Network Working Group
Internet-Draft
Updates: 4572 (if approved)
Intended status: Standards Track
Expires: September 22, 2016

C. Holmberg
Ericsson

March 21, 2016

Updates to RFC 4572
draft-holmberg-mmusic-4572-update-01.txt

Abstract

This document updates RFC 4572 by clarifying the usage of multiple SDP 'fingerprint' attributes with a single TLS connection. The document also updates the preferred cipher suite to be used, and removes the requirement to use the same hash function for calculating the certificate fingerprint that is used to calculate the certificate signature.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 22, 2016.

Copyright Notice

Table of Contents

1. Introduction

RFC 4572 [RFC4572] specifies how to establish Transport Layer Security (TLS) connections using the Session Description Protocol (SDP) [RFC4566].

RFC 4572 defines the SDP 'fingerprint' attribute, which is used to carry a secure hash value associated with a certificate. However, RFC 4572 is currently unclear on whether multiple 'fingerprint' can be associated with a single SDP media description ("m= line") [RFC4566], and the associated semantics. Multiple 'fingerprint' attributes are needed when an endpoint wants to provide multiple fingerprint, using different hash functions, for a certificate. Multiple 'fingerprint' attributes are also needed if an endpoint wants to provide fingerprints associated with multiple certificates. For example, with RTP-based media, an endpoint might use different certificates for RTP and RTCP.

RFC 4572 also specifies a preferred cipher suite. However, the currently preferred cipher suite is considered outdated, and the preference needs to be updated.

RFC 4572 mandates that the hash function used to calculate the fingerprint is the same hash function used to calculate the certificate signature. That requirement might prevent usage of newer, stronger and more collision-safe hash functions for calculating certificate fingerprints.

This document updates RFC 4572 [RFC4572] by clarifying the usage of multiple SDP 'fingerprint' attributes with a single TLS connection. The document also updates the preferred cipher suite to be used, and

removes the requirement to use the same hash function for calculating the certificiate fingerprint and certificate signature.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Update to RFC 4572

This section updates section 5 of RFC 4572.

3.1. Update to the sixth paragraph of section 5

OLD TEXT:

A certificate fingerprint MUST be computed using the same one-way hash function as is used in the certificate's signature algorithm. (This ensures that the security properties required for the certificate also apply for the fingerprint. It also guarantees that the fingerprint will be usable by the other endpoint, so long as the certificate itself is.) Following RFC 3279 [7] as updated by RFC 4055 [9], therefore, the defined hash functions are 'SHA-1' [11] [19], 'SHA-224' [11], 'SHA-256' [11], 'SHA-384' [11], 'SHA-512' [11], 'MD5' [12], and 'MD2' [13], with 'SHA-1' preferred. A new IANA registry of Hash Function Textual Names, specified in Section 8, allows for addition of future tokens, but they may only be added if they are included in RFCs that update or obsolete RFC 3279 [7]. Self-signed certificates (for which legacy certificates are not a consideration) MUST use one of the FIPS 180 algorithms (SHA-1, SHA-224, SHA-256, SHA-384, or SHA-512) as their signature algorithm, and thus also MUST use it to calculate certificate fingerprints.

NEW TEXT:

Following RFC 3279 [7] as updated by RFC 4055 [9], therefore, the defined hash functions are 'SHA-1' [11] [19], 'SHA-224' [11], 'SHA-256' [11], 'SHA-384' [11], 'SHA-512' [11], 'MD5' [12], and 'MD2' [13], with 'SHA-256' preferred. A new IANA registry of Hash Function Textual Names, specified in Section 8, allows for addition of future tokens, but they may only be added if they are included in RFCs that update or obsolete RFC 3279 [7].

3.2. New paragraphs to the end of section 5

NEW TEXT:

> Multiple SDP fingerprint attributes can be associated with an m- line. This can occur if multiple fingerprints have been calculated for a certificate, using different hash algorithms. It can also occur if multiple certificates might be used (e.g. separate certificates for RTP and RTCP). In such cases, the same number of fingerprints MUST be calculated for each certificate, and for each certificate the same set of hash algorithms MUST be used.
>
> An endpoint MUST be able to match at least one of the received fingerprints with the cerficiate(s) to be used. If there is no match, the endpoint MUST NOT establish the TLS connection.
>
> NOTE: The SDP fingerprint attribute does not contain a reference to a specific certificate. Endpoints need to compare all fingerprints with the cerficiate hash when looking for a match.

4. Security Considerations

This document improves security.

5. Acknowledgements

Martin Thompson, Paul Kyzivat and Jonathan Lennox provided valuable comments and input on this document.

6. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-holmberg-mmusic-4572-update-xx

- Add text

7. Normative References

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

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

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <http://www.rfc-editor.org/info/rfc4572>.

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

Author’s Address

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Network Working Group
Internet-Draft
Updates: 5763,7345 (if approved)
Intended status: Standards Track
Expires: September 22, 2016

C. Holmberg
Ericsson
R. Shpount
TurboBridge

March 21, 2016

Using the SDP Offer/Answer Mechanism for DTLS
draft-ietf-mmusic-dtls-sdp-11.txt

Abstract

This draft defines the SDP offer/answer procedures for negotiating and establishing a DTLS association. The draft also defines the criteria for when a new DTLS association must be established.

This draft defines a new SDP media-level attribute, 'dtls-association-id'.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 22, 2016.

Copyright Notice

Table of Contents

1. Introduction

[RFC5763] defines SDP Offer/Answer procedures for SRTP-DTLS. [RFC7345] defines SDP Offer/Answer procedures for UDPTL-DTLS. This specification defines general Offer/Answer procedures for DTLS, based on the procedures in [RFC5763]. Other specifications, defining specific DTLS usages, can then reference this specification, in order to ensure that the DTLS aspects are common among all usages. Having common procedures is essential when multiple usages share the same DTLS association [I-D.ietf-mmusic-sdp-bundle-negotiation].

As defined in [RFC5763], a new DTLS association MUST be established when transport parameters are changed. Transport parameter change is

not well defined when Interactive Connectivity Establishment (ICE) [RFC5245] is used. One possible way to determine a transport change is based on ufrag change, but the ufrag value is changed both when ICE is negotiated and when ICE restart [RFC5245] occurs. These events do not always require a new DTLS association to be established, but currently there is no way to explicitly indicate in an SDP offer or answer whether a new DTLS association is required. To solve that problem, this draft defines a new SDP attribute, 'dtls-association-id'. The attribute contains a unique value associated with a DTLS association, and by providing a new value in SDP offers and answers the attribute can be used to indicate whether a new DTLS association is to be established/re-established.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Establishing a new DTLS Association

3.1. General

A new DTLS association MUST be established in the following cases:

- The DTLS roles change;
- One or more fingerprint values are modified, added or removed; or
- The intent to establish a new DTLS association is explicitly signaled;

NOTE: The first two items list above are based on the procedures in [RFC5763]. This specification adds the support for explicit signaling.

Whenever an entity determines, based on the criteria above, that a new DTLS association is required, the entity MUST initiate an associated SDP offer/answer transaction, following the procedures in Section 5.

The sections below describe typical cases where a new DTLS association needs to be established.

3.2. Change of Local Transport Parameters

If an endpoint modifies its local transport parameters (address and/or port), and if the modification requires a new DTLS association, the endpoint MUST change its DTLS role, change its fingerprint value, and/or use the SDP 'dtls-association-id' attribute with a new value Section 4.

If the underlying transport explicitly prohibits a DTLS association to span multiple transports, and if the transport is changed, a new value MUST be assigned to the the SDP 'dtls-association-id' attribute. An example of such case is when DTLS is carried over SCTP, as described in [RFC6083].

3.3. Change of ICE ufrag value

If an endpoint uses ICE, and modifies a local ufrag value, and if the modification requires a new DTLS association, the endpoint MUST either change its DTLS role, a fingerprint value and/or assign a new value to the SDP 'dtls-association-id' attribute Section 4.

4. SDP dtls-association-id Attribute

The SDP 'dtls-association-id' attribute contains a unique value associated with a DTLS association.

Name: dtls-association-id

Value: conn-value

Usage Level: media

Charset Dependent: no

Syntax:

```
conn-value = 1*256alphanum
```

Example:

```
a=dtls-association-id:abc3dl
```

A 'dtls-association-id' attribute that contains a new value indicates an intention to establish a new DTLS association. A 'dtls-association-id' attribute that contains a previously assigned value indicates an intention to reuse an existing association.

There is no default value defined for the SDP 'dtls-association-id' attribute. Implementations that wish to use the attribute MUST explicitly include it in SDP offers and answers. If an offer or answer does not contain an attribute (this could happen if the offerer or answerer represents an existing implementation that has not been updated to support the attribute defined in this specification or an implementation which allocates a new temporary certificate for each association and uses change in fingerprint to indicate new association), other means needs to be used in order for endpoints to determine whether an offer or answer is associated with an event that requires the DTLS association to be re-established.

The mux category [I-D.ietf-mmusic-sdp-mux-attributes] for the 'dtls-association-id' attribute is 'IDENTICAL', which means that the attribute value must be identical across all media descriptions being multiplexed [I-D.ietf-mmusic-sdp-bundle-negotiation].

For RTP-based media, the 'dtls-association-id' attribute apply to whole associated media description. The attribute MUST NOT be defined per source (using the SDP 'ssrc' attribute [RFC5576]).

The SDP Offer/Answer [RFC3264] procedures associated with the attribute are defined in Section 5

5. SDP Offer/Answer Procedures

5.1. General

This section defines the generic SDP offer/answer procedures for negotiating a DTLS association. Additional procedures (e.g. regarding usage of specific SDP attributes etc) for individual DTLS usages (e.g. SRTP-DTLS) are outside the scope of this specification, and need to be specified in a usage specific specification.

NOTE: The procedures in this section are generalizations of procedures first specified in SRTP-DTLS [RFC5763], with the addition of usage of the SDP 'dtls-association-id' attribute. That document is herein revised to make use of these new procedures.

The procedures in this section apply to an SDP media description ("m=" line) associated a DTLS-protected media/data stream.

In order to negotiate a DTLS association, the following SDP attributes are used:

- The SDP 'setup' attribute, defined in [RFC4145], is used to negotiate the DTLS roles;

- The SDP 'fingerprint' attribute, defined in [RFC4572], is used to provide a fingerprint value; and

- The SDP 'dtls-association-id' attribute, defined in this specification, indicates a unique value associated with the DTLS association. The attribute value can be used to explicitly indicate the intention of establishing a new DTLS association.

This specification does not define the usage of the SDP 'connection' attribute [RFC4145] for negotiating a DTLS connection. However, the attribute MAY be used if the DTLS association is used together with another protocol, e.g. SCTP or TCP, for which the usage of the attribute has been defined.

Unlike for TCP and TLS connections, endpoints MUST NOT use the SDP 'setup' attribute 'holdconn' value when negotiating a DTLS association.

Endpoints MUST support SHA-256 for generating and verifying any fingerprint value associated with the DTLS association. The use of SHA-256 is preferred.

Endpoints MUST, at a minimum, support TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 and MUST support TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256. UDPTL over DTLS MUST prefer TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 and any other Perfect Forward Secrecy (PFS) cipher suites over non-PFS cipher suites. Implementations SHOULD disable TLS-level compression.

The certificate received during the DTLS handshake MUST match at least one of the certificate fingerprints received in SDP 'fingerprint' attributes. If no fingerprint matches the hashed certificate, then an endpoint MUST tear down the media session immediately. Note that it is permissible to wait until the other side's fingerprint has been received before establishing the connection; however, this may have undesirable latency effects.

SDP offerers and answerers might reuse certificates across multiple DTLS associations, and provide identical fingerprint values for each DTLS association. It MUST be ensured that the combination of the fingerprint values and the SDP 'dtls-association-id' attribute value is unique across all DTLS associations.

If an SDP offerer or answerer generates a new temporary self-signed certificate for each new DTLS association, they can omit the SDP 'dtls-association-id' attribute.

5.2. Generating the Initial SDP Offer

When the offerer sends the initial offer, and the offerer wants to establish a DTLS association, it MUST insert an SDP 'dtls-association-id' attribute with a new value in the offer. In addition, the offerer MUST insert an SDP 'setup' attribute according to the procedures in [RFC4145], and one or more SDP 'fingerprint' attributes according to the procedures in [RFC4572], in the offer.

If the offerer inserts the SDP 'setup' attribute with an 'actpass' or 'passive' value, the offerer MUST be prepared to receive a DTLS ClientHello message (if a new DTLS association is established by the answerer) from the answerer before it receives the SDP answer.

5.3. Generating the Answer

If an answerer receives an offer that contains an SDP 'dtls-association-id' attribute with a new value, or if the answerer receives an offer that contains an 'dtls-association-id' attribute with a previously assigned value and the answerer determines (based on the criteria for establishing a new DTLS association) that a new DTLS association is to be established, the answerer MUST insert a new value in the associated answer. In addition, the answerer MUST insert an SDP 'setup' attribute according to the procedures in [RFC4145], and one or more SDP 'fingerprint' attributes according to the procedures in [RFC4572], in the answer.

If an answerer receives an offer that contains an SDP 'dtls-association-id' attribute with a new value, and if the answerer does not accept the establishment of a new DTLS association, the answerer MUST reject the "m=" lines associated with the suggested DTLS association [RFC3264].

If an answerer receives an offer that contains a 'dtls-association-id' attribute with a previously assigned value, and if the answerer determines that a new DTLS association is not to be established, the answerer MUST insert a 'dtls-association-id' attribute with the previously assigned value in the associated answer. In addition, the answerer MUST insert an SDP 'setup' attribute with a value that does not change the previously negotiated DTLS roles, and one or more SDP 'fingerprint' attributes values that do not change the previously sent fingerprints, in the answer.

If the answerer receives an offer that does not contain an SDP 'dtls-association-id' attribute, the answerer MUST NOT insert a 'dtls-association-id' attribute in the answer.

If a new DTLS association is to be established, and if the answerer inserts an SDP 'setup' attribute with an 'active' value in the answer, the answerer MUST initiate a DTLS handshake by sending a DTLS ClientHello message towards the offerer.

5.4. Offerer Processing of the SDP Answer

When an offerer receives an answer that contains an SDP 'dtls-association-id' attribute with a new value, and if the offerer becomes DTLS client (based on the value of the SDP 'setup' attribute value [RFC4145]), the offerer MUST establish a DTLS association. If the offerer becomes DTLS server, it MUST wait for the answerer to establish the DTLS association.

If the answer contains an SDP 'dtls-association-id' attribute with a previously assigned value, the offerer will continue using the previously established DTLS association. It is considered an error case if the answer contains a 'dtls-association-id' attribute with a previously assigned value, and a DTLS association does not exist.

An offerer needs to be able to handle error conditions that can occur during an offer/answer transaction, e.g. if an answer contains an SDP 'dtls-association-id' attribute with a previosuly assigned value even if no DTLS association exists, or if the answer contains one or more new fingerprint values for an existing DTLS association. If such error case occurs, the offerer SHOULD terminate the associated DTLS association (if it exists) and send a new offer in order to terminate each media stream using the DTLS association, by setting the associated port value to zero [RFC4145].

5.5. Modifying the Session

When the offerer sends a subsequent offer, and if the offerer wants to establish a new DTLS association, the offerer MUST insert an SDP 'dtls-association-id' attribute with a new value in the offer. In addition, the offerer MUST insert an SDP 'setup' attribute according to the procedures in [RFC4145], and one or more SDP 'fingerprint' attributes according to the procedures in [RFC4572], in the offer.

when the offerer sends a subsequent offer, and the offerer does not want to establish a new DTLS association, and if a previously established DTLS association exists, the offerer MUST insert an SDP 'dtls-association-id' attribute with a previously assigned value in the offer. In addition, the offerer MUST insert an SDP 'setup' attribute with a value that does not change the previously negotiated DTLS roles, and one or more SDP 'fingerprint' attributes with values that do not change the previously sent fingerprints, in the offer.

NOTE: When a new DTLS association is established, each endpoint needs to be prepared to receive data on both the new and old DTLS associations as long as both are alive.

6. ICE Considerations

When ICE is used, the ICE connectivity checks are performed before the DTLS handshake begins. Note that if aggressive nomination mode is used, multiple candidate pairs may be marked valid before ICE finally converges on a single candidate pair.

An ICE restart [RFC5245] does not by default require a new DTLS association to be established.

As defined in [RFC5763], each ICE candidate associated with a component is treated as being part of the same DTLS association. Therefore, from a DTLS perspective it is not considered a change of local transport parameters when an endpoint switches between those ICE candidates.

7. Transport Protocol Considerations

7.1. Transport Re-Usage

If DTLS is transported on top of a connection-oriented transport protocol (e.g. TCP or SCTP), where all IP packets are acknowledged, all DTLS packets associated with a previous DTLS association MUST be acknowledged (or timed out) before a new DTLS association can be established on the same transport.

8. SIP Considerations

When the Session Initiation Protocol (SIP) [RFC3261] is used as the signal protocol for establishing a multimedia session, dialogs [RFC3261] might be established between the caller and multiple callees. This is referred to as forking. If forking occurs, separate DTLS associations MUST be established between the caller and each callee.

It is possible to send an INVITE request which does not contain an SDP offer. Such INVITE request is often referred to as an ’empty INVITE’, or an ’offer-less INVITE’. The receiving endpoint will include the SDP offer in a response associated with the response. When the endpoint generates such SDP offer, if a previously established DTLS association exists, the offerer SHOULD insert an SDP ’dtls-association-id’ attribute, and one or more SDP ’fingerprint’ attributes, with previously assigned attribute values. If a previously established DTLS association did not exists offer SHOULD

be generated based on the same rules as new offer Section 5.2. Regardless of the previous existence of DTLS association, the SDP 'setup' attribute MUST be included according to rules defiend in [RFC4145] and if ICE is used, ICE restart MUST be initiated as defined in [RFC5763].

9. RFC Updates

9.1. General

This section updates specifications that use DTLS-protected media, in order to reflect the procedures defined in this specification.

9.2. Update to RFC 5763

Update to section 5:

OLD TEXT:

5. Establishing a Secure Channel

The two endpoints in the exchange present their identities as part of the DTLS handshake procedure using certificates. This document uses certificates in the same style as described in "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)" [RFC4572].

If self-signed certificates are used, the content of the subjectAltName attribute inside the certificate MAY use the uniform resource identifier (URI) of the user. This is useful for debugging purposes only and is not required to bind the certificate to one of the communication endpoints. The integrity of the certificate is ensured through the fingerprint attribute in the SDP. The subjectAltName is not an important component of the certificate verification.

The generation of public/private key pairs is relatively expensive. Endpoints are not required to generate certificates for each session.

The offer/answer model, defined in [RFC3264], is used by protocols like the Session Initiation Protocol (SIP) [RFC3261] to set up multimedia sessions. In addition to the usual contents of an SDP [RFC4566] message, each media description ("m=" line and associated parameters) will also contain several attributes as specified in [RFC5764], [RFC4145], and [RFC4572].

When an endpoint wishes to set up a secure media session with another endpoint, it sends an offer in a SIP message to the other endpoint. This offer includes, as part of the SDP payload, the fingerprint of the certificate that the endpoint wants to use. The endpoint SHOULD send the SIP message containing the offer to the offerer's SIP proxy over an integrity protected channel. The proxy SHOULD add an Identity header field according to the procedures outlined in [RFC4474]. The SIP message containing the offer SHOULD be sent to the offerer's SIP proxy over an integrity protected channel. When the far endpoint receives the SIP message, it can verify the identity of the sender using the Identity header field. Since the Identity header field is a digital signature across several SIP header fields, in addition to the body of the SIP message, the receiver can also be certain that the message has not been tampered with after the digital signature was applied and added to the SIP message.

The far endpoint (answerer) may now establish a DTLS association with the offerer. Alternately, it can indicate in its answer that the offerer is to initiate the TLS association. In either case, mutual DTLS certificate-based authentication will be used. After completing the DTLS handshake, information about the authenticated identities, including the certificates, are made available to the endpoint application. The answerer is then able to verify that the offerer's certificate used for authentication in the DTLS handshake can be associated to the certificate fingerprint contained in the offer in the SDP. At this point, the answerer may indicate to the end user that the media is secured. The offerer may only tentatively accept the answerer's certificate since it may not yet have the answerer's certificate fingerprint.

When the answerer accepts the offer, it provides an answer back to the offerer containing the answerer's certificate fingerprint. At this point, the offerer can accept or reject the peer's certificate and the offerer can indicate to the end user that the media is secured.

Note that the entire authentication and key exchange for securing the media traffic is handled in the media path through DTLS. The signaling path is only used to verify the peers' certificate fingerprints.

The offer and answer MUST conform to the following requirements.

- The endpoint MUST use the setup attribute defined in [RFC4145]. The endpoint that is the offerer MUST use the setup attribute value of setup:actpass and be prepared to receive a client_hello before it receives the answer. The answerer MUST use either a setup attribute value of setup:active or setup:passive. Note that

if the answerer uses setup:passive, then the DTLS handshake will not begin until the answerer is received, which adds additional latency. setup:active allows the answer and the DTLS handshake to occur in parallel. Thus, setup:active is RECOMMENDED. Whichever party is active MUST initiate a DTLS handshake by sending a ClientHello over each flow (host/port quartet).

- The endpoint MUST NOT use the connection attribute defined in [RFC4145].
- The endpoint MUST use the certificate fingerprint attribute as specified in [RFC4572].
- The certificate presented during the DTLS handshake MUST match the fingerprint exchanged via the signaling path in the SDP. The security properties of this mechanism are described in Section 8.
- If the fingerprint does not match the hashed certificate, then the endpoint MUST tear down the media session immediately. Note that it is permissible to wait until the other side's fingerprint has been received before establishing the connection; however, this may have undesirable latency effects.

NEW TEXT:

5. Establishing a Secure Channel

The two endpoints in the exchange present their identities as part of the DTLS handshake procedure using certificates. This document uses certificates in the same style as described in "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)" [RFC4572].

If self-signed certificates are used, the content of the subjectAltName attribute inside the certificate MAY use the uniform resource identifier (URI) of the user. This is useful for debugging purposes only and is not required to bind the certificate to one of the communication endpoints. The integrity of the certificate is ensured through the fingerprint attribute in the SDP.

The generation of public/private key pairs is relatively expensive. Endpoints are not required to generate certificates for each session.

The offer/answer model, defined in [RFC3264], is used by protocols like the Session Initiation Protocol (SIP) [RFC3261] to set up multimedia sessions.

When an endpoint wishes to set up a secure media session with another endpoint, it sends an offer in a SIP message to the other endpoint. This offer includes, as part of the SDP payload, a fingerprint of a certificate that the endpoint wants to use. The endpoint SHOULD send the SIP message containing the offer to the offerer's SIP proxy over an integrity protected channel. The proxy SHOULD add an Identity header field according to the procedures outlined in [RFC4474]. The SIP message containing the offer SHOULD be sent to the offerer's SIP proxy over an integrity protected channel. When the far endpoint receives the SIP message, it can verify the identity of the sender using the Identity header field. Since the Identity header field is a digital signature across several SIP header fields, in addition to the body of the SIP message, the receiver can also be certain that the message has not been tampered with after the digital signature was applied and added to the SIP message.

The far endpoint (answerer) may now establish a DTLS association with the offerer. Alternately, it can indicate in its answer that the offerer is to initiate the DTLS association. In either case, mutual DTLS certificate-based authentication will be used. After completing the DTLS handshake, information about the authenticated identities, including the certificates, are made available to the endpoint application. The answerer is then able to verify that the offerer's certificate used for authentication in the DTLS handshake can be associated to the certificate fingerprint contained in the offer in the SDP. At this point, the answerer may indicate to the end user that the media is secured. The offerer may only tentatively accept the answerer's certificate since it may not yet have the answerer's certificate fingerprint.

When the answerer accepts the offer, it provides an answer back to the offerer containing the answerer's certificate fingerprint. At this point, the offerer can accept or reject the peer's certificate and the offerer can indicate to the end user that the media is secured.

Note that the entire authentication and key exchange for securing the media traffic is handled in the media path through DTLS. The signaling path is only used to verify the peers' certificate fingerprints.

The offerer and answerer MUST follow the SDP offer/answer procedures defined in [RFCXXXX].

Update to section 6.6:

OLD TEXT:

6.6. Session Modification

Once an answer is provided to the offerer, either endpoint MAY request a session modification that MAY include an updated offer. This session modification can be carried in either an INVITE or UPDATE request. The peers can reuse the existing associations if they are compatible (i.e., they have the same key fingerprints and transport parameters), or establish a new one following the same rules are for initial exchanges, tearing down the existing association as soon as the offer/answer exchange is completed. Note that if the active/passive status of the endpoints changes, a new connection MUST be established.

NEW TEXT:

6.6. Session Modification

Once an answer is provided to the offerer, either endpoint MAY request a session modification that MAY include an updated offer. This session modification can be carried in either an INVITE or UPDATE request. The peers can reuse an existing DTLS association, or establish a new one, following the procedures in [RFCXXXX].

Update to section 6.7.1:

OLD TEXT:

6.7.1. ICE Interaction

Interactive Connectivity Establishment (ICE), as specified in [RFC5245], provides a methodology of allowing participants in multimedia sessions to verify mutual connectivity. When ICE is being used, the ICE connectivity checks are performed before the DTLS handshake begins. Note that if aggressive nomination mode is used, multiple candidate pairs may be marked valid before ICE finally converges on a single candidate pair. Implementations MUST treat all ICE candidate pairs associated with a single component as part of the same DTLS association. Thus, there will be only one DTLS handshake even if there are multiple valid candidate pairs. Note that this may mean adjusting the endpoint IP addresses if the selected candidate pair shifts, just as if the DTLS packets were an ordinary media stream.

Note that Simple Traversal of the UDP Protocol through NAT (STUN) packets are sent directly over UDP, not over DTLS. [RFC5764]

describes how to demultiplex STUN packets from DTLS packets and SRTP packets.

NEW TEXT:

6.7.1. ICE Interaction

The Interactive Connectivity Establishment (ICE) [RFC5245] considerations for DTLS-protected media are described in [RFCXXXX].

Note that Simple Traversal of the UDP Protocol through NAT (STUN) packets are sent directly over UDP, not over DTLS. [RFC5764] describes how to demultiplex STUN packets from DTLS packets and SRTP packets.

9.3. Update to RFC 7345

Update to section 4:

OLD TEXT:

4. SDP Offerer/Answerer Procedures

4.1. General

An endpoint (i.e., both the offerer and the answerer) MUST create an SDP media description ("m=" line) for each UDPTL-over-DTLS media stream and MUST assign a UDP/TLS/UDPTL value (see Table 1) to the "proto" field of the "m=" line.

The procedures in this section apply to an "m=" line associated with a UDPTL-over-DTLS media stream.

In order to negotiate a UDPTL-over-DTLS media stream, the following SDP attributes are used:

- The SDP attributes defined for UDPTL over UDP, as described in [ITU.T38.2010]; and
- The SDP attributes, defined in [RFC4145] and [RFC4572], as described in this section.

The endpoint MUST NOT use the SDP "connection" attribute [RFC4145].

In order to negotiate the TLS roles for the UDPTL-over-DTLS transport connection, the endpoint MUST use the SDP "setup" attribute [RFC4145].

If the endpoint supports, and is willing to use, a cipher suite with an associated certificate, the endpoint MUST include an SDP "fingerprint" attribute [RFC4572]. The endpoint MUST support SHA-256 for generating and verifying the SDP "fingerprint" attribute value. The use of SHA-256 is preferred. UDPTL over DTLS, at a minimum, MUST support TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 and MUST support TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256. UDPTL over DTLS MUST prefer TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 and any other Perfect Forward Secrecy (PFS) cipher suites over non-PFS cipher suites. Implementations SHOULD disable TLS-level compression.

If a cipher suite with an associated certificate is selected during the DTLS handshake, the certificate received during the DTLS handshake MUST match the fingerprint received in the SDP "fingerprint" attribute. If the fingerprint does not match the hashed certificate, then the endpoint MUST tear down the media session immediately. Note that it is permissible to wait until the other side’s fingerprint has been received before establishing the connection; however, this may have undesirable latency effects.

4.2. Generating the Initial Offer

The offerer SHOULD assign the SDP "setup" attribute with a value of "actpass", unless the offerer insists on being either the sender or receiver of the DTLS ClientHello message, in which case the offerer can use either a value of "active" (the offerer will be the sender of ClientHello) or "passive" (the offerer will be the receiver of ClientHello). The offerer MUST NOT assign an SDP "setup" attribute with a "holdconn" value.

If the offerer assigns the SDP "setup" attribute with a value of "actpass" or "passive", the offerer MUST be prepared to receive a DTLS ClientHello message before it receives the SDP answer.

4.3. Generating the Answer

If the answerer accepts the offered UDPTL-over-DTLS transport connection, in the associated SDP answer, the answerer MUST assign an SDP "setup" attribute with a value of either "active" or "passive", according to the procedures in [RFC4145]. The answerer MUST NOT assign an SDP "setup" attribute with a value of "holdconn".

If the answerer assigns an SDP "setup" attribute with a value of "active" value, the answerer MUST initiate a DTLS handshake by

sending a DTLS ClientHello message on the negotiated media stream, towards the IP address and port of the offerer.

4.4. Offerer Processing of the Answer

When the offerer receives an SDP answer, if the offerer ends up being active it MUST initiate a DTLS handshake by sending a DTLS ClientHello message on the negotiated media stream, towards the IP address and port of the answerer.

4.5. Modifying the Session

Once an offer/answer exchange has been completed, either endpoint MAY send a new offer in order to modify the session. The endpoints can reuse the existing DTLS association if the key fingerprint values and transport parameters indicated by each endpoint are unchanged. Otherwise, following the rules for the initial offer/answer exchange, the endpoints can negotiate and create a new DTLS association and, once created, delete the previous DTLS association, following the same rules for the initial offer/answer exchange. Each endpoint needs to be prepared to receive data on both the new and old DTLS associations as long as both are alive.

NEW TEXT:

4. SDP Offerer/Answerer Procedures

An endpoint (i.e., both the offerer and the answerer) MUST create an SDP media description ("m=" line) for each UDPTL-over-DTLS media stream and MUST assign a UDP/TLS/UDPTL value (see Table 1) to the "proto" field of the "m=" line.

The offerer and answerer MUST follow the SDP offer/answer procedures defined in [RFCXXXX] in order to negotiate the DTLS association associated with the UDPTL-over-DTLS media stream. In addition, the offerer and answerer MUST use the SDP attributes defined for UDPTL over UDP, as defined in [ITU.T38.2010].

Update to section 5.2.1:

OLD TEXT:

5.2.1. ICE Usage

When Interactive Connectivity Establishment (ICE) [RFC5245] is being used, the ICE connectivity checks are performed before the DTLS

handshake begins. Note that if aggressive nomination mode is used, multiple candidate pairs may be marked valid before ICE finally converges on a single candidate pair. User Agents (UAs) MUST treat all ICE candidate pairs associated with a single component as part of the same DTLS association. Thus, there will be only one DTLS handshake even if there are multiple valid candidate pairs. Note that this may mean adjusting the endpoint IP addresses if the selected candidate pair shifts, just as if the DTLS packets were an ordinary media stream. In the case of an ICE restart, the DTLS handshake procedure is repeated, and a new DTLS association is created. Once the DTLS handshake is completed and the new DTLS association has been created, the previous DTLS association is deleted.

NEW TEXT:

5.2.1. ICE Usage

The Interactive Connectivity Establishment (ICE) [RFC5245] considerations for DTLS-protected media are described in [RFCXXXX].

10. Security Considerations

This specification does not modify the security considerations associated with DTLS, or the SDP offer/answer mechanism. In addition to the introduction of the SDP ’dtls-association-id’ attribute, the specification simply clarifies the procedures for negotiating and establishing a DTLS association.

11. IANA Considerations

This document updates the "Session Description Protocol Parameters" registry as specified in Section 8.2.2 of [RFC4566]. Specifically, it adds the SDP dtls-association-id attribute to the table for SDP media level attributes.

Attribute name: dtls-association-id
Type of attribute: media-level
Subject to charset: no
Purpose: Indicate whether a new DTLS association is to be established/re-established.
Appropriate Values: see Section 4
Contact name: Christer Holmberg
Category: IDENTICAL

12. Acknowledgements

Thanks to Justin Uberti, Martin Thomson, Paul Kyzivat, Jens Guballa, Charles Eckel and Gonzalo Salgueiro for providing comments and suggestions on the draft.

13. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-dtls-10

- Modified document to use dtls-association-id instead of dtls-connection
- Changes are based on comments from Eric Rescorla, Justin Uberti, and Paul Kyzivat.

Changes from draft-ietf-mmusic-sdp-dtls-08

- Offer/Answer section modified in order to allow sending of multiple SDP 'fingerprint' attributes.
- Terminology made consistent: 'DTLS connection' replaced with 'DTLS association'.
- Editorial changes based on comments from Paul Kyzivat.

Changes from draft-ietf-mmusic-sdp-dtls-07

- Reference to RFC 7315 replaced with reference to RFC 7345.

Changes from draft-ietf-mmusic-sdp-dtls-06

- Text on restrictions regarding spanning a DTLS association over multiple transports added.
- Mux category added to IANA Considerations.
- Normative text regarding mux category and source-specific applicability added.
- Reference to RFC 7315 added.
- Clarified that offerer/answerer that has not been updated to support this specification will not include the dtls-association-id attribute in offers and answers.

- Editorial corrections based on WGLC comments from Charles Eckel.

Changes from draft-ietf-mmusic-sdp-dtls-05

- Text on handling offer/answer error conditions added.

Changes from draft-ietf-mmusic-sdp-dtls-04

- Editorial nits fixed based on comments from Paul Kyzivat:

Changes from draft-ietf-mmusic-sdp-dtls-03

- Changes based on comments from Paul Kyzivat:

- - Modification of dtls-association-id attribute section.

- - Removal of IANA considerations subsection.

- - Making note into normative text in o/a section.

- Changes based on comments from Martin Thompson:

- - Abbreviations section removed.

- - Clarify that a new DTLS association requires a new o/a transaction.

Changes from draft-ietf-mmusic-sdp-dtls-02

- - Updated RFCs added to boilerplate.

Changes from draft-ietf-mmusic-sdp-dtls-01

- - Annex regarding ’dtls-association-id-id’ attribute removed.

- - Additional SDP offer/answer procedures, related to certificates, added.

- - Updates to RFC 5763 and RFC 7345 added.

- - Transport protocol considerations added.

Changes from draft-ietf-mmusic-sdp-dtls-00

- - SDP ’connection’ attribute replaced with new ’dtls-association-id’ attribute.

- - IANA Considerations added.

- o - E-mail regarding 'dtls-association-id-id' attribute added as Annex.

Changes from draft-holmberg-mmusic-sdp-dtls-01

- o - draft-ietf-mmusic version of draft submitted.
- o - Draft file name change (sdp-dtls -> dtls-sdp) due to collision with another expired draft.
- o - Clarify that if ufrag in offer is unchanged, it must be unchanged in associated answer.
- o - SIP Considerations section added.
- o - Section about multiple SDP fingerprint attributes added.

Changes from draft-holmberg-mmusic-sdp-dtls-00

- o - Editorial changes and clarifications.

14. References

14.1. Normative References

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

[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, <http://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, <http://www.rfc-editor.org/info/rfc3264>.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <http://www.rfc-editor.org/info/rfc4145>.

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

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <http://www.rfc-editor.org/info/rfc4572>.

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

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <http://www.rfc-editor.org/info/rfc5763>.

[RFC7345] Holmberg, C., Sedlacek, I., and G. Salgueiro, "UDP Transport Layer (UDPTL) over Datagram Transport Layer Security (DTLS)", RFC 7345, DOI 10.17487/RFC7345, August 2014, <http://www.rfc-editor.org/info/rfc7345>.

14.2. Informative References

[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, <http://www.rfc-editor.org/info/rfc5576>.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, DOI 10.17487/RFC6083, January 2011, <http://www.rfc-editor.org/info/rfc6083>.

[I-D.ietf-mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-12 (work in progress), January 2016.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-25 (work in progress), January 2016.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Roman Shpount
TurboBridge
4905 Del Ray Avenue, Suite 300
Bethesda, MD 20814
USA

Phone: +1 (240) 292-6632
Email: rshpount@turbobridge.com

Network Working Group
Internet-Draft
Updates: 5763,7345 (if approved)
Intended status: Standards Track
Expires: May 2, 2018

C. Holmberg
Ericsson
R. Shpount
TurboBridge

October 29, 2017

Session Description Protocol (SDP) Offer/Answer Considerations for Datagram Transport Layer Security (DTLS) and Transport Layer Security (TLS)

draft-ietf-mmusic-dtls-sdp-32.txt

Abstract

This document defines the Session Description Protocol (SDP) offer/answer procedures for negotiating and establishing a Datagram Transport Layer Security (DTLS) association. The document also defines the criteria for when a new DTLS association must be established. The document updates RFC 5763 and RFC 7345, by replacing common SDP offer/answer procedures with a reference to this specification.

This document defines a new SDP media-level attribute, 'tls-id'.

This document also defines how the 'tls-id' attribute can be used for negotiating and establishing a Transport Layer Security (TLS) connection, in conjunction with the procedures in RFC 4145 and RFC 8122.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 2, 2018.

Table of Contents

1. Introduction

[RFC5763] defines Session Description Protocol (SDP) offer/answer procedures for Secure Realtime Transport Protocol Using Datagram Transport Layer Security (DTLS-SRTP). [RFC7345] defines SDP offer/answer procedures for UDP Transport Layer over Datagram Transport Layer Security (UDPTL-DTLS). This specification defines general offer/answer procedures for DTLS, based on the procedures in [RFC5763]. Other specifications, defining specific DTLS usages, can then reference this specification, in order to ensure that the DTLS aspects are common among all usages. Having common procedures is essential when multiple usages share the same DTLS association [I-D.ietf-mmusic-sdp-bundle-negotiation]. The document updates [RFC5763] and [RFC7345], by replacing common SDP offer/answer procedures with a reference to this specification.

NOTE: Since the publication of [RFC5763], [RFC4474] has been obsoleted by [I-D.ietf-stir-rfc4474bis]. The updating of the references (and the associated procedures) within [RFC5763] is outside the scope of this document. However, implementers of [RFC5763] applications are encouraged to implement [I-D.ietf-stir-rfc4474bis] instead of [RFC4474].

As defined in [RFC5763], a new DTLS association MUST be established when transport parameters are changed. Transport parameter change is not well defined when Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] is used. One possible way to determine a transport change is based on ufrag [I-D.ietf-ice-rfc5245bis] change, but the ufrag value is changed both when ICE is negotiated and when ICE restart [I-D.ietf-ice-rfc5245bis] occurs. These events do not always require a new DTLS association to be established, but previously there was no way to explicitly indicate in an SDP offer or answer whether a new DTLS association is required. To solve that problem, this document defines a new SDP attribute, 'tls-id'. The pair of SDP 'tls-id' attribute values (the attribute values of the offerer and the answerer) uniquely identifies the DTLS association. Providing a new value of the 'tls-id' attribute in an SDP offer or answers can be used to indicate whether a new DTLS association is to be established.

The SDP 'tls-id' attribute can be specified when negotiating a Transport Layer Security (TLS) connection, using the procedures in this document in conjunction with the procedures in [RFC5763] and

[RFC8122]. The unique combination of SDP 'tls-id' attribute values can be used to identity the negotiated TLS connection. The unique value can be used, for example, within TLS protocol extensions to differentiate between multiple TLS connections and correlate those connections with specific offer/answer exchanges. The TLS specific considerations are described in Section 7.

2. Conventions

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. Establishing a new DTLS Association

3.1. General

A new DTLS association must be established between two endpoints after a successful SDP offer/answer exchange in the following cases:

- The negotiated DTLS setup roles change; or
- One or more fingerprint values are modified, added or removed in either an SDP offer or answer; or
- The intent to establish a new DTLS association is explicitly signaled using SDP, by changing the value of the SDP 'tls-id' attribute defined in this document;

NOTE: The first two items above are based on the procedures in [RFC5763]. This specification adds the support for explicit signaling using the SDP 'tls-id' attribute.

A new DTLS association can only be established as a result of the successful SDP offer/answer exchange. Whenever an entity determines that a new DTLS association is required, the entity MUST initiate an SDP offer/answer exchange, following the procedures in Section 5.

The sections below describe typical cases where a new DTLS association needs to be established.

In this document, a "new DTLS association" between two endpoints refers to either an initial DTLS association (when no DTLS association is currently established between the endpoints) or an DTLS association replacing a previously established DTLS association.

3.2. Change of Local Transport Parameters

If an endpoint modifies its local transport parameters (address and/or port), and if the modification requires a new DTLS association, the endpoint MUST change its local SDP 'tls-id' attribute value (see Section 4).

If the underlying transport protocol prohibits a DTLS association from spanning multiple 5-tuples (transport/source address/source port/destination address/destination port), and if the 5-tuple is changed, the endpoint MUST change its local SDP 'tls-id' attribute value (see Section 4). An example of such a case is when DTLS is carried over the Stream Control Transmission Protocol (SCTP), as described in [RFC6083].

3.3. Change of ICE ufrag value

If an endpoint uses ICE, and modifies a local ufrag value, and if the modification requires a new DTLS association, the endpoint MUST change its local SDP 'tls-id' attribute value (see Section 4).

4. SDP tls-id Attribute

The pair of SDP 'tls-id' attribute values (the attribute values of the offerer and the answerer) uniquely identifies the DTLS association or TLS connection.

Name: tls-id

Value: tls-id-value

Usage Level: media

Charset Dependent: no

Default Value: N/A

Syntax:

```
tls-id-value = 20*255(tls-id-char)
tls-id-char = ALPHA / DIGIT / "+" / "/" / "-" / "_"

<ALPHA and DIGIT defined in [RFC4566]>
```

Example:

```
a=tls-id:abc3de65cddef001be82
```

Every time an endpoint requests to establish a new DTLS association, the endpoint MUST generate a new local 'tls-id' attribute value. A non-changed local 'tls-id' attribute value, in combination with non-changed fingerprints, indicates that the endpoint intends to reuse the existing DTLS association.

The 'tls-id' attribute value MUST be generated using a strong random function and include at least 120 bits of randomness.

No default value is defined for the SDP 'tls-id' attribute. Implementations that wish to use the attribute MUST explicitly include it in SDP offers and answers. If an offer or answer does not contain a 'tls-id' attribute (this could happen if the offerer or answerer represents an existing implementation that has not been updated to support the 'tls-id' attribute), unless there is another mechanism to explicitly indicate that a new DTLS association is to be established, a modification of one or more of the following characteristics MUST be treated as an indication that an endpoint wants to establish a new DTLS association:

- DTLS setup role; or
- fingerprint set; or
- local transport parameters

NOTE: A modification of the ufrag value is not treated as an indication that an endpoint wants to establish a new DTLS assocation. In order to indicate that a new DTLS association is to be established, one or more of the characteristics listed above have to be modified.

The mux category [I-D.ietf-mmusic-sdp-mux-attributes] for the 'tls-id' attribute is 'IDENTICAL', which means that the attribute value applies to all media descriptions being multiplexed [I-D.ietf-mmusic-sdp-bundle-negotiation]. However, as described in [I-D.ietf-mmusic-sdp-bundle-negotiation], in order to avoid duplication the attribute is only associated with the "m=" line representing the offerer/answerer BUNDLE-tag.

For RTP-based media, the 'tls-id' attribute applies to the whole associated media description. The attribute MUST NOT be defined per source (using the SDP 'ssrc' attribute [RFC5576]).

The SDP offer/answer [RFC3264] procedures associated with the attribute are defined in Section 5.

5. SDP Offer/Answer Procedures

5.1. General

This section defines the generic SDP offer/answer procedures for negotiating a DTLS association. Additional procedures (e.g., regarding usage of specific SDP attributes etc.) for individual DTLS usages (e.g., DTLS-SRTP) are outside the scope of this specification, and need to be specified in a usage specific specification.

NOTE: The procedures in this section are generalizations of procedures first specified in DTLS-SRTP [RFC5763], with the addition of usage of the SDP 'tls-id' attribute. That document is herein updated to make use of these new procedures.

The procedures in this section apply to an SDP media description ("m=" line) associated with DTLS-protected media/data.

When an offerer or answerer indicates that it wants to establish a new DTLS association, it needs to make sure that media packets associated with any previously established DTLS association and the new DTLS association can be de-multiplexed. In case of an ordered transport (e.g., SCTP) this can be done simply by sending packets for the new DTLS association after all packets associated with a previously established DTLS association has been sent. In case of an unordered transport, such as UDP, packets associated with a previously established DTLS association can arrive after the answer

SDP was received and after the first packets associated with the new DTLS association were received. The only way to de-multiplex packets associated with with a previously established DTLS association and the new DTLS association is on the basis of the 5-tuple. Because of this, if an unordered transport is used for the DTLS association, a new 3-tuple (transport/source address/source port) MUST be allocated by at least one of the endpoints so that DTLS packets can be de-multiplexed.

When an offerer needs to establish a new DTLS association, and if an unordered transport (e.g., UDP) is used, the offerer MUST allocate a new 3-tuple for the offer in such a way that the offerer can disambiguate any packets associated with the new DTLS association from any packets associated with any other DTLS association. This typically means using a local address and/or port, or a set of ICE candidates (see Section 6), which were not recently used for any other DTLS association.

When an answerer needs to establish a new DTLS association, if an unordered transport is used, and if the offerer did not allocate a new 3-tuple, the answerer MUST allocate a new 3-tuple for the answer in such a way that it can disambiguate any packets associated with the new DTLS association from any packets associated with any other DTLS association. This typically means using a local address and/or port, or a set of ICE candidates (see Section 6), which were not recently used for any other DTLS association.

In order to negotiate a DTLS association, the following SDP attributes are used:

- The SDP ’setup’ attribute, defined in [RFC4145], is used to negotiate the DTLS roles;
- The SDP ’fingerprint’ attribute, defined in [RFC8122], is used to provide one or more fingerprint values; and
- The SDP ’tls-id’ attribute, defined in this specification, is used to identity the DTLS association.

This specification does not define the usage of the SDP ’connection’ attribute [RFC4145] for negotiating a DTLS association. However, the attribute MAY be used if the DTLS association is used together with another protocol (e.g., SCTP or TCP) for which the usage of the attribute has been defined.

Unlike for TCP and TLS connections, endpoints MUST NOT use the SDP ’setup’ attribute ’holdconn’ value when negotiating a DTLS association.

Endpoints MUST support the hash functions as defined in [RFC8122].

The certificate received during the DTLS handshake [RFC6347] MUST match a certificate fingerprint received in SDP 'fingerprint' attributes according to the procedures defined in [RFC8122]. If fingerprints do not match the hashed certificate, then an endpoint MUST tear down the media session immediately (see [RFC8122]).

SDP offerers and answerers might reuse certificates across multiple DTLS associations, and provide identical fingerprint values for each DTLS association. The combination of the SDP 'tls-id' attribute values of the SDP offerer and answerer identifies each individual DTLS association.

NOTE: There are cases where the SDP 'tls-id' attribute value generated by the offerer will end up being used for multiple DTLS associations. For that reason the combination of the attribute values of the offerer and answerer is needed in order to identity a DTLS association. An example of such case is where the offerer sends an updated offer (Section 5.5), without modifying its attribute value, but the answerer determines that a new DTLS association is to be created. The answerer will generate a new local attribute value for the new DTLS association (Section 5.3), while the offerer will use the same attribute value that it used for the current association. Another example is when the Session Initiation Protocol (SIP) [RFC3261] is used for signalling, and an offer is forked to multiple answerers. The attribute value generated by the offerer will be used for DTLS associations established by each answerer.

5.2. Generating the Initial SDP Offer

When an offerer sends the initial offer, the offerer MUST insert an SDP 'setup' attribute [RFC4145] with an 'actpass' attribute value, and one or more SDP 'fingerprint' attributes according to the procedures in [RFC8122]. In addition, the offerer MUST insert in the offer an SDP 'tls-id' attribute with a unique attribute value.

As the offerer inserts the SDP 'setup' attribute with an 'actpass' attribute value, the offerer MUST be prepared to receive a DTLS ClientHello message [RFC6347] (if a new DTLS association is established by the answerer) from the answerer before the offerer receives the SDP answer.

If the offerer receives a DTLS ClientHello message, and a DTLS association is established, before the offerer receives the SDP Answer carrying the fingerprint associated with the DTLS association, any data received on the DTLS association before the fingerprint MUST be considered coming from an unverified source. The processing of

such data, and sending of data by the offerer to the unverified source, is outside the scope of this document.

5.3. Generating the Answer

When an answerer sends an answer, the answerer MUST insert in the answer an SDP 'setup' attribute according to the procedures in [RFC4145], and one or more SDP 'fingerprint' attributes according to the procedures in [RFC8122]. If the answerer determines, based on the criteria specified in Section 3.1, that a new DTLS association is to be established, the answerer MUST insert in the associated answer an SDP 'tls-id' attribute with a new unique attribute value. Note that the offerer and answerer generate their own local 'tls-id' attribute values, and the combination of both values identify the DTLS association.

If the answerer receives an offer that requires establishment of a new DTLS association, and if the answerer does not accept the establishment of a new DTLS association, the answerer MUST reject the "m=" lines associated with the suggested DTLS association [RFC3264].

If an answerer receives an offer that does not require the establishment of a new DTLS association, and if the answerer determines that a new DTLS association is not to be established, the answerer MUST insert an SDP 'tls-id' attribute with the previously assigned attribute value in the associated answer. In addition, the answerer MUST insert an SDP 'setup' attribute with an attribute value that does not change the previously negotiated DTLS roles, and one or more SDP 'fingerprint' attributes values that do not change the previously sent fingerprint set, in the associated answer.

If the answerer receives an offer that does not contain an SDP 'tls-id' attribute, the answerer MUST NOT insert a 'tls-id' attribute in the answer.

If a new DTLS association is to be established, and if the answerer inserts an SDP 'setup' attribute with an 'active' attribute value in the answer, the answerer MUST initiate a DTLS handshake [RFC6347]) by sending a DTLS ClientHello message towards the offerer.

Even though an offerer is required to insert an 'SDP' setup attribute with an 'actpass' attribute value in initial offers (Section 5.2) and subsequent offers (Section 5.5), the answerer MUST be able to receive initial and subsequent offers with other attribute values, in order to be backward compatible with older implementations that might insert other attribute values in initial and subsequent offers.

5.4. Offerer Processing of the SDP Answer

When an offerer receives an answer that establishes a new DTLS association based on criteria defined in Section 3.1, and if the offerer becomes DTLS client (based on the value of the SDP 'setup' attribute value [RFC4145]), the offerer MUST establish a DTLS association. If the offerer becomes DTLS server, it MUST wait for the answerer to establish the DTLS association.

If the offerer indicated a desire to reuse an existing DTLS association and the answerer does not request the establishment of a new DTLS association, the offerer will continue to use the previously established DTLS association.

A new DTLS association can be established based on changes in either an SDP offer or answer. When communicating with legacy endpoints, an offerer can receive an answer that includes the same fingerprint set and setup role. A new DTLS association will still be established if such an answer was received as a response to an offer which requested the establishment of a new DTLS association, as the transport parameters would have been changed in the offer.

5.5. Modifying the Session

When an offerer sends a subsequent offer, and if the offerer wants to establish a new DTLS association, the offerer MUST insert an SDP 'setup' attribute [RFC4145] with an 'actpass' attribute value, and one or more SDP 'fingerprint' attributes according to the procedures in [RFC8122]. In addition, the offerer MUST insert in the offer an SDP 'tls-id' attribute with a new unique attribute value.

When an offerer sends a subsequent offer, and the offerer does not want to establish a new DTLS association, and if a previously established DTLS association exists, the offerer MUST insert an SDP 'setup' attribute with an 'actpass' attribute value, and one or more SDP 'fingerprint' attributes with attribute values that do not change the previously sent fingerprint set, in the offer. In addition, the offerer MUST insert an SDP 'tls-id' attribute with the previously assigned attribute value in the offer.

NOTE: When a new DTLS association is being established, each endpoint needs to be prepared to receive data on both the new and old DTLS associations as long as both are alive.

6. ICE Considerations

When the Interactive Connectivity Establishment (ICE) mechanism [I-D.ietf-ice-rfc5245bis] is used, the ICE connectivity checks are performed before the DTLS handshake begins. Note that if aggressive nomination mode is used, multiple candidate pairs may be marked valid before ICE finally converges on a single candidate pair.

NOTE: Aggressive nomination has been deprecated from ICE, but must still be supported for backwards compatibility reasons [I-D.ietf-ice-rfc5245bis].

When a new DTLS association is established over an unordered transport, in order to disambiguate any packets associated with the newly established DTLS association, at least one of the endpoints MUST allocate a completely new set of ICE candidates which were not recently used for any other DTLS association. This means the answerer cannot initiate a new DTLS association unless the offerer initiated ICE restart [I-D.ietf-ice-rfc5245bis]. If the answerer wants to initiate a new DTLS association, it needs to initiate an ICE restart and a new offer/answer exchange on its own. However, an ICE restart does not by default require a new DTLS association to be established.

NOTE: Simple Traversal of the UDP Protocol through NAT (STUN) packets are sent directly over UDP, not over DTLS. [RFC7983] describes how to demultiplex STUN packets from DTLS packets and SRTP packets.

Each ICE candidate associated with a component is treated as being part of the same DTLS association. Therefore, from a DTLS perspective it is not considered a change of local transport parameters when an endpoint switches between those ICE candidates.

7. TLS Considerations

The procedures in this document can also be used for negotiating and establishing a TLS connection, with the restriction described below.

As specified in [RFC4145], the SDP ’connection’ attribute is used to indicate whether to establish a new TLS connection. An offerer and answerer MUST ensure that the ’connection’ attribute value and the ’tls-id’ attribute value does not cause a conflict regarding whether a new TLS connection is to be established or not.

NOTE: Even though the SDP ’connection’ attribute can be used to indicate whether a new TLS connection is to be established, the unique combination of SDP ’tls-id’ attribute values can be used to identity a TLS connection. The unique value can be used e.g., within

TLS protocol extensions to differentiate between multiple TLS connections and correlate those connections with specific offer/answer exchanges. One such extension is defined in [I-D.ietf-mmusic-sdp-uks].

If an offerer or answerer inserts an SDP 'connection' attribute with a 'new' value in the offer/answer and also inserts an SDP 'tls-id' attribute, the value of tls-id' attribute MUST be new and unique.

If an offerer or answerer inserts an SDP 'connection' attribute with a 'existing' value in the offer/answer, if a previously established TLS connection exists, and if the offerer/answerer previously inserted an SDP 'tls-id' attribute associated with the same TLS connection in an offer/answer, the offerer/answerer MUST also insert an SDP 'tls-id' attribute with the previously assigned value in the offer/answer.

If an offerer or answerer receives an offer/answer with conflicting attribute values, the offerer/answerer MUST process the offer/answer as misformed.

An endpoint MUST NOT make assumptions regarding the support of the SDP 'tls-id' attribute by the peer. Therefore, to avoid ambiguity, both offerers and answerers MUST always use the 'connection' attribute in conjunction with the 'tls-id' attribute.

NOTE: As defined in [RFC4145], if the SDP 'connection' attribute is not explicitly present, the implicit default value is 'new'.

The SDP example below is based on the example in section 3.4 of [RFC8122], with the addition of the SDP 'tls-id' attribute.

```
m=image 54111 TCP/TLS t38
c=IN IP4 192.0.2.2
a=tls-id:abc3de65cddef001be82
a=setup:passive
a=connection:new
a=fingerprint:SHA-256 \
 12:DF:3E:5D:49:6B:19:E5:7C:AB:4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF: \
 3E:5D:49:6B:19:E5:7C:AB:4A:AD
a=fingerprint:SHA-1 \
 4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

8. SIP Considerations

When the Session Initiation Protocol (SIP) [RFC3261] is used as the signal protocol for establishing a multimedia session, dialogs [RFC3261] might be established between the caller and multiple callees. This is referred to as forking. If forking occurs, separate DTLS associations will be established between the caller and each callee.

When forking occurs, an SDP offerer can receive DTLS ClientHello messages and SDP answerers from multiple remote locations. Because of this, the offerer might have to wait for multiple SDP answers (from different remote locations) until it receives a certificate fingerprint that matches the certificate associated with a specific DTLS handshake. The offerer MUST NOT declare a fingerprint mismatch until it determines that it will not receive SDP answers from any additional remote locations.

It is possible to send an INVITE request which does not contain an SDP offer. Such an INVITE request is often referred to as an 'empty INVITE', or an 'offer-less INVITE'. The receiving endpoint will include the SDP offer in a response to the request. When the endpoint generates such SDP offer, if a previously established DTLS association exists, the offerer MUST insert an SDP 'tls-id' attribute, and one or more SDP 'fingerprint' attributes, with previously assigned attribute values. If a previously established DTLS association did not exist, the offer MUST be generated based on the same rules as a new offer (see Section 5.2). Regardless of the previous existence of a DTLS association, the SDP 'setup' attribute MUST be included according to the rules defined in [RFC4145]. Furthermore, if ICE is used, according to the third party call control considerations described in [I-D.ietf-mmusic-ice-sip-sdp], ICE restart MUST be initiated.

9. RFC Updates

9.1. General

This section updates specifications that use DTLS-protected media, in order to reflect the procedures defined in this specification.

9.2. Update to RFC 5763

9.2.1. Update to section 1

The reference to [RFC4572] is replaced with a reference to [RFC8122].

9.2.2. Update to section 5

The text in section 5 (Establishing a Secure Channel) is modified by replacing generic SDP offer/answer procedures for DTLS with a reference to this specification:

NEW TEXT:

The two endpoints in the exchange present their identities as part of the DTLS handshake procedure using certificates. This document uses certificates in the same style as described in "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)" [RFC8122].

If self-signed certificates are used, the content of the subjectAltName attribute inside the certificate MAY use the uniform resource identifier (URI) of the user. This is useful for debugging purposes only and is not required to bind the certificate to one of the communication endpoints. The integrity of the certificate is ensured through the fingerprint attribute in the SDP.

The generation of public/private key pairs is relatively expensive. Endpoints are not required to generate certificates for each session.

The offer/answer model, defined in [RFC3264], is used by protocols like the Session Initiation Protocol (SIP) [RFC3261] to set up multimedia sessions.

When an endpoint wishes to set up a secure media session with another endpoint, it sends an offer in a SIP message to the other endpoint. This offer includes, as part of the SDP payload, a fingerprint of a certificate that the endpoint wants to use. The endpoint SHOULD send the SIP message containing the offer to the offerer’s SIP proxy over an integrity protected channel. The proxy SHOULD add an Identity header field according to the procedures outlined in [RFC4474]. When the far endpoint receives the SIP message, it can verify the identity of the sender using the Identity header field. Since the Identity header field is a digital signature across several SIP header fields, in addition to the body of the SIP message, the receiver can also be certain that the message has not been tampered with after the digital signature was applied and added to the SIP message.

The far endpoint (answerer) may now establish a DTLS association with the offerer. Alternately, it can indicate in its answer that the offerer is to initiate the DTLS association. In either case, mutual DTLS certificate-based authentication will be used. After completing

the DTLS handshake, information about the authenticated identities, including the certificates, are made available to the endpoint application. The answerer is then able to verify that the offerer's certificate used for authentication in the DTLS handshake can be associated to a certificate fingerprint contained in the offer in the SDP. At this point, the answerer may indicate to the end user that the media is secured. The offerer may only tentatively accept the answerer's certificate since it may not yet have the answerer's certificate fingerprint.

When the answerer accepts the offer, it provides an answer back to the offerer containing the answerer's certificate fingerprint. At this point, the offerer can accept or reject the peer's certificate and the offerer can indicate to the end user that the media is secured.

Note that the entire authentication and key exchange for securing the media traffic is handled in the media path through DTLS. The signaling path is only used to verify the peers' certificate fingerprints.

The offerer and answerer MUST follow the SDP offer/answer procedures defined in [RFCXXXX].

9.2.3. Update to section 6.6

The text in section 6.6 (Session Modification) is modified by replacing generic SDP offer/answer procedures for DTLS with a reference to this specification:

NEW TEXT:

> Once an answer is provided to the offerer, either endpoint MAY request a session modification that MAY include an updated offer. This session modification can be carried in either an INVITE or UPDATE request. The peers can reuse an existing DTLS association, or establish a new one, following the procedures in [RFCXXXX].

9.2.4. Update to section 6.7.1

The text in section 6.7.1 (ICE Interaction) is modified by replacing the ICE procedures with a reference to this specification:

NEW TEXT:

The Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] considerations for DTLS-protected media are described in [RFCXXXX].

9.3. Update to RFC 7345

9.3.1. Update to section 4

The subsections (4.1.-4.5.) in section 4 (SDP Offerer/Answerer Procedures) are removed, and replaced with the new text below:

NEW TEXT:

An endpoint (i.e., both the offerer and the answerer) MUST create an SDP media description ("m=" line) for each UDPTL-over-DTLS media stream and MUST assign a UDP/TLS/UDPTL value (see Table 1) to the "proto" field of the "m=" line.

The offerer and answerer MUST follow the SDP offer/answer procedures defined in [RFCXXXX] in order to negotiate the DTLS association associated with the UDPTL-over-DTLS media stream. In addition, the offerer and answerer MUST use the SDP attributes defined for UDPTL over UDP, as defined in [ITU.T38.2010].

9.3.2. Update to section 5.2.1

The text in section 5.2.1 (ICE Usage) is modified by replacing the ICE procedures with a reference to this specification:

NEW TEXT:

The Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] considerations for DTLS-protected media are described in [RFCXXXX].

[RFC EDITOR NOTE: Throughout the document, please replace RFCXXXX with the RFC number of this document.]

9.3.3. Update to section 10.1

A reference to [RFC8122] is added to section 10.1 (Normative References):

NEW TEXT:

[RFC8122] Lennox, J. and C. Holmberg, "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 8122, DOI 10.17487/RFC8122, March 2017, <http://www.rfc-editor.org/info/rfc8122>.

10. Security Considerations

This specification does not modify the security considerations associated with DTLS, or the SDP offer/answer mechanism. In addition to the introduction of the SDP 'tls-id' attribute, the specification simply clarifies the procedures for negotiating and establishing a DTLS association.

This specification does not modify the actual TLS connection setup procedures. The SDP 'tls-is' attribute as such cannot be used to correlate an SDP Offer/Answer exchange with a TLS connection setup. Thus, this draft does not introduce new security considerations related to correlating an SDP Offer/Answer exchange with a TLS connection setup.

11. IANA Considerations

This document updates the "Session Description Protocol Parameters" registry as specified in Section 8.2.2 of [RFC4566]. Specifically, it adds the SDP 'tls-id' attribute to the table for SDP media level attributes.

Attribute name: tls-id
Type of attribute: media-level
Subject to charset: no
Purpose: Indicates whether a new DTLS association or TLS connection is to be established/re-established.
Appropriate Values: see Section 4
Contact name: Christer Holmberg
Mux Category: IDENTICAL

12. Acknowledgements

Thanks to Justin Uberti, Martin Thomson, Paul Kyzivat, Jens Guballa, Charles Eckel, Gonzalo Salgueiro and Paul Jones for providing comments and suggestions on the document. Ben Campbell performed an AD review. Paul Kyzivat performed a gen-art review.

13. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-dtls-31

- Changes based on IESG comments from Eric R

Changes from draft-ietf-mmusic-sdp-dtls-30

- Changes based on IESG comments from Mirja K

Changes from draft-ietf-mmusic-sdp-dtls-29

- Removal of ufrag value change as a trigger for a new DTLS association

Changes from draft-ietf-mmusic-sdp-dtls-28

- Changes based on IESG review by Adam Roach, Eric Rescorla, Alexey Melnikov and Mirja Kuhlewind:
- - Document title changed
- - Transport Protocol Considerations section removed
- - Additional text to Security Considerations section
- - Editorial changes

Changes from draft-ietf-mmusic-sdp-dtls-27

- Reference fixes based on Gen-ART review by Paul Kyzivat.

Changes from draft-ietf-mmusic-sdp-dtls-26

- Editorial fixes based on Gen-ART review by Paul Kyzivat.

Changes from draft-ietf-mmusic-sdp-dtls-25

- Minor editorial nits.

Changes from draft-ietf-mmusic-sdp-dtls-24

- Changes based on 2nd WGLC comments from Roman S and Martin T:
- - RFC update structure shortened (old text removed).
- - Guidance regarding receiving ClientHello before SDP answer added.
- - Additional SIP considerations regarding forking.
- - SDP setup attribute value restriction in initial and subsequent offers based on comment from Ekr.

Changes from draft-ietf-mmusic-sdp-dtls-23

- Editorial change to make it clear that the document does not modify the procedures in RFC 8122.

Changes from draft-ietf-mmusic-sdp-dtls-22

- Support for TLS added.
- Editorial changes based on sec-dir review by Rich Salz.
- Editorial changes based on gen-art review by Paul Kyzivat.
- Editorial changes based on ops-dir review by Carlos Pignataro.

Changes from draft-ietf-mmusic-sdp-dtls-21

- Changes based on AD review by Ben Campbell.
- (https://www.ietf.org/mail-archive/web/mmusic/current/msg17707.html)

Changes from draft-ietf-mmusic-sdp-dtls-20

- Change to length and randomness of tls-id attribute value.

Changes from draft-ietf-mmusic-sdp-dtls-19

- Change based on comment from Roman.

Changes from draft-ietf-mmusic-sdp-dtls-18

- Changes based on comments from Flemming.

- o - Change in tls-id value definition.
- o - Editorial fixes.

Changes from draft-ietf-mmusic-sdp-dtls-17

- o Reference fix.

Changes from draft-ietf-mmusic-sdp-dtls-16

- o Editorial changes based on 2nd WGLC comments from Christian Groves and Nevenka Biondic.

Changes from draft-ietf-mmusic-sdp-dtls-15

- o tls-id attribute value made globally unique

Changes from draft-ietf-mmusic-sdp-dtls-14

- o Changes based on comments from Flemming:
- o - Additional dtls-is clarifications
- o - Editorial fixes

Changes from draft-ietf-mmusic-sdp-dtls-13

- o Text about the updated RFCs added to Abstract and Introduction
- o Reference to RFC 5763 removed from section 6 (ICE Considerations)
- o Reference to RFC 5763 removed from section 8 (SIP Considerations)

Changes from draft-ietf-mmusic-sdp-dtls-12

- o "unreliable" changed to "unordered"

Changes from draft-ietf-mmusic-sdp-dtls-11

- o Attribute name changed to tls-id
- o Additional text based on comments from Roman Shpount.

Changes from draft-ietf-mmusic-sdp-dtls-10

- o Modified document to use tls-id instead of dtls-connection

- Changes are based on comments from Eric Rescorla, Justin Uberti, and Paul Kyzivat.

Changes from draft-ietf-mmusic-sdp-dtls-08

- Offer/Answer section modified in order to allow sending of multiple SDP 'fingerprint' attributes.

- Terminology made consistent: 'DTLS connection' replaced with 'DTLS association'.

- Editorial changes based on comments from Paul Kyzivat.

Changes from draft-ietf-mmusic-sdp-dtls-07

- Reference to RFC 7315 replaced with reference to RFC 7345.

Changes from draft-ietf-mmusic-sdp-dtls-06

- Text on restrictions regarding spanning a DTLS association over multiple transports added.

- Mux category added to IANA Considerations.

- Normative text regarding mux category and source-specific applicability added.

- Reference to RFC 7315 added.

- Clarified that offerer/answerer that has not been updated to support this specification will not include the tls-id attribute in offers and answers.

- Editorial corrections based on WGLC comments from Charles Eckel.

Changes from draft-ietf-mmusic-sdp-dtls-05

- Text on handling offer/answer error conditions added.

Changes from draft-ietf-mmusic-sdp-dtls-04

- Editorial nits fixed based on comments from Paul Kyzivat:

Changes from draft-ietf-mmusic-sdp-dtls-03

- Changes based on comments from Paul Kyzivat:

- - Modification of tls-id attribute section.

- o - Removal of IANA considerations subsection.
- o - Making note into normative text in o/a section.
- o Changes based on comments from Martin Thompson:
- o - Abbreviations section removed.
- o - Clarify that a new DTLS association requires a new o/a transaction.

Changes from draft-ietf-mmusic-sdp-dtls-02

- o - Updated RFCs added to boilerplate.

Changes from draft-ietf-mmusic-sdp-dtls-01

- o - Annex regarding 'tls-id-id' attribute removed.
- o - Additional SDP offer/answer procedures, related to certificates, added.
- o - Updates to RFC 5763 and RFC 7345 added.
- o - Transport protocol considerations added.

Changes from draft-ietf-mmusic-sdp-dtls-00

- o - SDP 'connection' attribute replaced with new 'tls-id' attribute.
- o - IANA Considerations added.
- o - E-mail regarding 'tls-id-id' attribute added as Annex.

Changes from draft-holmberg-mmusic-sdp-dtls-01

- o - draft-ietf-mmusic version of draft submitted.
- o - Draft file name change (sdp-dtls -> dtls-sdp) due to collision with another expired draft.
- o - Clarify that if ufrag in offer is unchanged, it must be unchanged in associated answer.
- o - SIP Considerations section added.
- o - Section about multiple SDP fingerprint attributes added.

Changes from draft-holmberg-mmusic-sdp-dtls-00

o - Editorial changes and clarifications.

14. References

14.1. Normative References

[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>.

[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>.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <https://www.rfc-editor.org/info/rfc4145>.

[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>.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <https://www.rfc-editor.org/info/rfc5763>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC7345] Holmberg, C., Sedlacek, I., and G. Salgueiro, "UDP Transport Layer (UDPTL) over Datagram Transport Layer Security (DTLS)", RFC 7345, DOI 10.17487/RFC7345, August 2014, <https://www.rfc-editor.org/info/rfc7345>.

[RFC8122] Lennox, J. and C. Holmberg, "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 8122, DOI 10.17487/RFC8122, March 2017, <https://www.rfc-editor.org/info/rfc8122>.

[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>.

[I-D.ietf-ice-rfc5245bis] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-13 (work in progress), October 2017.

[I-D.ietf-mmusic-sdp-mux-attributes] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-16 (work in progress), December 2016.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-39 (work in progress), August 2017.

14.2. Informative References

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <https://www.rfc-editor.org/info/rfc4474>.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <https://www.rfc-editor.org/info/rfc4572>.

[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>.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, DOI 10.17487/RFC6083, January 2011, <https://www.rfc-editor.org/info/rfc6083>.

[RFC7983] Petit-Huguenin, M. and G. Salgueiro, "Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)", RFC 7983, DOI 10.17487/RFC7983, September 2016, <https://www.rfc-editor.org/info/rfc7983>.

[I-D.ietf-stir-rfc4474bis] Peterson, J., Jennings, C., Rescorla, E., and C. Wendt, "Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-stir-rfc4474bis-16 (work in progress), February 2017.

[I-D.ietf-mmusic-ice-sip-sdp] Petit-Huguenin, M., Keranen, A., and S. Nandakumar, "Session Description Protocol (SDP) Offer/Answer procedures for Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-sip-sdp-14 (work in progress), October 2017.

[I-D.ietf-mmusic-sdp-uks] Thomson, M. and E. Rescorla, "Unknown Key Share Attacks on uses of Transport Layer Security with the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-uks-00 (work in progress), August 2017.

[ITU.T38.2010] International Telecommunications Union, "Procedures for real-time Group 3 facsimile communication over IP networks", ITU-T Recommendation T.38, September 2010.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Roman Shpount
TurboBridge
4905 Del Ray Avenue, Suite 300
Bethesda, MD 20814
USA

Phone: +1 (240) 292-6632
Email: rshpount@turbobridge.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2016

C. Holmberg
Ericsson

February 15, 2016

Indicating Exclusive Support of RTP/RTCP Multiplexing using SDP
draft-ietf-mmusic-mux-exclusive-03

Abstract

This document defines a new SDP media-level attribute, 'rtcp-mux-exclusive', that can be used by an endpoint to indicate exclusive support of RTP/RTCP multiplexing.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on August 18, 2016.

Copyright Notice

Table of Contents

1. Introduction

[RFC5761] defines how to multiplex RTP and RTCP on a single IP address and port, referred to as RTP/RTCP multiplexing. [RFC5761] also defines an Session Description Protocol (SDP) [RFC4566] attribute, 'rtcp-mux' that can be used by entities to indicate support, and negotiate usage of, RTP/RTCP multiplexing.

As defined in [RFC5761], if the peer endpoint does not support RTP/RTCP multiplexing, there must be a fallback to usage of separate ports for RTP and RTCP.

The RTCWEB WG has defined that support of the fallback mentioned above is optional. Therefore, there is a need for a mechanism that allows an endpoint to indicate exclusive support of RTP/RTCP multiplexing, meaning that endpoint only supports RTP/RTCP multiplexing and is not able to fallback to usage of separate ports for RTP and RTCP.

This document defines a new SDP media-level attribute, 'rtcp-mux-exclusive', that can be used to indicate exclusive support of RTP/RTCP multiplexing.

The document also describes the Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] considerations when indicating exclusive support of RTP/RTCP multiplexing.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. SDP rtcp-mux-exclusive Attribute

This section defines a new SDP media-level attribute, 'rtcp-mux-exclusive'.

Name: rtcp-mux-exclusive

Value: N/A

Usage Level: media

Charset Dependent: no

Syntax:

    rtcp-mux-exclusive

Example:

    a=rtcp-mux-exclusive

In an SDP offer, the offerer uses the SDP 'rtcp-mux-exclusive' attribute to indicate exclusive support of RTP/RTCP multiplexing for the RTP-based media associated with the SDP media description ("m=" line).

In an SDP answer, the 'rtcp-mux-exclusive' attribute indicates that the answerer supports, and accepts usage of, RTP/RTCP multiplexing for the RTP-based media associated with the SDP media description ("m=" line).

The usage of the SDP 'rtcp-mux-exclusive' attribute is only defined for RTP-based media.

The mux category [I-D.ietf-mmusic-sdp-mux-attributes] for the 'rtcp-mux-exclusive' attribute is 'NORMAL', which means that the attribute can be associated with an individual media description, even if the media description is multiplexed with other media descriptions [I-D.ietf-mmusic-sdp-bundle-negotiation] with which the attribute is not associated.

The 'rtcp-mux-exclusive' attribute applies to the whole associated media description. The attribute MUST NOT be defined per source (using the SDP 'ssrc' attribute [RFC5576]).

The SDP offer/answer [RFC3264] procedures associated with the attribute are defined in Section 4

4. SDP Offer/Answer Procedures

4.1. General

This section defines the SDP offer/answer [RFC3264] procedures for indicating exclusive support of, and negotiating usage of, RTP/RTCP multiplexing.

The procedures in this section apply to individual RTP-based SDP media descriptions ("m=" lines).

4.2. Generating the Initial SDP Offer

When an offerer sends the initial offer, if the offerer wants to indicate exclusive RTP/RTCP multiplexing for RTP-based media, the offerer MUST associate an SDP 'rtcp-mux-exclusive' attribute with the associated SDP media description ("m=" line).

In addition, if the offerer associates an SDP 'rtcp-mux-exclusive' attribute with an SDP media description ("m=" line), the offerer MUST also associate an SDP 'rtcp-mux' attribute with the same SDP media description ("m=" line), following the procedures in [RFC5761].

If the offerer associates an SDP 'rtcp' attribute [RFC3605] with an SDP media description ("m=" line), and if the offerer also associates an SDP 'rtcp-mux-exclusive' attribute with the same SDP media description ("m=" line), the address and port values of the SDP 'rtcp' attribute MUST match the corresponding values for RTP.

NOTE: This specification does not mandate the usage of the SDP 'rtcp' attribute for RTP/RTCP multiplexing.

4.3. Generating the Answer

When an answerer receives an offer that contains an SDP 'rtcp-mux-exclusive' attribute, associated with an RTP-based SDP media description ("m=" line), if the answerer accepts the usage of RTP/RTCP multiplexing, the answerer MUST associate an SDP 'rtcp-mux-exclusive' attribute with the corresponding SDP media description ("m=") in the associated answer. If the answerer does not accept the usage of RTP/RTCP multiplexing, the answerer MUST either reject the

SDP media description ("m=") by setting the port value to zero in the associated answer, or reject the whole offer, following the procedures in [RFC3264].

In addition, if the answerer associates an SDP 'rtcp-mux-exclusive' attribute with an SDP media description ("m=" line) in the answer, the answerer MUST also associate an SDP 'rtcp-mux' attribute with the same SDP media description ("m=" line), following the procedures in [RFC5761].

4.4. Offerer Processing of the SDP Answer

If an offerer associated an SDP 'rtcp-mux-exclusive' attribute with an RTP-based SDP media description ("m=" line) in an offer, and if the corresponding SDP media description ("m=" line) in the associated answer contains an SDP 'rtcp-mux-exclusive' attribute, and/or an SDP 'rtcp-mux' attribute, the offerer MUST apply the RTP/RTCP multiplexing procedures [RFC5761] to the associated RTP-based media. If the corresponding SDP media description ("m=" line) in the associated answer does not contain an SDP 'rtcp-mux-exclusive' attribute, nor an SDP 'rtcp-mux' attribute, the offerer MUST either take appropriate actions in order to disable the associated RTP-based media, or send a new offer without associating an SDP 'rtcp-mux-exclusive' attribute with the SDP media description ("m=" line).

NOTE: This document does not mandate specific actions on how to terminate the RTP media. The offerer might e.g. send a new offer, where the port value of the SDP media description is set to zero, in order to terminate the RTP media.

4.5. Modifying the Session

When an offerer sends a subsequent offer, if the offerer and answerer have previously negotiated usage of RTP/RTCP multiplexing for the media associated with an RTP-based SDP media description ("m=" line), the offerer SHOULD associate an SDP 'rtcp-mux-exclusive' attribute and an SDP 'rtcp-mux' attribute with the corresponding SDP media description ("m=" line). If the offerer does not associate the attributes with the corresponding SDP media description ("m=" line) it is an indication that the offerer no longer wants to use RTP/RTCP multiplexing, and instead MUST fallback to usage of separate ports for RTP and RTCP once the offer has been accepted by the answerer.

When an offerer sends a subsequent offer, if the offerer and answerer have not previously negotiated usage of RTP/RTCP multiplexing for the media associated with an RTP-based SDP media description ("m=" line), the offerer MAY indicate exclusive support of RTP/RTCP multiplexing,

following the procedures in Section 4.2. The offerer MUST process the associated answer following the procedures in Section 4.4.

NOTE: It is RECOMMENDED to not switch between usage of RTP/RTCP multiplexing and usage of separate ports for RTP and RTCP in a subsequent offer, unless there is a use-case that mandates it.

5. ICE Considerations

As defined in [I-D.ietf-ice-rfc5245bis], if an entity is aware that the remote peer supports, and is willing to use, RTP/RTCP multiplexing, the entity will only provide RTP candidates (component ID 1). However, only providing RTP candidates does not as such imply exclusive support of RTP/RTCP multiplexing. RTCP candidates would not be provided also in cases where RTCP is not supported at all. Therefore, additional information is needed in order to indicate support of exclusive RTP/RTCP multiplexing. This document defines such mechanism using the SDP 'rtcp-mux-exclusive' attributes.

6. Security Considerations

This document does not introduce new security considerations in additions to those specified in [RFC3605] and [RFC5761].

7. IANA Considerations

This document updates the "Session Description Protocol Parameters" registry as specified in Section 8.2.2 of [RFC4566]. Specifically, it adds the SDP 'rtcp-mux-exclusive' attribute to the table for SDP media level attributes.

Attribute name: rtcp-mux-exclusive
Type of attribute: media-level
Subject to charset: no
Purpose: Indicate exclusive support of RTP/RTCP multiplexing
Appropriate Values:
Contact name: Christer Holmberg
Category: NORMAL

8. Acknowledgments

Thanks to Roman Shpount, Paul Kyzivat, Ari Keranen, Bo Burman and Tomas Frankkila for their comments and input on the draft.

9. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-02

- Minor editorial fix.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-01

- Mux category and source-specific applicability added.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-00

- Defined new SDP attribute for indicating rtcp-mux-exclusive.
- Updates to RFC 5761 removed.
- IANA considerations added.

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-03

- Submitted as draft-ietf-mmusic-rtcp-mux-exclusive-00.

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-02

- Intended status changed to "Standards track".

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-01

- Clarified that the SDP rtcp attribute may contain the optional IP address part.

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-00

- Additional updates to Section 5.1.1 of RFC 5761.
- ICE considerations added.

10. References

10.1. Normative References

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

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

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

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <http://www.rfc-editor.org/info/rfc5761>.

[I-D.ietf-ice-rfc5245bis] Keranen, A. and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-00 (work in progress), October 2015.

10.2. Informative References

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

[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, <http://www.rfc-editor.org/info/rfc5576>.

[I-D.ietf-mmusic-sdp-mux-attributes] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-12 (work in progress), January 2016.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-25 (work in progress), January 2016.

Author’s Address

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Network Working Group
Internet-Draft
Updates: 5761 (if approved)
Intended status: Standards Track
Expires: November 6, 2017

C. Holmberg
Ericsson

May 5, 2017

Indicating Exclusive Support of RTP/RTCP Multiplexing using SDP
draft-ietf-mmusic-mux-exclusive-12.txt

Abstract

This document defines a new SDP media-level attribute, 'rtcp-mux-only', that can be used by an endpoint to indicate exclusive support of RTP/RTCP multiplexing. The document also updates RFC 5761, by clarifying that an offerer can use a mechanism to indicate that it is not able to send and receive RTCP on separate ports.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on November 6, 2017.

Table of Contents

1. Introduction

[RFC5761] defines how to multiplex RTP and RTCP on a single IP address and port, referred to as RTP/RTCP multiplexing. [RFC5761] also defines an Session Description Protocol (SDP) [RFC4566] attribute, 'rtcp-mux' that can be used by entities to indicate support, and negotiate usage of, RTP/RTCP multiplexing.

As defined in [RFC5761], if the peer endpoint does not support RTP/RTCP multiplexing, both endpoints should use separate ports for sending and receiving of RTCP (referred to as fallback to usage of separate ports for RTP and RTCP).

Some newer applications that do not require backward compatibility with peers that cannot multiplex RTCP might choose to not implement separation of RTP and RTCP. Examples of such applications are W3C WEBRTC [W3C.WD-webrtc-20120209] applications, that are not required to interoperate with non-WEBRTC clients. For such applications, this document defines an SDP attribute to signal intent to require multiplexing. The use of this attribute in SDP offers [RFC3264] by

entities that ever need to interoperate with peers that do not support RTC/RTCP multiplexing may harm interoperability. Also, while the SDP answerer [RFC3264] might support, and prefer usage of, fallback to non-multiplex, the attribute indicates that fallback to non-multiplex cannot be enabled. One example of where non-multiplex is preferred is where an endpoint is connected to a radio interface, and wants to use different bearers (possibly with different quality characteristics) for RTP and RTCP. Another example is where the one endpoint is acting as a gateway to a network where RTP/RTCP multiplexing cannot be used. In such case there endpoint may prefer non-multiplexing also towards the other network. Otherwise the endpoint would have to perform de-multiplexing of RTP and RTCP.

This document defines a new SDP media-level attribute, 'rtcp-mux-only', that can be used by an endpoint to indicate exclusive support of RTP/RTCP multiplexing. The document also updates [RFC5761], by clarifying that an offerer can use a mechanism to indicate that it is not able to send and receive RTCP on separate ports.

The document also describes the Interactive Connectivity Establishment (ICE) [RFC5245] considerations when indicating exclusive support of RTP/RTCP multiplexing.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. SDP rtcp-mux-only Attribute

This section defines a new SDP media-level attribute, 'rtcp-mux-only'.

Name: rtcp-mux-only

Value: N/A

Usage Level: media

Charset Dependent: no

Syntax:

rtcp-mux-only

Example:

a=rtcp-mux-only

In an SDP offer, the offerer uses the SDP 'rtcp-mux-only' attribute to indicate exclusive support of RTP/RTCP multiplexing for the RTP-based media associated with the SDP media description ("m=" line).

In an SDP answer, the 'rtcp-mux' attribute [RFC5761] indicates that the answerer supports, and accepts usage of, RTP/RTCP multiplexing for the RTP-based media associated with the SDP media description ("m=" line).

The usage of the 'rtcp-mux-only' attribute in an SDP answer is forbidden.

The usage of the SDP 'rtcp-mux-only' attribute is only defined for RTP-based media.

The mux category [I-D.ietf-mmusic-sdp-mux-attributes] for the 'rtcp-mux-only' attribute is 'IDENTICAL', which means that the attribute, if used within a BUNDLE group [I-D.ietf-mmusic-sdp-bundle-negotiation], must be associated with all multiplexed RTP-based media descriptions within the BUNDLE group.

The 'rtcp-mux-only' attribute applies to the whole associated media description. The attribute MUST NOT be defined per source (using the SDP 'ssrc' attribute [RFC5576]).

The SDP offer/answer [RFC3264] procedures associated with the attribute are defined in Section 4

4. SDP Offer/Answer Procedures

4.1. General

This section defines the SDP offer/answer [RFC3264] procedures for indicating exclusive support of, and negotiating usage of, RTP/RTCP multiplexing.

The procedures in this section apply to individual RTP-based SDP media descriptions ("m=" lines).

4.2. Generating the Initial SDP Offer

When an offerer sends the initial offer, if the offerer wants to indicate exclusive RTP/RTCP multiplexing for RTP-based media, the offerer MUST associate an SDP 'rtcp-mux-only' attribute with the associated SDP media description ("m=" line).

In addition, if the offerer associates an SDP 'rtcp-mux-only' attribute with an SDP media description ("m=" line, the offerer MUST also associate an SDP 'rtcp-mux' attribute with the same SDP media description ("m=" line), following the procedures in [RFC5761].

If the offerer associates an SDP 'rtcp' attribute [RFC3605] with an SDP media description ("m=" line), and if the offerer also associates an SDP 'rtcp-mux-only' attribute with the same SDP media description ("m=" line), the address and port values of the SDP 'rtcp' attribute MUST match the corresponding values for RTP.

NOTE: This specification does not mandate the usage of the SDP 'rtcp' attribute for RTP/RTCP multiplexing.

4.3. Generating the Answer

When an answerer receives an offer that contains an SDP 'rtcp-mux-only' attribute, associated with an RTP-based SDP media description ("m=" line), if the answerer accepts the usage of RTP/RTCP multiplexing, the answerer MUST associate an SDP 'rtcp-mux' attribute with the corresponding SDP media description ("m=") in the associated answer, following the procedures in [RFC5761]. If the answerer does not accept the usage of RTP/RTCP multiplexing, the answerer MUST either reject the SDP media description ("m=") by setting the port value to zero in the associated answer, or reject the whole offer, following the procedures in [RFC3264].

The answerer MUST NOT associate an SDP 'rtcp-mux-only' attribute with an SDP media description ("m=" line) in the answer.

4.4. Offerer Processing of the SDP Answer

If an offerer associated an SDP 'rtcp-mux-only' attribute with an RTP-based SDP media description ("m=" line) in an offer, and if the corresponding SDP media description ("m=" line) in the associated answer contains an SDP 'rtcp-mux' attribute, the offerer MUST apply the RTP/RTCP multiplexing procedures [RFC5761] to the associated RTP-based media. If the corresponding SDP media description ("m=" line) in the associated answer does not contain an SDP 'rtcp-mux' attribute, the offerer MUST either take appropriate actions in order to disable the associated RTP-based media, e.g., send a new offer with a zero port value associated with the SDP media description ("m=" line), or send a new offer without associating an SDP 'rtcp-mux-only' attribute with the SDP media description ("m=" line).

NOTE: This document does not mandate specific actions on how to terminate the RTP media. The offerer might e.g. send a new offer where the port value of the SDP media description is set to zero in order to terminate the RTP media.

4.5. Modifying the Session

When an offerer sends a subsequent offer, if the offerer and answerer have previously negotiated usage of exclusive RTP/RTCP multiplexing for the media associated with an RTP-based SDP media description ("m=" line), the offerer SHOULD associate an SDP 'rtcp-mux-only' with the corresponding SDP media description ("m=" line).

In addition, if the offerer associates an SDP 'rtcp-mux-only' attribute with an SDP media description ("m=" line), the offerer MUST also associate an SDP 'rtcp-mux' attribute with the same SDP media description ("m=" line), following the procedures in [RFC5761].

If the offerer does not associate the attributes with the corresponding SDP media description ("m=" line) it is an indication that the offerer no longer wants to use RTP/RTCP multiplexing, and instead MUST fallback to usage of separate ports for RTP and RTCP once the offer has been accepted by the answerer.

When an offerer sends a subsequent offer, if the offerer and answerer have not previously negotiated usage of RTP/RTCP multiplexing for the media associated with an RTP-based SDP media description ("m=" line), the offerer MAY indicate exclusive support of RTP/RTCP multiplexing, following the procedures in Section 4.2. The offerer MUST process the associated answer following the procedures in Section 4.4.

It is RECOMMENDED to not switch between usage of RTP/RTCP multiplexing and usage of separate ports for RTP and RTCP in a subsequent offer, unless there is a use-case that mandates it.

5. Update to RFC 5761

5.1. General

This section updates sections 5.1.1 and 5.1.3 of [RFC5761], by clarifying that an offerer can use a mechanism to indicate that it is not able to send and receive RTCP on separate ports, and that the offerer shall terminate the affected streams if the answerer does not indicate support of RTP/RTCP multiplexing. It also clarifies that, when the offerer is not able to send and receive RTCP on separate ports, the offerer will not provide an SDP 'candidate' attribute for RTCP, nor will the offerer provide a fallback port for RTCP (using the SDP 'rtcp' attribute).

5.2. Update to 4th paragraph of section 5.1.1

NOTE: [RFC8035] also updates section 5.1.1 of [RFC5761]. While the paragraph updated in this document is not updated by [RFC8035], the location of the paragraph within section 5.1.1 is moved.

OLD TEXT:

> If the answer does not contain an "a=rtcp-mux" attribute, the offerer MUST NOT multiplex RTP and RTCP packets on a single port. Instead, it should send and receive RTCP on a port allocated according to the usual port-selection rules (either the port pair, or a signalled port if the "a=rtcp:" attribute [10] is also included). This will occur when talking to a peer that does not understand the "a=rtcp-mux" attribute.

NEW TEXT:

> If the answer does not contain an "a=rtcp-mux" attribute, the offerer MUST NOT multiplex RTP and RTCP packets on a single port. Instead, it should send and receive RTCP on a port allocated according to the usual port-selection rules (either the port pair, or a signaled port if the "a=rtcp:" attribute [10] is also included). This will occur when talking to a peer that does not understand the "a=rtcp-mux" attribute. However, if the offerer indicated in the offer that it is not able to send and receive RTCP on a separate port, the offerer MUST disable the media streams associated with the attribute. The mechanism for indicating that the offerer is not able to send and receive RTCP on a separate port is outside the scope of this specification.

5.3. Update to 2nd paragraph of section 5.1.3

OLD TEXT:

> If it is desired to use both ICE and multiplexed RTP and RTCP, the initial offer MUST contain an "a=rtcp-mux" attribute to indicate that RTP and RTCP multiplexing is desired and MUST contain "a=candidate:" lines for both RTP and RTCP along with an "a=rtcp:" line indicating a fallback port for RTCP in the case that the answerer does not support RTP and RTCP multiplexing. This MUST be done for each media where RTP and RTCP multiplexing is desired.

NEW TEXT:

> If it is desired to use both ICE and multiplexed RTP and RTCP, the initial offer MUST contain an "a=rtcp-mux" attribute to indicate that RTP and RTCP multiplexing is desired and MUST contain "a=candidate:" lines for both RTP and RTCP along with an "a=rtcp:" line indicating a fallback port for RTCP in the case that the answerer does not support RTP and RTCP multiplexing. This MUST be done for each media where RTP and RTCP multiplexing is desired. However, if the offerer indicates in the offer that it is not able to send and receive RTCP on a separate port, the offerer MUST NOT include "a=candidate:" lines for RTCP, and the offerer MUST NOT provide a fallback port for RTCP using the "a=rtcp:" line.

6. ICE Considerations

As defined in [RFC5245], if an entity is aware that the remote peer supports, and is willing to use, RTP/RTCP multiplexing, the entity will only provide RTP candidates (component ID 1). However, only providing RTP candidates does not as such imply exclusive support of RTP/RTCP multiplexing. RTCP candidates would not be provided also in cases where RTCP is not supported at all. Therefore, additional information is needed in order to indicate support of exclusive RTP/RTCP multiplexing. This document defines such mechanism using the SDP ’rtcp-mux-only’ attributes.

7. Security Considerations

This document does not introduce new security considerations in additions to those specified in [RFC3605] and [RFC5761].

8. IANA Considerations

This document updates the "Session Description Protocol Parameters" registry as specified in Section 8.2.2 of [RFC4566]. Specifically,

it adds the SDP 'rtcp-mux-only' attribute to the table for SDP media level attributes.

Attribute name: rtcp-mux-only
Type of attribute: media-level
Subject to charset: no
Purpose: Indicate exclusive support of RTP/RTCP multiplexing
Appropriate Values: N/A
Contact name: Christer Holmberg (christer.holmberg@ericsson.com)
Mux Category: IDENTICAL

9. Acknowledgments

Thanks to Roman Shpount, Paul Kyzivat, Ari Keranen, Bo Burman, Tomas Frankkila and Martin Thomson for their comments and input on the document. Thanks to Francis Dupont for the genart review.

10. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-11

- Clarification note added to RFF 5761 update section.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-10

- Changes based on comments from Ekr:

- - 'rtcp-mux-only' attribute only defined for SDP offers

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-09

- Changes based on IESG review comments from Alexey Melnikov and Mirja Kuhlewind:

- - References to draft-mux-attributes and draft-sdp-bundle made normative.

- - Text added regarding cases where entities might want to use non-multiplexed RTP and RTCP.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-08

- Editorial changes based on genart comments from Francis Dupont.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-07

- Comments from Ben Campbell.
- - Additional text regarding applications for which the mechanism is suitable.
- - Removal of pre-RFC5378 disclaimer.
- - Editorial fixes.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-06

- - Editorial fix.
- - Addition of pre-RFC5378 disclaimer.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-05

- Editorial fix.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-04

- Changes based on comments from Flemming Andreasen.
- - Attribute mux category changed to IDENTICAL.
- - Reference to draft-5245bis changed to RFC 5245.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-03

- Editorial changes based on comments from Martin Thomson.
- Change of attribute name.
- RFC 5761 updates added.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-02

- Minor editorial fix.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-01

- Mux category and source-specific applicability added.

Changes from draft-ietf-mmusic-rtcp-mux-exclusive-00

- Defined new SDP attribute for indicating rtcp-mux-exclusive.

- Updates to RFC 5761 removed.
- IANA considerations added.

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-03

- Submitted as draft-ietf-mmusic-rtcp-mux-exclusive-00.

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-02

- Intended status changed to "Standards track".

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-01

- Clarified that the SDP rtcp attribute may contain the optional IP address part.

Changes from draft-holmberg-mmusic-rtcp-mux-exclusive-00

- Additional updates to Section 5.1.1 of RFC 5761.
- ICE considerations added.

11. References

11.1. Normative References

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

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

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

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <http://www.rfc-editor.org/info/rfc5761>.

[RFC8035] Holmberg, C., "Session Description Protocol (SDP) Offer/ Answer Clarifications for RTP/RTCP Multiplexing", RFC 8035, DOI 10.17487/RFC8035, November 2016, <http://www.rfc-editor.org/info/rfc8035>.

[I-D.ietf-mmusic-sdp-mux-attributes] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-16 (work in progress), December 2016.

[I-D.ietf-mmusic-sdp-bundle-negotiation] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-36 (work in progress), October 2016.

11.2. Informative References

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

[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, <http://www.rfc-editor.org/info/rfc5576>.

[W3C.WD-webrtc-20120209] Bergkvist, A., Burnett, D., Jennings, C., and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120209, February 2012, <http://www.w3.org/TR/2012/WD-webrtc-20120209>.

Author’s Address

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Network Working Group
Internet-Draft
Obsoletes: 4566 (if approved)
Intended status: Standards Track
Expires: May 6, 2016

M. Handley
UCL
V. Jacobson
PARC
C. Perkins
University of Glasgow
A. Begen
Unaffiliated

November 3, 2015

SDP: Session Description Protocol
draft-ietf-mmusic-rfc4566bis-16

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 Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 6, 2016.

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) [RFC2326], electronic mail using the MIME extensions, and the Hypertext Transport Protocol (HTTP).

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 limited to essential corrections, and are outlined in Section 10 of this memo.

2. Glossary of Terms

The following term is used in this document and has 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.

The terms "multimedia conference" and "multimedia session" are used in this document as defined in [I-D.ietf-avtext-rtp-grouping-taxonomy]. 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 [RFC2119].

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. 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) [RFC2326], 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 announcements of multicast sessions made only via email or the WWW do not have the property that the receiver of a session announcement can necessarily receive the session because the multicast sessions may be restricted in scope, and access to the WWW server or reception of email is possible outside this scope.

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 in

an internetwork, 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 non-participants that may need to know. (This latter feature is primarily useful when SDP is used with a multicast session announcement protocol.)

4.1. Media and Transport Information

An SDP 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 this address and port depend on the media and transport protocol defined. By default, this SHOULD be the remote address and remote port to which data is sent. Some media types may redefine this behaviour, but this is NOT RECOMMENDED since it 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) for more information about the session.

4.4. Categorisation

When many session descriptions are being distributed by SAP, or any other advertisement mechanism, it may be desirable to filter session announcements that are of interest from those that are not. SDP supports a categorisation mechanism for sessions that is capable of being automated (the "a=cat:" attribute; see Section 6).

4.5. Internationalisation

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 to be used when desired. Internationalisation only applies to free-text fields (session name and background information), and not to SDP as a whole.

5. SDP Specification

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

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

<type>=<value>

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

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

```
      Session description
         v=  (protocol version)
         o=  (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=" and "r=" lines; see below)
         z=* (time zone adjustments)
         k=* (encryption key)
         a=* (zero or more session attribute lines)
         Zero or more media descriptions

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

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

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

An SDP 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 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 233.252.0.2".

An example SDP description is:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 198.51.100.1
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 233.252.0.2
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=audio 49180 RTP/AVP 0
m=video 51372 RTP/AVP 99
c=IN IP4 233.252.0.1/127
a=rtpmap:99 h263-1998/90000
```

Text fields such as the session name and information 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 record, although parsers SHOULD be tolerant and also accept records terminated with a single newline character. If the "a=charset" attribute is not present, these octet strings MUST be interpreted as containing ISO-10646 characters in UTF-8 encoding (the presence of the "a=charset" attribute may force some fields to be interpreted differently).

A session description can contain domain names in the "o=", "u=", "e=", "c=", and "a=" lines. Any domain name used in SDP MUST comply with [RFC1034], [RFC1035]. Internationalised 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 internationalised domain names).

5.1. Protocol Version ("v=")

```
v=0
```

The "v=" line 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 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 it has been suggested that a Network Time Protocol (NTP) format timestamp be used to ensure uniqueness [RFC5905].

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

<nettype> is a text string giving the type of network. Initially "IN" is defined to have the meaning "Internet", but other values MAY be registered in 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 compressed textual representation of an IP version 6 address of the machine. For both IP4 and IP6, the fully qualified domain name is the form that SHOULD be given unless this is unavailable, in which case a globally unique address MAY be substituted. Unless an SDP extension for NAT traversal is used (e.g., ICE [RFC5245], ICE TCP [RFC6544]), a local IP address MUST NOT be used in any context where the SDP description might leave the scope in which the address is meaningful (for example, a local address MUST NOT be included in an application-level referral that might leave the scope).

In general, the "o=" line serves as a globally unique identifier for this version of this session description, and the subfields excepting the version taken together identify the session irrespective of any modifications.

For privacy reasons, it is sometimes desirable to obfuscate the username and IP address of the session originator. If this is a concern, an arbitrary <username> and private <unicast-address> MAY be chosen to populate the "o=" 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 is the textual session name. There MUST be one and only one "s=" line per session description. The "s=" line MUST NOT be empty and SHOULD contain ISO 10646 characters (but see also the "a=charset" attribute). If a session has no meaningful name, the value "s= " SHOULD be used (i.e., a single space as the session name).

5.4. Session Information ("i=")

i=<session description>

The "i=" line provides textual information about the session. There MUST be at most one session-level "i=" line per session description, and at most one "i=" line per media description/definition. Unless a media level "i="" line is used, 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.

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

The "i=" 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>
```

A URI is a Uniform Resource Identifier as used by WWW clients [RFC3986]. The URI should be a pointer to additional information about the session. This line is OPTIONAL. No more than one URI line is allowed per session description.

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

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

The "e=" and "p=" lines specify contact information for the person responsible for the 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 field. 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) 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 Data ("c=")

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

The "c=" line contains connection data.

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 "c=" line(s) per media description, in which case the per-media values override the session-level settings for the respective media.

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

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

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

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

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

- 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 IP6 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 at 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 IP6 example would be:

```
c=IN IP6 FF15::101/3
```

which is semantically equivalent to:

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

(remembering that the TTL field is not present in IP6 multicast).

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

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

5.8. Bandwidth ("b=")

b=<bwtype>:<bandwidth>

This OPTIONAL line denotes the proposed bandwidth to be used by the session or media. The <bwtype> is an alphanumeric modifier giving the meaning of the <bandwidth> figure. Two values are defined in this specification, but other values MAY be registered in 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 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 CT gives a total bandwidth figure for all the media at all endpoints.

AS The bandwidth is interpreted to be application specific (it will be the application's concept of maximum bandwidth). Normally, this will coincide with what is set on the application's "maximum bandwidth" control if applicable. For RTP-based applications, AS gives the RTP "session bandwidth" as defined in Section 6.2 of [RFC3550]. Note that AS gives a bandwidth figure for a single media at a single endpoint, although there may be many endpoints sending simultaneously.

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

b=X-YZ:128

Use of the "X-" prefix is NOT RECOMMENDED: instead new modifiers SHOULD be registered with IANA in the standard namespace. SDP parsers MUST ignore bandwidth fields with unknown modifiers. Modifiers MUST be 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. The definition of a new <bwtype> modifier MAY specify that the bandwidth is to be interpreted in some alternative unit (the "CT" and "AS" modifiers defined in this memo use the default units).

5.9. Timing ("t=")

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

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

The first and second sub-fields give the start and stop times, respectively, for the session. These values are the decimal representation of Network Time Protocol (NTP) time values in seconds since 1900 [RFC5905]. To convert these values to UNIX time, subtract decimal 2208988800.

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

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

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

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

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

5.10. Repeat Times ("r=")

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

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

```
t=3034423619 3042462419
r=604800 3600 0 90000
```

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)
```

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

```
r=7d 1h 0 25h
```

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

5.11. Time Zones ("z=")

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

To schedule a repeated session 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 NTP time 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:

```
z=2882844526 -1h 2898848070 0
```

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

If a session is likely to last several years, it is expected that the session 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>
```

If transported over a secure and trusted channel, the Session Description Protocol MAY be used to convey encryption keys. A simple mechanism for key exchange is provided by the key line ("k="), although this is primarily supported for compatibility with older implementations and its use is NOT RECOMMENDED. Work is in progress to define new key exchange mechanisms for use with SDP [RFC4567] [RFC4568], and it is expected that new applications will use those mechanisms.

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

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

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

k=clear:<encryption key>

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

k=base64:<encoded encryption key>

The encryption key is included in this key line but has been base64 encoded [RFC4648] because it includes characters that are prohibited in SDP. This method MUST NOT be used unless it can be guaranteed that the SDP is conveyed over a secure channel.

k=uri:<URI to obtain key>

A Uniform Resource Identifier is included in the key line. The URI refers to the data containing the key, and may require additional authentication before the key can be returned. When a request is made to the given URI, the reply should specify the encoding for the key. The URI is often an Secure Socket Layer/Transport Layer Security (SSL/TLS)-protected HTTP URI ("https:"), although this is not required.

k=prompt

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

The key line MUST NOT be used unless it can be guaranteed that the SDP is conveyed over a secure and trusted channel. An example of

such a channel might be SDP embedded inside an S/MIME message or a TLS-protected HTTP session. It is important to ensure that the secure channel is with the party that is authorised to join the session, not an intermediary: if a caching proxy server is used, it is important to ensure that the proxy is either trusted or unable to access the SDP.

5.13. Attributes ("a=")

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

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

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

Attribute lines may be of two forms:

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

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

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

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

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

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

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

5.14. Media Descriptions ("m=")

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

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

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

<port> is the transport port to which the media stream is sent. The meaning of the transport port depends on the network being used as specified in the relevant "c=" line, and on the transport protocol defined in the <proto> sub-field of the media field. Other ports used by the media application (such as the 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, "a=rtcp:" as defined in [RFC3605]).

If non-contiguous 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:" 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 below). If non-contiguous ports are required, they must be signalled using a separate attribute (for example, "a=rtcp:" as defined in [RFC3605]).

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:

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

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 semantics of multiple "m=" lines using the same transport address are undefined. This implies that, unlike limited past practice, there is no implicit grouping defined by such means and an explicit grouping framework (for example, [RFC5888]) should instead be used to express the intended semantics.

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

* udp: denotes an unspecified protocol running over UDP.

* RTP/AVP: denotes RTP [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.

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 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 sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.

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

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

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

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.

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 obsoleted.

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 cat attribute, this is to assist identifying wanted sessions at the receiver. This allows a receiver to select interesting session based on keywords describing the purpose of the session; 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 obsoleted.

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 ptime to decode RTP or vat audio, and it is intended as a recommendation for the encoding/packetisation 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 non-updated implementations 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
        ; do we want to define a limited range for this?
      encoding-params = channels
        ; 4566 is vague about what this can be. RFC4855 seems to be
        ; the authoritative source, and only allows the
        ; value of the media subtype "channels" parameter - the
        ; number of audio channels.
        ; Does anyone think this can be used for something else???
        ; (The implication that multiple parameters might be included
        ; seems a misdirection - additional parameters are
        ; to go into a=fmtp.)
        ; Does anyone have an example of other parameters
        ; using this field?
      channels = integer
        ; Is there any reason to make this less restrictive?
```

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 incusively 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 coded 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 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 parameters> 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) packetisation is required, the "ptime" attribute is used as given above.

6.7. Media Direction Attributes

At most one of recvonly/sendrecv/sendonly/inactive MAY appear at session level, and at most one MAY appear in each media section.

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

If none of the media direction attributes is present at either session level or media level, "sendrecv" SHOULD be assumed as the default for sessions that are not of the multimedia conference type "broadcast" or "H332" (see below).

Within the following SDP example, the "inactive" attribute applies to audio media and the "recvonly" attribute applies to video media.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 198.51.100.1
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 233.252.0.1/127
t=2873397496 2873404696
a=inactive
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
a=recvonly
```

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 recvonly applies to the media only, not to any associated control protocol (e.g., an RTP-based system in recvonly mode SHOULD still send RTCP packets).

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 sendonly and one recvonly. Note that sendonly 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 inactive.

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

Specifying the attribute "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 "recvonly".

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

6.10. charset (character set)

Name: charset

Value: charset-value

Usage Level: session

Charset Dependent: no

Syntax:

```
charset-value = mime-charset
(as defined in I-D.iana-charset-reg-procedure)
```

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 US-ASCII string and 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 recognised or not supported, all strings that are affected by it SHOULD be regarded as octet strings.

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

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 RFC5646
```

Example:

```
a=sdplang:fr
```

Multiple 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 sdplang attributes specified at session-level.

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

The "sdplang" attribute value must be a single [RFC5646] language tag in US-ASCII. An "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 RFC5646
```

Example:

```
a=lang:de
```

Multiple lang attributes can be provided either at session or media level if the session or media has capabilities to use multiple languages, 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, 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 "lang" attribute value must be a single [RFC5646] language tag in US-ASCII. A "lang" attribute SHOULD be specified when a session is of sufficient scope to cross geographic boundaries where the language of recipients cannot be assumed, or where the session has capabilities in languages different from the locally assumed norm.

Events during the session can influence which language(s) are used, and the participants are not strictly bound to only use the declared languages.

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

Editor's note: The usage should be checked with the SIP Forum to see whether anybody is using this. Otherwise, the recommendation will be not to use this attribute and the receiver ignores it upon reception.

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 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.

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 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 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 where media is sent to 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 and may decide to ask the user whether or not to accept the session.

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

On receiving a session description over an unauthenticated transport mechanism or from an untrusted party, software parsing the session should take a few precautions. Session descriptions contain information required to start software on the 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-authorised 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.

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 behaviour for an unknown attribute is to ignore it.

In certain environments, it has become common for intermediary systems to intercept and analyse session descriptions contained within other signalling 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 behaviours 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).

Use of the "k=" line poses a significant security risk, since it conveys session encryption keys in the clear. SDP MUST NOT be used to convey key material, unless it can be guaranteed that the channel over which the SDP is delivered is both private and authenticated. Moreover, the "k=" line provides no way to indicate or negotiate cryptographic key algorithms. As it provides for only a single symmetric key, rather than separate keys for confidentiality and integrity, its utility is severely limited. The use of the "k=" line is NOT RECOMMENDED, as discussed in Section 5.12.

8. IANA Considerations

8.1. The "application/sdp" Media Type

One media type registration from [RFC4566] is to be updated, as defined below.

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

Type name: application

Subtype name: sdp

Required parameters: None.

Optional parameters: None.

Encoding considerations:
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 XXXX). Arbitrary binary content cannot be directly represented in SDP.

Security considerations:
See Section 7 of RFC XXXX.

Interoperability considerations:
See RFC XXXX.

Published specification:
See RFC XXXX.

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

Additional information:

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

Person & email address to contact for further information:
IETF MMUSIC working group <mmusic@ietf.org>

Intended usage: COMMON

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

8.2. Registration of Parameters

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

The contact address for all parameters registered below is:

IETF MMUSIC working group <mmusic@ietf.org>

8.2.1. Media Types ("media")

The set of media types is intended to be small and SHOULD NOT be extended except under rare circumstances. The same rules should apply for media names as for top-level 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 [RFC5226].

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 (RFC 2327); 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]).

8.2.2. Transport Protocols ("proto")

The "proto" field describes the transport protocol used. This SHOULD reference a standards-track protocol RFC. This memo registers three values: "RTP/AVP" is a reference to [RFC3550] used under the RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551] running over UDP/IP, "RTP/SAVP" is a reference to the Secure Real-time Transport Protocol [RFC3711], and "udp" indicates an unspecified protocol over UDP.

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

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

8.2.3. Media Formats ("fmt")

Each transport protocol, defined by the "proto" field, has an associated "fmt" namespace that describes the media formats 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 "RTP/SAVP" profiles MUST use the payload type number as their "fmt" value. If the payload type number is dynamically assigned by this session description, an additional "rtpmap" attribute MUST be included to specify the format name and parameters as defined by the 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, new formats SHOULD be registered. Use of an existing media subtype for the format is encouraged. If no media subtype exists, it is RECOMMENDED that a suitable one be registered through the IETF process [RFC6838] by production of, or reference to, a standards-track RFC that defines the transport protocol for the format.

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

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

8.2.4. Attribute Names ("att-field")

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

New attribute registrations are accepted according to the "Specification Required" policy of [RFC5226], provided that the specification includes the following information:

- Contact name, email address, and telephone number.
- Attribute name (as it will appear in SDP). This MUST conform to the definition of <att-field>.
- 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 value. Enclosing the rule name in "[" and "]" indicates that a value is optional.
- Usage level of the attribute. (One or more of: session, media, source).
- Whether the attribute value is subject to the charset attribute.
- An ABNF definition of the attribute value rule. The rule MUST NOT match anything that is not also matched by <att-value>. The rule name MUST NOT be defined as an Incremental Alternative to <att-value>.
- An explanation of the purpose and usage of the attribute.
- A specification of appropriate attribute values for this attribute (If not included in syntax).
- Offer/Answer procedures as explained in [RFC3264].
- Indication of which "category" [I-D.ietf-mmusic-sdp-mux-attributes] an attribute is associated with.

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". 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 should be explicitly stated.

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

8.2.5. Bandwidth Specifiers ("bwtype")

A proliferation of bandwidth specifiers is strongly discouraged.

New bandwidth specifiers ("bwtype" fields) MUST be registered with IANA. The submission MUST reference a standards-track RFC specifying the semantics of the bandwidth specifier precisely, and indicating when it should be used, and why the existing registered bandwidth specifiers do not suffice.

IANA has registered the bandwidth specifiers "CT" and "AS" with definitions as in Section 5.8 of this memo (these definitions update those in [RFC4566]).

8.2.6. Network Types ("nettype")

New network types (the "nettype" field) MUST be registered with IANA if SDP needs to be used in the context of non-Internet environments. The registration is subject to the RFC Required - RFC publication policy of [RFC5226]. Although these are not normally the preserve of IANA, there may be circumstances when an Internet application needs to interoperate with a non-Internet application, such as when gatewaying an Internet 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 cannot be registered without registering at least one address type to be used with that network type. A new network type registration MUST reference an RFC that gives details of the network type and address type(s) and specifies how and when they would be used.

IANA has registered the network type "IN" to represent the Internet, with definition as in Sections 5.2 and 5.7 of this memo (these definitions update those in [RFC4566]).

8.2.7. Address Types ("addrtype")

New address types ("addrtype") MUST be registered with IANA. The registration is subject to the RFC Required - RFC publication policy of [RFC5226]. An address type is only meaningful in the context of a network type, and any registration of an address type MUST specify a registered network type or be submitted along with a network type registration. A new address type registration MUST reference an RFC giving details of the syntax of the address type. Address types are not expected to be registered frequently.

IANA has registered the address types "IP4" and "IP6" with definitions as in Sections 5.2 and 5.7 of this memo (these definitions update those in [RFC4566]).

8.2.8. Registration Procedure

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

- contact name, email address, and telephone number
- name being registered (as it will appear in SDP)
- long-form name in English
- type of name ("media", "proto", "fmt", "bwtype", "nettype", or "addrtype")
- a one-paragraph explanation of the purpose of the registered name
- a reference to the specification for the registered name (this will typically be an RFC number)

In the case of a new addrtype registration, the author has to check whether the new address type is usable with the existing network types. If yes, the "nettype" registry MUST be updated accordingly. In the case of a new nettype registration, the author MUST specify the usable address type(s).

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

8.3. Encryption Key Access Methods

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.

8.4. Reorganization of the nettype Registry

This document adds a new column in the "nettype" registry with the title "Usable addrtype Values" and updates the "nettype" registry as follows:

Type	SDP Name	Usable addrtype Values	Reference
nettype	IN	IP4, IP6	[RFC4566]
nettype	TN	RFC2543	[RFC2848]
nettype	ATM	NSAP, GWID, E164	[RFC3108]
nettype	PSTN	E164	[RFC7195]

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.

8.5. Reorganization of the att-field Registries

This document combines all the five "att-field" registries into one registry called "att-field" registry, and update the columns to reflect the name, usage level(s), charset dependency and reference. That is, the new registry uses the following columns:

Name | Usage Level | Dependent on charset? | Reference

The "Name" column reflects the attribute name (as it will appear in the SDP). The "Usage Level" column MUST indicate one or more of the following: session, media, source. The "Dependent on charset?" column MUST indicate "Yes" or "No" depending on whether the attribute value is subject to the charset attribute. Finally, the "Reference" column indicates the specification(s) where the attribute is defined.

Editor’s note: Will IANA reorganize this table based on what is in the registry now or should we provide the updated table in this document?

Editor’s note: [I-D.ietf-mmusic-sdp-mux-attributes] adds another column (muxing category) to this table. Should we add it here?

9. SDP Grammar

This section provides an Augmented BNF grammar for SDP. ABNF is defined in [RFC5234] and [RFC7405].

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

proto-version =       %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-fields =        *(%s"e" "=" email-address CRLF)

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

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

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

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

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

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

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

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

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

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

   fixed-len-time-unit = %s"d" / %s"h" / %s"m" / %s"s"
   ; NOTE: These units are case-sensitive.

   ; sub-rules of 'k='
   key-type =           %s"prompt"
                        %s"clear:"
                        %s"base64:"
                        %s"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 =          (att-field ":" att-value) / att-field
```

```
   att-field =           token

   att-value =           byte-string

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

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

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

   port =                1*DIGIT

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

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

   IP4-multicast =       m1 3( "." decimal-uchar )
                         "/" ttl [ "/" integer ]
                         ; 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 [ "/" integer ]
                         ; IP6 address starting with FF

   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, DIGIT, CRLF, SP, VCHAR: from RFC 5234
   ; URI-reference: from RFC 3986
   ; addr-spec: from RFC 5322
```

10. Summary of Changes from RFC 4566

The ABNF rule for IP6-address has been corrected. As a result, the ABNF rule for IP6-multicast has changed, and the (now unused) rules for hexpart, hexseq, and hex4 have been removed.

IP4 unicast and multicast addresses in the example SDP descriptions have been revised per RFCs 5735 and 5771.

Text in Section 5.2 has been revised to clarify the use of local addresses in case of ICE-like SDP extensions.

Normative and informative references have been updated.

The text regarding the session vs. media-level attribute usage has been clarified.

The case-insensitivity rules from RFC 4855 have been included in this document.

11. Acknowledgements

Many people in the IETF Multiparty Multimedia Session Control (MMUSIC) working group have made comments and suggestions contributing to this document.

12. References

12.1. Normative References

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

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://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, <http://www.rfc-editor.org/info/rfc2119>.

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

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://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, <http://www.rfc-editor.org/info/rfc3986>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <http://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, <http://www.rfc-editor.org/info/rfc5890>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <http://www.rfc-editor.org/info/rfc4648>.

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

[I-D.ietf-avtext-rtp-grouping-taxonomy] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-ietf-avtext-rtp-grouping-taxonomy-08 (work in progress), July 2015.

[I-D.iana-charset-reg-procedure] McFadden, M. and A. Melnikov, "IANA Charset Registration Procedures", draft-iana-charset-reg-procedure-01 (work in progress), April 2015.

[I-D.ietf-mmusic-sdp-mux-attributes] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-10 (work in progress), July 2015.

12.2. Informative References

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

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <http://www.rfc-editor.org/info/rfc5905>.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, DOI 10.17487/RFC2974, October 2000, <http://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, <http://www.rfc-editor.org/info/rfc3261>.

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, DOI 10.17487/RFC2326, April 1998, <http://www.rfc-editor.org/info/rfc2326>.

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

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

[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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc3890>.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, DOI 10.17487/RFC6544, March 2012, <http://www.rfc-editor.org/info/rfc6544>.

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

[ITU.H332.1998] International Telecommunication Union, "H.323 extended for loosely coupled conferences", ITU Recommendation H.332, September 1998.

[RFC4567] Arkko, J., Lindholm, F., Naslund, M., Norrman, K., and E. Carrara, "Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)", RFC 4567, DOI 10.17487/RFC4567, July 2006, <http://www.rfc-editor.org/info/rfc4567>.

[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, <http://www.rfc-editor.org/info/rfc4568>.

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

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

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

Authors' Addresses

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

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

Van Jacobson
PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
USA

EMail: van@parc.com

Colin Perkins
University of Glasgow
School of Computing Science
University of Glasgow
Glasgow G12 8QQ
UK

EMail: csp@csperkins.org

Ali Begen
Unaffiliated

EMail: acbegen@gmail.com

Network Working Group
Internet-Draft
Obsoletes: 4566 (if approved)
Intended status: Standards Track
Expires: December 20, 2019

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

June 18, 2019

SDP: Session Description Protocol
draft-ietf-mmusic-rfc4566bis-36

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 Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on December 20, 2019.

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 possible 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 non-participants that may need to know. (This latter feature is

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

4.1. Media and Transport Information

An SDP 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 sub-fields (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. 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 sub-fields 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 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 below) is the primary means for extending SDP and tailoring it to particular applications or media. Some attributes (the ones listed in Section 6 of this memo) have a defined meaning, but others may be added on a media- or session-specific basis. (Attribute scopes in addition to media-specific and session-specific may also be defined in extensions to this document. E.g., [RFC5576], [I-D.ietf-mmusic-data-channel-sdpneg].) 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 sub-fields 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 labelling media streams. As such, they are most likely to be useful when a single session has more than one distinct media stream of the same media type. An example would be two different whiteboards, one for slides and one for feedback and questions.

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

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

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

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. 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
```

(remembering that the TTL sub-field 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 giving the meaning of the <bandwidth> figure. 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 [I-D.ietf-mmusic-sdp-bundle-negotiation] 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 CT gives a total bandwidth figure for all the media at all endpoints.

AS The bandwidth is interpreted to be application specific (it will be the application's concept of maximum bandwidth). Normally, this will coincide with what is set on the application's "maximum bandwidth" control if applicable. For RTP-based applications, AS gives the RTP "session bandwidth" as defined in Section 6.2 of [RFC3550]. Note that AS gives a bandwidth figure for a single media at a single endpoint, although there may be many endpoints sending simultaneously.

[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 below) 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 sub-fields 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, an end-time SHOULD be given and modified as appropriate when new information becomes available about when the session should really end.

Permanent sessions may be shown to the user as never being active unless there are associated repeat times 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 sub-fields 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)

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

Attributes are the primary means for extending SDP. Attributes may be defined to be used as "session-level" attributes, "media-level" attributes, or both. (Attribute scopes in addition to media- and session- level may also be defined in extensions to this document. E.g., [RFC5576], [I-D.ietf-mmusic-data-channel-sdpneg].)

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

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

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

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

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

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

5.14. Media Descriptions ("m=")

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

A session description may contain a number of media descriptions. Each media description starts with an "m=" 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 sub-fields:

<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> sub-field 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, "a=rtcp:" as defined in [RFC3605]).

If non-contiguous 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:" 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 below).

This document does not include a mechanism for declaring hierarchically encoded streams using non-contiguous ports. (There is currently no attribute defined that can accomplish this. The "a=rtcp:" defined in [RFC3605] does not handle hierarchical encoding.) If a need arises to declare non-contiguous 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 [I-D.ietf-mmusic-sdp-bundle-negotiation].

<proto> is the transport protocol. The meaning of the transport protocol is dependent on the address type sub-field 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.

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 sub-fields describe the format of the media. The interpretation of the media format depends on the value of the <proto> sub-field.

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

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

For media using other transport protocols, the <fmt> sub-field is protocol specific. Rules for interpretation of the <fmt> sub-field 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 cat attribute, this was intended to assist identifying wanted sessions at the receiver. This allows a receiver to select interesting sessions based on keywords describing the purpose of the session; 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 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 non-updated implementations 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 coded 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 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 "ptime" attribute is used as given above.

6.7. Media Direction Attributes

At most one occurrence of recvonly, sendrecv, sendonly, or 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 appear 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, "sendrecv" SHOULD be assumed as the default.

Within the following SDP example, the "sendrecv" attribute applies to the first audio media and the "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 recvonly applies to the media only, not to any associated control protocol. An RTP-based system in recvonly 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 sendonly and one recvonly. Note that sendonly 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 inactive.

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 RFC5646
```

Example:

```
a=sdplang:fr
```

Multiple 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 sdplang attributes specified at session level.

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

The "sdplang" attribute value must be a single [RFC5646] language tag. An "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 RFC5646
```

Example:

a=lang:de

Multiple 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, 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 "lang" attribute value must be a single [RFC5646] language tag. 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 "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 'lang' attribute is specified, the "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.

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 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 where media is sent to 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). Use of some procedures and SDP extensions (e.g., ICE [RFC8445] and ICE-SIP-SDP [I-D.ietf-mmusic-ice-sip-sdp]) may avoid the need for intermediaries to modify SDP.

SDP MUST NOT be used to convey keying material (e.g., using "a=crypto" [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] is to be updated, as defined below.

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

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 XXXX). Arbitrary binary content cannot be directly represented in SDP.

Security considerations:
See Section 7 of RFC XXXX.

Interoperability considerations:
See RFC XXXX.

Published specification:
See RFC XXXX.

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

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 XXXX
IETF MMUSIC working group delegated from the IESG

8.2. Registration of Parameters

This specification replaces and updates the definitions of IANA parameter registries for seven named SDP sub-fields originally defined in [RFC4566]. Using the terminology in the SDP specification Augmented Backus-Naur Form (ABNF), they are "media", "proto", "att-field", "bwtype", "nettype", and "addrtype".

[EDITOR: Please change the RFC references to RFC4566 in these registries to instead refer to this document.]

The contact address for all parameters registered below is:

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

8.2.1. Media Types ("media")

The set of media types is intended to be small and SHOULD NOT be extended except under rare circumstances. The same rules should apply for media names as for top-level 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 (RFC 2327); 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.

8.2.2. Transport Protocols ("proto")

The "proto" sub-field 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, and "udp" indicates direct use of the UDP protocol.

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 the RTP protocol. For example, a profile whose short name is "XYZ" would be denoted by a "proto" sub-field of "RTP/XYZ".

Each transport protocol, defined by the "proto" sub-field, 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 "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, 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.

8.2.3. Attribute Names ("att-field")

8.2.3.1. New Attributes

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

New attribute registrations are accepted according to the "Specification Required" policy of [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 <att-field>.
- Attribute Syntax: For a value attribute (see clause 5.13), an ABNF definition of the attribute value <att-value> syntax (see Section 9) MUST be provided. The syntax MUST follow the rule form as 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: Usage level(s) of the attribute. One or more of: session, media, source, dcsa, dcsa(subprotocol). For a definition of source level attributes, see [RFC5576]. For a definition of dcsa attributes see: [I-D.ietf-mmusic-data-channel-sdpneg].

- Charset Dependent: Whether the attribute value is subject to the charset attribute or not (Yes/No).

- Purpose: An explanation of the purpose and usage of the attribute.

- O/A Procedures: Offer/Answer procedures as explained in [RFC3264].

- Mux Category: Indication of which multiplexing "category" [I-D.ietf-mmusic-sdp-mux-attributes] an attribute is associated with.

- 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 ("att-field" values) with definitions as in Section 6 of this memo (these definitions replace those in [RFC4566]).

8.2.3.2. 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 8.2.3.1 according to the following bullets:

- Contact Name: A name MUST be provided.
- Contact Email Address: An email address 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 [I-D.ietf-mmusic-sdp-mux-attributes]. 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 reference MUST be provided.

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

8.2.4. Bandwidth Specifiers ("bwtype")

A proliferation of bandwidth specifiers is strongly discouraged.

New bandwidth specifiers (<bwtype> sub-field 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.

IANA has registered the bandwidth specifiers "CT" and "AS" with definitions as in Section 5.8 of this memo (these definitions update those in [RFC4566]).

8.2.5. Network Types ("nettype")

New network types (<nettype> sub-field values) MUST be registered with IANA if SDP needs to be used in the context of non-Internet environments. The registration is subject to the "RFC Required" policy of [RFC8126]. Although these are not normally the preserve of IANA, there may be circumstances when an Internet application needs to interoperate with a non-Internet application, such as when gatewaying an Internet 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 cannot be registered without registering at least one address type to be used with that network type. A new network type registration MUST reference an RFC that gives details of the network type and address type(s) and specifies how and when they would be used.

IANA has registered the network type "IN" to represent the Internet, with definition as in Sections 5.2 and 5.7 of this memo (these definitions update those in [RFC4566]).

8.2.6. Address Types ("addrtype")

New address types ("addrtype") MUST be registered with IANA. The registration is subject to the "RFC Required" policy of [RFC8126]. An address type is only meaningful in the context of a network type, and any registration of an address type MUST specify a registered network type or be submitted along with a network type registration. A new address type registration MUST reference an RFC giving details of the syntax of the address type. Address types are not expected to be registered frequently.

Section 5.7 of this document gives new definitions of address types "IP4" and "IP6". The registries are to be correspondingly updated by IANA.

8.2.7. Registration Procedure

A specification document that defines new values for SDP "media", "proto", "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", "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.)

IANA will populate its registries with some or all of these values.

8.3. Encryption Key Access Methods

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.

8.4. Reorganization of the nettype and addrtype registries

This document adds a new column in the "nettype" registry with the title "Usable addrtype Values", replacing the separate "addrtype" registry. The following is the revised "nettype" registry:

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

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.

8.5. Reorganization of the att-field Registries

This document combines all of the (currently) five "att-field" registries into one registry called "att-field" registry, and updates the columns to reflect the name, usage level(s), charset dependency and reference. As such IANA is requested to create a new combined registry using the following columns:

Name	Usage Level	Dependent on Charset?	Mux Category	Reference

The "Name" column reflects the attribute name (as it will appear in the SDP). The "Usage Level" column MUST indicate one or more of the following: session, media, source, dcsa and dcsa(subprotocol). The "Dependent on Charset?" column MUST indicate "Yes" or "No" depending on whether the attribute value is subject to the charset attribute. The "Mux Category" column MUST indicate one of the following categories: NORMAL, NOT RECOMMENDED, IDENTICAL, SUM, TRANSPORT, INHERIT, IDENTICAL-PER-PT, SPECIAL or TBD as defined by [I-D.ietf-mmusic-sdp-mux-attributes]. Finally, the "Reference" column indicates the specification(s) where the attribute is defined.

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

Name	Usage Level	Dependent on Charset?	Mux Category	Reference
setup	session,media, dcsa,dcsa(msrp)	No	IDENTICAL	[RFC4145] [RFC6135] [I-D.mmusic-msrp-usage-data-channel]

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 =           (att-field ":" att-value) / att-field

   att-field =           token

   att-value =           byte-string

   ; 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" or "udp"

   port =                1*DIGIT

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

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

   IP4-multicast =       m1 3( "." decimal-uchar )
                         "/" ttl [ "/" 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 RFC5234>
DIGIT =               <DIGIT definition from RFC5234>
CRLF =                <CRLF definition from RFC5234>
HEXDIG =              <HEXDIG definition from RFC5234>
SP =                  <SP definition from RFC5234>
VCHAR =               <VCHAR definition from RFC5234>
URI-reference =       <URI-reference definition from RFC3986>
addr-spec =           <addr-spec definition from RFC5322>
```

10. Summary of Changes from RFC 4566

- Generally clarified and refined terminology.
- Identified now-obsolete items: "a=cat", "a=keywds", "k=".
- Updated normative and informative references, and added references to additional relevant related RFCs.
- Reformatted the SDP Attributes section for readability. The syntax of attribute values is now given in ABNF.
- Made mandatory the sending of RTCP with inactive media streams.

- Removed the section "Private Sessions". That section dates back to a time when the primary use of SDP was with SAP (Session Announcement Protocol). That 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 "lang" and "sdplang" attributes.

- Removed some references to SAP because it is no longer in widespread use.

- Changed the way <fmt> values for UDP transport are registered.

- Changed the mechanism and documentation required for registering new attributes.

- Tightened up IANA registration procedures for extensions. Removed phone number and long-form name.

- Reorganized the IANA nettype registry

- Reorganized the several IANA att-type registries into a single registry

- Revised ABNF syntax for clarity. Backward compatibility is maintained with a few exceptions:

  * 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. (This is incompatible with certain aberrant usage.)

  * Updated the "IP6-address" and "IP6-multicast" rules, consistent with the syntax in RFC3986. (This mirrors a bug fix made to RFC3261 by RFC5964.) Removed rules that were unused as a result of this change.

- 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 RFCs 5735 and 5771.

- Changed some examples to use IPv6 addresses, and added additional examples using IPv6.

- Incorporated case-insensitivity rules from RFC 4855.
- Revised sections that incorrectly referenced NTP.
- Clarified the explanation of the impact and use of a=charset.
- Revised the description of a=type to remove implication that it sometimes changes the default media direction to something other than sendrecv.

11. Acknowledgements

Many people in the IETF Multiparty Multimedia Session Control (MMUSIC) working group have made comments and suggestions contributing to this document.

In particular, we would like to thank 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, Joerg 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.

12. References

12.1. Normative References

[E164] International Telecommunication Union, "E.164 : The international public telecommunication numbering plan", ITU Recommendation E.164, November 2010.

[I-D.ietf-mmusic-data-channel-sdpneg] Drage, K., Makaraju, M., Ejzak, R., Marcon, J., and R. Even, "SDP-based Data Channel Negotiation", draft-ietf-mmusic-data-channel-sdpneg-28 (work in progress), May 2019.

[I-D.ietf-mmusic-sdp-mux-attributes] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-17 (work in progress), February 2018.

[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>.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <https://www.rfc-editor.org/info/rfc4145>.

[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>.

[RFC6135] Holmberg, C. and S. Blau, "An Alternative Connection Model for the Message Session Relay Protocol (MSRP)", RFC 6135, DOI 10.17487/RFC6135, February 2011, <https://www.rfc-editor.org/info/rfc6135>.

[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>.

12.2. Informative References

[I-D.ietf-mmusic-ice-sip-sdp]
Petit-Huguenin, M., Nandakumar, S., and A. Keranen, "Session Description Protocol (SDP) Offer/Answer procedures for Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-sip-sdp-36 (work in progress), June 2019.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-54 (work in progress), December 2018.

[ITU.H332.1998]
International Telecommunication Union, "H.323 extended for loosely coupled conferences", ITU Recommendation H.332, September 1998.

[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>.

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

[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>.

[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>.

Authors' Addresses

Ali Begen
Networked Media
Konya
Turkey

EMail: ali.begen@networked.media

Paul Kyzivat
USA

EMail: pkyzivat@alum.mit.edu

Colin Perkins
University of Glasgow
School of Computing Science
University of Glasgow
Glasgow G12 8QQ
UK

EMail: csp@csperkins.org

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

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