Stream:	Internet Engineering Task Force (IETF)
RFC:	8866
Obsoletes:	4566
Category:	Standards Track
Published:	January 2021
ISSN:	2070-1721

Authors:	A. Begen	P. Kyzivat	C. Perkins	M. Handley
	Networked Media		*University of Glasgow*	*UCL*

RFC 8866
SDP: Session Description Protocol

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. This document obsoletes RFC 4566.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8866.

Copyright Notice

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 it is intended to use different transport protocols as appropriate, including the Session Announcement Protocol (SAP) [RFC2974], Session Initiation Protocol (SIP) [RFC3261], Real-Time Streaming Protocol (RTSP) [RFC7826], electronic mail [RFC5322] using the MIME extensions [RFC2045], and the Hypertext Transport Protocol (HTTP) [RFC7230].

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 obsoletes [RFC4566]. The changes relative to [RFC4566] are outlined in Section 10 of this memo.

2. Glossary of Terms

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

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

Media Description: A Media Description contains the information needed for one party to establish an application-layer network protocol connection to another party. It starts with an "m=" line and is terminated by either the next "m=" line or by the end of the session description.

Session-Level Section: This refers to the parts that are not media descriptions, whereas the session description refers to the whole body that includes the session-level section and the media description(s).

The terms "multimedia conference" and "multimedia session" are used in this document as defined in [RFC7656]. The terms "session" and "multimedia session" are used interchangeably in this document.

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Examples of SDP Usage

3.1. Session Initiation

The Session Initiation Protocol (SIP) [RFC3261] 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 that allow participants to agree on a set of compatible media types [RFC6838]. These session descriptions are commonly formatted using SDP. When used with SIP, the offer/answer model [RFC3264] provides a limited framework for negotiation using SDP.

3.2. Streaming Media

The Real-Time Streaming Protocol (RTSP) [RFC7826], 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.

3.3. Email and the World Wide Web

Alternative means of conveying session descriptions include electronic mail and the World Wide Web (WWW). For both email and WWW distribution, the media 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 descriptions of multicast sessions sent only via email or the WWW do not have the property that the receiver of a session description 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 possibly outside this scope.

3.4. 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 used to implement such a distributed directory is the SAP [RFC2974]. SDP provides the recommended session description format for such session announcements.

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 with Internet protocols, although it is sufficiently general that it can describe multimedia conferences in other network environments. Media streams can be many-to-many. Sessions need not be continually active.

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 it 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 description includes the following:

- Session name and purpose
- Time(s) the session is active
- The media comprising the session
- Information needed to receive those media (addresses, ports, formats, etc.)

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

- Information about the bandwidth to be used by the session
- 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 nonparticipants 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 description includes the following media information:

- The type of media (video, audio, etc.)
- The media transport protocol (RTP/UDP/IP, H.320, etc.)
- 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:

- The multicast group address for media
- 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:

- The remote address for media
- The remote transport port for media

The semantics of the address and port depend on context. Typically, this **SHOULD** be the remote address and remote port to which media is to be sent or received. Details may differ based on the network type, address type, protocol, and media specified, and whether the SDP is being distributed as an advertisement or negotiated in an offer/answer [RFC3264] exchange. (E.g., Some address types or protocols may not have a notion of port.) Deviating from typical behavior should be done cautiously since this complicates implementations (including middleboxes that must parse the addresses to open Network Address Translation (NAT) or firewall pinholes).

4.2. Timing Information

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

- An arbitrary list of start and stop times bounding the session
- 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 (see Section 5.9).

4.3. Obtaining Further Information about a Session

A session description could convey enough information to decide whether or not to participate in a session. SDP may include additional pointers in the form of Uniform Resource Identifiers (URIs) [RFC3986] for more information about the session. (Note that use of URIs to indicate remote resources is subject to the security considerations from [RFC3986].)

4.4. Internationalization

The SDP specification recommends the use of the ISO 10646 character set in the UTF-8 encoding [RFC3629] to allow many different languages to be represented. However, to assist in compact representations, SDP also allows other character sets such as [ISO.8859-1.1998] to be used when desired. Internationalization only applies to free-text subfields (session name and background information), and not to SDP as a whole.

5. SDP Specification

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

An SDP description is entirely textual. SDP field names and attribute names use only the US-ASCII subset of UTF-8 [RFC3629], but textual fields and attribute values **MAY** use the full ISO 10646 character set in UTF-8 encoding, or some other character set defined by the "a=charset:" attribute (Section 6.10). Field and attribute values that use the full UTF-8 character set are never directly compared, hence there is no requirement for UTF-8 normalization. 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 descriptions 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 descriptions that could be detected easily and discarded.

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

```
<type>=<value>
```

where <type> is exactly one case-significant character and <value> is structured text whose format depends on <type>. In general, <value> is either a number of subfields delimited by a single space character or a free format string, and is case-significant unless a specific field defines otherwise. Whitespace separators are not used on either side of the "=" sign, however, the value can contain a leading whitespace as part of its syntax, i.e., that whitespace is part of the value.

An SDP description **MUST** conform to the syntax defined in Section 9. The following is an overview of the syntax.

An SDP description consists of a session-level section followed by zero or more media descriptions. The session-level section starts with a "v=" line and continues to the first media description (or the end of the whole description, whichever comes first). Each media description starts with an "m=" line and continues to the next media description or the end of the whole session description, whichever comes first. 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 when present, they must appear in exactly the order given here. (The fixed order greatly enhances error detection and allows for a simple parser). In the following overview, optional items are marked with a "*".

```
Session description
   v=  (protocol version)
   o=  (originator 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 descriptions)
   b=* (zero or more bandwidth information lines)
   One or more time descriptions:
     ("t=", "r=" and "z=" lines; see below)
   k=* (obsolete)
   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)
   z=* (optional time zone offset line)

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=* (obsolete)
   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 or reject any session description that contains a type letter that it does not understand. The attribute mechanism ("a=", described in Section 5.13) is the primary means for extending SDP and tailoring it to particular applications or media. Some attributes (the ones listed in Section 6) have a defined meaning, but others may be added on a media- or session-

specific basis. (Attribute scopes in addition to media-specific and session-specific scopes may also be defined in extensions to this document, e.g., [RFC5576] and [RFC8864].) An SDP parser **MUST** ignore any attribute it doesn't understand.

An SDP description may contain URIs that 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=") information in the session-level section applies to all the media descriptions of that session unless overridden by connection information in the media description. For instance, in the example below, each audio media description behaves as if it were given a "c=IN IP4 198.51.100.1".

An example SDP description is:

```
v=0
o=jdoe 3724394400 3724394405 IN IP4 198.51.100.1
s=Call to John Smith
i=SDP Offer #1
u=http://www.jdoe.example.com/home.html
e=Jane Doe <jane@jdoe.example.com>
p=+1 617 555-6011
c=IN IP4 198.51.100.1
t=0 0
m=audio 49170 RTP/AVP 0
m=audio 49180 RTP/AVP 0
m=video 51372 RTP/AVP 99
c=IN IP6 2001:db8::2
a=rtpmap:99 h263-1998/90000
```

Text-containing fields such as the session-name-field and information-field are octet strings that 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 line, although parsers **SHOULD** be tolerant and also accept lines 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. When the "a=charset:" attribute is present the session-name-field, information-field, and some attribute fields are interpreted according to the selected character set.

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 [RFC1034] and [RFC1035]. Internationalized domain names (IDNs) **MUST** be represented using the ASCII Compatible Encoding (ACE) form defined in [RFC5890] and **MUST NOT** be directly represented in UTF-8 or any other encoding (this requirement is for compatibility with [RFC2327] and other early SDP-related standards, which predate the development of internationalized domain names).

5.1. Protocol Version ("v=")

```
v=0
```

The "v=" line (version-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=" line (origin-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> forms a globally unique identifier for the session. The method of <sess-id> allocation is up to the creating tool, but a timestamp, in seconds since January 1, 1900 UTC, is recommended to ensure uniqueness.

<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 description. Again, as with <sess-id> it is **RECOMMENDED** that a timestamp be 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 the 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 the future (see Section 8).

<unicast-address> is an address of the machine from which the session was created. For an address type of "IP4", this is either a fully qualified domain name of the machine or the dotted-decimal representation of an IP version 4 address of the machine. For an address type of "IP6", this is either a fully qualified domain name of the machine or the address of the machine represented as specified in Section 4 of [RFC5952]. For both "IP4" and "IP6", the fully qualified domain name is the form that **SHOULD** be given unless this is unavailable, in which case a globally unique address **MAY** be substituted.

In general, the "o=" line serves as a globally unique identifier for this version of the 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=" line, provided that 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=" line (session-name-field) is the textual session name. There **MUST** be one and only one "s=" line per session description. The "s=" line **MUST NOT** be empty. If a session has no meaningful name, then "s= " or "s=-" (i.e., a single space or dash as the session name) is **RECOMMENDED**. If a session-level "a=charset:" attribute is present, it specifies the character set used in the "s=" field. If a session-level "a=charset:" attribute is not present, the "s=" field **MUST** contain ISO 10646 characters in UTF-8 encoding.

5.4. Session Information ("i=")

```
i=<session information>
```

The "i=" line (information-field) provides textual information about the session. There can be at most one session-level "i=" line per session description, and at most one "i=" line in each media description. Unless a media-level "i=" line is provided, the session-level "i=" line applies to that media description. If the "a=charset:" attribute is present, it specifies the character set used in the "i=" line. If the "a=charset:" attribute is not present, the "i=" line **MUST** contain ISO 10646 characters in UTF-8 encoding.

At most one "i=" line can be used for each media description. In media definitions, "i=" lines are primarily intended for labeling 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=" line 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>
```

The "u=" line (uri-field) provides a URI (Uniform Resource Identifier) [RFC3986]. The URI should be a pointer to additional human readable information about the session. This line is **OPTIONAL**. No more than one "u=" line is allowed per session description.

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

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

The "e=" line (email-field) and "p=" line (phone-field) specify contact information for the person responsible for the session. This is not necessarily the same person that created the session description.

Inclusion of an email address or phone number is **OPTIONAL**.

If an email address or phone number is present, it **MUST** be specified before the first media description. More than one email or phone line 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 [E164]) 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 parentheses if it is present. For example:

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

The alternative [RFC5322] 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 Information ("c=")

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

The "c=" line (connection-field) contains information necessary to establish a network connection.

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

The first subfield (<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 subfield (<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 subfield (<connection-address>) is the connection address. Additional subfields **MAY** be added after the connection address depending on the value of the <addrtype> subfield.

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

- 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, data relay, or data sink as determined by additional attribute-fields (Section 5.13). 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.
- Sessions using an "IP4" 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 session will be sent. TTL values **MUST** be in the range 0-255. Although 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 233.252.0.1/127
```

"IP6" multicast does not use TTL scoping, and hence the TTL value **MUST NOT** be present for "IP6" multicast. It is expected that IPv6 scoped addresses will be used to limit the scope of multimedia 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 233.252.0.1/127/3
```

would state that addresses 233.252.0.1, 233.252.0.2, and 233.252.0.3 are to be used with a TTL of 127. This is semantically identical to including multiple "c=" lines in a media description:

```
c=IN IP4 233.252.0.1/127
c=IN IP4 233.252.0.2/127
c=IN IP4 233.252.0.3/127
```

Similarly, an IPv6 example would be:

```
c=IN IP6 ff00::db8:0:101/3
```

which is semantically equivalent to:

```
c=IN IP6 ff00::db8:0:101
c=IN IP6 ff00::db8:0:102
c=IN IP6 ff00::db8:0:103
```

(remember that the TTL subfield is not present in "IP6" multicast).

Multiple addresses or "c=" lines **MAY** be specified on a per media description basis only if they provide multicast addresses for different layers in a hierarchical or layered encoding scheme. Multiple addresses or "c=" lines **MUST NOT** be specified at session level.

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

5.8. Bandwidth Information ("b=")

```
b=<bwtype>:<bandwidth>
```

The **OPTIONAL** "b=" line (bandwidth-field) denotes the proposed bandwidth to be used by the session or media description. The <bwtype> is an alphanumeric modifier that provides the meaning of the <bandwidth> number. Two values are defined in this specification, but other values **MAY** be registered in the future (see Section 8 and [RFC3556], [RFC3890]):

CT If the bandwidth of 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). Similarly, if the bandwidth of bundled media streams [RFC8843] in an "m=" line is different from the implicit value from the scope, a "b=CT:" line **SHOULD** be supplied in the media level. The primary purpose of this is to give an approximate idea as to whether two or more sessions (or bundled media streams) can coexist simultaneously. Note that a "b=CT:" line gives a total bandwidth figure for all the media at all endpoints.

The Mux Category for "b=CT:" is NORMAL. This is discussed in [RFC8859].

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, the "b=AS:" line gives the RTP "session bandwidth" as defined in Section 6.2 of [RFC3550]. Note that a "b=AS:" line gives a bandwidth figure for a single media at a single endpoint, although there may be many endpoints sending simultaneously.

The Mux Category for "b=AS:" is SUM. This is discussed in [RFC8859].

[RFC4566] defined an "X-" prefix for <bwtype> names. This was intended for experimental purposes only. For example:

```
b=X-YZ:128
```

Use of the "X-" prefix is **NOT RECOMMENDED**. Instead new (non "X-" prefix) <bwtype> names **SHOULD** be defined, and then **MUST** be registered with IANA in the standard namespace. SDP parsers **MUST** ignore bandwidth-fields with unknown <bwtype> names. The <bwtype> names **MUST** be alphanumeric and, although no length limit is given, it is recommended that they be short.

The <bandwidth> is interpreted as kilobits per second by default (including the transport and network-layer, but not the link-layer, overhead). 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. Time Active ("t=")

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

A "t=" line (time-field) begins a time description that specifies the start and stop times for a session. Multiple time descriptions **MAY** be used if a session is active at multiple irregularly spaced times; each additional time description specifies additional periods of time for which the session will be active. If the session is active at regular repeat times, a repeat description, begun by an "r=" line (see Section 5.10) can be included following the time-field -- in which case the time-field specifies the start and stop times of the entire repeat sequence.

The following example specifies two active intervals:

```
t=3724394400 3724398000 ; Mon 8-Jan-2018 10:00-11:00 UTC
t=3724484400 3724488000 ; Tue 9-Jan-2018 11:00-12:00 UTC
```

The first and second subfields of the time-field give the start and stop times, respectively, for the session. These are the decimal representation of time values in seconds since January 1, 1900 UTC. To convert these values to Unix time (UTC), subtract decimal 2208988800.

Some time representations will wrap in the year 2036. Because SDP uses an arbitrary length decimal representation, it does not have this issue. Implementations of SDP need to be prepared to handle these larger values.

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 behavior other than this is required, a <stop-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 that state precisely when the session will be active.

5.10. Repeat Times ("r=")

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

An"r=" line (repeat-field) specifies repeat times for a session. If needed to express complex schedules, multiple repeat-fields may be included. 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=" line would be the 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=" line stop time would be the representation of the end of the last session three months later. By default, all subfields are in seconds, so the "r=" and "t=" lines might be the following:

```
t=3724394400 3730536000 ; Mon 8-Jan-2018 10:00-11:00 UTC
                        ; Tues 20-Mar-2018 12:00 UTC
r=604800 3600 0 90000   ; 1 week, 1 hour, zero, 25 hours
```

To make the 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)

Table 1: Time Unit Specification Characters

Thus, the above repeat-field 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 time-descriptions should be used to explicitly list the session times.

5.11. Time Zone Adjustment ("z=")

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

A "z=" line (zone-field) is an optional modifier to the repeat-fields it immediately follows. It does not apply to any other fields.

To schedule a repeated session that 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 saving 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 time (represented as seconds since January

1, 1900 UTC) that a time zone adjustment happens and the offset from the time when the session was first scheduled. The "z=" line allows the sender to specify a list of these adjustment times and offsets from the base time.

An example might be the following:

```
t=3724394400 3754123200       ; Mon 8-Jan-2018 10:00 UTC
                              ; Tues 18-Dec-2018 12:00 UTC
r=604800 3600 0 90000         ; 1 week, 1 hour, zero, 25 hours
z=3730928400 -1h 3749680800 0 ; Sun 25-Mar-2018 1:00 UTC,
                              ; offset 1 hour,
                              ; Sun 28-Oct-2018 2:00 UTC,
                              ; no offset
```

This specifies that at time 3730928400 (Sun 25-Mar-2018 1:00 UTC, the onset of British Summer Time) the time base by which the session's repeat times are calculated is shifted back by 1 hour, and that at time 3749680800 (Sun 28-Oct-2018 2:00 UTC, the end of British Summer Time) the session's original time base is restored. Adjustments are always relative to the specified start time -- they are not cumulative.

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

5.12. Encryption Keys ("k=")

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

The "k=" line (key-field) is obsolete and **MUST NOT** be used. It is included in this document for legacy reasons. One **MUST NOT** include a "k=" line in an SDP, and **MUST** discard it if it is received in an SDP.

5.13. Attributes ("a=")

```
a=<attribute-name>
a=<attribute-name>:<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. (Attribute scopes in addition to media-level and session-level scopes may also be defined in extensions to this document, e.g., [RFC5576] and [RFC8864].)

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

Attribute-fields may be of two forms:

- A property attribute is simply of the form "a=<attribute-name>". 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".
- A value attribute is of the form "a=<attribute-name>:<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 "a=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=" line (media-field) and is terminated by either the next "m=" line or by the end of the session description. A media-field has several subfields:

<media> is the media type. This document defines the values "audio", "video", "text", "application", and "message". This list is extended by other memos and may be further extended by additional memos registering media types in the future (see Section 8). For example, [RFC6466] defined the "image" media type.

<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=" line, and on the transport protocol defined in the <proto> subfield of the media-field. Other ports used by the

media application (such as the RTP Control Protocol (RTCP) port [RFC3550]) **MAY** be derived algorithmically from the base media port or **MAY** be specified in a separate attribute (for example, the "a=rtcp:" attribute as defined in [RFC3605]).

If noncontiguous ports are used or if they don't follow the parity rule of even RTP ports and odd RTCP ports, the "a=rtcp:" attribute **MUST** be used. Applications that are requested to send media to a <port> that is odd and where the "a=rtcp:" attribute is present **MUST NOT** subtract 1 from the RTP port: that is, they **MUST** send the RTP to the port indicated in <port> and send the RTCP to the port indicated in the "a=rtcp:" attribute.

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=" line:

```
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 the description of <fmt> below).

This document does not include a mechanism for declaring hierarchically encoded streams using noncontiguous ports. (There is currently no attribute defined that can accomplish this. The "a=rtcp:" attribute defined in [RFC3605] does not handle hierarchical encoding.) If a need arises to declare noncontiguous ports then it will be necessary to define a new attribute to do so.

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

```
m=video 49170/2 RTP/AVP 31
c=IN IP4 233.252.0.1/127/2
```

would imply that address 233.252.0.1 is used with ports 49170 and 49171, and address 233.252.0.2 is used with ports 49172 and 49173.

The mapping is similar if multiple addresses are specified using multiple "c=" lines. For example:

```
m=video 49170/2 RTP/AVP 31
c=IN IP6 ff00::db8:0:101
c=IN IP6 ff00::db8:0:102
```

would imply that address ff00::db8:0:101 is used with ports 49170 and 49171, and address ff00::db8:0:102 is used with ports 49172 and 49173.

This document gives no meaning to assigning the same media address to multiple media descriptions. Doing so does not implicitly group those media descriptions in any way. An explicit grouping framework (for example, [RFC5888]) should instead be used to express the intended semantics. For instance, see [RFC8843].

<proto> is the transport protocol. The meaning of the transport protocol is dependent on the address type subfield in the relevant "c=" line. Thus a "c=" line with an address type of "IP4" indicates that the transport protocol runs over IPv4. The following transport protocols are defined, but may be extended through registration of new protocols with IANA (see Section 8):

- udp: denotes that the data is transported directly in UDP with no additional framing.
- RTP/AVP: denotes RTP [RFC3550] used under the RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551] running over UDP.
- RTP/SAVP: denotes the Secure Real-time Transport Protocol [RFC3711] running over UDP.
- RTP/SAVPF: denotes SRTP with the Extended SRTP Profile for RTCP-Based Feedback [RFC5124] 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 (MBone's popular multimedia audio tool) Pulse Code Modulation (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 subfields describe the format of the media. The interpretation of the media format depends on the value of the <proto> subfield.

If the <proto> subfield is "RTP/AVP" or "RTP/SAVP", the <fmt> subfields 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, and these payload formats are listed in order of preference, with the first format listed being preferred. When multiple payload formats are listed, the first acceptable payload format from the beginning of the list **SHOULD** be used for the session. For dynamic payload type assignments, the "a=rtpmap:" attribute (see Section 6.6)

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 6.15).

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

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

Section 3 of [RFC4855] states that the payload format (encoding) names defined in the RTP profile are commonly shown in upper case, while media subtype names are commonly shown in lower case. It also states that both of these names are case-insensitive in both places, similar to parameter names which are case-insensitive both in media type strings and in the default mapping to the SDP "a=fmtp:" attribute.

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. Syntax is provided using ABNF [RFC7405] with some of the rules defined further in Section 9.

6.1. cat (Category)

Name: cat

Value: cat-value

Usage Level: session

Charset Dependent: no

Syntax:

```
cat-value = category
category = non-ws-string
```

Example:

```
a=cat:foo.bar
```

This attribute gives the dot-separated hierarchical category of the session. This is to enable a receiver to filter unwanted sessions by category. There is no central registry of categories. This attribute is obsolete and **SHOULD NOT** be used. It **SHOULD** be ignored if received.

6.2. keywds (Keywords)

Name: keywds

Value: keywds-value

Usage Level: session

Charset Dependent: yes

Syntax:

```
keywds-value = keywords
keywords = text
```

Example:

```
a=keywds:SDP session description protocol
```

Like the "a=cat:" attribute, this was intended to assist identifying wanted sessions at the receiver, and to allow a receiver to select interesting sessions based on keywords describing the purpose of the session; however, there is no central registry of keywords. 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. This attribute is obsolete and **SHOULD NOT** be used. It **SHOULD** be ignored if received.

6.3. tool

Name: tool

Value: tool-value

Usage Level: session

Charset Dependent: no

Syntax:

```
tool-value = tool-name-and-version
tool-name-and-version = text
```

Example:

```
a=tool:foobar V3.2
```

This gives the name and version number of the tool used to create the session description.

6.4. ptime (Packet Time)

Name: ptime

Value: ptime-value

Usage Level: media

Charset Dependent: no

Syntax:

```
ptime-value = non-zero-int-or-real
```

Example:

```
a=ptime:20
```

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 "a=ptime:" to decode RTP or vat audio, and it is intended as a recommendation for the encoding/packetization of audio.

6.5. maxptime (Maximum Packet Time)

Name: maxptime

Value: maxptime-value

Usage Level: media

Charset Dependent: no

Syntax:

```
maxptime-value = non-zero-int-or-real
```

Example:

```
a=maxptime:20
```

This gives the maximum amount of media that 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. Note that this attribute was introduced after [RFC2327], and implementations that have not been updated will ignore this attribute.

6.6. rtpmap

Name: rtpmap

Value: rtpmap-value

Usage Level: media

Charset Dependent: no

Syntax:

```
rtpmap-value = payload-type SP encoding-name
  "/" clock-rate [ "/" encoding-params ]
payload-type = zero-based-integer
encoding-name = token
clock-rate = integer
encoding-params = channels
channels = integer
```

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. Note that the payload type number is indicated in a 7-bit field, limiting the values to inclusively between 0 and 127.

Although an RTP profile can 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 encoded single-channel audio sampled at 8 kHz. 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 "a=rtpmap:" attribute can be defined for each media format specified. Thus, we might have the following:

```
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 media subtypes 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-params indicates the number of audio channels. This parameter is **OPTIONAL** and may be omitted if the number of channels is one, provided that 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 [RFC3551]) packetization is required, the "a=ptime:" attribute is used as given in Section 6.4.

6.7. Media Direction Attributes

At most one occurrence of "a=recvonly", "a=sendrecv", "a=sendonly", or "a=inactive" **MAY** appear at session level, and at most one **MAY** appear in each media description.

If any one of these appears in a media description, then it applies for that media description. If none appears in a media description, then the one from session level, if any, applies to that media description.

If none of the media direction attributes is present at either session level or media level, "a=sendrecv" **SHOULD** be assumed as the default.

Within the following SDP example, the "a=sendrecv" attribute applies to the first audio media and the "a=inactive" attribute applies to the others.

```
v=0
o=jdoe 3724395000 3724395001 IN IP6 2001:db8::1
s=-
c=IN IP6 2001:db8::1
t=0 0
a=inactive
m=audio 49170 RTP/AVP 0
a=sendrecv
m=audio 49180 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

6.7.1. recvonly (Receive-Only)

Name: recvonly

Value:

Usage Level: session, media

Charset Dependent: no

Example:

```
a=recvonly
```

This specifies that the tools should be started in receive-only mode where applicable. Note that receive-only mode applies to the media only, not to any associated control protocol. An RTP-based system in receive-only mode **MUST** still send RTCP packets as described in [RFC3550], Section 6.

6.7.2. sendrecv (Send-Receive)

Name: sendrecv

Value:

Usage Level: session, media

Charset Dependent: no

Example:

```
a=sendrecv
```

This specifies that the tools should be started in send and receive mode. This is necessary for interactive multimedia conferences with tools that default to receive-only mode.

6.7.3. sendonly (Send-Only)

Name: sendonly

Value:

Usage Level: session, media

Charset Dependent: no

Example:

```
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 in send-only mode and one in receive-vonly mode. Note that send-only mode applies only to the media, and any associated control protocol (e.g., RTCP) **SHOULD** still be received and processed as normal.

6.7.4. inactive

Name: inactive

Value:

Usage Level: session, media

Charset Dependent: no

Example:

```
a=inactive
```

This specifies that the tools should be started in inactive mode. This is necessary for interactive multimedia conferences where users can put other users on hold. No media is sent over an inactive media stream. Note that an RTP-based system **MUST** still send RTCP (if RTCP is used), even if started in inactive mode.

6.8. orient (Orientation)

Name: orient

Value: orient-value

Usage Level: media

Charset Dependent: no

Syntax:

```
orient-value = portrait / landscape / seascape
portrait  = %s"portrait"
landscape = %s"landscape"
seascape  = %s"seascape"
   ; NOTE: These names are case-sensitive.
```

Example:

```
a=orient:portrait
```

Normally this is only used for a whiteboard or presentation tool. It specifies the orientation of the workspace on the screen. Permitted values are "portrait", "landscape", and "seascape" (upside-down landscape).

6.9. type (Conference Type)

Name: type

Value: type-value

Usage Level: session

Charset Dependent: no

Syntax:

```
type-value = conference-type
conference-type = broadcast / meeting / moderated / test /
                  H332
broadcast = %s"broadcast"
meeting   = %s"meeting"
moderated = %s"moderated"
test      = %s"test"
H332      = %s"H332"
   ; NOTE: These names are case-sensitive.
```

Example:

```
a=type:moderated
```

This specifies the type of the multimedia conference. Allowed values are "broadcast", "meeting", "moderated", "test", and "H332". These values have implications for other options that are likely to be appropriate:

- When "a=type:broadcast" is specified, "a=recvonly" is probably appropriate for those connecting.
- When "a=type:meeting" is specified, "a=sendrecv" is likely to be appropriate.
- "a=type:moderated" suggests the use of a floor control tool and that the media tools be started so as to mute new sites joining the multimedia conference.
- Specifying "a=type:H332" indicates that this loosely coupled session is part of an H.332 session as defined in the ITU H.332 specification [ITU.H332.1998]. Media tools should be started using "a=recvonly".
- Specifying "a=type:test" is suggested as a hint that, unless explicitly requested otherwise, receivers can safely avoid displaying this session description to users.

6.10. charset (Character Set)

Name: charset

Value: charset-value

Usage Level: session

Charset Dependent: no

Syntax:

```
charset-value = <defined in [RFC2978]>
```

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
```

The charset specified **MUST** be one of those registered in the IANA Character Sets registry (http://www.iana.org/assignments/character-sets), such as ISO-8859-1. The character set identifier is a string that **MUST** be compared against identifiers from the "Name" or "Preferred MIME Name" field of the registry using a case-insensitive comparison. If the identifier is not recognized or not supported, all strings that are affected by it **SHOULD** be regarded as octet strings.

Charset-dependent fields **MUST** contain only sequences of bytes that are valid according to the definition of the selected character set. Furthermore, charset-dependent fields **MUST NOT** contain the bytes 0x00 (Nul), 0x0A (LF), and 0x0d (CR).

6.11. sdplang (SDP Language)

Name: sdplang

Value: sdplang-value

Usage Level: session, media

Charset Dependent: no

Syntax:

```
sdplang-value = Language-Tag
; Language-Tag defined in RFC 5646
```

Example:

```
a=sdplang:fr
```

Multiple "a=sdplang:" attributes can be provided either at session or media level if the session description or media use multiple languages.

As a session-level attribute, it specifies the language for the session description (not the language of the media). As a media-level attribute, it specifies the language for any media-level SDP information-field associated with that media (again not the language of the media), overriding any "a=sdplang:" attributes specified at session level.

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

The "a=sdplang:" attribute value must be a single language tag [RFC5646]. An "a=sdplang:" attribute **SHOULD** be specified when a session is distributed with 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.

6.12. lang (Language)

Name: lang

Value: lang-value

Usage Level: session, media

Charset Dependent: no

Syntax:

```
lang-value = Language-Tag
; Language-Tag defined in RFC 5646
```

Example:

```
a=lang:de
```

Multiple "a=lang:" attributes can be provided either at session or media level if the session or media has capabilities in more than one language, in which case the order of the attributes indicates the order of preference of the various languages in the session or media, from most preferred to least preferred.

As a session-level attribute, "a=lang:" specifies a language capability for the session being described. As a media-level attribute, it specifies a language capability for that media, overriding any session-level language(s) specified.

The "a=lang:" attribute value must be a single [RFC5646] language tag. An "a=lang:" attribute **SHOULD** be specified when a session is of sufficient scope to cross geographic boundaries where the language of participants cannot be assumed, or where the session has capabilities in languages different from the locally assumed norm.

The "a=lang:" attribute is supposed to be used for setting the initial language(s) used in the session. Events during the session may influence which language(s) are used, and the participants are not strictly bound to only use the declared languages.

Most real-time use cases start with just one language used, while other cases involve a range of languages, e.g., an interpreted or subtitled session. When more than one "a=lang:" attribute is specified, the "a=lang:" attribute itself does not provide any information about multiple languages being intended to be used during the session, or if the intention is to only select one of the languages. If needed, a new attribute can be defined and used to indicate such intentions. Without such semantics, it is assumed that for a negotiated session one of the declared languages will be selected and used.

6.13. framerate (Frame Rate)

Name: framerate

Value: framerate-value

Usage Level: media

Charset Dependent: no

Syntax:

```
framerate-value = non-zero-int-or-real
```

Example:

```
a=framerate:60
```

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 are allowed. It is defined only for video media.

6.14. quality

Name: quality

Value: quality-value

Usage Level: media

Charset Dependent: no

Syntax:

```
quality-value = zero-based-integer
```

Example:

```
a=quality:10
```

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 is 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 behavior given no quality suggestion.
0	the worst still-image quality the codec designer thinks is still usable.

Table 2: Encoding Quality Values

6.15. fmtp (Format Parameters)

Name: fmtp

Value: fmtp-value

Usage Level: media

Charset Dependent: no

Syntax:

```
fmtp-value = fmt SP format-specific-params
format-specific-params = byte-string
  ; Notes:
  ; - The format parameters are media type parameters and
  ;   need to reflect their syntax.
```

Example:

```
a=fmtp:96 profile-level-id=42e016;max-mbps=108000;max-fs=3600
```

This attribute allows parameters that are specific to a particular format to be conveyed in a way that SDP does not have to understand them. The format must be one of the formats specified for the media. Format-specific parameters, semicolon separated, 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.

The "a=fmtp:" attribute may be used to specify parameters for any protocol and format that defines use of such parameters.

7. Security Considerations

SDP is frequently used with the Session Initiation Protocol [RFC3261] using the offer/answer model [RFC3264] to agree on 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. Entities receiving and acting upon an SDP message **SHOULD** be aware that a session description cannot be trusted unless it has been obtained by an authenticated and integrity-protected transport protocol from a known and trusted source. Many different transport protocols may be used to distribute session descriptions, and the nature of the authentication and integrity protection will differ from transport to transport. For some transports, security features are often not deployed. In case a session description has not been obtained in a trusted manner, the endpoint **SHOULD** exercise care because, among other attacks, the media sessions received may not be the intended ones, the destination to where the media is sent may not be the expected one, any of the parameters of the session may be incorrect, or the media security may be compromised. It is up

to the endpoint to make a sensible decision, taking into account the security risks of the application and the user preferences - the endpoint may decide to ask the user whether or not to accept the session.

On receiving a session description over an unauthenticated transport mechanism or from an untrusted party, software parsing the session description should take a few precautions. Similar concerns apply if integrity protection is not in place. Session descriptions contain information required to start software on the receiver's 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 the user's 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 or not a session is interactive, applications that are unsure should assume sessions are interactive. Software processing URLs contained in session descriptions should also heed the security considerations identified in [RFC3986].

In this specification, there are no attributes that 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 behavior for an unknown attribute is to ignore it.

In certain environments, it has become common for intermediary systems to intercept and analyze session descriptions contained within other signaling protocols. This is done for a range of purposes, including but not limited to opening holes in firewalls to allow media streams to pass, or to mark, prioritize, or block traffic selectively. In some cases, such intermediary systems may modify the session description, for example, to have the contents of the session description match NAT bindings dynamically created. These behaviors are **NOT RECOMMENDED** unless the session description is conveyed in such a manner that allows the intermediary system to conduct proper checks to establish the authenticity of the session description, and the authority of its source to establish such communication sessions. SDP by itself does not include sufficient information to enable these checks: they depend on the encapsulating protocol (e.g., SIP or RTSP). The use of some procedures and SDP extensions (e.g., Interactive Connectivity Establishment (ICE) [RFC8445] and ICE-SIP-SDP [RFC8839]) may avoid the need for intermediaries to modify SDP.

SDP **MUST NOT** be used to convey keying material (e.g., using the "a=crypto:" attribute [RFC4568]) 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 media type registration from [RFC4566] has been updated, as defined below.

Type name: application

Subtype name: sdp

Required parameters: None.

Optional parameters: None.

Encoding considerations: 8-bit text. SDP files are primarily UTF-8 format text. The "a=charset:" attribute may be used to signal the presence of other character sets in certain parts of an SDP file (see Section 6 of RFC 8866). Arbitrary binary content cannot be directly represented in SDP.

Security considerations: See Section 7 of RFC 8866.

Interoperability considerations: See RFC 8866.

Published specification: See RFC 8866.

Applications which use this media type:
Voice over IP, video teleconferencing, streaming media, instant messaging, among others. See also Section 3 of RFC 8866.

Fragment identifier considerations: None

Additional information:

Deprecated alias names for this type: N/A
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:
IETF MMUSIC working group
<mmusic@ietf.org>

Intended usage: COMMON

Restrictions on usage: None

Author/Change controller:
Authors of RFC 8866
IETF MMUSIC working group delegated from the IESG

8.2. Registration of SDP Parameters with IANA

This document specifies IANA parameter registries for six named SDP subfields. Using the terminology in the SDP specification Augmented Backus-Naur Form (ABNF), they are <media>, <proto>, <attribute-name>, <bwtype>, <nettype>, and <addrtype>.

This document also replaces and updates the definitions of all those parameters previously defined by [RFC4566].

IANA has changed all references to RFC 4566 in these registries to instead refer to this document.

The contact name and email address for all parameters registered in this document is:

> The IETF MMUSIC working group <mmusic@ietf.org> or its successor as designated by the IESG.

All of these registries have a common format:

Type	SDP Name	[other fields]	Reference

Table 3: Common Format for SDP Registries

8.2.1. Registration Procedure

A specification document that defines values for SDP <media>, <proto>, <attribute-name>, <bwtype>, <nettype>, and <addrtype> parameters **MUST** include the following information:

- Contact name
- Contact email address
- Name being defined (as it will appear in SDP)
- Type of name (<media>, <proto>, <attribute-name>, <bwtype>, <nettype>, or <addrtype>)
- A description of the purpose of the defined name
- A stable reference to the document containing this information and the definition of the value. (This will typically be an RFC number.)

The subsections below specify what other information (if any) must be specified for particular parameters, and what other fields are to be included in the registry.

8.2.2. 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 well as for top-level media types, and where possible the same name should be registered for SDP as for MIME. For media other than existing top-level media types, a Standards Track RFC **MUST** be produced for a new top-level media type to be registered, and the registration **MUST** provide good justification why no existing media name is appropriate (the "Standards Action" policy of [RFC8126]).

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

Note: The media types "control" and "data" were listed as valid in an early version of this specification [RFC2327]; 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 [RFC3840]. If these media types are considered useful in the 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 [RFC3840]). Also note that [RFC6466] defined the "image" media type.

8.2.3. Transport Protocols (<proto>)

The <proto> subfield describes the transport protocol used. The registration procedure for this registry is "RFC Required".

This document registers two values:

- "RTP/AVP" is a reference to [RFC3550] used under the RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551] running over UDP/IP.
- "udp" indicates direct use of UDP.

New transport protocols **MAY** be defined, and **MUST** be registered with IANA. Registrations **MUST** reference an RFC describing the protocol. Such an RFC **MAY** be Experimental or Informational, although it is preferable that it be Standards Track. The RFC defining a new protocol **MUST** define the rules by which the <fmt> (see below) namespace is managed.

<proto> names starting with "RTP/" **MUST** only be used for defining transport protocols that are profiles of RTP. For example, a profile whose short name is "XYZ" would be denoted by a <proto> subfield of "RTP/XYZ".

Each transport protocol, defined by the <proto> subfield, has an associated <fmt> namespace that describes the media formats that may be conveyed by that protocol. Formats cover all the possible encodings that could be transported in a multimedia session.

RTP payload formats under the "RTP/AVP" and other "RTP/*" 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 "a=rtpmap:" attribute **MUST** be included to specify the format name and parameters as defined by the media type registration for the payload format. It is **RECOMMENDED** that other RTP profiles that are registered (in combination with RTP) as SDP transport protocols specify the same rules for the <fmt> namespace.

For the "udp" protocol, the allowed <fmt> values are media subtypes from the IANA Media Types registry. The media type and subtype combination <media>/<fmt> specifies the format of the body of UDP packets. Use of an existing media subtype for the format is encouraged. If no suitable media subtype exists, it is **RECOMMENDED** that a new one be registered through the IETF process [RFC6838] by production of, or reference to, a Standards Track RFC that defines 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 (<attribute-name>)

Attribute-field names (<attribute-name>) **MUST** be registered with IANA and documented to avoid any issues due to conflicting attribute definitions under the same name. (While unknown attributes in SDP are simply ignored, conflicting ones that fragment the protocol are a serious problem.)

The format of the <attribute-name> registry is:

Type	SDP Name	Usage Level	Mux Category	Reference

Table 4: Format of the <attribute-name> Registry

For example, the attribute "a=lang:", which is defined for both session and media level, will be listed in the new registry as follows:

Type	SDP Name	Usage Level	Mux Category	Reference
attribute	lang	session, media	TRANSPORT	[RFC8866] [RFC8859]

Table 5: <attribute-name> Registry Example

This one <attribute-name> registry combines all of the previous usage-level-specific "att-field" registries, including updates made by [RFC8859], and renames the "att-field" registry to the "attribute-name (formerly "att-field")" registry. IANA has completed the necessary reformatting.

Section 6 of this document replaces the initial set of attribute definitions made by [RFC4566]. IANA has updated the registry accordingly.

Documents can define new attributes and can also extend the definitions of previously defined attributes.

8.2.4.1. New Attributes

New attribute registrations are accepted according to the "Specification Required" policy of [RFC8126], provided that the specification includes the following information:

- Contact name
- Contact email address
- Attribute name: the name of the attribute that will appear in SDP. This **MUST** conform to the definition of <attribute-name>.
- Attribute syntax: for a value attribute (see Section 5.13), an ABNF definition of the attribute value <attribute-value> syntax (see Section 9) **MUST** be provided. The syntax **MUST** follow the rule form per Section 2.2 of [RFC5234] and [RFC7405]. This **SHALL** define the allowable

values that the attribute might take. It **MAY** also define an extension method for the addition of future values. For a property attribute, the ABNF definition is omitted as the property attribute takes no values.

- Attribute semantics: for a value attribute, a semantic description of the values that the attribute might take **MUST** be provided. The usage of a property attribute is described under Purpose below.
- Attribute value: the name of an ABNF syntax rule defining the syntax of the value. Absence of a rule name indicates that the attribute takes no values. Enclosing the rule name in "[" and "]" indicates that a value is optional.
- Usage level: the usage level(s) of the attribute. This **MUST** be one or more of the following: session, media, source, dcsa, and dcsa(subprotocol). For a definition of source-level attributes, see [RFC5576]. For a definition of dcsa attributes see [RFC8864].
- Charset dependent: this **MUST** be "Yes" or "No" depending on whether the attribute value is subject to the "a=charset:" attribute.
- Purpose: an explanation of the purpose and usage of the attribute.
- O/A procedures: offer/answer procedures as explained in [RFC3264].
- Mux Category: this **MUST** indicate one of the following categories: NORMAL, NOT RECOMMENDED, IDENTICAL, SUM, TRANSPORT, INHERIT, IDENTICAL-PER-PT, SPECIAL, or TBD as defined by [RFC8859].
- Reference: a reference to the specification defining the 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.

Submitters of registrations should also carefully choose the attribute usage level. They should not choose only "session" when the attribute can have different values when media is disaggregated, i.e., when each "m=" section has its own IP address on a different endpoint. In that case, the attribute type chosen should be "session, media" or "media" (depending on desired semantics). The default rule is that for all new SDP attributes that can occur both in session and media level, the media level overrides the session level. When this is not the case for a new SDP attribute, it **MUST** be explicitly stated.

IANA has registered the initial set of attribute names (<attribute-name> values) with definitions as in Section 6 of this memo (these definitions replace those in [RFC4566]).

8.2.4.2. Updates to Existing Attributes

Updated attribute registrations are accepted according to the "Specification Required" policy of [RFC8126].

The Designated Expert reviewing the update is requested to evaluate whether the update is compatible with the prior intent and use of the attribute, and whether the new document is of sufficient maturity and authority in relation to the prior document.

The specification updating the attribute (for example, by adding a new value) **MUST** update registration information items from Section 8.2.4.1 according to the following constraints:

- Contact name: a name for an entity responsible for the update **MUST** be provided.
- Contact email address: an email address for an entity responsible for the update **MUST** be provided.
- Attribute name: **MUST** be provided and **MUST NOT** be changed. Otherwise it is a new attribute.
- Attribute syntax: the existing rule syntax with the syntax extensions **MUST** be provided if there is a change to the syntax. A revision to an existing attribute usage **MAY** extend the syntax of an attribute, but **MUST** be backward compatible.
- Attribute semantics: a semantic description of new additional attribute values or a semantic extension of existing values. Existing attribute values semantics **MUST** only be extended in a backward compatible manner.
- Usage level: updates **MAY** only add additional levels.
- Charset dependent: **MUST NOT** be changed.
- Purpose: **MAY** be extended according to the updated usage.
- O/A procedures: **MAY** be updated in a backward compatible manner and/or it applies to a new usage level only.
- Mux Category: no change unless from "TBD" to another value (see [RFC8859]. It **MAY** also change if media level is being added to the definition of an attribute that previously did not include it.
- Reference: a new (additional or replacement) reference **MUST** be provided.

Items **SHOULD** be omitted if there is no impact to them as a result of the attribute update.

8.2.5. Bandwidth Specifiers (<bwtype>)

A proliferation of bandwidth specifiers is strongly discouraged.

New bandwidth specifiers (<bwtype> subfield values) **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.

The RFC **MUST** specify the Mux Category for this value as defined by [RFC8859].

The format of the <bwtype> registry is:

Type	SDP Name	Mux Category	Reference

Table 6: Format of the <bwtype> Registry

IANA has updated the <bwtype> registry entries for the bandwidth specifiers "CT" and "AS" with the definitions in Section 5.8 of this memo (these definitions replace those in [RFC4566]).

8.2.6. Network Types (<nettype>)

Network type "IN", representing the Internet, is defined in Section 5.2 and Section 5.7 of this memo (this definition replaces that in [RFC4566]).

To enable SDP to reference a new non-Internet environment, a new network type (<nettype> subfield value) **MUST** be registered with IANA. The registration is subject to the "RFC Required" policy of [RFC8126]. Although non-Internet environments 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 telephone call into the Public Switched Telephone Network (PSTN). The number of network types should be small and should be rarely extended. A new network type registration **MUST** reference an RFC that gives details of the network type and the address type(s) that may be used with it.

The format of the <nettype> registry is:

Type	SDP Name	Usable addrtype Values	Reference

Table 7: Format of the <nettype> Registry

IANA has updated the <nettype> registry to this new format. The following is the updated content of the registry:

Type	SDP Name	Usable addrtype Values	Reference
nettype	IN	IP4, IP6	[RFC8866]
nettype	TN	RFC2543	[RFC2848]
nettype	ATM	NSAP, GWID, E164	[RFC3108]
nettype	PSTN	E164	[RFC7195]

Table 8: Content of the <nettype> registry

Note that both [RFC7195] and [RFC3108] registered "E164" as an address type, although [RFC7195] mentions that the "E164" address type has a different context for ATM and PSTN networks.

8.2.7. Address Types (<addrtype>)

New address types (<addrtype>) **MUST** be registered with IANA. The registration is subject to the "RFC Required" policy of [RFC8126]. 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.

Section 5.7 of this document gives new definitions of address types "IP4" and "IP6".

8.3. Encryption Key Access Methods (OBSOLETE)

The IANA previously maintained a table of SDP encryption key access method ("enckey") names. This table is obsolete, 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 [RFC5234] and [RFC7405].

```
; SDP Syntax
session-description = version-field
                      origin-field
                      session-name-field
                      [information-field]
                      [uri-field]
                      *email-field
                      *phone-field
                      [connection-field]
                      *bandwidth-field
                      1*time-description
                      [key-field]
                      *attribute-field
                      *media-description

version-field =       %s"v" "=" 1*DIGIT CRLF
                          ;this memo describes version 0

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

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

information-field =   %s"i" "=" text CRLF

uri-field =           %s"u" "=" uri CRLF

email-field =         %s"e" "=" email-address CRLF

phone-field =         %s"p" "=" phone-number CRLF

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

bandwidth-field =     %s"b" "=" bwtype ":" bandwidth CRLF

time-description =    time-field
                          [repeat-description]

repeat-description =  1*repeat-field
                          [zone-field]

time-field =          %s"t" "=" start-time SP stop-time CRLF

repeat-field =        %s"r" "=" repeat-interval SP typed-time
                          1*(SP typed-time) CRLF

zone-field =          %s"z" "=" time SP ["-"] typed-time
                          *(SP time SP ["-"] typed-time) CRLF

key-field =           %s"k" "=" key-type CRLF

attribute-field =     %s"a" "=" attribute CRLF
```

```
media-description =   media-field
                      [information-field]
                      *connection-field
                      *bandwidth-field
                      [key-field]
                      *attribute-field

media-field =         %s"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 RFC 3986

; sub-rules of 'e=', see RFC 5322 for definitions
email-address        = address-and-comment / dispname-and-address
                       / addr-spec
address-and-comment  = addr-spec 1*SP "(" 1*email-safe ")"
dispname-and-address = 1*email-safe 1*SP "<" addr-spec ">"

; 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 time in
```

```
                      ; seconds since January 1, 1900 UTC.
                      ; The representation is an unbounded
                      ; length field containing at least
                      ; 10 digits. Unlike some representations
                      ; 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 = %s"d" / %s"h" / %s"m" / %s"s"
; NOTE: These units are case-sensitive.

; sub-rules of 'k='
key-type =            %s"prompt" /
                      %s"clear:" text /
                      %s"base64:" base64 /
                      %s"uri:" uri
                      ; NOTE: These names are case-sensitive.

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

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

attribute-name =      token

attribute-value =     byte-string

att-field =           attribute-name ; for backward compatibility

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

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

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

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 [ "/" numaddr ]
                      ; IP4 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 =       IP6-address [ "/" numaddr ]
                      ; IP6 address starting with FF

numaddr =             integer

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

FQDN =                4*(alpha-numeric / "-" / ".")
                      ; fully qualified domain name as specified
                      ; in RFC 1035 (and updates)

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

b1 =                  decimal-uchar
                      ; less than "224"

IP6-address =                                    6( h16 ":" ) ls32
                      /                       "::" 5( h16 ":" ) ls32
                      / [               h16 ] "::" 4( h16 ":" ) ls32
                      / [ *1( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32
                      / [ *2( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
                      / [ *3( h16 ":" ) h16 ] "::"    h16 ":"   ls32
                      / [ *4( h16 ":" ) h16 ] "::"              ls32
                      / [ *5( h16 ":" ) h16 ] "::"              h16
                      / [ *6( h16 ":" ) h16 ] "::"

h16 =                 1*4HEXDIG

ls32 =                ( h16 ":" h16 ) / IP4-address

; 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 =          ALPHA / DIGIT
                              / "!" / "#" / "$" / "%" / "&"
                              / "'" ; (single quote)
                              / "*" / "+" / "-" / "." / "^" / "_"
                              / "`" ; (Grave accent)
```

```
                              / "{" / "|" / "}" / "~"

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

zero-based-integer = "0" / integer

non-zero-int-or-real = integer / non-zero-real

non-zero-real = zero-based-integer "." *DIGIT POS-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 =                 <ALPHA definition from RFC 5234>
DIGIT =                 <DIGIT definition from RFC 5234>
CRLF =                  <CRLF definition from RFC 5234>
HEXDIG =                <HEXDIG definition from RFC 5234>
SP =                    <SP definition from RFC 5234>
VCHAR =                 <VCHAR definition from RFC 5234>
URI-reference =         <URI-reference definition from RFC 3986>
addr-spec =             <addr-spec definition from RFC 5322>
```

10. Summary of Changes from RFC 4566

- Generally clarified and refined terminology. Aligned terms used in text with the ABNF. The terms <attribute>, <att-field>, and "att-field" are now <attribute-name>. The terms <value> and <att-value> are now <attribute-value>. The term "media" is now <media>.
- Identified now-obsolete items: "a=cat:" (Section 6.1), "a=keywds:" (Section 6.2), and "k=" (Section 5.12).
- Updated normative and informative references, and added references to additional relevant related RFCs.
- Reformatted the SDP Attributes section (Section 6) for readability. The syntax of attribute values is now given in ABNF.
- Made mandatory the sending of RTCP with inactive media streams (Section 6.7.4).
- Removed the section "Private Sessions". That section dated back to a time when the primary use of SDP was with SAP (Session Announcement Protocol), which has fallen out of use. Now

the vast majority of uses of SDP is for establishment of private sessions. The considerations for that are covered in Section 7.

- Expanded and clarified the specification of the "a=lang:" (Section 6.12) and "a=sdplang:" (Section 6.11) attributes.
- Removed some references to SAP because it is no longer in widespread use.
- Changed the way <fmt> values for UDP transport are registered (Section 8.2.3).
- Changed the mechanism and documentation required for registering new attributes (Section 8.2.4.1).
- Tightened up IANA registration procedures for extensions. Removed phone number and long-form name (Section 8.2).
- Expanded the IANA <nettype> registry to identify valid <addrtype> subfields (Section 8.2.6).
- Reorganized the several IANA "att-field" registries into a single <attribute-name> registry (Section 8.2.4).
- Revised ABNF syntax (Section 9) for clarity and for alignment with text. Backward compatibility is maintained with a few exceptions. Of particular note:
  - Revised the syntax of time descriptions ("t=", "r=", "z=") to remove ambiguities. Clarified that "z=" only modifies the immediately preceding "r=" lines. Made "z=" without a preceding "r=" a syntax error (Section 5.11). (This is incompatible with certain aberrant usage.)
  - Updated the "IP6-address" and "IP6-multicast" rules, consistent with the syntax in [RFC3986], mirroring a bug fix made to [RFC3261] by [RFC5954]. Removed rules that were unused as a result of this change.
  - The "att-field" rule has been renamed "attribute-name" because elsewhere "*-field" always refers to a complete line. However, the rulename "att-field" remains defined as a synonym for backward compatibility with references from other RFCs.
  - The "att-value" rule has been renamed "attribute-value".
- Revised normative statements that were redundant with ABNF syntax, making the text non-normative.
- Revised IPv4 unicast and multicast addresses in the example SDP descriptions per [RFC5735] and [RFC5771].
- Changed some examples to use IPv6 addresses, and added additional examples using IPv6.
- Incorporated case-insensitivity rules from [RFC4855].
- Revised sections that incorrectly referenced NTP (Section 5.2, Section 5.9, Section 5.10, and Section 5.11).
- Clarified the explanation of the impact and use of the "a=charset:" attribute (Section 6.10).
- Revised the description of the "a=type:" attribute to remove implication that it sometimes changes the default media direction to something other than "a=sendrecv" (Section 6.9).

11. References

11.1. Normative References

[E164] International Telecommunication Union, "E.164 : The international public telecommunication numbering plan", ITU Recommendation E.164, November 2010, <https://www.itu.int/rec/T-REC-E.164-201011-I/en>.

[ISO.8859-1.1998] International Organization for Standardization, "Information technology - 8-bit single byte coded graphic - character sets - Part 1: Latin alphabet No. 1, JTC1/SC2", ISO/IEC Standard 8859-1, 1998.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2848] Petrack, S. and L. Conroy, "The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services", RFC 2848, DOI 10.17487/RFC2848, June 2000, <https://www.rfc-editor.org/info/rfc2848>.

[RFC2978] Freed, N. and J. Postel, "IANA Charset Registration Procedures", BCP 19, RFC 2978, DOI 10.17487/RFC2978, October 2000, <https://www.rfc-editor.org/info/rfc2978>.

[RFC3108] Kumar, R. and M. Mostafa, "Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections", RFC 3108, DOI 10.17487/RFC3108, May 2001, <https://www.rfc-editor.org/info/rfc3108>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <https://www.rfc-editor.org/info/rfc5576>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <https://www.rfc-editor.org/info/rfc5646>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <https://www.rfc-editor.org/info/rfc5890>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

[RFC7195] Garcia-Martin, M. and S. Veikkolainen, "Session Description Protocol (SDP) Extension for Setting Audio and Video Media Streams over Circuit-Switched Bearers in the Public Switched Telephone Network (PSTN)", RFC 7195, DOI 10.17487/RFC7195, May 2014, <https://www.rfc-editor.org/info/rfc7195>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8859] Nandakumar, S., "A Framework for Session Description Protocol (SDP) Attributes When Multiplexing", RFC 8859, DOI 10.17487/RFC8859, January 2021, <https://www.rfc-editor.org/info/rfc8859>.

[RFC8864] Drage, K., Makaraju, M., Ejzak, R., Marcon, J., and R. Even, Ed., "Negotiation Data Channels Using the Session Description Protocol (SDP)", RFC 8864, DOI 10.17487/RFC8864, January 2021, <https://www.rfc-editor.org/info/rfc8864>.

11.2. Informative References

[ITU.H332.1998] International Telecommunication Union, "H.332 : H.323 extended for loosely coupled conferences", ITU Recommendation H.332, September 1998, <https://www.itu.int/rec/T-REC-H.332-199809-I/en>.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <https://www.rfc-editor.org/info/rfc2045>.

[RFC2327] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, DOI 10.17487/RFC2327, April 1998, <https://www.rfc-editor.org/info/rfc2327>.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <https://www.rfc-editor.org/info/rfc2974>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

[RFC3556] Casner, S., "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556, DOI 10.17487/RFC3556, July 2003, <https://www.rfc-editor.org/info/rfc3556>.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <https://www.rfc-editor.org/info/rfc3605>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <https://www.rfc-editor.org/info/rfc3840>.

[RFC3890] Westerlund, M., "A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)", RFC 3890, DOI 10.17487/RFC3890, September 2004, <https://www.rfc-editor.org/info/rfc3890>.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <https://www.rfc-editor.org/info/rfc4568>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/info/rfc4855>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <https://www.rfc-editor.org/info/rfc5124>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[RFC5735] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", RFC 5735, DOI 10.17487/RFC5735, January 2010, <https://www.rfc-editor.org/info/rfc5735>.

[RFC5771] Cotton, M., Vegoda, L., and D. Meyer, "IANA Guidelines for IPv4 Multicast Address Assignments", BCP 51, RFC 5771, DOI 10.17487/RFC5771, March 2010, <https://www.rfc-editor.org/info/rfc5771>.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <https://www.rfc-editor.org/info/rfc5888>.

[RFC5954] Gurbani, V., Ed., Carpenter, B., Ed., and B. Tate, Ed., "Essential Correction for IPv6 ABNF and URI Comparison in RFC 3261", RFC 5954, DOI 10.17487/RFC5954, August 2010, <https://www.rfc-editor.org/info/rfc5954>.

[RFC6466] Salgueiro, G., "IANA Registration of the 'image' Media Type for the Session Description Protocol (SDP)", RFC 6466, DOI 10.17487/RFC6466, December 2011, <https://www.rfc-editor.org/info/rfc6466>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

[RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <https://www.rfc-editor.org/info/rfc7405>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

[RFC7826] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, Ed., "Real-Time Streaming Protocol Version 2.0", RFC 7826, DOI 10.17487/RFC7826, December 2016, <https://www.rfc-editor.org/info/rfc7826>.

[RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

[RFC8839] Petit-Huguenin, M., Nandakumar, S., Holmberg, C., Keränen, A., and R. Shpount, "Session Description Protocol (SDP) Offer/Answer Procedures for Interactive Connectivity Establishment (ICE)", RFC 8839, DOI 10.17487/RFC8839, January 2021, <https://www.rfc-editor.org/info/rfc8839>.

[RFC8843] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", RFC 8843, DOI 10.17487/RFC8843, January 2021, <https://www.rfc-editor.org/info/rfc8843>.

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 the following people who contributed to the creation of this document or one of its predecessor documents: Adam Roach, Allison Mankin, Bernie Hoeneisen, Bill Fenner, Carsten Bormann, Eve Schooler, Flemming Andreasen, Gonzalo Camarillo, Jörg Ott, John Elwell, Jon Peterson, Jonathan Lennox, Jonathan Rosenberg, Keith Drage, Peter Parnes, Rob Lanphier, Ross Finlayson, Sean Olson, Spencer Dawkins, Steve Casner, Steve Hanna, Van Jacobson.

Authors' Addresses

Ali Begen
Networked Media
Turkey
Email: ali.begen@networked.media

Paul Kyzivat
United States of America
Email: pkyzivat@alum.mit.edu

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow
G12 8QQ
United Kingdom
Email: csp@csperkins.org

Mark Handley
University College London
Department of Computer Science
London
WC1E 6BT
United Kingdom
Email: M.Handley@cs.ucl.ac.uk