for the media ("m=") line. Conversely, it would of course not have been valid to provide the rtpmap in the capability description and then omit the "a=rtpmap" line.

Below, we show another example of the simple capability declaration mechanism, this time with multiple media streams:

```
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 18
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18
m=video 3458 RTP/AVP 31
a=cdsc: 3 video RTP/AVP 31 34
```

The sender of this session description is currently prepared to send and receive G.729 audio and H.261 video. The sender is furthermore capable of supporting:

- * PCMU encoding for the audio media stream,
- * H.263 for the video media stream.

Note that the first capability number specified is 1, whereas the next is 3, since two media formats were included in the first capability description. Also note that the sequence number applies to the entire capability set, i.e. both audio and video, and hence is only supplied once. Finally, note that the media formats 18 and 31 are listed in both the media lines and the capability set as required. The above session description could have equally been supplied as follows:

```

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=
c=IN IP4 128.96.41.1
t=0 0
a=sgn: 0
a=cdsc: 1 audio RTP/AVP 0 18
a=cdsc: 3 video RTP/AVP 31 34
m=audio 3456 RTP/AVP 18
m=video 3458 RTP/AVP 31

```

i.e., with the capability set provided at the session-level.

4. Security Considerations

Capability negotiation of security-sensitive parameters is a delicate process, and should not be done without careful evaluation of the design, including the possible susceptibility to downgrade attacks. Use of capability re-negotiation may make the session susceptible to denial of service, without design care as to authentication.

5. IANA Considerations

This document defines the following new SDP parameters of type "att-field" (attribute names):

Attribute name: sgn
 Long form name: Sequence number.
 Type of attribute: Session-level and media-level.
 Subject to charset: No.
 Purpose: Capability set numbering.
 Appropriate values: See Section 3.

Attribute name: cdsc
 Long form name: Capability description.
 Type of attribute: Session-level and media-level.
 Subject to charset: No.
 Purpose: Describe capabilities in a capability set.
 Appropriate values: See Section 3.

Attribute name: cpar
 Long form name: Capability parameter line.
 Type of attribute: Session-level and media-level.
 Subject to charset: No.
 Purpose: Provide capability description parameters.
 Appropriate values: See Section 3.

Attribute name: cparmin
Long form name: Minimum capability parameter line.
Type of attribute: Session-level and media-level.
Subject to charset: No.
Purpose: Provide minimum capability description parameters.
Appropriate values: See Section 3.

Attribute name: cparmax
Long form name: Maximum capability parameter line.
Type of attribute: Session-level and media-level.
Subject to charset: No.
Purpose: Provide maximum capability description parameters.
Appropriate values: See Section 3.

6. Normative References

- [1] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] Handley, M. and V. Jacobson, "SDP: session description protocol", Request for Comments 2327, April 1998.

7. Informative References

- [4] Kutscher, Ott, Bormann and Curcio, "Requirements for Session Description and Capability Negotiation", Work in Progress.
- [5] Kutscher, Ott and Borman, "Session Description and Capability Negotiation", Work in Progress.
- [6] Handley, M., Schulzrinne, H., Schooler, E. and J. Rosenberg, "SIP: session initiation protocol", RFC 2543, March 1999.
- [7] Arango, M., Dugan, A., Elliott, I., Huitema, C. and S. Pickett, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 2705, October 1999.
- [8] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with SDP", Work in Progress.
- [9] ITU-T Recommendation T.38 Annex D, "SIP/SDP Call Establishment Procedures".

8. Acknowledgments

This work draws upon the ongoing work on SDPng in the IETF MMUSIC Working Group; in particular [4]. Furthermore this work was inspired by the CableLabs PacketCable project. The author would like to recognize and thank Joerg Ott and Jonathan Rosenberg who provided many detailed comments and suggestions to improve this specification. Colin Perkins, Orit Levin and Tom Taylor provided valuable feedback as well.

9. Author's Address

Flemming Andreasen
Cisco Systems
499 Thornall Street, 8th floor
Edison, NJ
EMail: fandreas@cisco.com

10. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

