                    SDP: Session Description Protocol
                     draft-ietf-mmusic-sdp-new-23.txt

Status of this Memo

Copyright Notice

Abstract

   This memo defines the Session Description Protocol (SDP). SDP is
   intended for describing multimedia sessions for the purposes of
   session announcement, session invitation, and other forms of
   multimedia session initiation.

Table of Contents

1. Introduction

    When initiating multimedia teleconferences, voice-over-IP calls,
    streaming video, or other sessions, there is a requirement to convey
    media details, transport addresses, and other session description
    metadata to the participants.

    SDP provides a standard representation for such information,
    irrespective of how that information is transported. SDP is purely a
    format for session description - it does not incorporate a transport
    protocol, and is intended to use different transport protocols as
    appropriate, including the Session Announcement Protocol [14],
    Session Initiation Protocol [15], Real-Time Streaming Protocol [16],
    electronic mail using the MIME extensions, and the Hypertext
    Transport Protocol.

    SDP is intended to be general purpose so that it can be used in a
    wide range of network environments and applications. However, it is
    not intended to support negotiation of session content or media
    encodings: this is viewed as outside the scope of session
    description.

    This memo updates RFC 2327 [6] in the light of implementation
    experience, and adds a small number of new features. Section 10
    outlines the changes introduced in this memo.

2. Glossary of Terms

    The following terms are used in this document, and have specific
    meaning within the context of this document.

    Conference: A multimedia conference is a set of two or more
       communicating users along with the software they are using to
       communicate.

    Session: A multimedia session is a set of multimedia senders and
       receivers and the data streams flowing from senders to receivers.
       A multimedia conference is an example of a multimedia session.

    Session Description: A well defined format for conveying sufficient
       information to discover and participate in a multimedia session.

    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 [3].

3. Examples of SDP Usage

3.1 Multicast Session Announcement

   In order to assist the advertisement of multicast multimedia
   conferences and other multicast sessions, and to communicate the
   relevant session setup information to prospective participants, a
   distributed session directory may be used. An instance of such a
   session directory periodically sends packets containing a description
   of the session to a well known multicast group. These advertisements
   are received by other session directories such that potential remote
   participants can use the session description to start the tools
   required to participate in the session.

   One protocol commonly used to implement such a distributed directory
   is the Session Announcement Protocol, SAP [14]. SDP provides the
   recommended session description format for such session
   announcements.

3.2 Session Initiation

   The Session Initiation Protocol, SIP [15] is an application layer
   control protocol for creating, modifying and terminating sessions
   such as Internet multimedia conferences, Internet telephone calls and
   multimedia distribution. The SIP messages used to create sessions
   carry session descriptions which allow participants to agree on a set
   of compatible media types. These session descriptions are commonly
   formatted using SDP. When used with SIP, the offer/answer model [17]
   provides a limited framework for negotiation using SDP.

3.3 Streaming media

   The Real Time Streaming Protocol, RTSP [16], is an application-level
   protocol for control over the delivery of data with real-time
   properties. RTSP provides an extensible framework to enable
   controlled, on-demand delivery of real-time data, such as audio and
   video. An RTSP client and server negotiate an appropriate set of
   parameters for media delivery, partially using SDP syntax to describe
   those parameters.

3.4 Email and the World Wide Web

   Alternative means of conveying session descriptions include
   electronic mail and the World Wide Web. For both email and WWW
   distribution, the MIME content type "application/sdp" is used. This
   enables the automatic launching of applications for participation in
   the session from the WWW client or mail reader in a standard manner.

Note that announcements of multicast sessions made only via email or
the World Wide Web (WWW) do not have the property that the receiver
of a session announcement can necessarily receive the session because
the multicast sessions may be restricted in scope, and access to the
WWW server or reception of email is possible outside this scope.
Session announcements made using SAP do not suffer this mismatch.

4. Requirements and Recommendations

The purpose of SDP is to convey information about media streams in
multimedia sessions to allow the recipients of a session description
to participate in the session. SDP is primarily intended for use in
an internetwork, although it is sufficiently general that it can
describe conferences in other network environments. Media streams
can be many-to-many. The times during which the session is active
need not be continuous.

Thus far, multicast based sessions on the Internet have differed from
many other forms of conferencing in that anyone receiving the traffic
can join the session (unless the session traffic is encrypted). In
such an environment, SDP serves two primary purposes. It is a means
to communicate the existence of a session, and is a means to convey
sufficient information to enable joining and participating in the
session. In a unicast environment, only the latter purpose is likely
to be relevant.

An SDP session description includes:

o Session name and purpose

o Time(s) the session is active

o The media comprising the session

o Information needed to receive those media (addresses, ports,
   formats and so on)

As resources necessary to participate in a session may be limited,
some additional information may also be desirable:

o Information about the bandwidth to be used by the session

o Contact information for the person responsible for the session

In general, SDP must convey sufficient information to enable
applications to join a session (with the possible exception of
encryption keys), and to announce the resources to be used to any
non-participants that may need to know (this latter feature is

primarily useful when SDP is used with a multicast session
announcement protocol).

4.1 Media and Transport Information

An SDP session description includes the following media information:

o   The type of media (video, audio, etc.)

o   The transport protocol (RTP/UDP/IP, H.320, etc.)

o   The format of the media (H.261 video, MPEG video, etc.)

In addition to media format and transport protocol, SDP conveys
address and port details. For an IP multicast session, these
comprise:

o   The multicast group address for media

o   The transport port for media

This address and port are the destination address and destination
port of the multicast stream, whether being sent, received, or both.

For unicast IP sessions, the following are conveyed:

o   The remote address for media

o   The remote transport port for media

The semantics of this address and port depend on the media and
transport protocol defined. By default, this SHOULD be the remote
address and remote port to which data is sent. Some media types MAY
redefine this behaviour, but this is NOT RECOMMENDED.

4.2 Timing Information

Sessions may either be bounded or unbounded in time. Whether or not
they are bounded, they may be only active at specific times. SDP can
convey:

o   An arbitrary list of start and stop times bounding the session

o   For each bound, repeat times such as "every Wednesday at 10am for
    one hour"

This timing information is globally consistent, irrespective of local
time zone or daylight saving time.

4.3 Private Sessions

   It is possible to create both public sessions and private sessions.
   SDP itself does not distinguish between these: private sessions are
   typically conveyed by encrypting the session description during
   distribution. The details of how encryption is performed are
   dependent on the mechanism used to convey SDP: mechanisms are
   currently defined for SDP transported using SAP [14] and SIP [15],
   others may be defined in future.

   If a session announcement is private it is possible to use that
   private announcement to convey encryption keys necessary to decode
   each of the media in a conference, including enough information to
   know which encryption scheme is used for each media.

4.4 Obtaining Further Information about a Session

   A session description should convey enough information to decide
   whether or not to participate in a session. SDP may include
   additional pointers in the form of Universal Resources Identifiers
   (URIs) for more information about the session.

4.5 Categorisation

   When many session descriptions are being distributed by SAP, or any
   other advertisement mechanism, it may be desirable to filter session
   announcements that are of interest from those that are not. SDP
   supports a categorisation mechanism for sessions that is capable of
   being automated.

4.6 Internationalisation

   The SDP specification recommends the use of the ISO 10646 character
   sets in the UTF-8 encoding [5] to allow many different languages to
   be represented. However, to assist in compact representations, SDP
   also allows other character sets such as ISO 8859-1 to be used when
   desired. Internationalisation only applies to free-text fields
   (session name and background information), and not to SDP as a whole.

5. SDP Specification

   An SDP session description is denoted by the MIME content type
   "application/sdp" (See Section 8).

   An SDP session description is entirely textual using the ISO 10646
   character set in UTF-8 encoding. SDP field names and attribute names
   use only the US-ASCII subset of UTF-8, but textual fields and
   attribute values MAY use the full ISO 10646 character set. Field and

attribute values which use the full UTF-8 character set are never
directly compared, hence there is no requirement for UTF-8
normalisation. The textual form, as opposed to a binary encoding
such as ASN.1 or XDR, was chosen to enhance portability, to enable a
variety of transports to be used, and to allow flexible, text-based
toolkits to be used to generate and process session descriptions.
However, since SDP may be used in environments where the maximum
permissible size of a session description is limited, the encoding is
deliberately compact. Also, since announcements may be transported
via very unreliable means or damaged by an intermediate caching
server, the encoding was designed with strict order and formatting
rules so that most errors would result in malformed session
announcements which could be detected easily and discarded. This
also allows rapid discarding of encrypted session announcements for
which a receiver does not have the correct key.

An SDP session description consists of a number of lines of text of
the form:

    <type>=<value>

where <type> MUST be exactly one case-significant character and
<value> is structured text whose format depends on <type>. In
general <value> is either a number of fields delimited by a single
space character, or a free format string. Whitespace MUST NOT be
used either side of the "=" sign.

An SDP session description consists of a session-level section
followed by zero or more media-level sections. The session-level
part starts with a "v=" line and continues to the first media-level
section. The media description starts with an "m=" line and
continues to the next media description or end of the whole session
description. In general, session-level values are the default for
all media unless overridden by an equivalent media-level value.

Some lines in each description are REQUIRED and some are OPTIONAL but
all MUST appear in exactly the order given here (the fixed order
greatly enhances error detection and allows for a simple parser).
OPTIONAL items are marked with a "*".

```
      Session description
         v=  (protocol version)
         o=  (owner/creator and session identifier)
         s=  (session name)
         i=* (session information)
         u=* (URI of description)
         e=* (email address)
         p=* (phone number)
         c=* (connection information - not required if included in
               all media)
         b=* (zero or more bandwidth information lines)
         One or more time descriptions ("t=" and "r=" lines, see below)
         z=* (time zone adjustments)
         k=* (encryption key)
         a=* (zero or more session attribute lines)
         Zero or more media descriptions

      Time description
         t=  (time the session is active)
         r=* (zero or more repeat times)

      Media description, if present
         m=  (media name and transport address)
         i=* (media title)
         c=* (connection information - optional if included at
               session-level)
         b=* (zero or more bandwidth information lines)
         k=* (encryption key)
         a=* (zero or more media attribute lines)
```

The set of type letters is deliberately small and not intended to be
extensible -- an SDP parser MUST completely ignore any session
description that contains a type letter that it does not understand.
The attribute mechanism ("a=" described below) is the primary means
for extending SDP and tailoring it to particular applications or
media. Some attributes (the ones listed in Section 6 of this memo)
have a defined meaning, but others may be added on an application-,
media- or session-specific basis. An SDP parser MUST ignore any
attribute it doesn't understand.

An SDP session description may contain URIs which reference external
content in the "u=", "k=" and "a=" lines. These URIs may be
dereferenced in some cases, making the session description non-self
contained.

The connection ("c=") and attribute ("a=") information in the
session-level section applies to all the media of that session unless
overridden by connection information or an attribute of the same name

in the media description. For instance, in the example below, each
media behaves as if it were given a "recvonly" attribute.

An example SDP description is:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

Text fields such as the session name and information are octet
strings which may contain any octet with the exceptions of 0x00
(Nul), 0x0a (ASCII newline) and 0x0d (ASCII carriage return). The
sequence CRLF (0x0d0a) is used to end a record, although parsers
SHOULD be tolerant and also accept records terminated with a single
newline character. If the "a=charset" attribute is not present,
these octet strings MUST be interpreted as containing ISO-10646
characters in UTF-8 encoding (the presence of the "a=charset"
attribute MAY force some fields to be interpreted differently).

A session description can contain domain names in the "o=", "u=",
"e=", "c=" and "a=" lines. Any domain name used in SDP MUST comply
with [1], [2]. Internationalised domain names (IDNs) MUST be
represented using the ASCII Compatible Encoding (ACE) form defined in
[11] and MUST NOT be directly represented in UTF-8 or any other
encoding (this requirement is for compatibility with RFC 2327 and
other SDP-related standards, which predate the development of
internationalized domain names).

5.1 Protocol Version ("v=")

```
v=0
```

The "v=" field gives the version of the Session Description Protocol.
This memo defines version 0. There is no minor version number.

5.2 Origin ("o=")

```
o=<username> <sess-id> <sess-version> <nettype> <addrtype>
  <unicast-address>
```

The "o=" field gives the originator of the session (her username and
the address of the user's host) plus a session identifier and version
number:

<username> is the user's login on the originating host, or it is "-"
   if the originating host does not support the concept of user ids.
   The <username> MUST NOT contain spaces.

<sess-id> is a numeric string such that the tuple of <username>,
   <sess-id>, <nettype>, <addrtype> and <unicast-address> form a
   globally unique identifier for the session. The method of
   <sess-id> allocation is up to the creating tool, but it has been
   suggested that a Network Time Protocol (NTP) format timestamp be
   used to ensure uniqueness [13].

<sess-version> is a version number for this session description. Its
   usage is up to the creating tool, so long as <sess-version> is
   increased when a modification is made to the session data. Again,
   it is RECOMMENDED that an NTP format timestamp is used.

<nettype> is a text string giving the type of network. Initially
   "IN" is defined to have the meaning "Internet", but other values
   MAY be registered in future (see Section 8).

<addrtype> is a text string giving the type of the address that
   follows. Initially "IP4" and "IP6" are defined, but other values
   MAY be registered in future (see Section 8).

<unicast-address> is the address of the machine from which the
   session was created. For an address type of IP4, this is either
   the fully-qualified domain name of the machine, or the
   dotted-decimal representation of the IP version 4 address of the
   machine. For an address type of IP6, this is either the
   fully-qualified domain name of the machine, or the compressed
   textual representation of the IP version 6 address of the machine.
   For both IP4 and IP6, the fully-qualified domain name is the form
   that SHOULD be given unless this is unavailable, in which case the
   globally unique address MAY be substituted. A local IP address
   MUST NOT be used in any context where the SDP description might
   leave the scope in which the address is meaningful.

In general, the "o=" field serves as a globally unique identifier for
this version of this session description, and the subfields excepting

the version taken together identify the session irrespective of any
modifications.

For privacy reasons, it is sometimes desirable to obfuscate the
username and IP address of the session originator. If this is a
concern, an arbitrary <username> and private <unicast-address> MAY be
chosen to populate the "o=" field, provided these are selected in a
manner that does not affect the global uniqueness of the field.

5.3 Session Name ("s=")

     s=<session name>

The "s=" field is the textual session name. There MUST be one and
only one "s=" field per session description. The "s=" field MUST NOT
be empty and SHOULD contain ISO 10646 characters (but see also the
"a=charset" attribute). If a session has no meaningful name, the
value "s= " SHOULD be used (i.e. a single space as the session
name).

5.4 Session Information ("i=")

     i=<session description>

The "i=" field provides textual information about the session. There
MUST be at most one session-level "i=" field per session description,
and at most one "i=" field per media. If the "a=charset" attribute
is present, it specifies the character set used in the "i=" field.
If the "a=charset" attribute is not present, the "i=" field MUST
contain ISO 10646 characters in UTF-8 encoding.

A single "i=" field MAY also be used for each media definition. In
media definitions, "i=" fields are primarily intended for labelling
media streams. As such, they are most likely to be useful when a
single session has more than one distinct media stream of the same
media type. An example would be two different whiteboards, one for
slides and one for feedback and questions.

The "i=" field is intended to provide a free-form human readable
description of the session or the purpose of a media stream. It is
not suitable for parsing by automata.

5.5 URI ("u=")

     u=<uri>

A URI is a Universal Resource Identifier as used by WWW clients [7],
[9]. The URI should be a pointer to additional information about the

session. This field is OPTIONAL, but if it is present it MUST be
specified before the first media field. No more than one URI field
is allowed per session description.

5.6 Email Address and Phone Number ("e=" and "p=")

      e=<email-address>
      p=<phone-number>

The "e=" and "p=" lines specify contact information for the person
responsible for the conference. This is not necessarily the same
person that created the conference announcement.

Inclusion of an email address or phone number is OPTIONAL. Note that
the previous version of SDP specified that either an email field or a
phone field MUST be specified, but this was widely ignored. The
change brings the specification into line with common usage.

If the email address or phone number are present, they MUST be
specified before the first media field. More than one email or phone
field can be given for a session description.

Phone numbers SHOULD be given in the form of an international public
telecommunication number (see ITU-T Recommendation E.164) preceded by
a "+". Spaces and hyphens may be used to split up a phone field to
aid readability if desired. For example:

   p=+1 617 555-6011

Both email addresses and phone numbers can have an OPTIONAL free text
string associated with them, normally giving the name of the person
who may be contacted. This MUST be enclosed in parenthesis if it is
present. For example:

   e=j.doe@example.com (Jane Doe)

The alternative RFC 2822 name quoting convention is also allowed for
both email addresses and phone numbers. For example:

   e=Jane Doe <j.doe@example.com>

The free text string SHOULD be in the ISO-10646 character set with
UTF-8 encoding, or alternatively in ISO-8859-1 or other encodings if
the appropriate session-level "a=charset" attribute is set.

5.7 Connection Data ("c=")

     c=<nettype> <addrtype> <connection-address>

   The "c=" field contains connection data.

   A session description MUST contain either at least one "c=" field in
   each media description or a single "c=" field at the session level.
   It MAY contain a single session-level "c=" field and additional "c="
   field(s) per media description, in which case the per-media values
   override the session-level settings for the respective media.

   The first sub-field ("<nettype>") is the network type, which is a
   text string giving the type of network. Initially "IN" is defined to
   have the meaning "Internet", but other values MAY be registered in
   the future (see Section 8).

   The second sub-field ("<addrtype>") is the address type. This allows
   SDP to be used for sessions that are not IP based. This memo only
   defines IP4 and IP6, but other values MAY be registered in the future
   (see Section 8).

   The third sub-field ("<connection-address>") is the connection
   address. OPTIONAL sub-fields MAY be added after the connection
   address depending on the value of the <addrtype> field.

   When the <addrtype> is IP4 and IP6, the connection address is defined
   as follows:

   o If the session is multicast, the connection address will be an IP
      multicast group address. If the session is not multicast, then
      the connection address contains the unicast IP address of the
      expected data source or data relay or data sink as determined by
      additional attribute fields. It is not expected that unicast
      addresses will be given in a session description that is
      communicated by a multicast announcement, though this is not
      prohibited.

   o Sessions using an IPv4 multicast connection address MUST also have
      a time to live (TTL) value present in addition to the multicast
      address. The TTL and the address together define the scope with
      which multicast packets sent in this conference will be sent. TTL
      values MUST be in the range 0-255. While the TTL MUST be
      specified, its use to scope multicast traffic is deprecated;
      applications SHOULD use an administratively scoped address
      instead.

   The TTL for the session is appended to the address using a slash as a

separator. An example is:

    c=IN IP4 224.2.36.42/127

IPv6 multicast does not use TTL scoping, and hence the TTL value MUST
NOT be present for IPv6 multicast. It is expected that IPv6 scoped
addresses will be used to limit the scope of conferences.

Hierarchical or layered encoding schemes are data streams where the
encoding from a single media source is split into a number of layers.
The receiver can choose the desired quality (and hence bandwidth) by
only subscribing to a subset of these layers. Such layered encodings
are normally transmitted in multiple multicast groups to allow
multicast pruning. This technique keeps unwanted traffic from sites
only requiring certain levels of the hierarchy. For applications
requiring multiple multicast groups, we allow the following notation
to be used for the connection address:

    <base multicast address>[/<ttl>]/<number of addresses>

If the number of addresses is not given it is assumed to be one.
Multicast addresses so assigned are contiguously allocated above the
base address, so that, for example:

    c=IN IP4 224.2.1.1/127/3

would state that addresses 224.2.1.1, 224.2.1.2 and 224.2.1.3 are to
be used at a TTL of 127. This is semantically identical to including
multiple "c=" lines in a media description:

    c=IN IP4 224.2.1.1/127
    c=IN IP4 224.2.1.2/127
    c=IN IP4 224.2.1.3/127

Similarly, an IPv6 example would be:

    c=IN IP6 FF15::101/3

which is semantically equivalent to:

    c=IN IP6 FF15::101
    c=IN IP6 FF15::102
    c=IN IP6 FF15::103

(remembering that the TTL field is not present in IPv6 multicast).

Multiple addresses or "c=" lines MAY be specified on a per-media
basis only if they provide multicast addresses for different layers

in a hierarchical or layered encoding scheme. They MUST NOT be
specified for a session-level "c=" field.

The slash notation for multiple addresses described above MUST NOT be
used for IP unicast addresses.

5.8 Bandwidth ("b=")

    b=<bwtype>:<bandwidth>

This OPTIONAL field denotes the proposed bandwidth to be used by the
session or media. The <bwtype> is an alphanumeric modifier giving
the meaning of the <bandwidth> figure. Two values are defined in
this specification, but other values MAY be registered in future (see
Section 8 and [20], [24]):

CT If the bandwidth of a session or media in a session is different
   from the bandwidth implicit from the scope, a "b=CT:..." line
   SHOULD be supplied for the session giving the proposed upper limit
   to the bandwidth used (the "conference total" bandwidth). The
   primary purpose of this is to give an approximate idea as to
   whether two or more sessions can co-exist simultaneously. When
   using the CT modifier with RTP, if several RTP sessions are part
   of the conference, the conference total refers to total bandwidth
   of all RTP sessions.

AS The bandwidth is interpreted to be application-specific (it will
   be the application's concept of maximum bandwidth). Normally this
   will coincide with what is set on the application's "maximum
   bandwidth" control if applicable. For RTP based applications, AS
   gives the RTP "session bandwidth" as defined in Section 6.2 of
   [18].

Note that CT gives a total bandwidth figure for all the media at all
sites. AS gives a bandwidth figure for a single media at a single
site, although there may be many sites sending simultaneously.

A prefix "X-" is defined for <bwtype> names. This is intended for
experimental purposes only. For example:

    b=X-YZ:128

Use of the "X-" prefix is NOT RECOMMENDED: instead new modifiers
SHOULD be registered with IANA in the standard namespace. SDP
parsers MUST ignore bandwidth fields with unknown modifiers.
Modifiers MUST be alpha-numeric and, although no length limit is
given, they are recommended to be short.

   The <bandwidth> is interpreted as kilobits per second by default.
   The definition of a new <bwtype> modifier MAY specify that the
   bandwidth is to be interpreted in some alternative unit (the "CT" and
   "AS" modifiers defined in this memo use the default units).

5.9 Timing ("t=")

     t=<start-time> <stop-time>

   The "t=" lines specify the start and stop times for a session.
   Multiple "t=" lines MAY be used if a session is active at multiple
   irregularly spaced times; each additional "t=" lines specifies an
   additional period of time for which the session will be active. If
   the session is active at regular times, an "r=" line (see below)
   should be used in addition to, and following, a "t=" line - in which
   case the "t=" line specifies the start and stop times of the repeat
   sequence.

   The first and second sub-fields give the start and stop times for the
   session respectively. These values are the decimal representation of
   Network Time Protocol (NTP) time values in seconds since 1900 [13].
   To convert these values to UNIX time, subtract decimal 2208988800.

   NTP timestamps are elsewhere represented by 64 bit values which wrap
   sometime in the year 2036. Since SDP uses an arbitrary length
   decimal representation, this should not cause an issue (SDP
   timestamps MUST continue counting seconds since 1900, NTP will use
   the value modulo the 64 bit limit).

   If the <stop-time> is set to zero, then the session is not bounded,
   though it will not become active until after the <start-time>. If
   the <start-time> is also zero, the session is regarded as permanent.

   User interfaces SHOULD strongly discourage the creation of unbounded
   and permanent sessions as they give no information about when the
   session is actually going to terminate, and so make scheduling
   difficult.

   The general assumption may be made, when displaying unbounded
   sessions that have not timed out to the user, that an unbounded
   session will only be active until half an hour from the current time
   or the session start time, whichever is the later. If behaviour
   other than this is required, an end-time SHOULD be given and modified
   as appropriate when new information becomes available about when the
   session should really end.

   Permanent sessions may be shown to the user as never being active
   unless there are associated repeat times which state precisely when

the session will be active. In general, permanent sessions SHOULD
NOT be created for any session expected to have a duration of less
than 2 months, and should be discouraged for sessions expected to
have a duration of less than 6 months.

5.10 Repeat Times ("r=")

    r=<repeat-interval> <active duration> <offsets from start-time>

"r=" fields specify repeat times for a session. For example, if a
session is active at 10am on Monday and 11am on Tuesday for one hour
each week for three months, then the <start-time> in the
corresponding "t=" field would be the NTP representation of 10am on
the first Monday, the <repeat interval> would be 1 week, the <active
duration> would be 1 hour, and the offsets would be zero and 25
hours. The corresponding "t=" field stop time would be the NTP
representation of the end of the last session three months later. By
default all fields are in seconds, so the "r=" and "t=" fields might
be:

    t=3034423619 3042462419
    r=604800 3600 0 90000

To make description more compact, times may also be given in units of
days, hours or minutes. The syntax for these is a number immediately
followed by a single case-sensitive character. Fractional units are
not allowed - a smaller unit should be used instead. The following
unit specification characters are allowed:

    d - days (86400 seconds)
    h - hours (3600 seconds)
    m - minutes (60 seconds)
    s - seconds (allowed for completeness but NOT RECOMMENDED)

Thus, the above session announcement could also have been written:

    r=7d 1h 0 25h

Monthly and yearly repeats cannot be directly specified with a single
SDP repeat time - instead separate "t=" fields should be used to
explicitly list the session times.

5.11 Time Zones ("z=")

    z=<adjustment time> <offset> <adjustment time> <offset> ....

To schedule a repeated session which spans a change from daylight
saving time to standard time or vice-versa, it is necessary to

specify offsets from the base time. This is required because
different time zones change time at different times of day, different
countries change to or from daylight time on different dates, and
some countries do not have daylight saving time at all.

Thus in order to schedule a session that is at the same time winter
and summer, it must be possible to specify unambiguously by whose
time zone a session is scheduled. To simplify this task for
receivers, we allow the sender to specify the NTP time that a time
zone adjustment happens and the offset from the time when the session
was first scheduled. The "z=" field allows the sender to specify a
list of these adjustment times and offsets from the base time.

An example might be:

    z=2882844526 -1h 2898848070 0

This specifies that at time 2882844526 the time base by which the
session's repeat times are calculated is shifted back by 1 hour, and
that at time 2898848070 the session's original time base is restored.
Adjustments are always relative to the specified start time – they
are not cumulative. Adjustments apply to all "t=" and "r=" lines in
a session description.

If a session is likely to last several years, it is expected that the
session announcement will be modified periodically rather than
transmit several years worth of adjustments in one session
announcement.

5.12 Encryption Keys ("k=")

    k=<method>
    k=<method>:<encryption key>

If transported over a secure and trusted channel, the session
description protocol MAY be used to convey encryption keys. A simple
mechanism for key exchange is provided by the key field ("k=")
although this is primarily supported for compatibility with older
implementations and its use is NOT RECOMMENDED. Work is in progress
to define new key exchange mechanisms for use with SDP [26][27] and
it is expected that new applications will use those mechanisms.

A key field is permitted before the first media entry (in which case
it applies to all media in the session), or for each media entry as
required. The format of keys and their usage is outside the scope of
this document, and the key field provides no way to indicate the
encryption algorithm to be used, key type, or other information about
the key: this is assumed to be provided by the higher-level protocol

using SDP. If there is a need to convey this information within SDP,
the extensions mentioned previously SHOULD be used. Many security
protocols require two keys: one for confidentiality, another for
integrity. This specification does not support transfer of two keys.

The method indicates the mechanism to be used to obtain a usable key
by external means, or from the encoded encryption key given. The
following methods are defined:

    k=clear:<encryption key>

        The encryption key is included untransformed in this key field.
        This method MUST NOT be used unless it can be guaranteed that
        the SDP is conveyed over a secure channel. The encryption key
        is interpreted as text according to the charset attribute, use
        the "k=base64:" method to convey characters that are otherwise
        prohibited in SDP.

    k=base64:<encoded encryption key>

        The encryption key is included in this key field but has been
        base64 encoded [12] because it includes characters that are
        prohibited in SDP. This method MUST NOT be used unless it can
        be guaranteed that the SDP is conveyed over a secure channel.

    k=uri:<URI to obtain key>

        A Universal Resource Identifier is included in the key field.
        The URI refers to the data containing the key, and may require
        additional authentication before the key can be returned. When
        a request is made to the given URI, the reply should specify
        the encoding for the key. The URI is often a secure HTTP URI,
        although this is not required.

    k=prompt

        No key is included in this SDP description, but the session or
        media stream referred to by this key field is encrypted. The
        user should be prompted for the key when attempting to join the
        session, and this user-supplied key should then be used to
        decrypt the media streams. The use of user-specified keys is
        NOT RECOMMENDED, since such keys tend to have weak security
        properties.

The key field MUST NOT be used unless it can be guaranteed that the
SDP is conveyed over a secure and trusted channel. An example of
such a channel might be SDP embedded inside an S/MIME message or a
TLS-protected HTTP session. It is important to ensure that the

secure channel is with the party that is authorised to join the
session, not an intermediary: if a caching proxy server is used, it
is important to ensure that the proxy is either trusted or unable to
access the SDP. Definition of appropriate security measures is
beyond the scope of this specification, and should be defined by the
users of SDP.

5.13 Attributes ("a=")

       a=<attribute>
       a=<attribute>:<value>

   Attributes are the primary means for extending SDP. Attributes may
   be defined to be used as "session-level" attributes, "media-level"
   attributes, or both.

   A media description may have any number of attributes ("a=" fields)
   which are media specific. These are referred to as "media-level"
   attributes and add information about the media stream. Attribute
   fields can also be added before the first media field; these
   "session-level" attributes convey additional information that applies
   to the conference as a whole rather than to individual media.

   Attribute fields may be of two forms:

   o A property attribute is simply of the form "a=<flag>". These are
      binary attributes, and the presence of the attribute conveys that
      the attribute is a property of the session. An example might be
      "a=recvonly".

   o A value attribute is of the form "a=<attribute>:<value>". For
      example, a whiteboard could have the value attribute
      "a=orient:landscape"

   Attribute interpretation depends on the media tool being invoked.
   Thus receivers of session descriptions should be configurable in
   their interpretation of session descriptions in general and of
   attributes in particular.

   Attribute names MUST use the US-ASCII subset of ISO-10646/UTF-8.

   Attribute values are octet strings, and MAY use any octet value
   except 0x00 (Nul), 0x0A (LF), and 0x0D (CR). By default, attribute
   values are to be interpreted as in ISO-10646 character set with UTF-8
   encoding. Unlike other text fields, attribute values are NOT
   normally affected by the "charset" attribute as this would make
   comparisons against known values problematic. However, when an
   attribute is defined, it can be defined to be charset-dependent, in

which case its value should be interpreted in the session charset
rather than in ISO-10646.

Attributes MUST be registered with IANA (see Section 8). If an
attribute is received that is not understood, it MUST be ignored by
the receiver.

5.14 Media Descriptions ("m=")

     m=<media> <port> <proto> <fmt> ...

A session description may contain a number of media descriptions.
Each media description starts with an "m=" field, and is terminated
by either the next "m=" field or by the end of the session
description. A media field has several sub-fields:

<media> is the media type. Currently defined media are "audio",
    "video", "text", "application" and "message", although this list
    may be extended in future (see Section 8).

<port> is the transport port to which the media stream is sent. The
    meaning of the transport port depends on the network being used as
    specified in the relevant "c=" field, and on the transport
    protocol defined in the <proto> sub-field of the media field.
    Other ports used by the media application (such as the RTCP port
    [18]) MAY be derived algorithmically from the base media port or
    MAY be specified in a separate attribute (for example "a=rtcp:" as
    defined in [21]).

    For applications where hierarchically encoded streams are being
    sent to a unicast address, it may be necessary to specify multiple
    transport ports. This is done using a similar notation to that
    used for IP multicast addresses in the "c=" field:

        m=<media> <port>/<number of ports> <proto> <fmt> ...

    In such a case, the ports used depend on the transport protocol.
    For RTP, the default is that only the even numbered ports are used
    for data with the corresponding one-higher odd ports used for the
    RTCP belonging to the RTP session, and the <number of ports>
    denoting the number of RTP sessions. For example:

        m=video 49170/2 RTP/AVP 31

    would specify that ports 49170 and 49171 form one RTP/RTCP pair
    and 49172 and 49173 form the second RTP/RTCP pair. RTP/AVP is the
    transport protocol and 31 is the format (see below). If
    non-contiguous ports are required, they must be signalled using a

separate attribute (for example "a=rtcp:" as defined in [21]).

If multiple addresses are specified in the "c=" field and multiple
ports are specified in the "m=" field, a one-to-one mapping from
port to the corresponding address is implied. For example:

    c=IN IP4 224.2.1.1/127/2
    m=video 49170/2 RTP/AVP 31

would imply that address 224.2.1.1 is used with ports 49170 and
49171, and address 224.2.1.2 is used with ports 49172 and 49173.

The combination of ports specified in "m=" lines and IP addresses
specified in "c=" lines MUST comply with the following rules for
RTP-based media streams (other protocols SHOULD define similar
rules):

1. If two media sessions have the same transport address (i.e.
    identical IP address and port numbers), the associated payload
    types (e.g. given in the "a=rtpmap:" attribute) MUST NOT be
    in conflict, i.e. the same payload type MUST NOT be mapped to
    different media types.

2. If two media sessions have the same transport address, they
    MUST use compatible media (e.g. both audio or both video).

3. If two media sessions have the same transport address, they
    SHOULD operate under the same RTP profile. The sessions MAY
    use two different RTP profiles only if those profiles are
    specifically designed to be compatible.

4. If two media sessions have the same RTP transport address,
    they MUST also use the same RTCP address and vice versa.

Two media sessions with the same transport address indicate
alternatives for the same media stream, i.e. all profiles, media
types, and payload types provided in any of the "m=" lines are
valid.

<proto> is the transport protocol. The meaning of the transport
   protocol is dependent on the address type field in the relevant
   "c=" field. Thus a "c=" field of IP4 indicates that the transport
   protocol runs over IP4. The following transport protocols are
   defined, but may be extended through registration of new protocols
   with IANA (see Section 8):

        * udp: denotes an unspecified protocol running over UDP.

        * RTP/AVP: denotes RTP [18] used under the RTP Profile for Audio
           and Video Conferences with Minimal Control [19] running over
           UDP.

        * RTP/SAVP: denotes the Secure Real-time Transport Protocol [22]
           running over UDP.

     The main reason to specify the transport-protocol in addition to
     the media format is that the same standard media formats may be
     carried over different transport protocols even when the network
     protocol is the same – a historical example is vat PCM audio and
     RTP PCM audio, another might be TCP/RTP PCM audio. In addition,
     relays and monitoring tools that are transport-protocol-specific
     but format-independent are possible.

   <fmt> is a media format description. The fourth and any subsequent
     sub-fields describe the format of the media. The interpretation
     of the media format depends on the value of the <proto> sub-field.

     If the <proto> sub-field is "RTP/AVP" or "RTP/SAVP" the <fmt>
     sub-fields contain RTP payload type numbers. When a list of
     payload type numbers is given, this implies that all of these
     payload formats MAY be used in the session, but the first of these
     formats SHOULD be used as the default format for the session. For
     dynamic payload type assignments the "a=rtpmap:" attribute (see
     Section 6) SHOULD be used to map from an RTP payload type number
     to a media encoding name that identifies the payload format. The
     "a=fmtp:" attribute MAY be used to specify format parameters (see
     Section 6).

     If the <proto> sub-field is "udp" the <fmt> sub-fields MUST
     reference a media type describing the format under the "audio",
     "video", "text", "application" or "message" top-level MIME types.
     The media type registration SHOULD define the packet format for
     use with UDP transport.

     For media using other transport protocols, the <fmt> field is
     protocol specific. Rules for interpretation of the <fmt>
     sub-field MUST be defined when registering new protocols (see
     section 8.2.2).

6. SDP Attributes

   The following attributes are defined. Since application writers may
   add new attributes as they are required, this list is not exhaustive.

Registration procedures for new attributes are defined in Section
8.2.4.

   a=cat:<category>

      This attribute gives the dot-separated hierarchical category
      of the session. This is to enable a receiver to filter
      unwanted sessions by category. It is a session-level
      attribute, and is not dependent on charset.

   a=keywds:<keywords>

      Like the cat attribute, this is to assist identifying wanted
      sessions at the receiver. This allows a receiver to select
      interesting session based on keywords describing the purpose
      of the session. It is a session-level attribute. It is a
      charset dependent attribute, meaning that its value should be
      interpreted in the charset specified for the session
      description if one is specified, or by default in ISO
      10646/UTF-8.

   a=tool:<name and version of tool>

      This gives the name and version number of the tool used to
      create the session description. It is a session-level
      attribute, and is not dependent on charset.

   a=ptime:<packet time>

      This gives the length of time in milliseconds represented by
      the media in a packet. This is probably only meaningful for
      audio data, but may be used with other media types if it makes
      sense. It should not be necessary to know ptime to decode RTP
      or vat audio, and it is intended as a recommendation for the
      encoding/packetisation of audio. It is a media attribute, and
      is not dependent on charset.

   a=maxptime:<maximum packet time>

      The maximum amount of media which can be encapsulated in each
      packet, expressed as time in milliseconds. The time SHALL be
      calculated as the sum of the time the media present in the
      packet represents. For frame based codecs, the time SHOULD
      be an integer multiple of the frame size. This attribute is
      probably only meaningful for audio data, but may be used with
      other media types if it makes sense. It is a media attribute,
      and is not dependent on charset. Note that this attribute was
      introduced after RFC 2327, and non updated implementations will

      ignore this attribute.

   a=rtpmap:<payload type> <encoding name>/<clock rate>
     [/<encoding parameters>]

      This attribute maps from an RTP payload type number (as used in
      an "m=" line) to an encoding name denoting the payload format
      to be used. It also provides information on the clock rate and
      encoding parameters. It is a media level attribute that is not
      dependent on charset.

      While an RTP profile may make static assignments of payload
      type numbers to payload formats, it is more common for that
      assignment to be done dynamically using "a=rtpmap:" attributes.
      As an example of a static payload type, consider u-law PCM
      coded single channel audio sampled at 8kHz. This is completely
      defined in the RTP Audio/Video profile as payload type 0, so
      there is no need for an "a=rtpmap: attribute, and the media for
      such a stream sent to UDP port 49232 can be specified as:

          m=audio 49232 RTP/AVP 0

      An example of a dynamic payload type is 16 bit linear encoded
      stereo audio sampled at 16 kHz. If we wish to use the dynamic
      RTP/AVP payload type 98 for this stream, additional information
      is required to decode it:

          m=audio 49232 RTP/AVP 98
          a=rtpmap:98 L16/16000/2

      Up to one rtpmap attribute can be defined for each media format
      specified. Thus we might have:

          m=audio 49230 RTP/AVP 96 97 98
          a=rtpmap:96 L8/8000
          a=rtpmap:97 L16/8000
          a=rtpmap:98 L16/11025/2

      RTP profiles that specify the use of dynamic payload types MUST
      define the set of valid encoding names and/or a means to
      register encoding names if that profile is to be used with SDP.
      The "RTP/AVP" and "RTP/SAVP" profiles use MIME sub-types for
      encoding names, under the top-level media type denoted in the
      "m=" line. In the example above, the media types are "audio/l8"
      and "audio/l16".

      For audio streams, <encoding parameters> indicates the
      number of audio channels. This parameter is OPTIONAL and

   may be omitted if the number of channels is one, provided
   no additional parameters are needed.

   For video streams, no encoding parameters are currently
   specified.

   Additional encoding parameters MAY be defined in the future,
   but codec specific parameters SHOULD NOT be added. Parameters
   added to an "a=rtpmap:" attribute SHOULD only be those required
   for a session directory to make the choice of appropriate media
   to participate in a session. Codec-specific parameters should
   be added in other attributes (for example, "a=fmtp:").

   Note: RTP audio formats typically do not include information
   about the number of samples per packet. If a non-default (as
   defined in the RTP Audio/Video Profile) packetisation is
   required, the "ptime" attribute is used as given below.

a=recvonly

   This specifies that the tools should be started in receive
   only mode where applicable. It can be either a session or
   media attribute, and is not dependent on charset. Note that
   recvonly applies to the media only, not to any associated
   control protocol (e.g. an RTP based system in recvonly mode
   SHOULD still send RTCP packets).

a=sendrecv

   This specifies that the tools should be started in send and
   receive mode. This is necessary for interactive conferences
   with tools that default to receive only mode. It can be either
   a session or media attribute, and is not dependent on charset.

   If none of the attributes "sendonly", "recvonly", "inactive",
   and "sendrecv" is present, "sendrecv" SHOULD be assumed as the
   default for sessions which are not of the conference type
   "broadcast" or "H332" (see below).

a=sendonly

   This specifies that the tools should be started in send-only
   mode. An example may be where a different unicast address is
   to be used for a traffic destination than for a traffic
   source. In such a case, two media descriptions may be used,
   one sendonly and one recvonly. It can be either a session or
   media attribute, but would normally only be used as a media
   attribute. It is not dependent on charset. Note that sendonly

applies only to the media, and any associated control protocol
(e.g. RTCP) SHOULD still be received and processed as normal.

a=inactive

This specifies that the tools should be started in inactive
mode. This is necessary for interactive conferences where
users can put other users on hold. No media is sent over an
inactive media stream. Note that an RTP based system SHOULD
still send RTCP, even if started inactive. It can be either a
session or media attribute, and is not dependent on charset.

a=orient:<orientation>

Normally this is only used for a whiteboard or presentation
tool. It specifies the orientation of a the workspace on
the screen. It is a media attribute. Permitted values are
"portrait", "landscape" and "seascape" (upside down landscape).
It is not dependent on charset.

a=type:<conference type>

This specifies the type of the conference. Suggested values
are "broadcast", "meeting", "moderated", "test" and "H332".
"recvonly" should be the default for "type:broadcast"
sessions, "type:meeting" should imply "sendrecv" and
"type:moderated" should indicate the use of a floor control
tool and that the media tools are started so as to mute new
sites joining the conference.

Specifying the attribute "type:H332" indicates that this
loosely coupled session is part of a H.332 session as defined
in the ITU H.332 specification [15]. Media tools should be
started "recvonly".

Specifying the attribute "type:test" is suggested as a hint
that, unless explicitly requested otherwise, receivers can
safely avoid displaying this session description to users.

The type attribute is a session-level attribute, and is not
dependent on charset.

a=charset:<character set>

This specifies the character set to be used to display the
session name and information data. By default, the ISO-10646
character set in UTF-8 encoding is used. If a more compact
representation is required, other character sets may be used.

For example, the ISO 8859-1 is specified with the following
SDP attribute:

    a=charset:ISO-8859-1

This is a session-level attribute and is not dependent on
charset. The charset specified MUST be one of those registered
with IANA, such as ISO-8859-1. The character set identifier is
a US-ASCII string and MUST be compared against the IANA
identifiers using a case insensitive comparison. If the
identifier is not recognised or not supported, all strings that
are affected by it SHOULD be regarded as octet strings.

Note that a character set specified MUST still prohibit the
use of bytes 0x00 (Nul), 0x0A (LF) and 0x0d (CR). Character
sets requiring the use of these characters MUST define a
quoting mechanism that prevents these bytes appearing within
text fields.

a=sdplang:<language tag>

    This can be a session level attribute or a media level
    attribute. As a session level attribute, it specifies the
    language for the session description. As a media level
    attribute, it specifies the language for any media-level SDP
    information field associated with that media. Multiple
    sdplang attributes can be provided either at session or media
    level if multiple languages in the session description or
    media use multiple languages, in which case the order of the
    attributes indicates the order of importance of the various
    languages in the session or media from most important to least
    important.

    In general, sending session descriptions consisting of
    multiple languages is discouraged. Instead, multiple
    descriptions SHOULD be sent describing the session, one in
    each language. However this is not possible with all
    transport mechanisms, and so multiple sdplang attributes are
    allowed although NOT RECOMMENDED.

    The "sdplang" attribute value must be a single RFC 3066
    language tag in US-ASCII [6]. It is not dependent on
    the charset attribute. An "sdplang" attribute SHOULD be
    specified when a session is of sufficient scope to cross
    geographic boundaries where the language of recipients cannot
    be assumed, or where the session is in a different language
    from the locally assumed norm.

a=lang:<language tag>

   This can be a session level attribute or a media level
   attribute. As a session level attribute, it specifies the
   default language for the session being described. As a media
   level attribute, it specifies the language for that media,
   overriding any session-level language specified. Multiple
   lang attributes can be provided either at session or media
   level if the session description or media use multiple
   languages, in which case the order of the attributes indicates
   the order of importance of the various languages in the
   session or media from most important to least important.

   The "lang" attribute value must be a single RFC 3066 language
   tag in US-ASCII [6]. It is not dependent on the charset
   attribute. A "lang" attribute SHOULD be specified when a
   session is of sufficient scope to cross geographic boundaries
   where the language of recipients cannot be assumed, or where
   the session is in a different language from the locally
   assumed norm.

a=framerate:<frame rate>

   This gives the maximum video frame rate in frames/sec. It is
   intended as a recommendation for the encoding of video data.
   Decimal representations of fractional values using the
   notation "<integer>.<fraction>" are allowed. It is a
   media attribute, defined only for video media, and is not
   dependent on charset.

a=quality:<quality>

   This gives a suggestion for the quality of the encoding as an
   integer value. The intention of the quality attribute for
   video is to specify a non-default trade-off between frame-rate
   and still-image quality. For video, the value in the range 0
   to 10, with the following suggested meaning:

      10 - the best still-image quality the compression scheme
           can give.
       5 - the default behaviour given no quality suggestion.
       0 - the worst still-image quality the codec designer
           thinks is still usable.

   It is a media attribute, and is not dependent on charset.

a=fmtp:<format> <format specific parameters>

This attribute allows parameters that are specific to a
particular format to be conveyed in a way that SDP doesn't
have to understand them. The format must be one of the
formats specified for the media. Format-specific parameters
may be any set of parameters required to be conveyed by SDP
and given unchanged to the media tool that will use this
format. At most one instance of this attribute is allowed
for each format.

It is a media attribute, and is not dependent on charset.

7. Security Considerations

   SDP is frequently used with the Session Initiation Protocol [15]
   using the offer/answer model [17] to agree parameters for unicast
   sessions. When used in this manner, the security considerations of
   those protocols apply.

   SDP is a session description format that describes multimedia
   sessions. A session description SHOULD NOT be trusted unless it has
   been obtained by an authenticated transport protocol from a trusted
   source. Many different transport protocols may be used to distribute
   session description, and the nature of the authentication will differ
   from transport to transport.

   One transport that will frequently be used to distribute session
   descriptions is the Session Announcement Protocol (SAP). SAP
   provides both encryption and authentication mechanisms but due to the
   nature of session announcements it is likely that there are many
   occasions where the originator of a session announcement cannot be
   authenticated because they are previously unknown to the receiver of
   the announcement and because no common public key infrastructure is
   available.

   On receiving a session description over an unauthenticated transport
   mechanism or from an untrusted party, software parsing the session
   should take a few precautions. Session descriptions contain
   information required to start software on the receivers system.
   Software that parses a session description MUST NOT be able to start
   other software except that which is specifically configured as
   appropriate software to participate in multimedia sessions. It is
   normally considered inappropriate for software parsing a session
   description to start, on a user's system, software that is
   appropriate to participate in multimedia sessions, without the user
   first being informed that such software will be started and giving
   their consent. Thus a session description arriving by session
   announcement, email, session invitation, or WWW page MUST NOT deliver

the user into an interactive multimedia session unless the user has
explicitly pre-authorized such action. As it is not always simple to
tell whether a session is interactive or not, applications that are
unsure should assume sessions are interactive.

In this specification, there are no attributes which would allow the
recipient of a session description to be informed to start multimedia
tools in a mode where they default to transmitting. Under some
circumstances it might be appropriate to define such attributes. If
this is done an application parsing a session description containing
such attributes SHOULD either ignore them, or inform the user that
joining this session will result in the automatic transmission of
multimedia data. The default behaviour for an unknown attribute is
to ignore it.

Session descriptions may be parsed at intermediate systems such as
firewalls for the purposes of opening a hole in the firewall to allow
participation in multimedia sessions. This SHOULD NOT be done unless
the SDP is conveyed in a manner that allows proper authentication and
authorization checks to ensure that firewall holes are only opened in
accordance with applicable security policy. SDP by itself does not
include sufficient information to enable these checks: they depend on
the encapsulating protocol (e.g. SIP or RTSP).

Use of the "k=" field poses a significant security risk, since it
conveys session encryption keys in the clear. SDP MUST NOT be used
to convey key material, unless it can be guaranteed that the channel
over which the SDP is delivered is both private and authenticated.

8. IANA Considerations

8.1 The "application/sdp" media type

One MIME type is to be registered, as defined below. This updates
the previous definition from RFC 2327.

   To: ietf-types@iana.org
   Subject: Registration of media type "application/sdp"

   MIME media type name: application

   MIME subtype name: sdp

   Required parameters: None.

   Optional parameters: None.

   Encoding considerations:

        SDP files are primarily 7-bit ASCII text. The "a=charset:"
        attribute may be used to signal the presence of other,
        possibly 8-bit, text in certain parts of an SDP file (see
        section 6 of RFC XXXX). Arbitrary binary content cannot
        be directly represented in SDP.

     Security considerations:
        See section 7 of RFC XXXX

     Interoperability considerations:
        See RFC XXXX

     Published specification:
        See RFC XXXX

     Applications which use this media type:
        Voice over IP, video teleconferencing, streaming media, instant
        messaging, etc. See also section 3 of RFC XXXX.

     Additional information:

     Magic number(s):   None.
     File extension(s): The extension ".sdp" is commonly used.
     Macintosh File Type Code(s): "sdp "

     Person & email address to contact for further information:
        Mark Handley <M.Handley@cs.ucl.ac.uk>
        Colin Perkins <csp@csperkins.org>
        IETF MMUSIC working group <mmusic@ietf.org>

     Intended usage: COMMON

     Author/Change controller:
        Authors of RFC XXXX
        IETF MMUSIC working group delegated from the IESG

8.2 Registration of Parameters

   There are seven field names that may be registered with IANA. Using
   the terminology in the SDP specification BNF, they are "media",
   "proto", "fmt", "att-field", "bwtype", "nettype" and "addrtype".

8.2.1 Media types ("media")

   The set of media types is intended to be small and SHOULD NOT be
   extended except under rare circumstances. The same rules should
   apply for media names as for top-level MIME content types, and where

possible the same name should be registered for SDP as for MIME. For
media other than existing MIME top-level content types, a
standards-track RFC MUST be produced for a new top-level content type
to be registered, and the registration MUST provide good
justification why no existing media name is appropriate (the
"Standards Action" policy of RFC 2434 [8].

This memo registers the media types "audio", "video", "text",
"application" and "message".

Note: The media types "control" and "data" were listed as valid in
the previous version of this specification [6], however their
semantics were never fully specified and they are not widely used.
These media types have been removed in this specification, although
they still remain valid media type capabilities for a SIP user agent
as defined in RFC 3840 [23]. If these media types are considered
useful in future, a Standards Track RFC MUST be produced to document
their use. Until that is done, applications SHOULD NOT use these
types and SHOULD NOT declare support for them in SIP capabilities
declarations (even though they exist in the registry created by RFC
3840).

8.2.2 Transport protocols ("proto")

The "proto" field describes the transport protocol used. This SHOULD
reference a standards-track protocol RFC. This memo registers three
values: "RTP/AVP" is a reference to RTP [18] used under the RTP
Profile for Audio and Video Conferences with Minimal Control [19]
running over UDP/IP, "RTP/SAVP" is a reference to the Secure
Real-time Transport Protocol [22], and "udp" indicates an unspecified
protocol over UDP.

If other RTP profiles are defined in the future, their "proto" name
SHOULD be specified in the same manner. For example, an RTP profile
whose short name is "XYZ" would be denoted by a "proto" field of
"RTP/XYZ".

New transport protocols SHOULD be registered with IANA.
Registrations MUST reference an RFC describing the protocol. Such an
RFC MAY be Experimental or Informational, although it is preferable
if it is Standards-Track. Registrations MUST also define the rules
by which their "fmt" namespace is managed (see below).

8.2.3 Media formats ("fmt")

Each transport protocol, defined by the "proto" field, has an
associated "fmt" namespace that describes the media formats which may
conveyed by that protocol. Formats cover all the possible encodings

that might want to be transported in a multimedia session.

RTP payload formats under the "RTP/AVP" and "RTP/SAVP" profiles MUST
use the payload type number as their "fmt" value. If the payload
type number is dynamically assigned by this session description, an
additional "rtpmap" attribute MUST be included to specify the format
name and parameters as defined by the MIME type registration for the
payload format. It is RECOMMENDED that other RTP profiles which are
registered (in combination with RTP) as SDP transport protocols
specify the same rules for the "fmt" namespace.

For the "udp" protocol, new formats SHOULD be registered. Use of an
existing MIME subtype for the format is encouraged. If no MIME
subtype exists, it is RECOMMENDED that a suitable one is registered
through the IETF process (RFC 2048) by production of, or reference
to, a standards-track RFC that defines the transport protocol for the
format.

For other protocols, formats MAY be registered according to the rules
of the associated "proto" specification.

Registrations of new formats MUST specify which transport protocols
they apply to.

8.2.4 Attribute names ("att-field")

Attribute field names ("att-field") MUST be registered with IANA and
documented, because of noticeable issues due to conflicting
attributes under the same name. Unknown attributes in SDP are simply
ignored, but conflicting ones that fragment the protocol are a
serious problem.

New attribute registrations are accepted according to the
"Specification Required" policy of RFC 2434, provided that the
specification includes the following information:

o   contact name, email address and telephone number

o   attribute-name (as it will appear in SDP)

o   long-form attribute name in English

o   type of attribute (session level, media level, or both)

o   whether the attribute value is subject to the charset attribute.

o   a one paragraph explanation of the purpose of the attribute.

   o a specification of appropriate attribute values for this
      attribute.

   The above is the minimum that IANA will accept. Attributes that are
   expected to see widespread use and interoperability, SHOULD be
   documented with a standards-track RFC that specifies the attribute
   more precisely.

   Submitters of registrations should ensure that the specification is
   in the spirit of SDP attributes, most notably that the attribute is
   platform independent in the sense that it makes no implicit
   assumptions about operating systems and does not name specific pieces
   of software in a manner that might inhibit interoperability.

   IANA is requested to register the following initial set of attribute
   names ("att-field" values), with definitions as in Section 6 of this
   memo (these definitions update those in RFC 2327):

Name	Session or Media level?	Dependent on charset?
cat	Session	No
keywds	Session	Yes
tool	Session	No
ptime	Media	No
maxptime	Media	No
rtpmap	Media	No
recvonly	Either	No
sendrecv	Either	No
sendonly	Either	No
inactive	Either	No
orient	Media	No
type	Session	No
charset	Session	No
sdplang	Either	No
lang	Either	No
framerate	Media	No
quality	Media	No
fmtp	Media	No

8.2.5 Bandwidth specifiers ("bwtype")

   A proliferation of bandwidth specifiers is strongly discouraged.

   New bandwidth specifiers ("bwtype" fields) MUST be registered with
   IANA. The submission MUST reference a standards-track RFC specifying
   the semantics of the bandwidth specifier precisely, and indicating
   when it should be used, and why the existing registered bandwidth

specifiers do not suffice.

IANA is requested to register the bandwith specifiers "CT" and "AS" with definitions as in Section 5.8 of this memo (these definitions update those in RFC 2327).

8.2.6 Network types ("nettype")

New network types (the "nettype" field) may be registered with IANA if SDP needs to be used in the context of non-Internet environments. Whilst these are not normally the preserve of IANA, there may be circumstances when an Internet application needs to interoperate with a non- Internet application, such as when gatewaying an Internet telephony call into the PSTN. The number of network types should be small and should be rarely extended. A new network type cannot be registered without registering at least one address type to be used with that network type. A new network type registration MUST reference an RFC which gives details of the network type and address type and specifies how and when they would be used.

IANA is requested to register the network type "IN" to represent the Internet, with definition as in Sections 5.2 and 5.7 of this memo (these definitions update those in RFC 2327).

8.2.7 Address types ("addrtype")

New address types ("addrtype") may be registered with IANA. An address type is only meaningful in the context of a network type, and any registration of an address type MUST specify a registered network type, or be submitted along with a network type registration. A new address type registration MUST reference an RFC giving details of the syntax of the address type. Address types are not expected to be registered frequently.

IANA is requested to register the address types "IP4" and "IP6" with definitions as in Sections 5.2 and 5.7 of this memo (these definitions update those in RFC 2327).

8.2.8 Registration Procedure

In the RFC documentation that registers SDP "media", "proto", "fmt", "bwtype", "nettype" and "addrtype" fields, the authors MUST include the following information for IANA to place in the appropriate registry:

o   contact name, email address and telephone number

o   name being registered (as it will appear in SDP)

   o long-form name in English

   o type of name ("media", "proto", "fmt", "bwtype", "nettype", or
      "addrtype")

   o a one paragraph explanation of the purpose of the registered name.

   o a reference to the specification for the registered name (this
      will typically be an RFC number).

   IANA may refer any registration to the IESG Transport Area Directors
   for review, and may request revisions to be made before a
   registration will be made.

8.3 Encryption Key Access Methods

   The IANA currently maintains a table of SDP encryption key access
   method ("enckey") names. This table is obsolete and SHOULD be
   removed, since the "k=" line is not extensible. New registrations
   MUST NOT be accepted.

9. SDP Grammar

   This section provides an Augmented BNF grammar for SDP. ABNF is
   defined in [4].

```
      ; SDP Syntax
      session-description = proto-version
                            origin-field
                            session-name-field
                            information-field
                            uri-field
                            email-fields
                            phone-fields
                            connection-field
                            bandwidth-fields
                            time-fields
                            key-field
                            attribute-fields
                            media-descriptions

      proto-version =       "v=" 1*DIGIT CRLF
                            ;this memo describes version 0

      origin-field =        "o=" username SP sess-id SP sess-version SP
                            nettype SP addrtype SP unicast-address CRLF

      session-name-field =  "s=" text CRLF
```

```
     information-field =   ["i=" text CRLF]

     uri-field =           ["u=" uri CRLF]

     email-fields =        *("e=" email-address CRLF)

     phone-fields =        *("p=" phone-number CRLF)

     connection-field =    ["c=" nettype SP addrtype SP
                           connection-address CRLF]
                           ;a connection field must be present
                           ;in every media description or at the
                           ;session-level

     bandwidth-fields =    *("b=" bwtype ":" bandwidth CRLF)

     time-fields =         1*( "t=" start-time SP stop-time
                           *(CRLF repeat-fields) CRLF)
                           [zone-adjustments CRLF]

     repeat-fields =       "r=" repeat-interval SP typed-time
                           1*(SP typed-time)

     zone-adjustments =    "z=" time SP ["-"] typed-time
                           *(SP time SP ["-"] typed-time)

     key-field =           ["k=" key-type CRLF]

     attribute-fields =    *("a=" attribute CRLF)

     media-descriptions =  *( media-field
                           information-field
                           *connection-field
                           bandwidth-fields
                           key-field
                           attribute-fields )

     media-field =         "m=" media SP port ["/" integer]
                           SP proto 1*(SP fmt) CRLF

     ; sub-rules of 'o='
     username =            non-ws-string
                           ;pretty wide definition, but doesn't
                           ;include space

     sess-id =             1*DIGIT
                           ;should be unique for this username/host
```

```
     sess-version =        1*DIGIT

     nettype =             token
                           ;typically "IN"

     addrtype =            token
                           ;typically "IP4" or "IP6"

     ; sub-rules of 'u='
     uri =                 URI-reference
                           ; see RFC2396 and RFC2732

     ; sub-rules of 'e='
     email-address =       email *SP "(" 1*email-safe ")" /
                           1*email-safe "<" email ">" /
                           email

     email =               addr-spec ; defined in RFC2822
                                     ; modified to remove CFWS

     ; sub-rules of 'p='
     phone-number =        phone *SP "(" 1*email-safe ")" /
                           1*email-safe "<" phone ">" /
                           phone

     phone =               ["+"] DIGIT 1*(SP / "-" / DIGIT)

     ; sub-rules of 'c='
     connection-address =  multicast-address / unicast-address

     ; sub-rules of 'b='
     bwtype =              token

     bandwidth =           1*DIGIT

     ; sub-rules of 't='
     start-time =          time / "0"

     stop-time =           time / "0"

     time =                POS-DIGIT 9*DIGIT
                           ; Decimal representation of NTP time in
                           ; seconds since 1900.  The representation
                           ; of NTP time is an unbounded length field
                           ; containing at least 10 digits.  Unlike the
                           ; 64-bit representation used elsewhere, time
                           ; in SDP does not wrap in the year 2036.
```

```
    ; sub-rules of 'r=' and 'z='
    repeat-interval =     POS-DIGIT *DIGIT [fixed-len-time-unit]

    typed-time =          1*DIGIT [fixed-len-time-unit]

    fixed-len-time-unit = "d" / "h" / "m" / "s"

    ; sub-rules of 'k='
    key-type =            "prompt" /
                          "clear:" text /
                          "base64:" base64 /
                          "uri:" uri

    base64      =         *base64-unit [base64-pad]
    base64-unit =         4base64-char
    base64-pad  =         2base64-char "==" / 3base64-char "="
    base64-char =         ALPHA / DIGIT / "+" / "/"

    ; sub-rules of 'a='
    attribute =           (att-field ":" att-value) / att-field

    att-field =           token

    att-value =           byte-string

    ; sub-rules of 'm='
    media =               token
                          ;typically "audio", "video", "text" or
                          ;"application"

    fmt =                 token
                          ;typically an RTP payload type for audio
                          ;and video media

    proto  =              token *("/" token)
                          ;typically "RTP/AVP" or "udp"

    port =                1*DIGIT

    ; generic sub-rules: addressing
    unicast-address =     IP4-address / IP6-address / FQDN / extn-addr

    multicast-address =   IP4-multicast / IP6-multicast / FQDN
                          / extn-addr

    IP4-multicast =       m1 3( "." decimal-uchar )
                          "/" ttl [ "/" integer ]
                          ; IPv4 multicast addresses may be in the
```

```
                              ; range 224.0.0.0 to 239.255.255.255

      m1 =                    ("22" ("4"/"5"/"6"/"7"/"8"/"9")) /
                              ("23" DIGIT )

      IP6-multicast =         hexpart [ "/" integer ]
                              ; IPv6 address starting with FF

      ttl =                   (POS-DIGIT *2DIGIT) / "0"

      FQDN =                  4*(alpha-numeric / "-" / ".")
                              ; fully qualified domain name as specified
                              ; in RFC1035 (and updates)

      IP4-address =           b1 3("." decimal-uchar)

      b1 =                    decimal-uchar
                              ; less than "224"

      ; The following is from RFC2373 Appendix B.  It is a direct copy.
      IP6-address =           hexpart [ ":" IP4-address ]

      hexpart =               hexseq / hexseq "::" [ hexseq ] /
                              "::" [ hexseq ]

      hexseq  =               hex4 *( ":" hex4)

      hex4    =               1*4HEXDIG

      ; Generic for other address families
      extn-addr =      non-ws-string

      ; generic sub-rules: datatypes
      text =                  byte-string
                              ;default is to interpret this as UTF8 text.
                              ;ISO 8859-1 requires "a=charset:ISO-8859-1"
                              ;session-level attribute to be used

      byte-string =           1*(%x01-09/%x0B-0C/%x0E-FF)
                              ;any byte except NUL, CR or LF

      non-ws-string =         1*(VCHAR/%x80-FF)
                              ;string of visible characters

      token-char =            %x21 / %x23-27 / %x2A-2B / %x2D-2E / %x30-39
                              / %x41-5A / %x5E-7E

      token =            1*(token-char)
```

```
    email-safe =           %x01-09/%x0B-0C/%x0E-27/%x2A-3B/%x3D/%x3F-FF
                           ;any byte except NUL, CR, LF, or the quoting
                           ;characters ()<>

    integer =              POS-DIGIT *DIGIT

    ; generic sub-rules: primitives
    alpha-numeric =        ALPHA / DIGIT

    POS-DIGIT =            %x31-39 ; 1 - 9

    decimal-uchar =        DIGIT
                           / POS-DIGIT DIGIT
                           / ("1" 2*(DIGIT))
                           / ("2" ("0"/"1"/"2"/"3"/"4") DIGIT)
                           / ("2" "5" ("0"/"1"/"2"/"3"/"4"/"5"))

    ; external references:
    ; ALPHA, DIGIT, CRLF, SP, VCHAR: from RFC 2234
    ; URI-reference: from RFC2396 and RFC2732
    ; addr-spec: from RFC 2822
```

10. Summary of Changes from RFC 2327

   The memo has been significantly restructured, incorporating a large
   number of clarifications to the specification in light of use. With
   the exception of those items noted below, the changes to the memo are
   intended to be backwards compatible clarifications. However, due to
   inconsistencies and unclear definitions in RFC 2327 it is likely that
   some implementations interpreted that memo in ways that differ from
   this version of SDP.

   The ABNF grammar in Section 9 has been extensively revised and
   updated, correcting a number of mistakes and incorporating the RFC
   3266 IPv6 extensions. Known inconsistencies between the grammar and
   the specification text have been resolved.

   A media type registration for SDP is included. Requirements for the
   registration of attributes and other parameters with IANA have been
   clarified and tightened (Section 8). It is noted that "text" and
   "message" are valid media types for use with SDP, but that "control"
   and "data" are under-specified and deprecated.

   RFC 2119 terms are now used throughout to specify requirements
   levels. Certain of those requirements, in particular in relation to
   parameter registration, are stricter than those in RFC 2327.

The "RTP/SAVP" RTP profile and its "fmt" namespace are registered.

The attributes "a=inactive" and "a=maxptime" have been added.

RFC 2327 mandated that either "e=" or "p=" was required. Both are
now optional, to reflect actual usage.

The significant limitations of the "k=" field are noted, and its use
is deprecated.

Most uses of the "x-" prefix notation for experimental parameters are
disallowed and the other uses are deprecated.

11. Acknowledgements

Many people in the IETF Multiparty Multimedia Session Control
(MMUSIC) working group have made comments and suggestions
contributing to this document. In particular, we would like to thank
Eve Schooler, Steve Casner, Bill Fenner, Allison Mankin, Ross
Finlayson, Peter Parnes, Joerg Ott, Carsten Bormann, Steve Hanna,
Jonathan Lennox, Keith Drage, Sean Olson, Bernie Hoeneisen and
Jonathan Rosenberg.

12. References

12.1 Normative References

   [1]   Mockapetris, P., "Domain names - concepts and facilities", STD
          13, RFC 1034, November 1987.

   [2]   Mockapetris, P., "Domain names - implementation and
          specification", STD 13, RFC 1035, November 1987.

   [3]   Bradner, S., "Key words for use in RFCs to Indicate Requirement
          Levels", BCP 14, RFC 2119, March 1997.

   [4]   Crocker, D. and P. Overell, "Augmented BNF for Syntax
          Specifications: ABNF", RFC 2234, November 1997.

   [5]   Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC
          2279, January 1998.

   [6]   Handley, M. and V. Jacobson, "SDP: Session Description
          Protocol", RFC 2327, April 1998.

   [7]   Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform
          Resource Identifiers (URI): Generic Syntax", RFC 2396, August
          1998.

   [8]   Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA
         Considerations Section in RFCs", BCP 26, RFC 2434, October
         1998.

   [9]   Hinden, R., Carpenter, B. and L. Masinter, "Format for Literal
         IPv6 Addresses in URL's", RFC 2732, December 1999.

   [10] Alvestrand, H., "Tags for the Identification of Languages", BCP
         47, RFC 3066, January 2001.

   [11] Faltstrom, P., Hoffman, P. and A. Costello, "Internationalizing
         Domain Names in Applications (IDNA)", RFC 3490, March 2003.

   [12] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings",
         RFC 3548, July 2003.

12.2 Informative References

   [13] Mills, D., "Network Time Protocol (Version 3) Specification,
         Implementation", RFC 1305, March 1992.

   [14] Handley, M., Perkins, C. and E. Whelan, "Session Announcement
         Protocol", RFC 2974, October 2000.

   [15] 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.

   [16] Schulzrinne, H., Rao, A. and R. Lanphier, "Real Time Streaming
         Protocol (RTSP)", RFC 2326, April 1998.

   [17] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with
         Session Description Protocol (SDP)", RFC 3264, June 2002.

   [18] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson,
         "RTP: A Transport Protocol for Real-Time Applications", STD 64,
         RFC 3550, July 2003.

   [19] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video
         Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

   [20] Casner, S., "Session Description Protocol (SDP) Bandwidth
         Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556,
         July 2003.

   [21] Huitema, C., "Real Time Control Protocol (RTCP) attribute in
         Session Description Protocol (SDP)", RFC 3605, October 2003.

   [22] Baugher, M., McGrew, D., Naslund, M., Carrara, E. and K.
         Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC
         3711, March 2004.

   [23] Rosenberg, J., Schulzrinne, H. and P. Kyzivat, "Indicating User
         Agent Capabilities in the Session Initiation Protocol (SIP)",
         RFC 3840, August 2004.

   [24] Westerlund, M., "A Transport Independent Bandwidth Modifier for
         the Session Description Protocol (SDP)", RFC 3890, September
         2004.

   [25] International Telecommunications Union, "H.323 extended for
         loosely coupled conferences", ITU Recommendation H.332,
         September 1998.

   [26] Arkko, J., Carrara, E., Lindholm, F., Naslund, M. and K.
         Norrman, "Key Management Extensions for Session Description
         Protocol (SDP) and Real Time Streaming Protocol (RTSP)",
         draft-ietf-mmusic-kmgmt-ext-12 (work in progress), November
         2004.

   [27] Andreasen, F., Baugher, M. and D. Wing, "Session Description
         Protocol Security Descriptions for Media Streams",
         draft-ietf-mmusic-sdescriptions-07 (work in progress), July
         2004.

Authors' Addresses

   Mark Handley
   University College London
   Department of Computer Science
   Gower Street
   London WC1E 6BT
   UK

   EMail: M.Handley@cs.ucl.ac.uk

   Van Jacobson
   Packet Design
   2465 Latham Street
   Mountain View, CA 94040
   USA

   EMail: van@packetdesign.com

Colin Perkins
University of Glasgow
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ
UK

EMail: csp@csperkins.org

Intellectual Property Statement

Acknowledgment