

Network Working Group  
Internet-Draft  
Expires: October 16, 2001

Kutscher  
Ott  
Bormann  
TZI, Universitaet Bremen  
Curcio  
Nokia Mobile Phones  
April 17, 2001

**Requirements for Session Description and Capability Negotiation**  
**draft-ietf-mmusic-sdpng-req-01.txt**

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To view the entire list of Internet-Draft Shadow Directories, see <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 16, 2001.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

This document defines some terminology and lists a set of requirements that are relevant for a framework for session description and endpoint capability negotiation in multiparty multimedia conferencing scenarios.

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 [confctrl@isi.edu](mailto:confctrl@isi.edu) and/or the authors.

Document Revision



\$Revision: 2.1 \$

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Use Cases . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Terminology . . . . .	<a href="#">6</a>
<a href="#">4.</a>	General Requirements . . . . .	<a href="#">9</a>
<a href="#">4.1</a>	Simplicity . . . . .	<a href="#">9</a>
<a href="#">4.2</a>	Extensibility . . . . .	<a href="#">9</a>
<a href="#">4.3</a>	Firewall Friendliness . . . . .	<a href="#">9</a>
<a href="#">4.4</a>	Authentication, Authorization, Media Keys . . . . .	<a href="#">9</a>
<a href="#">4.5</a>	Text encoding . . . . .	<a href="#">10</a>
<a href="#">4.6</a>	Session vs. Media Description . . . . .	<a href="#">10</a>
<a href="#">4.7</a>	Mapping (of a Subset) to SDP . . . . .	<a href="#">10</a>
<a href="#">5.</a>	Session Description Requirements . . . . .	<a href="#">11</a>
<a href="#">5.1</a>	Media Description . . . . .	<a href="#">11</a>
<a href="#">5.1.1</a>	Medium Type . . . . .	<a href="#">11</a>
<a href="#">5.1.2</a>	Media Stream Packetization . . . . .	<a href="#">11</a>
<a href="#">5.1.3</a>	Transport . . . . .	<a href="#">11</a>
<a href="#">5.1.4</a>	QoS . . . . .	<a href="#">12</a>
<a href="#">5.1.5</a>	Resource Utilization . . . . .	<a href="#">12</a>
<a href="#">5.1.6</a>	Dependencies . . . . .	<a href="#">13</a>
<a href="#">5.1.7</a>	Other parameters (media-specific) . . . . .	<a href="#">13</a>
<a href="#">5.1.7.1</a>	Media Codec Bit Rates . . . . .	<a href="#">13</a>
<a href="#">5.1.7.2</a>	Advanced codec modes . . . . .	<a href="#">13</a>
<a href="#">5.1.8</a>	Naming Hierarchy and/or Scoping . . . . .	<a href="#">13</a>
<a href="#">5.1.9</a>	Profiles . . . . .	<a href="#">14</a>
<a href="#">5.1.10</a>	Registrations of Names . . . . .	<a href="#">14</a>
<a href="#">5.1.11</a>	Meta-Information . . . . .	<a href="#">14</a>
<a href="#">5.1.11.1</a>	Scheduling . . . . .	<a href="#">14</a>
<a href="#">5.1.11.2</a>	Optional Meta-Information Packages . . . . .	<a href="#">15</a>
<a href="#">6.</a>	Requirements for Capability Description and Negotiation	<a href="#">16</a>
<a href="#">6.1</a>	Capability Description . . . . .	<a href="#">16</a>
<a href="#">6.2</a>	Capability Constraints . . . . .	<a href="#">16</a>
<a href="#">6.3</a>	Processing Rules . . . . .	<a href="#">17</a>
<a href="#">7.</a>	Open Issues . . . . .	<a href="#">19</a>
<a href="#">8.</a>	Remarks . . . . .	<a href="#">20</a>
	References . . . . .	<a href="#">21</a>
	Authors' Addresses . . . . .	<a href="#">22</a>
	Full Copyright Statement . . . . .	<a href="#">24</a>



## **1. Introduction**

Multiparty multimedia conferencing is one application that requires the dynamic interchange of end system capabilities and the negotiation of a parameter set that is appropriate for all sending and receiving end systems in a conference. For some applications, e.g., for loosely coupled conferences, it may be sufficient to simply have session parameters be fixed by the initiator of a conference. In such a scenario no negotiation is required because only those participants with media tools that support the predefined settings can join a media session and/or a conference.

This approach is applicable for conferences that are announced some time ahead of the actual start date of the conference. Potential participants can check the availability of media tools in advance and tools like session directories can configure media tools on startup. This procedure however fails to work for conferences initiated spontaneously like Internet phone calls or ad-hoc multiparty conferences. Fixed settings for parameters like media types, their encoding etc. can easily inhibit the initiation of conferences, for example in situations where a caller insists on a fixed audio encoding that is not available at the callee's end system.

To allow for spontaneous conferences, the process of defining a conference's parameter set must therefore be performed either at conference start (for closed conferences) or maybe (potentially) even repeatedly every time a new participant joins an active conference. The latter approach may not be appropriate for every type of conference without applying certain policies: For conferences with TV-broadcast or lecture characteristics (one main active source) it is usually not desired to re-negotiate parameters every time a new participant with an exotic configuration joins because it may inconvenience existing participants or even exclude the main source from media sessions. But conferences with equal "rights" for participants that are open for new participants on the other hand would need a different model of dynamic capability negotiation, for example a telephone call that is extended to a 3-parties conference at some time during the session.

SDP [[1](#)] allows to specify multimedia sessions (i.e. conferences, "session" as used here is not to be confused with "RTP session"!) by providing general information about the session as a whole and specifications for all the media streams (RTP sessions and others) to be used to exchange information within the multimedia session.

Currently, media descriptions in SDP are used for two purposes:

o to describe session parameters for announcements and invitations

Kutscher, et. al.

Expires October 16, 2001

[Page 3]

(the original purpose of SDP)

- o to describe the capabilities of a system (and possibly provide a choice between a number of alternatives). Note that SDP was not designed to facilitate this.

A distinction between these two "sets of semantics" is only made implicitly.

In the following we first introduce a model for session description and capability negotiation and define some terms that are later used to express some requirements. Note that this list of requirements is possibly incomplete. The purpose of this document is to initiate the development of a session description and capability negotiation framework.





## **2. Use Cases**

This is an initial list of use cases:

And endpoint is a device attached to an IP network via a fixed or wireless connection (LAN, WLAN, GPRS, IMT-2000, etc.).

Case 1: Point-to-point connection

Case 2: Point-to-point connection with use of proxy/CPS

Case 3: 1-to-n connection (multicast distribution such as a lecture or video streaming or music)

Case 4: n-party connection (such as a speech and/or video and/or data call with a variable number of participants over time)



### 3. Terminology

Any (computer) system has, at a time, a number of rather fixed hardware as well as software resources. These resources ultimately define the limitations on what can be captured, displayed, rendered, replayed, etc. with this particular device. We term features enabled and restricted by these resources "system capabilities".

Example: System capabilities may include: a limitation of the screen resolution for true color by the graphics board; available audio hardware or software may offer only certain media encodings (e.g. G.711 and G.723.1 but not GSM); and CPU processing power and quality of implementation may constrain the possible video encoding algorithms.

In multiparty multimedia conferences, participants employ different "components" in conducting the conference.

Example: In lecture multicast conferences one component might be the voice transmission for the lecturer, another the transmission of video pictures showing the lecturer and the third the transmission of presentation material.

Depending on system capabilities, user preferences and other technical and political constraints, different configurations can be chosen to accomplish the ``deployment'' of these components.

Each component can be characterized at least by (a) its intended use (i.e. the function it shall provide) and (b) a one or more possible ways to realize this function. Each way of realizing a particular function is referred to as a "configuration".

Example: A conference component's intended use may be to make transparencies of a presentation visible to the audience on the Mbone. This can be achieved either by a video camera capturing the image and transmitting a video stream via some video tool or by loading a copy of the slides into a distributed electronic whiteboard. For each of these cases, additional parameters may exist, variations of which lead to additional configurations (see below).

Two configurations are considered different regardless of whether they employ entirely different mechanisms and protocols (as in the previous example) or they choose the same and differ only in a single parameter.

Example: In case of video transmission, a JPEG-based still image protocol may be used, H.261 encoded CIF images could be sent as could H.261 encoded QCIF images. All three cases constitute



different configurations. Of course there are many more detailed protocol parameters.

Each component's configurations are limited by the participating system's capabilities. In addition, the intended use of a component may constrain the possible configurations further to a subset suitable for the particular component's purpose.

Example: In a system for highly interactive audio communication the component responsible for audio may decide not to use the available G.723.1 audio codec to avoid the additional latency but only use G.711. This would be reflected in this component only showing configurations based upon G.711. Still, multiple configurations are possible, e.g. depending on the use of A-law or u-Law, packetization and redundancy parameters, etc.

In this system model, we distinguish two types of configurations:

- o potential configurations  
(a set of any number of configurations per component) indicating a system's functional capabilities as constrained by the intended use of the various components;
- o actual configurations  
(exactly one per instance of a component) reflecting the mode of operation of this component's particular instantiation.

Example: The potential configuration of the aforementioned video component may indicate support for JPEG, H.261/CIF, and H.261/QCIF. A particular instantiation for a video conference may use the actual configuration of H.261/CIF for exchanging video streams.

In summary, the key terms of this model are:

- o A multimedia session (streaming or conference) consists of one or more conference components for multimedia "interaction".
- o A component describes a particular type of interaction (e.g. audio conversation, slide presentation) that can be realized by means of different applications (possibly using different protocols).
- o A configuration is a set of parameters that are required to implement a certain variation (realization) of a certain component. There are actual and potential configurations.
  - \* Potential configurations describe possible configurations that are supported by an end system.



- \* An actual configuration is an "instantiation" of one of the potential configurations, i.e. a decision how to realize a certain component.

In less abstract words, potential configurations describe what a system can do ("capabilities") and actual configurations describe how a system is configured to operate at a certain point in time (media stream spec).

To decide on a certain actual configuration, a negotiation process needs to take place between the involved peers:

1. to determine which potential configuration(s) they have in common, and
2. to select one of this shared set of common potential configurations to be used for information exchange (e.g. based upon preferences, external constraints, etc.).

In SAP [[11](#)] based session announcements on the Mbone, for which SDP was originally developed, the negotiation procedure is non-existent. Instead, the announcement contains the media stream description sent out (i.e. the actual configurations) which implicitly describe what a receiver must understand to participate.

In point-to-point scenarios, the negotiation procedure is typically carried out implicitly: each party informs the other about what it can receive and the respective sender chooses from this set a configuration that it can transmit.

Capability negotiation must not only work for 2-party conferences but is also required for multi-party conferences. Especially for the latter case it is required that the process of determining the subset of allowable potential configurations is deterministic to reduce the number of required round trips before a session can be established.

In the following, we elaborate on requirements for an SDPng specification, subdivided into general requirements and requirements for session descriptions, potential and actual configurations as well as negotiation rules.





## **4. General Requirements**

Note that the order in which these requirements are presented does not imply their relative importance.

### **4.1 Simplicity**

The SDPng syntax shall be simple to parse and the protocol rules shall be easy to implement.

### **4.2 Extensibility**

SDPng shall be extensible in a backward compatible fashion. Extensions should be doable without modifying the SDPng specification itself. The spec should preclude two independent extensions from clashing with each other (e.g. in the naming of attributes).

Along with extensibility comes the requirement to identify certain extensions as mandatory in a given context while others as optional.

It should be possible to define subsets or profiles of the SDPng so that simple implementations understands only minimal parts of SDPng, and are able to interwork with more refined and complex SDPng implementations.

### **4.3 Firewall Friendliness**

It should be theoretically possible for firewalls (and other network infrastructure elements) to process announcements etc. that contain SDPng content. The concrete procedures have to be defined but if possible the processing of the SDPng content should be doable without interpretation of the textual descriptions.

In general, there should not be any problem for a gateway or a proxy server to execute the capability computation, and this operation has not to be limited to the endpoints only.

### **4.4 Authentication, Authorization, Media Keys**

SDPng should allow independent security attributes for parts of a session description. In particular, signing and/or encrypting parts of a session description should be supported.

The originator of the session may authenticate the session description or parts of it, or encrypt it so that only authorized users may access it.

In addition to the media encryption keys, the session description



should include also authentication keys, or include ways for authorized session participants to derive these keys.

In order to support rapid re-keying, the session description should include way to specify multiple encryption contexts and indicate how and when these encryption contexts will be used. For instance, the session description would indicate the current key and destination group, and then that the key will be changed at 20010401Z084000, and the media encoded with the new key will be sent to group 243.243.12.8.

#### **4.5 Text encoding**

A concise text representation is desirable in order to enhance portability and allow for simple implementations. At run time, size of encoded packets should be minimized, processing as well.

A language that allows specifications to be formally validated is desirable.

#### **4.6 Session vs. Media Description**

In many application scenarios (particularly with SIP and MEGACO/H.248), only media descriptions are needed and there is no need for session description parameters. SDPng should make parameter sets optional where it is conceivable that not all application will need them.

#### **4.7 Mapping (of a Subset) to SDP**

It shall be possible to translate a subset of SDPng into standard SDP session description to enable a certain minimal degree of interoperability between SDP-based and SDPng-based systems. However, as SDPng will provide enhanced functionality compared to SDP, a full mapping to SDP is not possible.

Note: Backwards compatibility to the SDP syntax has been discussed and it was found that this is not goal for SDPng, as it is felt that [RFC 2327](#) is too limiting.

Since several flavors of SDP have been developed (e.g., the MEGACO WG uses certain non-SDP enhancements) it needs to be discussed which of these flavors need to be considered for some kind of mapping.



## **5. Session Description Requirements**

For now, we only consider requirements for media (stream) descriptions.

### **5.1 Media Description**

It must be possible to express the following information with SDPng:

#### **5.1.1 Medium Type**

Payload types and format parameters for audio and video are well-defined and the basic semantics are clear (as defined in [RFC1889](#) [2] and [RFC2327](#) [1]).

Format descriptions for text and whiteboard are currently only defined in the context of specific applications, this is probably going to change in the future (not an SDPng work item).

Non-standard (in terms of defined as a non-standard payload type) codecs and format parameters can be accomplished by using dynamic payload type mappings. This is a crucial feature of SDP that needs to be preserved for RTP applications.

Current SDP only provides a= (a=fmtp) as means to specify codec parameters but actually gives little support on how to do this. Schemes for expressing more sophisticated parameters (e.g. supporting nesting) may be necessary. Nevertheless, it is imperative to keep the overall structure of a codec description manageable.

Note that there is a conflict between the desire to be able to use any old SDP and translate it in SDPng and the desire to have a useful structure in the SDPng data.

#### **5.1.2 Media Stream Packetization**

SDPng needs to be able to take care of more sophisticated payload descriptions than simple payload type assignment. Audio/video redundancy coding schemes need to be supported as need other mechanisms for FEC ([RFC 2733](#) [7]) and media stream repair ([RFC 2354](#) [8]). Also, layered coding schemes need to be supported.

Finally, a separation of the media encoding scheme, the packetization format, and possible repair schemes (and their respective parameters) is required.

#### **5.1.3 Transport**

Since session descriptions are not only used to describe sessions



that use IPv4/RTP for media transport it must be possible to specify different transport protocols (and their corresponding mandatory parameters). This means SDPng must support different address formats (IPv4, IPv6, E.164, NSAP, ...), multiplexing schemes (e.g. to identify a channel on a TDM link), and different transport protocol stacks (RTP/UDP/IP, RTP/AAL5/ATM, ...). Potential further parameters and interdependencies for multiplexed transports should be considered.

Additionally the requirement for expressing multiple addresses per actual configuration (layered coding support) has emerged, as well as the requirement for expressing multiple addresses per potential configuration (one port per payload type to simplify processing at the receiver). See also [Section 6.1](#) for a requirement to separate alternative configurations and simultaneous media sessions.

In multi-unicast-scenarios it must be possible to specify more than one transport address for a single media stream in an actual configuration, i.e. by specifying address lists.

In "broadcast"- or "lecture"-like sessions source filters might be needed that allow receivers to verify the source and apply filters in multicast sessions. Similarly, for SSM, the transport address includes an (Sender,Group) pair of IP addresses.

The definition of codecs and the definitions of transport parameters should be strictly separated from each other. See also [Section 5.1.9](#).

#### [5.1.4](#) QoS

QoS-Parameters for different protocol domains (e.g. traffic specification and flow specification or TOS bits for IP QoS) need to be specified. [draft-ietf-mmusic-sdp-qos-00.txt](#) [10] has provided a proposal for a syntax that can be used with SDP to describe network and security preconditions that have to be met in order to establish a session.

#### [5.1.5](#) Resource Utilization

Capability descriptions should not be based on available resources and resource requirements (in terms of CPU cycles, DSPs, etc.) for the following reasons:

- o Device manufacturers might not like to let hardware information go out from their devices.
- o The resource utilization function is not bijective since, for example, to support a certain media, a device A could require a





quantity X of resources, while another device B of a different manufacturer could require a quantity Y of resource, where  $X \neq Y$ . This is an implementation dependent issue, and it is related with how efficiently/inefficiently a manufacturer is able to implement a feature in its device.

#### **5.1.6 Dependencies**

Certain codes may depend on other resources being available (e.g. a G.723.1 audio codec may need a DTMF codec as well while a G.711 codec does not). Such interdependencies need to be expressed.

#### **5.1.7 Other parameters (media-specific)**

Extension mechanisms that allow to describe arbitrary other parameters of media codecs and formats are mandatory. It is possibly required to distinguish between mandatory and optional extension parameters.

In particular, it must be possible to introduce new (optional) parameters for a payload format and have old implementations still parse the parameters correctly.

Some audio/video specific parameters can be defined as suggested in [\[16\]](#).

##### **5.1.7.1 Media Codec Bit Rates**

There should be the possibility to configure ranges of bit rates (for example 32-64 kbps) or a discrete set of rates (i.e. 24, 32, 48, 64, 128, ... kbps). This is to allow an efficient resource allocation and allow as well interworking with systems where only a number of discrete bit rates is available. Resource reservation and QoS configuration mechanisms in general should be available as optional packages (see [Section 5.1.4](#)). It is conceivable that there be a separation into generic and transport specific QoS packages.

##### **5.1.7.2 Advanced codec modes**

Special advanced codec modes may be announced depending, for example, on the network conditions, to activate special settings in order to preserve good quality of media and to reduce the probability of call dropping.

##### **5.1.8 Naming Hierarchy and/or Scoping**

Parameter names should be constructed in a way to avoid clashes and thereby simplify independent development of e.g. codec parameter descriptions in different groups.



### **5.1.9 Profiles**

The configuration options for the different aspects of session descriptions (codec definitions, transport parameters etc.) should be defined in different orthogonal profiles ("packages"). Two different types of profiles are required:

- o A profile can define the data structures and collapsing rules for a specific function, e.g., for transport parameters. Capability and session descriptions can refer to such a profile and define concrete values for the profile parameters.
- o Instantiations of such profiles, that already contain concrete parameters, say for specific codec definitions, can be referenced by capability and session descriptions in order to allow for combining different aspects to a final description that is synthetic and of lower computational complexity.

An open issue is the question whether profiles should be referenced by name, i.e., by creating a well-known registry, or whether profiles should be referenced by address, i.e., by creating the possibility to retrieve them on demand. It is conceivable that a combination of both is useful: Some basic definitions are registered and well known and some other, uncommon definitions can be referenced by URIs.

### **5.1.10 Registrations of Names**

SDP uses top-level MIME content types [[15](#)] for media names. These convention should be adopted in order to avoid the unnecessary creation of a new namespace.

SDP also defines a registration procedure that allows to register new names for "media", "proto", "fmt", "bwtpe", "nettype", or "addrtype" field names (defined in [[1](#)]). If possible, the names that are already defined should be re-used. The definition of SDPng should contain a specification that states the registration procedures for SDPng.

### **5.1.11 Meta-Information**

#### **5.1.11.1 Scheduling**

In order to be usable for conference announcements, e.g. by means of SAP [[11](#)] it also required to provide a means for expressing scheduling information, i.e. to express the date (or the recurring dates) when a conference is taking place.

Two alternatives for implementing scheduling functions are



conceivable:

- o SDP-style (using the SDP model that is implemented as t= and r= lines); and
- o Using ICalender [\[14\]](#).

#### **[5.1.11.2](#) Optional Meta-Information Packages**

Location Meta-Information: In case of usage of SAP [\[11\]](#) as content or channel directory, the session description should include the location information, including physical location and the L1/L2 addressing information required to access the session. L1/L2 info may include things like transmission media used, frequencies, L2 multiplexing information etc. This makes it possible to broadcast session descriptions sent using broadband radio on, e.g., narrowband radio network. The recipients can derive their location from the location sent in the SAP session description, and then decide if and how they can receive the media sessions.

Pricing Information: The session description need to specify the pricing information for the session, if participating in the session requires payment.



## **6. Requirements for Capability Description and Negotiation**

### **6.1 Capability Description**

When describing the capabilities of endsystem by providing a list of different potential configurations, it must be possible to distinguish alternatives (different potential configuration) from different simultaneous sessions of a conference. A clear separation of these two concepts must be made that should also be realized in the description language.

### **6.2 Capability Constraints**

Capability negotiation is used to gain a session description (an actual configuration) that is compatible with the different end system capabilities and user preferences of the potential participants of a conference.

A media capability description is the same as a potential configuration, as it contains a set of allowable configurations for different components that could be used to implement the corresponding component. A capability description should allow specifying a number of interdependencies among capabilities. Traditional SDP only supports alternative capabilities and the specification implicitly assumed that all capabilities could be combined and basically used at the same time (looking at the pure session description, at least).

Processing power, hardware, link, or other resources may preclude the simultaneous use of certain configurations and/or limit the number of simultaneous instantiations of one or more configurations. This has led to a need to express in more detail constraints on combinations of configurations including the following constraints:

- o grouping capabilities (-> capability set);
- o expressing simultaneous capability sets;
- o expressing alternative capability sets; and
- o constraining the number of uses of a certain capability (set).

It needs to be carefully investigated how much more sophistication (if any) than simply listing alternatives needs to go into a base specification of SDPng (and which extension mechanisms for certain applications or for future revisions should be allowed).

Examples are known where complex capability descriptions are available but are simply not used (at least not at the level of





sophistication that would be possible). This strongly calls for keeping requirements on capability constraints rather modest (KISS).

The description of capabilities and the specification of capacity limits (maximum numbers of instantiations at a time) should be separated. This allows for greater modularity, since the common descriptions of capabilities can be referenced and imported, while the constraints (that are usually unique for a specific endsystem) can be provided inline and can be applied across single capability definitions. In order to allow for simple basic implementations, this also allows to treat the constraints as optional sections that do not have to be processed by an implementations.

The capabilities should be expressed as limitations on codec support, transport capability restrictions but not implicitly as limitations on machine resources, such as CPU type, clock rates, memory etc., that describe internal limitations in order to infer the supported capabilities.

### **6.3 Processing Rules**

The processing of potential configurations includes the process of "collapsing" sets of potential configurations offered by participants, i.e. the computation of the intersection of these potential configurations.

The processing (i.e. collapsing, forwarding etc.) of different potential configurations in order to find a compatible subset must work without having to know the semantics of the individual parameters. This is a key requirement for extensibility. Endsistemas, conference bridges, proxies and gateways are thus only required to understand the basic SDPng semantics of session and capability description in order to compute the supported subset of capabilities for a conference.

Additionally it must be possible to make use of different negotiation policies in order to reflect different conference types. For example in a lecture-style conference the policy might be to ensure that a capability collapsing process does not yield an actual configuration that excludes the main source (i.e. the lecturer and her end system) from the conference.

Preferences may also be considered in the negotiation process. This may need to be considered at the SDPng level (e.g. to express preferences, priorities).

Of course, the negotiation of configurations must not only work in peer-to-peer-conference scenarios but also be usable in multi party scenarios. The collapsing rules should work commutatively and



associatively, that means if given 3 end systems A,B,C the result for computing the supported configurations should be same when computing  $(A*B)*C$  and  $(B*C)*A$  (let "\*" be the collapsing function).

Negotiation of capabilities should take no longer than two or three message exchanges. The description format must enable such efficiency.

In order to allow for concise capability specification it will probably be required to group descriptions of, say, codecs and to establish a kind of hierarchy that allows to attach a certain attribute or parameter to a whole group of codecs.

It might then also be required to have a naming scheme that allows to name definitions in order to be able to later reference them in subsequent definitions. This is useful in situations where some definition extends a previous definition by just one parameter or in situations where codecs are combined, for example for expressing redundancy or layered codings. Different models of re-use are conceivable.



## **7. Open Issues**

This section contains a list of items that need further work and/or discussion:

It needs to distinguished more precisely between mandatory baseline functionality and optional extensions.

## **8. Remarks**

Explicitly addressing the issue of capability negotiation when drafting the new session description language generates new sets of requirements, some of which might conflict with other important goals, such as simplicity, conciseness and SDP-compatibility.

However, we think that it's worthwhile to sketch a reasonably complete and powerful solution first and then later develop a migration path from today's technology instead of imposing limitations at the outset to minimize the possibly necessary changes.



## References

- [1] Handley, M. and V. Jacobsen, "SDP: Session Description Protocol", [RFC 2327](#), April 1998.
- [2] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobsen, "RTP: A Transport Protocol for Real-Time Applications", [RFC 1889](#), January 1996.
- [3] Schulzrinne, H., "RTP Profile for Audio and Video Conferences with Minimal Control", [RFC 1890](#), January 1996.
- [4] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A. and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", [RFC 2198](#), September 1997.
- [5] Klyne, G., "A Syntax for Describing Media Feature Sets", [RFC 2533](#), March 1999.
- [6] Klyne, G., "Protocol-independent Content Negotiation Framework", [RFC 2703](#), September 1999.
- [7] Rosenberg, J. and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", [RFC 2733](#), December 1999.
- [8] Perkins, C. and O. Hodson, "Options for Repair of Streaming Media", [RFC 2354](#), June 1998.
- [9] Camarillo, G., Holler, J. and G. AP Eriksson, "SDP media alignment in SIP", Internet Draft [draft-camarillo-sip-sdp-01.txt](#), November 2000.
- [10] Rosenberg, J., Schulzrinne, H. and S. Donovan, "Establishing QoS and Security Preconditions for SDP Sessions", Internet Draft [draft-ietf-mmusic-sdp-qos-00.txt](#), June 1999.
- [11] Handley, M., Perkins, C. and E. Whelan, "Session Announcement Protocol", [RFC 2974](#), October 2000.
- [12] Kumar, R. and M. Mostafa, "Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections", Internet Draft [draft-ietf-mmusic-sdp-atm-05.txt](#), February 2001.
- [13] Casner, S., "SDP Bandwidth Modifiers for RTCP Bandwidth", Internet Draft [draft-ietf-avt-rtcp-bw-02.txt](#), November 2000.
- [14] Dawson, F., Stenerson, D., MISSING ORGANIZATION and , "Internet Calendaring and Scheduling Core Object Specification





(iCalendar)", [RFC 2445](#), November 1998.

- [15] Freed, N., Klensin, J. and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", [BCP 13](#), [RFC 2048](#), November 1996.
- [16] Levin, O., "SIP Requirements for support of Multimedia and Video", Internet Draft [draft-levin-sip-for-video-00.txt](#), February 2001.

#### Authors' Addresses

Dirk Kutscher  
TZI, Universitaet Bremen  
Bibliothekstr. 1  
Bremen 28359  
Germany

Phone: +49.421.218-7595  
Fax: +49.421.218-7000  
EMail: [dku@tzi.org](mailto:dku@tzi.org)

Joerg Ott  
TZI, Universitaet Bremen  
Bibliothekstr. 1  
Bremen 28359  
Germany

Phone: +49.421.201-7028  
Fax: +49.421.218-7000  
EMail: [jo@tzi.org](mailto:jo@tzi.org)

Carsten Bormann  
TZI, Universitaet Bremen  
Bibliothekstr. 1  
Bremen 28359  
Germany

Phone: +49.421.218-7024  
Fax: +49.421.218-7000  
EMail: [cabo@tzi.org](mailto:cabo@tzi.org)



Igor Curcio  
Nokia Mobile Phones  
P.O. Box 83 (Visiokatu 1)  
33721 Tampere  
Finland

Phone: +358.3.272.5372  
Fax: +358.10.505.7662  
EMail: igor.curcio@nokia.com

## Full Copyright Statement

Copyright (C) The Internet Society (2001). 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.

